

**DOKTORI (PhD) ÉRTEKEZÉS**

**TESCHNER GERGELY**

**MOSONMAGYARÓVÁR**

**2014.**

NYUGAT MAGYARORSZÁGI EGYETEM  
MEZŐGAZDASÁG- ÉS ÉLELMISZERTUDOMÁNY KAR  
MOSONMAGYARÓVÁR  
GAZDASÁGTUDOMÁNYI INTÉZET

Doktori Iskola vezető:  
DR. SZABÓ FERENC DSc  
az MTA doktora, egyetemi tanár

Programvezető:  
DR. TENK ANTAL CSc  
professor emeritus

Tudományos vezetők:  
DR. CSATAI RÓZSA CSc                      DR. TENK ANTAL CSc  
egyetemi docens                              professor emeritus

**TEJTERMELŐ VÁLLALKOZÁSOK MENEDZSMENT-  
INFORMÁCIÓS RENDSZEREINEK JELENLEGI ÉS  
TOVÁBBFEJLESZTÉSI LEHETŐSÉGEI**

Írta:  
TESCHNER GERGELY

MOSONMAGYARÓVÁR  
2014

**TEJTERMELŐ VÁLLALKOZÁSOK MENEDZSMENT-  
INFORMÁCIÓS RENDSZEREINEK JELENLEGI ÉS  
TOVÁBBFEJLESZTÉSI LEHETŐSÉGEI**

Értekezés doktori (PhD) fokozat elnyerése érdekében  
a Nyugat- magyarországi Egyetem  
Mezőgazdaság- és Élelmiszertudományi Kar  
Ujhelyi Imre Állattudományi Doktori Iskolája  
Az állati termék előállítás, feldolgozás és forgalmazás ökonómiai kérdései  
alprogramja keretében

Írta:  
TESCHNER GERGELY

Témavezető: Dr. Csatai Rózsa  
Témavezető: Dr. Tenk Antal

Elfogadásra javaslom (igen / nem)  
(alíírás)

A jelölt a doktori szigorlaton \_\_\_\_% -ot ért el,  
Mosonmagyaróvár .....  
a Szigorlati Bizottság elnöke

Az értekezést bírálóként elfogadásra javaslom (igen /nem)

Első bíráló (Dr. ....) igen /nem  
(alíírás)

Második bíráló (Dr. ....) igen /nem  
(alíírás)

A jelölt az értekezés nyilvános vitáján.....% - ot ért el

Mosonmagyaróvár, .....  
a Bírálóbizottság elnöke

A doktori (PhD) oklevél minősítése.....  
Az EDT elnöke

## TARTALOMJEGYZÉK

---

TARTALOMJEGYZÉK	4
ÁBRAJEGYZÉK	6
TÁBLÁZATOK JEGYZÉKE	7
1. BEVEZETÉS	8
2. HIPOTÉZISEK	12
3. A DISSZERTÁCIÓ CÉLKITŰZÉSEI	14
4. SZAKIRODALMI ÁTTEKINTÉS	15
4.1. Főbb tendenciák a világ- és az EU szarvasmarha tartásában	15
4.1.1. A világ szarvasmarha ágazatának kilátásai	15
4.1.2. A szarvasmarha ágazat helyzete az EU-ban	17
4.2. A hazai szarvasmarha ágazat jelenlegi helyzete és kilátásai	19
4.3. Számítógéppel támogatott döntések a mezőgazdaságban	27
4.4. A Magyarországon legelterjedtebb telepírányítási rendszerek alkalmazási területei	40
4.4.1. Az ALPRO rendszer.	40
4.4.2. A RISKÁ rendszer	42
4.4.3. A TALP telepírányító rendszer	46
4.4.4. Tapasztalatok a telepírányítási rendszerekkel összefüggésben	49
4.5. Vezetői döntéshozatal	55
4.5.1. A döntés	55
4.5.2. A döntéshozatal szakaszai	56
4.5.3. Döntések csoportosítása	60
4.6. Adat és információ	64
4.6.1. Adat és információ értelmezése	64
4.6.2. Az információ szerepe a döntéshozás folyamatában	65
4.7. Döntéstámogató rendszerek	67

5. ANYAG ÉS MÓDSZER	74
5.1. Mélyinterjú kutatás	75
5.2. Szekunder kutatás a vizsgálatba vont vállalatokról	76
5.3. Szoftverfejlesztés	76
6. EREDMÉNYEK ÉS ÉRTÉKELÉSÜK	79
6.1. Mélyinterjú kutatás bevont vállalkozások főbb jellemzői	79
6.2. Mélyinterjú kiértékelése	85
6.3. Adatbázis kialakítása a szarvasmarha ágazattal kapcsolatos döntések támogatásához	95
6.4. Tervezést és döntést támogató információs rendszer kialakítása	105
7. KÖVETKEZTETÉSEK, JAVASLATOK	114
8. ÚJ ÉS ÚJSZERŰ TUDOMÁNYOS EREDMÉNYEK	120
9. ÖSSZEFOGLALÁS	121
10. KÖSZÖNETNYILVÁNÍTÁS	123
11. IRODALOMJEGYZÉK	124
12. Mellékletek	134
1. Melléklet: Komplex nemzetközi vállalatirányítási rendszer mezőgazdasági célú szolgáltatásokat megvalósító speciális illesztési protokolljainak fejlesztése	<b>Hiba! A könyvjelző nem létezik.</b>
2. Melléklet: A mélyinterjú során használt vázlat	134
3. Melléklet: Titoktartási megállapodás	<b>Hiba! A könyvjelző nem létezik.</b>
4. Melléklet:	138
4.2. Data Access Layer	141
4.3. Import Layer	157
4.4. Models Layer	158
4.5. User Interface Layer	187



## ÁBRAJEGYZÉK

---

1. ábra: A világ tejtermelése 1996-2010-ig.	16
2. ábra: Szarvasmarha állomány változása Magyarországon 1990-2010-ig	20
3. ábra: Tejtermékek külkereskedelme – mennyiségben	24
4. ábra: A döntéshozás szakaszai, normatív modell	57
5. ábra: Döntéshozatal szakaszai, nem programozott döntési helyzetben	58
6. ábra: Információ jellemzői az egyes vezetői szintek és problémaszerkezetek esetében	67
7. ábra: Fejt tehénlétszám adatok	81
8. ábra: A vizsgált minta területi elhelyezkedése	82
9. ábra: Vizsgált minta megoszlása vállalkozási forma szerint	83
10. ábra: Az árbevétel alakulása 2008-2011 között vizsgált vállalkozásonként	84
11. ábra: A válaszadók munkaköri megoszlása	85
12. ábra: Nincs kapcsolat a főkönyvi rendszerrel	89
13. ábra: Támogatja-e a döntéshozatalt?	90
14. ábra: Terv-tény elemzés gyakorisága	91
15. ábra: A kialakított rendszerstruktúra	100
16. ábra: SQL tábla kapcsolatok	103
17. ábra: Az elkészült rendszer felhasználói felülete	107
18. ábra: A hónapok kiválasztása, és a jelszó bekérése	109
19. ábra: Importálandó file kiválasztása	111
20. ábra: Az elkészült új rendszer elhelyezkedése és információ forrásai a szarvasmarhatartó telep működésében	112

## TÁBLÁZATOK JEGYZÉKE

---

1. táblázat: Magyarország 15 legnagyobb tehenlétszámmal rendelkező tehenészeti telepe rangsorolva	21
2. táblázat: A nyerstej termelői és kiviteli ára Magyarországon	26
3. táblázat: Szarvasmarhát tartók száma régióként (ezer db)	23
4. táblázat: Szoftverek megoszlása feladatok alapján	34
5. táblázat: A rendszer által kezelt listázások	45
6. táblázat: A döntéstámogató rendszerek fejlődése	69
7. táblázat: TEAOR-0141 tevékenységet végző vállalkozások száma (2008-2010)	79
8. táblázat: Szoftverhasználat a tejelő szarvasmarhát tartó gazdaságoknál	86
9. táblázat: A vezetői döntések megalapozása	93
10. táblázat: Egy döntést támogató rendszertől elvárható igények a tervezésben	94
11. táblázat: Importálandó CSV file (részlet)	102



## KIVONAT

---

Az elmúlt 20 évben a döntéstámogatás területén számos, a gyakorlatba átültetett informatikai fejlesztés történt, melyek felmérése országos szinten aktuálissá vált, különösen az ágazat számítástechnikai rendszereinek használatában jelentkező hiányosságok, fejlesztésre szoruló területek miatt. Az informatikai szektorban a folyamatosan csökkenő adattárolási egységre- és sebességnövekményre jutó költségek következtében, egyre szélesebb körben kerülnek bevezetésre informatikai megoldások.

A szarvasmarha ágazat komplex menedzsment rendszere több évtizede beépíti a vezetés, irányítás témakörével kapcsolatos vizsgálatokat, kutatási eredményeket a racionális viszonyokhoz közelítő döntések meghozatalához. Az információ mint erőforrás mindig is fontos szerepet töltött be a vállalatirányítás egyes szintjein, ezért az újabb információs technológiák megkövetelik a további kutatásokat és az eredmények publikálását.

A vizsgálatok alapján elkészült egy döntéstámogató információs rendszer a tejtermelő vállalkozások számára, kifejezetten a tervezési feladatok támogatására. A szoftver tartalmazza mindazon adatbázis elemeket, melyek igényként jelentek meg a kutatás során. A kifejlesztett alkalmazás által kalkulált mutatószámok megfelelőnek bizonyulnak az operatív döntések meghozatalához és a tervezési feladatok elvégzéséhez.

Az aggregált, egy felületen megjelenített adatok jobban segítik a vállalatok irányítását és hatékonyabb működését.

## ABSTRACT

---

Over the past 20 years, many decision-supporting IT improvements have been put into practice and it became necessary to conduct a nationwide survey, especially due to weaknesses and the areas needing improvement in the use of the sector's IT systems. In the IT sector, due to the continuously decreasing cost per unit of data storage and speed increment, the IT solutions are introduced more and more widely.

The complex management system of beef industry has been incorporating the studies and the research results connected to the topics of leadership and management into the decision-making, which approaches the rational circumstances, for several decades.

Information as a resource has always played an important role at each level of corporate management; therefore, the new information technologies require further research and publication of their results.

Based on the studies, a decision-supporting information system for the dairy businesses, specifically to support their design tasks, has been developed. The software contains all the elements of the data base, which have appeared as a requirement during research. The indicators calculated by the developed application prove to be adequate to make operational decisions and to perform design tasks.

The aggregated data, displayed on the same surface, help the management of the companies and their more effective operation much better.

## 1. BEVEZETÉS

---

A világon mindenütt, így Magyarországon is megfigyelhető (kutatásokkal igazolható), hogy egyre kevesebb olyan vállalkozás tevékenykedik, ahol nem alkalmaznak számítástechnikai eszközöket. Az újabb technológiák segítségével tárolt adatok és információk mennyisége exponenciálisan növekszik, ezáltal a vállalkozásoknak a növekvő információtengerben kell versenybe szállni a konkurens cégekkel, megfelelni a tulajdonosok, szakhatóságok elvárásainak.

A vezetők feladata, hogy megalapozott és jó döntéseket hozzanak, a vállalkozás céljait szem előtt tartva. Ehhez releváns adatokra és információkra van szükségük, gyorsan feldolgozható formában.

Egy nagyobb szervezet esetében elképzelhetetlen, hogy a működés során keletkezett adatok tárolására és gyors feldolgozására ne használjanak tranzakció-feldolgozó rendszert. A keletkező nagy mennyiségű adatot adatbázisba rendezve - további vezetői információs rendszerek használatával- értékes információk nyerhetők. Kívánatos állapot, hogy az üzleti tevékenység minden részletéről átfogó ismeretek álljanak rendelkezésre. Ez jelen technikai feltételek mellett nem lehetséges és az ember is csak korlátozott mértékben képes egy döntési szituációban teljesen objektíven választani a lehetséges alternatívák közül.

A vezetői információs rendszerek fejlesztése területén számos kutatás történt, ezek eredményének egy részét már átültették a gyakorlatba. Az egyes rendszerek fejlettségét tekintve szó lehet akár egyszerű táblázatkezelő szoftverről, vagy komplex vállalatirányítási rendszerről is. Mindezek segítséget nyújtanak a vezető számára az alternatívák vizsgálatában, az egyre alaposabb és gyorsabb döntéshozatalban és a döntések következményeinek feltárásában. Így a döntéshozó nem csak

a tapasztalatára, gyakorlatára, megérzéseire támaszkodhat, hanem a racionális viszonyokhoz közelítő döntést tud hozni.

A témával kapcsolatos vizsgálataimat képező tejtermelő vállalkozásokról ugyanúgy, mint a mezőgazdaság egészéről elmondható, hogy több évtizede jelennek meg kutatási eredmények, fejlesztések döntéstámogató rendszerekkel kapcsolatosan. Ennek ellenére a kialakított és használt rendszerek nem állnak azon a fejlettségi szinten, ahogy azt a szakemberek előzetesen remélték.

A mikroszámítógépek és az újabb információs technológiák megkövetelik a témakör folyamatos kutatását és az eredmények publikálását.

A kutatás magyarországi tejtermelő vállalkozások döntéstámogató rendszereire terjed ki. A szakirodalom áttekintést ad a tejtermelő ágazat helyzetéről, problémáiról világviszonylatban, illetőleg az Európai Unióban és Magyarországon. A téma megkívánja, hogy az elmúlt időszak kutatásai bemutatásra kerüljenek, ezért a szakirodalom feldolgozása során a magyarországi problémákra kialakított eljárások, informatikai rendszerek bemutatása már az 1960-as évektől kezdődik.

A szakirodalom foglalkozik a döntés, a döntéstámogatás, az adat és az információ tárgyköreivel. A vezetőt növekvő mennyiségű információ éri és fontos, hogy ki tudja szűrni a számára fontos és a döntéshozásban releváns elemeket. A kutatás során egy olyan döntéstámogató rendszer kifejlesztésére került sor, amely - reményeink szerint - az eddigieknél jobban segítheti a magyarországi tejtermelő vállalkozásokat döntéseik megalapozásában. A rendszer önálló használata, vagy más rendszerekbe való beépítése - és ezzel együtt a magyarországi elterjedés - érdekében a dolgozat bemutatja a szoftver forráskódját is.

## 2. HIPOTÉZISEK

---

A szakirodalmi források – mint szekunder információk-, valamint a kutatásba bevont vállalkozásokról és az általuk használt számítógépes döntéstámogató informatikai rendszerekről előzetesen gyűjtött információk figyelembe vételével az elvégzendő kutatások irányának kijelöléséhez az alábbi hipotézisek állíthatók fel:

**Első hipotézis (H1):** A tejtermeléssel foglalkozó hazai vállalkozások által **alkalmazott számítástechnikai eszközállomány mind kapacitás, mind pedig korszerűség tekintetében igen különböző.** Hasonlóan jelentős különbségek vannak az egyes gazdaságokban használt szoftverek alkalmazhatósága, illetve effektív alkalmazási területei között is.

**Második hipotézis (H2):** A **döntéstámogató programok (szoftverek) használatának hatékonyságát** mind az állományméret, mind pedig az adatforrások (adatbázis) köre és terjedelme korlátok közé szorítja.

A rendelkezésre álló **technikai feltételek csak korlátozott lehetőséget nyújtanak** a teljesen objektív és a rendszer minden elemét átfogó döntés-előkészítéshez.

**Harmadik hipotézis (H3):** A hazai mezőgazdasági vállalkozásokra - így a **tejtermelő vállalkozásokra is - a szubjektív elemekkel operáló döntés-előkészítés, döntéstámogatás jellemző.** Ez azt jelenti, hogy a viszonylag korszerű számítástechnikai eszközök és módszerek mellett – gyakran azok kikerülésével – készítenek elő, illetve hoznak meg döntéseket.

**Negyedik hipotézis (H4):** A vállalatokban zajló **döntési folyamatok nem alkotnak konzisztens rendszert** és különösen a vállalati (ágazati) tervezés területén van szükség a döntéstámogató rendszerek fejlesztésére.

**Ötödik hipotézis (H5):** Olyan döntéstámogató rendszer kidolgozására van szükség, amely az adott rendszerről rendelkezésre álló információk (adatbázis) felhasználásával **önállóan és/vagy más rendszerekkel összekapcsolva ad lehetőséget tejtermelő vállalkozások menedzsmentjének tervezésével kapcsolatos döntéstámogatására, illetve – előkészítésre.**

**Hatodik hipotézis (H6):** A kutatásokból nyert és a termelői gyakorlatot pontosan leképező információk lehetőséget adnak egy **saját fejlesztésű számítógépes szoftvernek a létrehozására és a gyakorlat számára történő átadására** (kipróbálására). Az új szoftvernek egyik előnye (újszerűsége) abban van, hogy **összekapcsolható a vállalatban alkalmazott többi szoftverrel.**

Ezek a hipotézisek, valamint az ezek alapján megfogalmazott célkitűzések alapul szolgálhatnak a kutatások anyagának és módszerének megválasztásában, továbbá a saját vizsgálatok tárgyának (anyagának) és területeinek kiválasztásához. Egyben előrevetítik annak a lehetőségét, hogy a kutatásokból hol, milyen területeken várhatók új, illetve újszerű tudományos eredmények.

### 3. A DISSZERTÁCIÓ CÉLKITŰZÉSEI

---

A disszertációt előkészítő primer kutatások végső célja egy **tudományosan megalapozott innovatív fejlesztés**, mely a gyakorlatban elvégzett széleskörű felméréseken alapszik és amely a saját vizsgálatokból származó információk felhasználásával szolgáltat **új döntéstámogató rendszert** szarvasmarhatartó telepek, illetve tejtermeléssel foglalkozó vállalkozások számára.

Ennek a végső célnak az elérését az alábbi részcélok, illetve kutatási fázisok hivatottak megalapozni:

1. A kidolgozandó döntéstámogató informatikai rendszer adaptálásának megalapozását előkészítendő e rendszert alkalmazó **tejtermelő vállalkozások gazdálkodási jellemzőinek, főbb paramétereinek bemutatása.**

2. Országos szinten annak felmérése, hogy a tehéntej-termeléssel foglalkozó **vállalkozásokban aktuálisan hol tart a termeléssel összefüggő vezetői döntéshozás informatikai megalapozottsága**, különös tekintettel a tervezésnek információ-technológiai támogatottságára.

3. A tejtermelésben jelenleg alkalmazott döntés-előkészítő és támogató **informatikai rendszerek, illetve szoftverek használatában jelentkező hiányosságok, fejlesztésre szoruló területek feltárása.**

4. A kifejlesztésre kerülő új döntéstámogató informatikai rendszer felépítéséhez és hatékony működtetéséhez **szükséges adatbázis kijelölése.**

5. Az előző négy pontban vázolt célok teljesítése következtében született információkat alapul véve egy – a gyakorlatban elterjedt információs rendszerekhez képest **új – döntéstámogató szoftver kialakítása, amely első sorban a tervezést segíti, úgy, hogy összhangban van a vállalkozás többi szoftverével és döntéseiknél alternatívákat kínál a menedzsment számára.**

## 4. SZAKIRODALMI ÁTTEKINTÉS

---

A gazdasági állatfajok közül a szarvasmarha világ-, Európai Unió (a továbbiakban: EU) és hazai viszonylatban is igen nagy jelentőséggel bír. A szarvasmarha ágazat egyik, talán legfontosabb célja a lakosság tej- és tejtermékekkel történő ellátása. A szarvasmarha-tartás másik terméke a hús. Harmadsorban megemlítendő, hogy a szarvasmarha-ágazat ipari nyersanyagokat is előállít (pl. bőripari nyersanyag). A szarvasmarha a legelő, a tömegtakarmányok és melléktermékek fontos hasznosítója és mint ilyen, a fenntartható környezet fontos tényezője.

### 4.1. Főbb tendenciák a világ- és az EU szarvasmarha tartásában

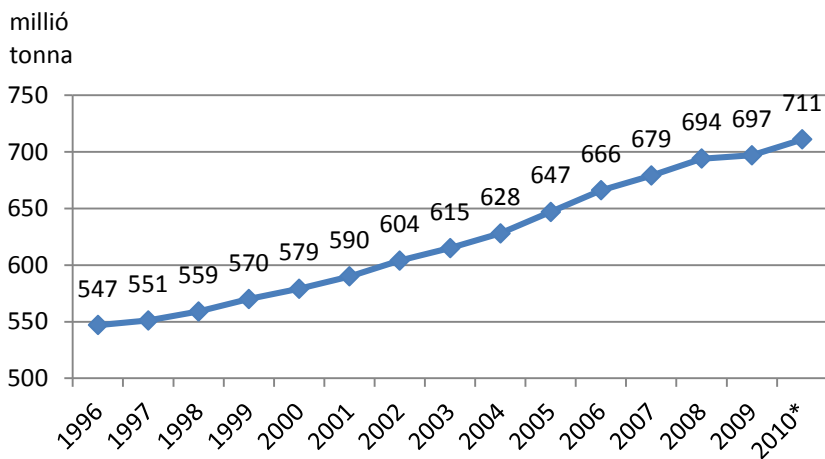
#### 4.1.1. A világ szarvasmarha ágazatának kilátásai

A világon a tejelő szarvasmarhák száma folyamatosan emelkedik: míg 2000-ben 220 millió, 2010-ben már 264 millió egyed tartottak nyilván.

A világ tehéntej termelése 2000-ben 490 millió tonna volt, ez 2010-re 600 millió tonnára emelkedett, ami 22%-os növekedést jelent. A 2000-ben termelt mennyiségnek közel háromnegyedét Észak- és Dél Amerika, valamint Európa adta (143 és 210 millió tonna). Ázsia 94 millió tonna tehéntejet termelt, míg Afrika és Óceánia 20-20 millió tonnával részesült. 2010-re az ázsiai termelés emelkedett jelentősen: 158 millió tonna, a növekedés 68 %, kiemelkedő a Kelet-ázsiai növekedés: 20 millió tonnáról 46 millióra (130 %). Emelkedés figyelhető meg Észak- és Dél Amerikában is: 176 millió tonna (23 %) tehéntejet termeltek. Európa, Afrika és Óceánia termelése változatlan maradt (FAOSTAT, 2000-2010).



BLASKÓ (2011) szerint a globális gazdasági válság a tejágazatban is hanyatlást okozott 2008 végén, ami 2009 első felében jelentős hatással volt a termékárakra. Az 1. ábrán megfigyelhető, hogy a világ tejtermelésének növekedése lelassult 2009-ben. 2009. második fele és a 2010-es év azonban változást hozott. A világkereskedelem jelentős fellendülése következett be 2009. év második felében. 2010. első féléve során az árak emelkedése mellett a termelés is növekvő tendenciát mutatott. A tejtermelés volumene várhatóan tovább emelkedik, 2017-re eléri, illetve meg is haladja majd a 794 millió tonnát. A világ tejtermelésének tendenciáját mutatja az 1. ábra 1996. és 2010. között. 2010-ben 1,6 %-os növekedés figyelhető meg, de ez elmarad az elmúlt évtizedekben tapasztalt 2,1 %-os átlagos éves növekedéstől. A 2010. évi növekedéshez legnagyobb mértékben India és Kína járult hozzá, de Brazília, az EU 27 és az USA is növelni tudta a tejtermelését.



**1. ábra: A világ tejtermelése 1996-2010-ig.**

Forrás: FAO, 2010

A rendelkezésre álló adatok alapján folyamatos emelkedés figyelhető meg a tej-, és tejtermékfogyasztásban. 2009-ben azonban az egy főre jutó tejfogyasztás 0,4 %-kal csökkent. A csökkenés elsősorban a gazdasági világválságra vezethető vissza. A fogyasztás visszaesésének másik oka a kínai fogyasztás emelkedésének mérséklődése a „melamin botrány” miatt (BLASKÓ, 2011).

A FAO (2010) adatai alapján 2009-ben a tej- és tejtermékfogyasztás világviszonylatban 103,0 kg/fő/év volt. Amíg a fejlett országokban az átlagos fogyasztás 245 kg/fő/év, addig a fejlődő országokban ez az érték csupán 66,2 kg/fő/év. A tej-, és tejtermékek fogyasztási szintje 1 %-kal várhatóan emelkedni fog és eléri a 104,3 kg/fő/éves szintet.

#### **4.1.2. A szarvasmarha ágazat helyzete az EU-ban**

FAOSTAT (2010) adatai alapján a világon megtermelt tej mennyiségének 24,5%-át (147 millió tonna) az EU-27 országai adták 2010-ben. 2000-ben a világon megtermelt tej mennyiségének 31%-át adta az EU, mely 2004-ben tíz új tagállammal bővült. Ennek eredményeként jelentősen növekedett a tejtermelés volumene. 2007-ben újabb két tagállam csatlakozott, az ő csatlakozásukkal azonban a tejtermelésben számottevő változás nem történt.

A világ tejelő szarvasmarha állományának 9%-ával rendelkezik az EU, ami 23,8 millió tejelő szarvasmarhát jelent

A tejtermelésből legnagyobb hányadot képviselő országokban található a tejelő tehénállomány több, mint 60%-a. A legnagyobb tejelő tehénállomány Németországban található (4,2 millió), ezt követi Franciaország (3,7 millió), Lengyelország (2,5 millió), Egyesült Királyság (1,9 millió), Olaszország (1,9 millió), majd Hollandia 1,5 millió egyeddel. Magyarország tejelő tehénállománya 248 ezer, ami 34%-os csökkenést mutat a 2000. évhez képest (FAOSTAT, 2010).

Az EU országaiban több, mint száz szarvasmarha fajtát tartanak, azonban a tejtermelés szempontjából a Holstein-fríz adja a legnagyobb hányadot. A Holstein-frízt elsősorban fogyasztói tej előállítására alkalmazzák, mely nagy tejmennyiséggel, gépi fejésre alkalmas tőggel, jó technológiatűrő képességgel rendelkezik, így az iparszerű tejtermelő telepek legfontosabb fajtájává vált.

A tej-, és tejtermékek az EU-ban a közvetlenül szabályozott termékek körébe tartoznak. A piaci szabályozás keretfeltételeit az 1255/99/EK- a termelés szabályozását pedig a 3950/92/EK tanácsi rendelet tartalmazza. E rendeletek értelmében a termékpálya egyensúlyi helyzetének a megőrzéséhez a következő eszközök állnak rendelkezésre:

- belpiaci intézkedések (intervenciók felvásárlás és magántárolási támogatás);
- tejtermékek belső felhasználásának támogatása;
- kvótaszabályozás;
- külkereskedelem szabályozása (export-importengedélyezési rendszer, import vámok alkalmazása, export visszatérítés).

(NYÁRS et.al., 2004).

Világpiaci kitekintésben elmondható, hogy a vaj és a teljes tejpor esetében Óceánia jelentős versenyelőnyben van alacsony árai miatt. A sovány tejpor tekintetében az USA és Óceánia versenyképesebb, de a Single CMO (Mezőgazdasági Piacok Közös Szervezésével Foglalkozó Irányító Bizottság) ülésén történt export visszatérítés mértékének emelése már jelentős eredményeket mutat az exportban az EU számára. Az ACP (African, Caribbean, Pacific) országok felé a sűrített tej és a tejpor exportja jelentősnek mondható. Az exporttal kapcsolatban megállapítható, hogy az EU-27 számára a legnagyobb kihívást a sajt exportja jelenti (NAGY, 2009).

Az EU tejipara az egyesülések és a felvásárlások révén az egyre nagyobb méretű és költséghatékonyabb vállalati struktúra felé halad. Ezek a nagyvállalatok jellemzően a saját márkanéveik felhasználásával a fontosabb tejtermékek gyártására és forgalmazására összpontosítanak. A világ legnagyobb feldolgozóiból (12 óriáscég) hét európai, és hatnak a központja is az EU-ban található. Európában a tejtermékpiac legfontosabb szereplői a regionális, akár egész országokat átfogó szövetkezetek (pl.: Arla Foods, Campina Melkunie, Friesland Coberco, Sodiaal, Nordmilch), valamint a transznacionális óriáscégek (pl.: Nestlé, Danone, Kraft) (NYÁRS et. al., 2002).

Az EU tejtermelésére vonatkozó előrejelzések szerint, 2015-re 4-5%-os növekedés várható, aminek következtében a uniós belső árak 10 %-kal csökkenhetnek. A tejtermékek fogyasztáscsökkenését is prognosztizálják. Várhatóan számos tagállam nem fogja kitölteni a tejkvótáját, mivel csökkent a tejtermelésük (URL<sup>1</sup>).

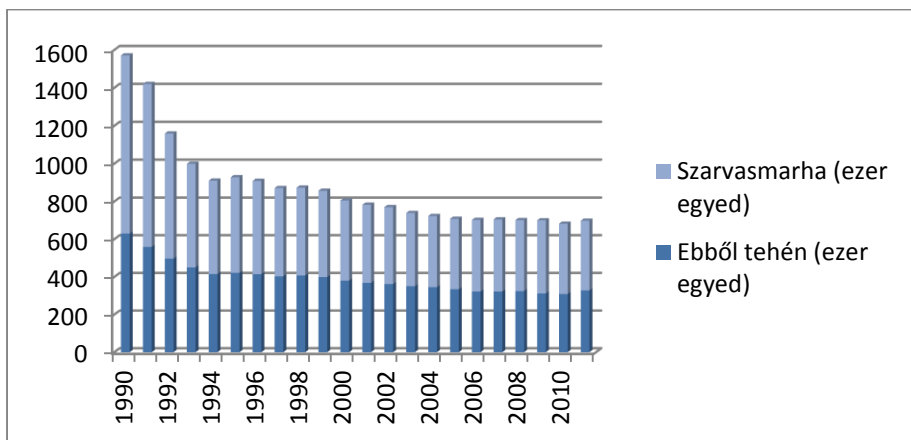
#### **4.2. A hazai szarvasmarha ágazat jelenlegi helyzete és kilátásai**

A tejtermelés valamennyi agrárágazattal rendelkező országban az egyik leginkább figyelemmel kísért kérdés. A piacgazdasági verseny közismerten szelektálja a termelőket, s közülük csak azok tudnak fennmaradni, akik fajlagosan nagy hozamokat és extra minőségű, viszonylag nagy tömegű tejet a legalacsonyabb fajlagos ráfordítással tudnak előállítani. Ennek eléréséhez a nagy biológiai értékű fajta és az igényes takarmányozás meghatározó szerepének hangsúlyozása mellett olyan tartási és üzemeltetési technológiák szükségesek, amelyek kielégítik a nagy termelési potenciállal rendelkező tehének termelőkörnyezeti, valamint a szakképzett munkaerő munkakörnyezeti igényeit, és lehetővé teszik a korszerű takarmányozás-technológia

alkalmazását is, miközben a fajlagos üzemeltetési költségek a lehető legalacsonyabbak (PATKÓS, 2007).

Magyarországon az 1990-es évtizedben igen jelentős változás következett be a szarvasmarhatartás területén. SZÉLES (1993) szerint az állattenyésztés egészére a hanyatlás volt jellemző, ami megnyilvánul az állatállomány jelentős csökkenésében és az állati eredetű termék előállításának és fogyasztásának visszaesésében egyaránt.

A 2. ábrán látható, hogy a hazai szarvasmarha állomány létszáma 1990-2000 között 1571 ezerről 805 ezerre (51%), a tehénlétszám pedig 630 ezerről (ebből tejhasznú 394 ezer) 380 ezerre csökkent (ebből tejhasznú 292 ezer darab) (60%). Ez a csökkenés európai viszonylatban igen jelentős volt.



## 2. ábra: Szarvasmarha állomány változása Magyarországon 1990-2010-ig

Forrás: KSH, 2011

Az utóbbi tizenöt évben jelentős állománykoncentráció zajlott le az ágazatban. Magyarország 15 legnagyobb tehénlétszámmal rendelkező tehenészeti telepének az állománylétszámát az 1. táblázat tartalmazza. A koncentráció hatására a kisebb gazdaságok kiestek a termelésből és

helyüket a nagy, 300 tehénnél többet tartó gazdaságok vették át. Ez a folyamat a minőségi, magas színvonalú termelésnek kedvezett (SZŰCS et. al., 2010).

### 1. táblázat: Magyarország 15 legnagyobb tehénlétszámmal rendelkező gazdasága

Rang-sor	Telep	Összes tehén	Tej kg	Zsír kg	Zsír %	Fehérje kg	Fehérje %
1	Enyingi Agrár Zrt.	1718	9621	370	3,84	314	3,26
2	Hunland Dairy Kft	1531	9288	337	3,62	307	3,31
3	Bóly Zrt.	1427	1008	364	3,61	331	3,28
4	Agroprodukt Zrt. Marcaltó-Ihász	1394	1036	360	3,47	334	3,22
5	Hód-Mezőgazda Zrt	1368	9969	370	3,71	318	3,19
6	Alcsiszigeti Zrt.	1315	8384	324	3,87	270	3,22
7	NAGISZ-TEJ Kft.	1266	8398	332	3,96	277	3,3
8	Hidasháti Mg. Zrt.	1147	7923	297	3,75	264	3,33
9	Solum Zrt.	1130	1076	369	3,43	346	3,22
10	Hunland Farm Kft.	1087	9112	343	3,77	307	3,37
11	Tedej Zrt.	1086	8887	313	3,52	289	3,25
12	Geo-Milk Kft.	1080	1037	388	3,74	337	3,25
13	Sárvári Mg. Zrt.	1078	1046	397	3,79	342	3,27
14	Pélpusztai Mg. Kft.	1037	9023	373	4,13	292	3,24
15	Agroprodukt Zrt. Marcalgergelyi	1020	9837	328	3,33	320	3,25

Forrás: HFTE-NÉBIH-ÁT Kft., 2012

Hazánkban a termelt tej 80%-át a 300-nál nagyobb egyedszámmal rendelkező telepek termelik, melyek a gazdaságok mindössze 3, az állománynak azonban 68%-át képviselik (SALAMON et. al., 2008).

A fajlagos hozamok 1990 után folyamatosan emelkedtek, melynek hatására az állományi létszám csökkenése ellenére sem csökkent az országos tejhozam. Ez a folyamat azonban megállt, és Magyarország tejtermelése az EU csatlakozást követően tovább csökkent. A tej- és

tejtermékek külkereskedelmében 2004. óta értékben nettó importóri pozícióba kerültünk (POPP et. al. 2010).

Magyarország szarvasmarha állománya a 2000-2011. év közötti időszakot figyelembe véve csökkenő tendenciát mutat. A KSH adatai szerint 2000-ben 805 ezer szarvasmarhát számláltak (ebből gazdasági szervezetnél: 543 ezer, egyéni gazdaságnál: 263 ezer egyed volt). 2010-ben összesen 682 ezer (ebből gazdasági szervezetnél: 449 ezer, egyéni gazdaságnál: 233 ezer), azonban 2011-ben már 698 ezer szarvasmarha volt Magyarországon (ebből gazdasági szervezetnél: 450 ezer, egyéni gazdaságnál: 248 ezer), így az állomány az előző évihez viszonyítva 2,4%-kal gyarapodott. A szarvasmarha-állomány 64,5%-át a gazdasági szervezetek, 35,5%-át az egyéni gazdaságok tartották, ez az arány nem változott lényegesen a 2010. évi adatokhoz képest. A tehénállományon belül folytatódott a korábbi években megfigyelhető szerkezeti átalakulás, mert 2010-hez képest a húshasznú tehenek száma 11,6%-kal (77 ezer egyedre), a kettős hasznosítású állomány 14,9%-kal (54 ezer tehenre) bővült (EGRI, 2012).

A 2. táblázat adatai szerint Magyarországon 2010-ben összesen 934 gazdasági szervezet foglalkozott szarvasmarhatartással. Ebből a tehéntartó gazdasági szervezetek száma: 799 volt. A legtöbb szarvasmarhatartó gazdasági szervezet az Észak-alföldi és Dél-alföldi régióban található. A szarvasmarhatartó egyéni gazdaságok száma 2010-ben összesen 18168, ebből a tehéntartó egyéni gazdaságok száma: 12587. A többi országrészhez viszonyítva jelentősen több az egyéni gazdaságok száma az Észak-alföldi és a Dél-alföldi régióban.

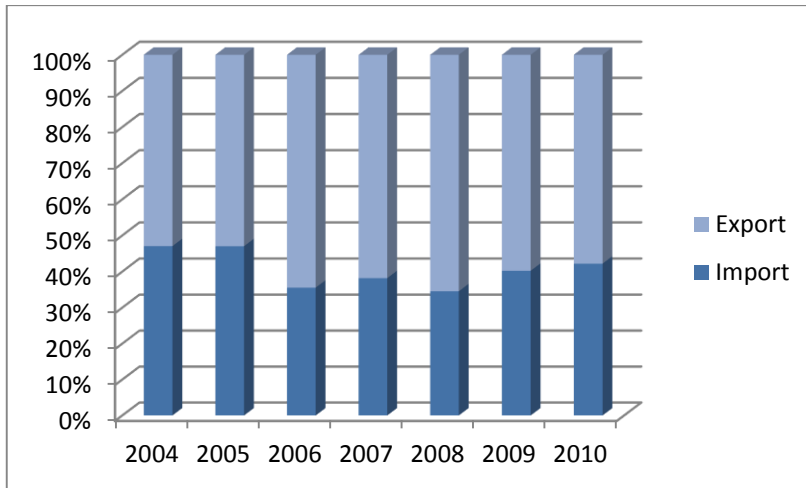
**2. táblázat: Szarvasmarhát tartók száma régióként (ezer db)**

<i>Régió</i>	<i>Szarvasmarha</i>	<i>Ebből: tehén</i>
<b><i>Gazdasági szervezetek</i></b>		
Közép-Magyarország	105	87
Közép-Dunántúl	119	107
Nyugat-Dunántúl	127	105
Dél-Dunántúl	131	104
Dunántúl	377	316
Észak-Magyarország	100	85
Észak-Alföld	198	176
Dél-Alföld	154	135
Alföld és Észak	452	396
<b><i>Összesen</i></b>	<b><i>934</i></b>	<b><i>799</i></b>
<b><i>Egyéni gazdaságok</i></b>		
Közép-Magyarország	1292	853
Közép-Dunántúl	1252	704
Nyugat-Dunántúl	1958	1235
Dél-Dunántúl	1418	904
Dunántúl	4628	2843
Észak-Magyarország	1677	1335
Észak-Alföld	4942	3569
Dél-Alföld	5629	3987
Alföld és Észak	12248	8891
<b><i>Összesen</i></b>	<b><i>18168</i></b>	<b><i>12587</i></b>

Forrás: KSH, 2012

2010-ben közel 40 milliárd forinttal haladta meg az import (85 milliárd forint) értéke az export (47 milliárd forint) nagyságát, 2011 első félévében a behozott tej és tejtermékek értéke 35%-kal múlta felül az exportot. Annak ellenére, hogy a tej és tejtermék importunk értéke kb. 40 %-kal meghaladja az export értékét, a 3. ábráról leolvasható, hogy 2004 óta az export mennyisége nagyobb az import mennyiségénél.





### 3. ábra: Tejtermékek külkereskedelme – mennyiségben

Forrás: KSH, 2011

A 2010. évi 400 millió kg-os export közel 90%-át a tej (jelentős mértékben nyerstej) adta, míg az importban a tej (dobozos) mennyiségi aránya alig 60%.

A sajt és túró exportunk mind mennyiségben, mind értékben 1/3-a ugyanezen kategória importjának, a 2010-es 85 milliárd forintos importból 36 milliárd forint ebben a termékkörben keletkezett.

2011-ben a legtöbb importtermék Németországból, Szlovákiából és Lengyelországból érkezett Magyarországra. A Németországból származó import döntő hányadát a magas hozzáadott értékű termékek (sajt, túró, joghurt, stb.) tették ki, míg Szlovákiából főleg dobozos tartós tej érkezett.

A tej- és tejtermék exportunk fele 2010-ben 2 országba irányult. Legjelentősebb exportpartnerünk (Olaszországot is megelőzve) Románia volt. Az ide irányuló 12,1 milliárd forintos kivitel több mint 50%-át a tej (dobozos és nyerstej) tette ki, de közel 6 milliárd forint értékben magasabb hozzáadott értékű terméket is sikerült elhelyeznünk

a román piacon. Olaszországba lényegében csak nyerstejet exportálunk, a 2010. évi 9 milliárd forint értékű kivitellel szemben 2009-ben még közel 12 milliárd forintnyi export irányult ide (OLÁH, 2011).

A tejkvóta-rendszer alapvetően a megtermelt és értékesítésre kerülő nyerstej mennyiségének szabályozásával, illetve egyéb piacszabályozási eszközökkel garantálja a piaci egyensúly megteremtését és fenntartását. Az EU a nemzeti kvóta mértékén felül termelt tehéntej mennyiségek piacon történő elhelyezésének vagy tárolásának költségeit a túltermelést előidéző tejtermelőkkel térítteti meg, fizetési kötelezettség (illeték) kiszabása révén. (177/2011. (XII.19.) számú MVH Közlemény).

2009/2010-ben az EU által meghatározott tejkvóta mennyiség 75 %-át termelte meg Magyarország. A 2010/2011-es kvótaévben az EU tagállamai közül Magyarországon csökkent a legnagyobb mértékben a tejtermelés. 2010-es évben 1,64 milliárd liter tejet állítottunk elő, a kvótánkat alig 80%-os mértékben használtuk ki. A megtermelt tej több, mint 75%-át a gazdasági szervezetek adták.

A visszaesés két fő okra vezethető vissza: a minőségi követelmények szigorodása és az árszínvonal csökkenése, ami végsősoron csökkenő jövedelmezőséget okozott (VÁGÓ, 2005).

Magyarországon 2011-ben - az előző évihez képest - a nyerstej felvásárlása 7%-kal csökkent. A nyerstej termelői-és kiviteli árakat a 3. táblázat tartalmazza. A nyerstej átlagára 21%-kal, a tej és tejtermékek feldolgozói ára 9%-kal nőtt. A nyerstej teljes kivitele 26%-kal emelkedett (a tejtermelés költségeinek döntő hányadát kitevő takarmányköltségek ezt jóval meghaladó mértékben nőttek. (OLÁH, 2011).

2012-ben a nyerstej 67%-át a termelők és a kereskedők, 33%-át a feldolgozók vitték ki külföldre. A termelők és kereskedők nyerstej kivitele 13%-kal, a feldolgozóké 78%-kal nőtt egy év alatt. A

felvásárlás csökkenésével a tejtermékgyártás bruttó termelése 5, a belföldi értékesítés 7, az export 2%-kal csökkent az év első tizenegy hónapjában az egy évvel korábbihoz viszonyítva (EGRI, 2012).

### 3. táblázat: A nyerstej termelői és kiviteli ára Magyarországon

Me: HUF/kg

Megnevezés	2004	2005	2006	2007	2008	2009	2010	2011
Nyerstej termelői alapára	62,1	62,82	62,29	70,46	79,6	58,64	69,04	83,55
Nyerstej termelői átlagára	62,47	62,98	62,35	70,7	81,01	60,14	71,3	85,96
Teljes nyerstej kiviteli ára*	-	-	-	-	-	65,54	78,83	92,96

\* A nyerstej kiviteli árának figyelése 2009-ben kezdődött

Forrás: AKI PÁIR, 2012

A tejtermék értékesítésben az üzletláncok a domináns szereplők, piaci stratégiájuk egyoldalúan fogyasztó-centrikus, árkövető magatartást folytatnak, valamint kereskedelmi márkás termékeikkel gyengítik a feldolgozók pozícióit. A tej- és tejtermékek növekvő behozatala miatt tovább csökkenhet a hazai feldolgozók belpiaci részesedése, ami elkerülhetetlenné teszi a cégek közötti szelekciót és valószínűsíti a külföldi tőke térnyerését.

A magyar termékek iránti hűség nem jellemző, a fogyasztók legfeljebb egy-egy „bevált”, régi hazai márkához ragaszkodnak és nyitottak a kereskedelmi láncok által preferált, olcsóbb importtermékek iránt. Persze van néhány olyan hazai termék, amely ma is annyira népszerű, hogy még az importtermékekre építő diszkontok sem hagyhatják ki a kínálatukból. Ezek jól megkülönböztethető, magas feldolgozottsági szintű és nagy hozzáadott értékű termékek, többnyire joghurtok,

szeletes termékek és tejdesszertek. Igazi növekedést csak a hazai produktumok értek el a magyar termékek a piacon (POPP et. al, 2010). Tej és tejtermékek esetében a rendszerváltás előtti fogyasztás (193,3 kg/fő/év) nagyobb volt, mint napjainkban. A mélypontot az 1995-ös év jelenti, 129,4 kg/fő/év fogyasztással, ezt követően is ingadozó fogyasztás tapasztalható (BALOGH, 2008).

A tej-, és tejtermékfogyasztás alapvetően jövedelmi kérdés. Ezt bizonyítja, hogy míg hazánkban a magasabb átlagkeresettel rendelkezők éves tejfogyasztása mintegy 250 kg/fő körüli, addig az alacsonyabb jövedelmű réteg alig 90 kg/fő mennyiséget fogyaszt (SZŰCS et. al, 2010). Részleteiben vizsgálva a kérdést, a folyadék formájában történő fogyasztás (natúr, ízesített és savanyú tejfélék) 66 kg, a vajfogyasztás 1,2 kg, a sajt fogyasztás (túróval együtt) pedig 7,1 kg/fő/év, ami az EU országai közül igen alacsony tej és tejtermék-fogyasztási szintet mutat (URL<sup>2</sup>).

### **4.3. Számítógéppel támogatott döntések a mezőgazdaságban**

A számítógép- és információtudomány eredményeinek alkalmazása a mezőgazdaságban hosszabb időszakra tekint vissza. A kereskedelmi forgalmú számítógépek az egyetemeken az 1960-as években kezdtek megjelenni. Ezek első felhasználói a mezőgazdasági kutatók és a szaktanácsadói szolgáltató szervezetek dolgozói voltak. Az 1975-1990.-es években a mezőgazdaságban is nagyobb mértékű informatikai fejlesztésekre került sor, elsősorban a nagyobb állami gazdaságokban és termelőszövetkezetekben. A fejlesztések során több helyen törekedtek komplex információs rendszerek kialakítására, melyekben fontos célként szerepelt a különböző szintű vezetői döntéstámogatás (HEGYI et. al., 2011).

A számítógépek használata az állattartásban a szarvasmarhatartás területén jutott a legmagasabb szintre. Ennek oka részben az állatok magas egyedi értékében, részben a gépesítés magas szintjében kereshető. Az állatok egyedi kezelésére éppen a tejtermelő tehenészetekben jelentkezett először igény az eltérő és napi szinten értékelhető tejelőteljesítmény miatt (TAKÁTSY, 2000; ARTMANN, 1997).

Az 1950-60-as években a gazdasági vezetők munkájuk során mindinkább fontos tényezőként tekintettek a matematikai elemzésekre, számításokra, mint döntéseiket támogató eszközre. Általánossá vált az a szemlélet, hogy a vezetők elsődleges feladata a lehetőségek közül az adott viszonyok között a legmegfelelőbb és a lehető legnagyobb eredményt biztosító alternatíva kiválasztása. A tapasztalat azonban önmagában még nem garantálja a legjobb megoldás kiválasztását. Ebben az időben több tanulmány jelent meg, melyek a lineáris programozás alkalmazásával foglalkoztak. Így - többek között - alkalmazásra került a takarmányadagok összeállításánál (TÓTH, 1964; BARTOS, 1967). Bemutatásra kerültek a készletgazdálkodás, sorbanállási jelenségek, berendezések, gépek használata során felmerülő problémákat kezelő matematikai, lineáris programozási modellek. Útmutatást adtak a számítógépen való megoldás lehetőségeiről és módszereiről is (MEGYERI, 1965).

CSETE (1965) már valószínűsíti az "elektronikus számítógépek" gyors elterjedését és a nagytömegű számítási feladatokat igénylő módszerek szélesebb körű alkalmazását. Külön kitér mind a növénytermesztés, mind az állattenyésztés terén alkalmazható optimalizáló eljárásokra.

Csete és munkatársai évente rendszeresen 300-400 termelőszövetkezetben végeztek felméréseket, vizsgálatokat, segítették a vállalati tervek tudományos megalapozását és számítógépekkel végezték el a manuális feladatok többségét. Ebben az időben még

lyukkártya segítségével programozták a műveleteket és azon tárolták az adatokat (KLÁTYIK, 1969).

Több kutató foglalkozott a számítógépek alkalmazásával a hálótervezésben, ahol 100-120 eseményből álló hálódigramok esetében kézzel elvégezhetőek a számítások, ennél nagyobb hálódigramok, illetve mélyebb számítások elvégzésénél szükség volt az "elektronikus számítógépek" alkalmazására (KOVÁCS et.al., 1967; KOVÁCS, 1969).

PINTÉR (1974) a számítógépek akkori teljesítményével foglalkozik a többperiódusú lineáris programozás módszerének alkalmazásával, ezen belül is a programozási modellel.

A MAE Agrárgazdasági Társaság Üzemszervezési Szakosztályának vitaülésén a lineáris programozás szerepéről szólva Székely (1975) azt hangsúlyozta, hogy a hagyományos tervezési eljárásnál költségesebbek a lineáris programozással készült tervek, de a többletköltség megtérül. A hozzászólók azt emelték ki, hogy egy-egy programozással készült termelési terv 50-80 ezer Ft-os számítógépköltsége ugyan nem hagyható figyelmen kívül, de mérlegelni kell azt is, hogy 3,5-4 millió forint eredményjavulás érhető el velük (KISS et.al., 1975).

1979-ben az SAE Afikim bemutatta az első gyakorlatban alkalmazható elektronikus tejmérőt, majd az azt követő években egy teljes telepírányítási rendszert (PINHAS et.al., 2008).

A Debreceni Agrártudományi Egyemen 1978. szeptemberében rendezték „Az operációkutatás<sup>1</sup> és számítástechnika mezőgazdasági alkalmazása” című II. országos tudományos konferenciát. A 84

---

<sup>1</sup> operációkutatás: Az operációkutatást olyan tudományos és gyakorlati tevékenységnek tekintjük, amelynek célja a döntések előkészítése a bonyolult rendszerekben, ehhez tudományos módszerekkel közelítenek, amely kapcsolódik a számítógépek alkalmazásához is, ezáltal kvantitatív eredményekhez jutnak (MÉSZÁROS 1981).

előadásban elhangzottak alapján elmondható, hogy a számítógép alkalmazási területének kiterjesztésére számos koncepció, terv született. A különböző szintű döntéshozás során felmerülő koherens információigény kielégítésére több módszert mutattak be a konferencia előadói. A számítástechnika alkalmassá tette számviteli rendszerek létrehozását, az információgyűjtés és feldolgozás gépesítését. Ezzel összefüggésben ajánlasként hangzott el a felsőoktatás és a kutatóintézetek számára, hogy nem csak a mérnökképzésben, hanem a gyakorlatban dolgozó szakemberek számára is szükséges a számítástechnikai képzés illetve továbbképzés bevezetése. Emellett foglalkozni kell a vállalatok belső információs rendszerének a számítástechnika alkalmazásához szükséges kialakításával, információs rendszerek létrehozásával, melyek a gazdasági döntések matematikai megalapozásához használható adatokat szolgáltatnak (ERTSEY, 1978).

Csáki és munkatársai 1978-ban a Bábolnai Mezőgazdasági Kombinát részére számítógépes irányítási rendszert dolgoznak ki. A fő célkitűzésük egy olyan rendszer kialakítása, mely kielégíti a vállalati vezetés információigényét, megfelel az állami előírásoknak és rendelkezéseknek, továbbá a lehető legrövidebb időn belül dokumentálja a gazdasági folyamatok számszerű hatásait és nem eredményez létszámtöbbletet.

A rendszert három fő területre tagolták:

- információfeldolgozási alrendszer;
- döntéselőkészítési alrendszer;
- technikai ellátási alrendszer.

Az információfeldolgozási alrendszer további öt részrendszerből épült fel. Funkciójuk a vállalat éves tevékenységével összefüggő információk elsődleges feldolgozása, rendszerezése.

A döntéselőkészítő alrendszer feladata a vállalat különböző vezetési szintjein születő döntések megalapozása volt.

A technikai ellátási alrendszer egy UNIVAC 1005-ös kisteljesítményű számítógépből, egy R-22-es számítógépből és 4 Robotron 1372 kazettás adatrögzítő egységből állt (CSÁKI et.al., 1980).

CSETE, (1981) a számítógépek rohamos elterjedéséről, a számítástechnika mezőgazdasági alkalmazásának szakemberigényéről ír. A korszakot az jellemzi, hogy a hardwaregyártó cégek folyamatosan fejlesztik, növelik a számítógépek teljesítményét, viszont az alkalmazás széles elterjedésének gátat szab az ágazatban felmerülő software hiány. Megoldást a szakember képzésben látja, kiemelten a rendszerszervezés és a számítástechnika gyakorlati alkalmazásának területén. Ebben az időben a számítógépek kihasználatlanságának egyik oka az alkalmazó szakemberek kis száma, emiatt a meglévő számítógépek nem voltak kihasználva, sok esetben csak rendkívül egyszerű feladatok megoldására használták őket.

Az 1981-ben megrendezésre kerülő "A számítógép felhasználása a mezőgazdasági vállalatok tervezésében és irányításában" című konferencián, az előadók beszámoltak az e téren végzett elméleti és gyakorlati munkájukról. A rendezvény előadásai alapján elmondható, hogy a számítógépet a gyakorlatban legelterjedtebben a könyvelésben alkalmazzák. Ezen kívül készültek számítógéppel támogatott tervek is, így ötéves, éves és havi terveket készítettek a Debreceni és a Gödöllői



Agrártudományi Egyetemen. Számos elméleti modellt is bemutatnak, melyek komplett irányítási feladatokat láttak el (MÉSZÁROS, 1981). A számítógépek térhódítása nem volt zavarmentes, ezért szólni kell a 80-as években a számítógép alkalmazása során fellépő problémákról is. Magyarországon az akkor beszerezhető számítógépek választéka kicsi volt, a szomszédos országokban jóval nagyobb hardware és software választékkal rendelkeztek. A software-k többnyire nem egységesített és kis szériában készített termékek voltak, egyedi megrendelés alapján készültek. Gyakran előfordult, hogy ugyanarra az ügyviteli feladatra vagy gazdasági funkcióra több szoftver született. Nem volt kielégítő az információáramlás, a nagyüzemek nem szereztek tudomást a már elkészült szoftverekről. A kutatóintézetek nem elégítették ki a gazdasági szereplők igényeit, mivel a software-fejlesztés során első sorban a döntés-előkészítés területére koncentráltak, ott alkalmazták az operációkutatás módszereit, így ezen a területen értek el eredményeket. Az adatfeldolgozás háttérbe szorult, nem fejlesztették az ügyviteli, számviteli rendszereket. Az akkori szervezőintézetek szakemberei viszont az adatszolgáltató, ügyviteli területtel foglalkoztak. A szervező és kutatóintézetek nem működtek együtt, így a fejlesztés több irányban történt. A fejlesztés egységesítése halaszthatatlanná vált. Az 1980-ban megalakult az Állami Gazdaságok Információs Rendszerfejlesztő Együttműködése, az AGSZIR-COOP. Az AGSZIR-COOP feladat kettős volt, egyrészt valamennyi mezőgazdasági nagyüzem számára hasznosítható egységes, magas színvonalú számítógépes rendszer kidolgozása, másrészt a KSH-SZÜV-vel együttműködve biztosítani azt, hogy a kidolgozott alrendszerek az ország bármely részén, az azt alkalmazni kívánó mezőgazdasági nagyüzemben vagy az illetékes SZÜV számítóközpontban rendelkezésre álljon.

Az AGSZIR-COOP-nál fejlesztések történtek mind a készletgazdálkodás, mind a bérleggazdálkodás területén (MONOSTORI, 1982).

NYITRAI et. al. (1982) vizsgálta a döntéshozók információ ellátását a készlet és bérleggazdálkodással kapcsolatosan. Egy vállalatot vizsgáltak, amely nem rendelkezett önálló számítógéppel, hanem egy szomszédos mezőgazdasági nagyüzem R-22-es számítógépének szabad kapacitását használta. A vállalat havonta körülbelül 10 ezer darab bizonylatot dolgozott fel. Számítógép nélküli feldolgozás esetén 8-10 fő munkaidejét kötötte le, ezért indokolt volt a számítógép alkalmazása. A tanulmány rávilágított arra, hogy számos hibalehetőség adódik a számítógépes adatfeldolgozás során, mivel a rendszer hibás bizonylatok bevitelét engedélyezte. Több területen szakaszos adatfeldolgozás történt, így nem álltak naprakész információk a vezetők rendelkezésére. KOVÁCSNÉ et. al (1986) a software ellátottságával foglalkozó írásukban a mezőgazdasági célú és elérhető, forgalmazásban lévő mikroszámítógépes software termékeket mutatják be. Ez volt az első olyan tanulmány, amely összegyűjtötte a mezőgazdaságra vonatkozó teljes software kínálatot. A vizsgálatba 27 számítógéptípusra kidolgozott, 450 programot vettek be.

A tanulmány az IBM-PC alacsony részesedése ellenére nagy áttörést prognosztizál, mivel kedvező paramétereivel alkalmas a mezőgazdaságban való széleskörű elterjedésre.

A felmérés kiterjedt a vállalati feladatokhoz igazodó téma szerinti csoportosításra is.

A szerzők által készített összeállítás (4. táblázat) alapján megállapítható, hogy a felső vezetést támogató programok alkalmazása a mezőgazdaságban meglehetősen szűk körben terjedt el. Az állattenyésztés 50%-os részesedése a termelési folyamatokat leíró programokból viszont kifejezetten jónak mondható.

#### 4. táblázat: Szoftverek megoszlása alkalmazási területek alapján

Témakörök	Szoftvertermékek megoszlása
<b>A vállalat irányításához felhasznált programok:</b>	<b>10%</b>
ebből:	
tervezés	53%
szervezés	13%
ellenőrzés	16%
döntés	18%
<b>A termelés feltételrendszerét elemző programok:</b>	<b>7,90%</b>
ebből:	
állószerkezet	50%
forgószerkezet	-
anyagok	3%
munkaügy	27%
energia	20%
<b>A termelési folyamatot leíró programok</b>	<b>42,10%</b>
ebből:	
növénytermelés	31%
állattenyésztés	50%
egyéb	19%
<b>Az eredmények számbavételéhez felhasználható programok</b>	<b>27,90%</b>
ebből:	
számvitel	49%
pénzügy	8%
értékesítés	3%
munkabér	17%
adók	14%
ár	6%
egyéb	6%
<b>A gazdálkodási tevékenységet szolgáló programok</b>	<b>12,10%</b>
ebből:	
matematikai módszerek	62%
nyilvántartói rendszerek	13%
külső információk	10%
rendező módszerek	4%
külső-belső kapcsolatok	11%

Forrás: KOVÁCSNÉ et.al., 1986

A szerzők kiemelik a vállalatirányítás számára közvetlenül hatékonyan alkalmazható szoftverek alacsony számát, javasolják egy teljes gazdálkodást vagy legalább egy-egy komplex fázisát átfogó számítógépes rendszer kialakítását.

SZÉKELY (1986) szerint a termelés és vállalatirányítás gyakorlata elavult, korszerűtlen módszereken alapszik. Ez a megállapítás egy 1983-ban végzett reprezentatív kutatáson alapul, mely szerint a termelészövetkezetek 3-, az állami gazdaságok 14-, a mezőgazdasági nagyvállalatok átlagosan 5%-a használt korszerű számítástechnikai megoldásokat középható tervének kidolgozásához. A tanulmány- a felsorolt operációkutatási módszereken túlmenően - arról szól, hogy a gyors adatkezelési és számítástechnikai megoldások új, táblázat-kalkulációs vagy adatkezelési eljárások kidolgozását tették lehetővé.

Abban az időben a mezőgazdaságban nagyobb volt a fizetőképes kereslet a számítógépek és szoftverek iránt, mint ami programmal, számítógéppel kielégíthető lett volna. Az ágazatban több, mint 80féle személyi és mikroszámítógépet és közel 300-féle programot használnak. A szoftvereket illetően nagy átfedések voltak. 1986-ban 728 vállalatnál 1641 számítógép működött, beleértve az üzemgazdasági munkát segítő COMMODORE-64 típusú gépeket is. Ezeknek a gépeknek tárolási kapacitása alacsony volt, 90%-a a személyi és mikro kategóriába tartozott (TÓTH, 1986).

Egy 1987-ben bemutatott, számítástechnikára alapozott vállalatirányítási modell a szarvasmarha ágazat számára készült, amelyben a kiemelendő irányítási feladatok közé tartoztak a

- szaporodásbiológiai folyamatok (borjúszaporulat);
- takarmányozási folyamatok.

A modell a tervezést a "tényhelyzet"-re alapozta. A pontos tervek készítésének előfeltétele, hogy az aktuális állapotról megbízható adatok álljanak rendelkezésre. Az irányítási rendszer lényeges eleme egy

kidolgozott tervezőrendszer, amely segítségével bármikor lehet tervet készíteni. A rendszert kísérleti jelleggel a Gödöllői Agrártudományi Egyetem Üzemtani Tanszékén próbálták ki és a keresési és rendezési feladatokat leszámítva tökéletesen működött (SZELÉNYI, 1987).

Mester (1989) egy, a mezőgazdaságban alkalmazható ügyviteli rendszert mutat be. A szoftver mezőgazdasági specialitásokat tartalmazó komplex ügyviteli rendszer, melyet a megrendelő igényeihez igazítva vezettek be. Elsőként VT-20-as, majd IBM-XT és AT gépeken használták. Az elkészült számítógépes rendszer az alábbi feladatokat ellátó alrendszerekből állt:

- főkönyvi,
- anyag,
- pénztár elszámolási,
- bankelszámolási,
- vevő-szállító folyószámla vezetés,
- bér- és munkaügyi,
- költségfelosztás,
- állóeszköz,
- befejezetlen termelés.

Érdekes összehasonlítani az előzőekben bemutatott rendszer lehetőségeit egy 5 évvel későbbi írásából megismerttel, amely egy termelőszövetkezet számítógépes rendszerét mutatja be.

ROBOTRON 5110 és COMMODORE 64 típusú számítógép és összesen 15 software állt rendelkezésre. Ezek a szoftverek egymáshoz képest függetlenek, működésükben nincs kapcsolódási pont, szigetrendszerek.

Többek között az alábbi programokat használták:

- készletkönyvelési,
- főkönyvi könyvelési,
- munkabér-nyilvántartási,
- takarmányadag optimalizálási,
- segédüzemi teljesítményfelosztó,
- integrált sertéstartók nyilvántartása és elszámolása.

A szerző szerint továbblépést jelenthet a számítógépek cseréje nagyobb teljesítményű gépekre, de erre anyagi okok miatt nem volt lehetőség. Emiatt a rendelkezésre álló eszközök, rendszerek implementálása akadályokba ütközött (FEHÉR, 1991).

A 20. század utolsó évtizedének nagy áttörése az internet elterjedése volt és egyre több úgynevezett vékony-kliens eszközökön keresztül elérhető számítógépes rendszerek, döntéstámogatási információk használatára nyílt lehetőség. Az internet az akkori nemzetközi tendenciákat figyelembe véve nagy hatással volt az agrárüzleti tevékenység számos területére (HERDON, 2004).

Az EUROVET állatokat nyomonkövető rendszer egy Web alapú moduláris keretrendszer, alkalmas az állattartó hely és állomány nyilvántartására, állatazonosításra és- nyilvántartásra, szállítással, állategészségüggyel kapcsolatban felmerülő információk kezelésére (SZILÁGYI, 2006).

Az 1990-es évek elején kezdődő gazdasági átalakulás jelentős változásokat hozott Magyarország agrárágazatában. A tulajdonosi szerkezetváltozás hatására nemcsak a mezőgazdasági tevékenység, hanem az azt kiszolgáló számítástechnikai alkalmazások igénye, jellege, típusa is jelentősen megváltozott (HERDON, 2004).

A mai globalizálódó piaci környezetben a tejtermelő gazdaságnak olyan hatékony menedzsment eszközökre van szüksége, melyek

leegyszerűsítik a mindennapi munkát, feltárják az aktuális helyzetet és nélkülözhetetlen adatokat nyújtanak a hosszabttávú terveikhez.

A modern szoftverfejlesztéseket és technológiai újításokat felhasználva relációs adatbázisok segítségével olyan rendszerek készülnek, amelyekben az aktuális és múltbéli adatokon statisztikai mutatók, jelentések, szűrések végezhetőek.

Az állománymenedzsment modul főbb tulajdonságai:

- napi állományjelentések és általános termékenységi vizsgálat,
- egészségügyi ellenőrzés,
- fejési hatékonyság jelentések,
- automatikus állatorvosi ellenőrző lista készítése megadott paraméterek alapján,
- egyszerűen elkészíthető állatorvosi jelentések,
- selejtezési terv,
- állományméret és tejtermelés tervezése,
- könnyedén követhető automatizált folyamatok,
- fejlett szűrők tehén kiválasztáshoz, komplett adatbázison alapuló, átfogó jelentések állatorvosi vizsgálat és egyéb célokból,
- felhasználói jelentések.

(PINHAS, 2008; XIONG et.al., 2008).

Saját vizsgálataink alapján elmondható, hogy a modern fejlesztések között hiányzik a megfelelő kapcsolat, egymástól elkülönülő szigetrendszerek alakultak ki. (HEGYI et.al, 2009).

A menedzsment számára fejlesztett szoftvert a Szent István Egyetem Vállalati Gazdaságtudományi Intézete. A Józsefmajori Kísérleti Tangazdaságban alkalmazott OMIR (Operatív Menedzsment Információs Rendszer) szoftver, a tervezési és döntési folyamatokat segíti. Mindezt úgy, hogy külső és belső információs bázisaként

szolgálnak a felhasználók számára. A rendszer képes az éves tervezést segíteni azáltal, hogy támaszkodik a vállalat működése során keletkező belső információkra. Az állattenyésztést érintő tervkészítés felülről lefelé halad, a piac elemzésétől az erőforrás tervezésen keresztül egészen a terv-tény elemzésig (SZÉKELY, 2004). Saját vizsgálati eredményeink is alátámasztják a tervezés és terv-tény elemzés kiemelt jelentőségét és alkalmazási lehetőségeit. A menedzsment számára releváns belső információk rendszerben való megjelenítése segíti a döntési eljárást (TESCHNER, 2014).



#### **4.4. A Magyarországon legelterjedtebb telepírányítási rendszerek alkalmazási területei**

Annak érdekében, hogy a kutatások során kifejlesztett saját döntéstámogató informatikai rendszerünk újszerűségét bemutathassuk, az alábbiakban összefoglaljuk a nálunk jelenleg elterjedtebb telepírányítási rendszerek funkcióit és a gyakorlati alkalmazásuk során szerzett tapasztalatokat. Ilyen rendszerek: az ALPRO, a RISKÁ és a TALP elnevezésű, több alapfunkciójában hasonló, bizonyos funkciókban azonban különböző telepírányítási rendszerek.

##### **4.4.1. Az ALPRO rendszer.**

A tejtermelő telepek napi működését támogató szoftver és eszközzrendszer. Segítségével olyan információkhoz juthat a döntéshozó, amelyek segítik az operatív döntések releváns információkkal való támogatására.

Lehetővé teszi a pontos tehénazonosítást, segítségével nyomon lehet követni az egyedi tejhozamokat. Folyamatos információt biztosít a takarmányozás, a reprodukció, a testtömeg, a tejhozamok vonatkozásában. Grafikonok segítségével a különféle adatok könnyebben és gyorsabban értelmezhetőek, lehetőség van egyedi és csoportos összehasonlítások elkészítésére.

Az ALPRO Windows funkciókat az alábbi fő menüpontokban csoportosították:

##### **Fejés**

A fejési funkciókkal ellenőrizhető a napi és a hosszú távú fejési teljesítmény. Alapvető funkciók: tejhozamok, laktációs grafikon, tejfolyás és fejési időtartam grafikonok, illetve tejsír és tejfehérje adatok. A program kiértékeli a fejési műveletet is.

## **Takarmányozás**

Ez a menüpont az alábbi funkciókat tartalmazza: takarmányadag számítása, takarmányfogyasztás, főbb teljesítménymutatók (takarmányköltséget meghaladó árbevétel, a takarmányozás fajlagos költsége)

## **Tenyésztés**

Az ALPRO Windows tenyésztési modulja nyújt segítséget a tenyésztéssel kapcsolatos feladatok ellátásához. Ezek a funkciók lehetővé teszik a szaporodási és tenyésztési események regisztrálását, és segítséget nyújtanak az állatok szaporodásának megfigyeléséhez és ellenőrzéséhez. A főbb teljesítménymutatók (mint például a szervizperiódus, vemhesülési arány, vemhességi arány és ellési időköz) értékelése alapján lehetséges a műveletek hosszú távú nyereségességének biztosítása, illetve nyomonkövetése.

## **Állategészség**

Az ALPRO Windows segítségével minden egyes állat egészségi állapotáról nyilvántartás készíthető. Ehhez az alábbi adatok adhatók meg: szomatikus sejtszám, testkondíció-pont és az előforduló betegségek. Egy vagy több válogatókapu alkalmazása esetén az ALPRO Windows felhasználható egyes állatok elkülönítésére, egy adott fejési műszak után vagy akár hosszabb időszakra is. Ezen kívül kritériumok adhatók meg az állatok automatikus elkülönítésére; például ilyen lehet a nagy aktivitás vagy a várható termékenyítési idő.

## **Ivarzás észlelése**

Az aktivitási görbe és az aktivitási jelek nyújtanak segítséget az ivarzás felismerésében. Az ivarzás észlelő funkciókhoz szükség van a DeLaval aktivitásmérő rendszerre.

## **Figyelmeztetések és naptár**

A Figyelmeztetések ablak és a Naptár hívja fel a figyelmet az esedékes termékenyítésre, ellésre vagy szárazra állításra; ilyen figyelmeztető jel

lehet a tejhozam csökkenése vagy a túl kicsi takarmányfogyasztás. A naptár felhasználható a tenyésztési események regisztrálására, feljegyzések hozzáadására és a „teendők lista” kinyomtatására.

### **Állományadminisztráció**

Ide tartozik az állatok regisztrálása és törlése, a transzponderek és az aktivitási érzékelő számok hozzárendelése, a csoportok definiálása és a csoportok közötti áthelyezése.

### **Kimutatások**

A kimutatások különösen rugalmas és hasznos eszközként használhatók. Felhasználhatók az adatok megtekintésére és a feladatok végrehajtására is. Az ALPRO Windows több előre definiált és azonnal felhasználható kimutatást készít. Ha ennél többre van szükség, akkor a meglévő kimutatások módosíthatók vagy teljesen új, egyéni igények szerint kialakított kimutatások generálhatók.

### **Grafikonok**

Az ALPRO Windows laktációs diagramjai fontos jellemzőket osztanak meg: trendek, eltérések megjelenítése és információk összegzése. A kimutatásokhoz hasonlóan a grafikonok módosíthatók és új, egyéni igény szerint kialakított grafikonok hozhatók létre.

### **Riasztások**

A hibák kimutatásának képessége fontos kívánalom bármely komplex rendszerrel szemben. Az ALPRO Windows az ALPRO processzor által kibocsátott riasztásokat listázza (DeLaval, 2006).

#### **4.4.2. A RISKÁ rendszer**

RISKA korszerű Szarvasmarha Telepirányítási rendszer A SYSTO Kft.-ben 25 éve fejlesztették ki és mára az Internetet is felhasználják információ-szolgáltatásai javításához (URL<sup>3</sup>).

A RISKA program kialakítása óta (a nyolcvanas évek vége) sokat fejlődött a világ. Olyan funkciók és elvárások merültek fel, amelyek

akkor még elképzelhetetlenek voltak. A gépek sebességének és kapacitásának növekedése olyan elgondolások megvalósítását is lehetővé teszi, amelyre a kezdetekben a fejlesztők nem is gondoltak. Ezért vált szükségessé a régi alapokon nyugvó, de teljesen új program készítése, amely a több felhasználós hálózatokon való munkát is lehetővé teszi (SYSTO, 2008).

Jelentős mérföldkő volt 1997 májusában az Állattenyésztési Teljesítmény Vizsgáló Kft. és a SYSTO Kft. között létrejött együttműködési szerződés. Ennek keretében minden ÁT Kft. Központban kitelepített RISKÁ verzió működik (Kovács, 2004).

A RISKOM program a telepírányítási program szerves része. Ez a program biztosítja az ÁT KFT.-vel való kétirányú adatcserét. Úgy mint az általuk végzett ellenőrző befejeésekhez használt PSION Workabout kézi számítógépekről az adatok átvételét; a tejminták laborvizsgálati eredményeinek fogadását mágneslemezről vagy modemről; a termelési és tenyésztési adatok átadását-átvételét, illetve összehasonlítását az ÁT KFT. központi adatbázisával. Ez a program teszi lehetővé a telepen használt PSION Oragniser II. vagy Workabout kézi számítógépnek a tehén- és növendékadatokkal való feltöltését és a módosítások visszatöltését a RISKÁ program adatbázisába (SYSTO, 2008).

A programrendszer fő modulja a tenyésztési program. Része a tehénállomány nyilvántartó, amely a telepi alapnyilvántartásból gyűjti a származási, termelési, tenyésztési adatokat, tárolja a szaporodásbiológiai, a tőgy, a láb, az anyagcsereforgalmi és egyéb kezeléseket. A folyamatosan felvitt adatokból a program eseménynaptárt készít, ahol a tehén összes adata időrendi sorrendben lekérdezhető. A szakmai összefüggések és folyamatok mélyen leprogramozottak, ezért a visszkapott információk nagyon jól és biztonságosan használhatók a mindennapi munkában. Így válik lehetővé a tenyésztési és termelési adatok, a gazdaságossági mutatók

állandó követése, összegzése, eloszlások, összefüggések vizsgálata (pl.: termékenyítési index, produktivitás, szervizperiódus, két ellés közti idő, 305 napos laktáció, várható termelés stb.). A program 20 rokonsági szintig keres anyai, illetve nagyanyai háttérrel, értékes tehéncsaládok adatainak gyűjtése által. Az adatállomány kezelése és áttekinthetősége garancia a kötetlen tehenészetekben a tehén egyediségének a megőrzésére (KOVÁCS, 2004).

A Riska program által kínált lehetőségeket az alábbi felsorolásban és az 5. táblázatban foglaltuk össze:

- bevitel,
- kiválasztás,
- eseménynaptár,
- laktációs görbe,
- kezelések,
- diagnosztika,
- kimutatások készítése,
- paraméteres lekérdezés.

### 5. táblázat: A rendszer által létrehozott output adatok (listák)

<b>Riska program által nyújtott gyors lekérdezési lehetőségek:</b>	<b>Szaporodás-biológiai listák:</b>	<b>Állategészségügyi listák:</b>
Szaporodás-biológiai táblák	Várható ellések	Kezelésre visszarendeltek időrendi sorrendben
Ellési gyorsinformációk	100 napon belül ellett tehének	Szaporodási betegségek
Konténer-térkép	Külföldi azonosítójú egyedek	Tőgybetegségek
Napi gyorsinformációk értékek	Termékenyített tehének	Lábbetegségek
Termékenyítés gyorsinformációk	Várhatóan ivarzó tehének	Egyéb betegségek
Gyorsinformációk üres és üresre bírált tehenekről	Vemhességvizsgálat gyorsinformációk	Vakcinázás
Az üres tehének utolsó szap.biol. kezeléseinek eloszlása	Mérlegelés gyorsinformációk	Minden betegség
Laktáció zárási gyorsinformációk	305-napos zárt laktációk adatai zárás dátuma szerint sorrendben	Kezelt tehének listája
Termelési állapot gyorsinformációk	Termelési állapot	
Szárazra állítási gyorsinformációk	Várható szárazra állítások időrendi sorrendben	
Gyorsinformációk a növendék állományról	Állategészségügyi tábla:	
	Selejtezés - kikerülés táblák	
	Kikerülés gyorsinformációk	
	Kikerülési gyorsinformációk (db,kg,Ft)	
	Selejtezés gyorsinformációk	
	Selejtezés - kikerülés listák	
	Kikerülések	
	Selejtezések	

Forrás: (URL<sup>3</sup>)

#### 4.4.3. A TALP telepírányító rendszer

A TALP program Windows XP és Win 7 környezetben telepíthető.

Külön menücsoportokban található az adatbeviteli, a lekérdezési (lista készítés), az adatkapcsolati és technikai beállítási feladatok elvégzését biztosító funkciók.

Lehetőség van egy-egy állat összes adatának egy képernyőn történő megtekintésére és szükség esetén egy korábbi adat módosítására.

A program modulokból épül fel, az egyes modulok közül igény szerint lehet kiválasztani a felhasználónak, hogy a program biztosította összes modul közül melyeket szeretné használni.

A program az alábbi területek nyilvántartására és karbantartására alkalmas:

- törzsadatok nyilvántartása (azonosító, születési idő, ivar, apa, anya, kikerülés, stb.),
- bikatörzs adatok (tenyészbikák adatai),
- tenyésztési adatok nyilvántartása (termékenyítések, vemhességek, ellések, stb.),
- termelési adatok nyilvántartása (termelt tej, zsír, fehérje, szomatikus sejtszám, karbamid, stb.),
- termelési (takarmányozási) csoportok kialakítása áttekinthető, paraméterezhető táblázatok használatával,
- állategészségügyi adatok (egyedi és csoportos kezelések, egyéb beavatkozások, TBC vizsgálathoz nyilvántartások),
- tőgyegészségügyi modul (szomatikus sejtszámvizsgálati adatokon alapuló szűrőlista, mastitest vizsgálatok, bakteriológiai vizsgálatok eredményeinek és az okszerű kezeléseknél a nyilvántartása),

- gyógyszer nyilvántartás (gyógyszertörzs adatok, gyógyszer paraméterek, készletek, élelmezés-egészségügyi várakozási idők figyelése, stb.),
- ENAR modul (fűljelző készletnyilvántartás, marhalevél adatok kezelése, ENAR bejelentések nyilvántartása, ENAR leltár, stb.),
- termékenyítő anyag nyilvántartás (spermakészletek, adag, termelési szám, stb.),
- állományváltozások kezelése (korcsoportok kialakítása, korosbítások, állományváltozások napi, havi, éves),
- takarmányozási napló vezetése (takarmánytörzs adatok, takarmány paraméterek, takarmánykészletek, takarmányozási napló készítés),
- tehén egyedi érték nyilvántartás (tenyésztésbe állításkori érték, értékcsökkenési futamidő, maradvány érték, egyedi érték adott időpontban stb.),
- tejházi nyilvántartások (termelt tej fejésenként, naponta, dekádonként, havonta, felhasználások, értékesítések.),
- adatkapcsolatok (termelésellenőrzés adatainak átvétele, közvetlenül ellenőrzőfejés után, labor adatok elektronikusan, TALP-at használó telepek közötti teljes körű adatátadás, Tenyésztő szervezetek adatai (küllemi bírálati adatok, tenyészértékek) Fejőrendszerekkel kölcsönös adatkapcsolat (Al Pro, Boumatic),
- származás elemzések (származási adatok értékelése, beltenyésztési együttható számítás, adatok leválogatása és átadása párosítási tervkészítő programoknak, párosítási adatok átvétele).



A TALP az adatbázisban tárolt adatok lekérdezésére többféle, a felhasználó igényeihez igazodó lehetőséget biztosít. A lekérdezések menü alatt az alábbi fő csoportosításban kereshetők az adatok:

- tenyésztési adatok,
- termelési adatok,
- termelési csoport listák,
- állatorvosi kezelések,
- szaporodásbiológiai értékelések,
- laktációs listák,
- napi teendők listái.

Az egyes csoportokon belül választható listák (megadott adatkörben) egyedileg is paraméterezhetők, azaz szűrések és rendezési szempontok tetszés szerint beállíthatóak.

A témakörönként választható listák mellett készíthetők szabadon összeállítható listák is.

Ezeknél a listáknál a felhasználó dönti el, hogy milyen adatok, milyen sorrendben és milyen szűrési feltételek szerint kerüljenek listázásra, mi legyen a lista neve, stb. Ezek a szabadon összeállítható listák igény esetén elmenthetők és bármikor indíthatók a későbbiekben.

Az ÁT kft. által közel két évtizede elektronikusan gyűjtött tenyésztési és termelési adatok egy központi adatbázisban kerülnek tárolásra. Ebből az adatbázisból igény szerint lehetőség van egy tenyészet adatbázisának a leválogatására. A TALP programmal egy-egy telep adatbázisa egyszerűen és gyorsan betölthető a telepi számítógépre. Így a telepi szakemberek az aktuális adatok ismeretében kezdenek meg a TALP használatát, készíthetnek kimutatásokat és egyedi igények szerint kibővített tartalommal folytathatják az adatok rögzítését. (atkft.hu, ÁT Kft., 2009)

#### **4.4.4. Tapasztalatok a telepírányítási rendszerek gyakorlati alkalmazásáról**

2011-ben felmérést végeztek a magyarországi tehenészetek számítógépes irányításával összefüggésben. Tizenöt hazai nagyüzemi tehenészetben szerzett saját-, illetve tehenészeti szakemberek tapasztalatainak összegzését végezték el. A tehenészet irányításához a tehenészetekben a következő számítógépes programcsomagok valamelyikét alkalmazták: Farm 2.000, Milka, RISKA, és TALP. A megkérdezettek becslése szerint a tehenészetek többségében (amely több száz hazai tehenészetet jelent) a RISKA programcsomagot használják. A tapasztalatok úgy összegezhetőek, hogy a RISKA programot használó tehenészetek a papíralapú nyilvántartásról (pl. a tenyésztési napló, a fedeztetési jegyzék, a szaporulati napló, az ellési napló, az involúciós és méhkezelési jegyzék, az ENAR leltár, a krotália nyilvántartás, a marhalevél nyilvántartás stb. esetében) áttértek a számítógépen történő, elektronikus nyilvántartásra.

Valamennyi vizsgált tehenészetben használják a Microsoft Word és Excel programokat. Microsoft Worddel készítik a tehenészeti leveleket, szerződéseket, meghatalmazásokat, belső felhasználású szakmai anyagokat, melyek elkészítéséhez korábban írógépet használtak. A tehenészet helyi igényeinek megfelelő nyomtatványokhoz és táblázatokhoz, a tejtermeléshez beszerzett anyagok felhasználásának nyilvántartásához, a tehenészeti dolgozók munkabeosztásához, a havi, negyedéves-, fél éves-, háromnegyed éves-, évenkénti összesítésekhez és a tehenészeti statisztikák-, elemzések-, grafikonok-, összefüggések készítésekhöz az Excel programot tartják nélkülözhetetlenek napjainkban.

A megkérdezettek becslése szerint jelenleg mintegy száz hazai tehenészet (köztük a felkeresett 15 tehenészet is) alkalmaz tehenészeti telepírányítási rendszert.

A telepírányítási rendszerek - típusától függetlenül - automatikus működésű mérő-adatgyűjtő, illetve beavatkozó hardver és szoftver eszközöket tartalmaznak.

A felmérésben résztvevő tehenészetek valamennyi telepírányítási rendszere lényegét tekintve vezetékes, számítógépes mérő-adatgyűjtő és adatfeldolgozó hálózat.

A tehenészeti szakemberek egybehangzó véleménye szerint a meglévő telepírányítási rendszerük teljesebb körű használatával, további, hasznos, termelési információ elérése válna lehetővé. Jelenleg a rendszereiket leginkább a fejési adatok összegyűjtésére használják. A tehenészetek számítógépes rendszereit jellemzően az alábbi szakemberek használják: - telepi rendszergazdák, állatorvosok, felcserek, inszeminátorok, műszakvezetők, telepvezetők (kevés idejük jut a számítógépes adathalmaz hasznosítására). A rendszergazdák rendelkeznek a legteljesebb ismerettel a tehenészetekben alkalmazott számítógépes rendszerről, de saját bevallásuk szerint tudásuk hiányos és részleges. A rendszergazdák véleménye alapján a rendelkezésre álló gépkönyvek többnyire a saját verziójukat megelőző verziószámú rendszerről szólnak és főként angol nyelvűek. A nyilatkozó telepi szakemberek szívesen fogadnák a szakmájukra szabott, rendszeres, a tehenészetben tartott, a számítógépes rendszerről szóló továbbképzést, a számítógépes táv-szaktanácsadást.

A közelmúltban 5 hazai tehenészetben a fejőházi rendszerük üzemeltetésében megoldásként jelent meg a gép-gép kommunikáció (M2M machine to machine kommunikáció), amely új lehetőséget jelent a távoli gépek (távoli fejőgépek) üzemeltetésében, illetve az üzemeltetés irányításában. Az M2M kommunikáció révén megvalósulhat a fejőgépek hatékony távfelügyelete, távdiagnosztikája, valamint a számítógéppel támogatott táv-szaktanácsadás. Az M2M

kommunikációban ugyanis valósidejű adatok juthatnak el a központba a fejőgépek állapotáról, üzemeltetési jellemzőiről.

A tehenészet részéről kulcsfontosságú tényező a termelékenység, ezért a gépek, berendezések, illetve a rendszer hibamentes rendelkezésre állása kerül előtérbe. Ennek okán a tehenészet számára egyre jelentősebbé válik a szervizszolgáltató kínálata. Az a szervizszolgáltató, amely megfelelő alkatrész-, és tudásbázist, távfelügyeleti, távdiagnosztikai, táv-szaktanácsadási, távkarbantartási koncepciókat kínál partner tehenészetei számára, jelentős versenyelőnyhöz juthat és egyben a saját erőforrásait is óvhatja (VOJTELA, 2011).

A Józsefmajori Tangazdaság a számítógépes „telepirányítási rendszert” (ALPRO) a tejtermelés elindításával egyidejűleg szerezte be. A rendszer megvalósításához teljesíteni kellett az alapvető követelményeket: az állatok egyedi nyilvántartását, megfelelő, számítógépes szenzorokkal ellátott fejőház kiépítését, automatikus abrakadagoló berendezések üzembe állítását. Később, a rendszer fejlesztésekor további műszaki berendezések felszereléséről kellett gondoskodni, amelyek az állatok automatikus elkülönítését (válogató kapu), mérését (digitális állatmérleg) és aktivitásának megfigyelését (rádiójel-adók antennarendszerrel), az adatok átvitelét (belső hálózat) stb. tették lehetővé. A számítógépes szoftver testre szabásával és részben saját fejlesztésekkel (belső hálózat menedzsment, archiválási rendszer, a digitális tömegmérésekhez kidolgozott program stb.) alakult ki a számítógépes tejtermelésirányítás Józsefmajorban.

A számítógépes rendszerbe részben automatikusan (pl. állatazonosító szám, kifejt tejmennyiség, elfogyasztott abrakmennyiség), részben programozott vagy kézi adatbevitellel (pl. inszeminálási, ellési paraméterek, takarmányadagok előírása stb.) kerülnek be az adatok. Az adatbevitelt nagyrészt az állatgondozást ellátó dolgozók végzik. Az

ellenőrzés és irányítás, továbbá az szükséges programfrissítés és adatszervezés a gazdaság operatív vezetőjének, illetőleg a programot szállító cégnek a feladata.

A számítógépes rendszer többféle módon is támogatja az irányítási feladatokat. A telepírányítási programot magában tartalmazó processzor kijelzője a legfontosabb információk bevitelét és leolvasását teszi lehetővé. Az adatokat ugyanakkor átveszi egy kapcsolt számítógép is, amely MS Windows felületen, felhasználóbarát módon segíti a kommunikációt. A program a tejtermeléssel, az abrakfogyasztással, a tenyésztési, szaporodásbiológiai feladatokkal kapcsolatos, továbbá egyéb kiegészítő információk gyors áttekintését teszi lehetővé. Emellett az archivált adatokból különböző tenyésztési és termelési elemzések, értékelések is készíthetők.

A fejéssel és tejtermeléssel kapcsolatos adatok a fejőberendezésről kerülnek a számítógépes program által irányított adatbázisba. A képernyőn bármikor leolvashatók a lekérdezett tehénszámhoz tartozó fejési adatok (aktuális fejés, napi, heti, havi, laktációs és utolsó laktációs termelés adatai), továbbá a fejési sebesség, időtartam és további hasznos információk.

A számítógép a termelési adatokból listát is készít, amely képernyőn tanulmányozható, vagy kinyomtatható. E listák fontos segítséget jelentenek a tehenek közötti rangsor felállításában, a selejtezésben és a tenyésztésben, de ezek az adatok szolgálhatnak alapul az aktuális takarmányozási, állategészségügyi problémák feltárásában is. Külön ábrákon szemléltethetők az egyes tehenekhez tartozó laktációs görbék, amelyek a tejtermelés fontos tulajdonságait mutatják.

A tenyésztéssel, szaporodásbiológiával kapcsolatos feladatok szakszerű végrehajtása teszi lehetővé a szaporulat és ezen keresztül az állatállomány terveknek megfelelő alakulását, a tejtermelés gazdaságos megvalósítását, továbbá az állomány állandó genetikai javítását. Az

ehhez szükséges információk több számítógépes táblán érhetők el. Ennek középpontjában a tenyésztési táblázat áll, amely a tehenek legfontosabb tenyésztési paramétereit, továbbá mindenkori aktuális szaporodásbiológiai helyzetét szemlélteti.

Az állomány alakulását és termelését minden kétséget kizáróan a helyes időpontban történő termékenyítéssel lehet befolyásolni. A tehenállomány jelentős részénél ez az ivarzás külső jelekből is megállapítható, ugyanakkor mindig jelentős a „csendesen ivarzó” tehenek száma is. Ennél a csoportnál fokozott figyelemre van szükség, amelyet a telepírányítási program két szempontból is segít. Az egyik lehetőség az ivarzás időpontjának előrejelzése, amely statisztikai átlagokon alapul, amiről a korábban bemutatott figyelmeztető listák egyike ad tájékoztatást. A másik lehetőség a tehenek aktivitásának korszerű rádió-telemetriai eszközökkel történő megfigyelése.

A telepírányítási rendszer kifejlesztői a Józsefmajori Kísérleti- és Tangazdaság részvételével olyan állat aktivitásmérési rendszert dolgoztak ki, amely az egyedekre helyezett rádiójel-adó berendezésen és a tehenistállóban három egymástól távoli ponton elhelyezett antennákból, valamint az adatokat értékelő szoftverből áll. A berendezés emberi közbeavatkozás nélkül, folyamatosan jelzi az állatok helyváltoztatását.

A gazdaságban végzett mérések bebizonyították, hogy nem csak az istállóban, hanem a kifutóban, sőt, az istálló mellett fekvő legelőterületen, tehát a feltételezettnél nagyobb hatótávolságban is működőképes a rendszer. Az aktivitás méréshez először az állatokra jellemző, standard aktivitási görbéket kellett kialakítani, amelyhez viszonyított eltérések jelentik a nagyobb vagy a kisebb aktivitást. Az előző (az ivarzás előre jelzett időpontjával együtt) általában az ivarzás tényét mutatja, míg a kisebb aktivitásból betegségekre vagy a várható ellésre lehet következtetni. Az egyedi aktivitásmérési adatok

vizsgálatával fontos kiegészítő információkhoz lehet jutni a tehenek ivarzási időpontjának pontosabb meghatározásához.

A számítógépes rendszeren keresztül történik a termelési fázistól és szinttől függő egyedi takarmányadagok meghatározása és az abrakfogyasztás ellenőrzése is. A takarmányadag napi mennyiségét a telepírányítási rendszerben rendelkezésre álló számítógépes program határozza meg, ami különböző szakmai szempontok alapján felülbírálható.

A táblázatban szereplő előírt abrakfogyasztási adatok a tehenek azonosítási számaival együtt eljutnak az abrakadagoló automatákhoz, amelyeknél a tehenek naponta többször abraktakarmányhoz juthatnak. A teljes napi takarmányadagot tehát többszöri, tetszőleges időpontban fogyasztják el az állatok. Az abrakadagoló automata visszajelzést ad az elfogyasztott abrak mennyiségéről, így az addigi tényleges fogyasztás mindenkor megjelenik a számítógép képernyőjén.

A tervezett mennyiséggel és az előző napi fogyasztással történő összehasonlításból fontos következtetéseket lehet levonni az operatív irányítás számára és ennek alapján a szükséges intézkedéseket meg lehet tenni.

A számítógépes telepírányítási rendszerhez kapcsolódóan a gazdaság egy saját fejlesztés eredményeként lehetővé tette a tejlő tehenek folyamatos testtömeg mérését is, egy FVM K+F keretében.

A testtömeg mérés segítségével meg lehet állapítani a nagyhozamú tehenek „önfeláldozó” tejleadásának következményét: a kondíció esetleges leromlását, az élőtömeg jelentős csökkenését. A megfelelő kísérleti adatok birtokában takarmányozási intézkedéseket lehet kidolgozni, amelyek lehetővé teszik egyrészt a nagyhozamú tehenek képességeinek kihasználását, másrészt egyensúlyban tartják termelési és szaporodásbiológiai paramétereit is (SZÉKELY, 2004).

## **4.5. Vezetői döntéshozatal**

A vállalati hierarchia minden szintjén kulcsfontosságú az időben meghozott helyes döntés, amelynél a döntéshozó saját intuícióján, tapasztalatán kívül leginkább az általa elérhető információkra támaszkodik. Ezek az információk - a döntések inputjaiként - a vállalati működés fontos erőforrásai közé tartoznak. A vállalati gyakorlatban ugyanolyan nélkülözhetetlenek, mint a többi hagyományosan is vállalati erőforrásként kezelt tényező: a munkaerő, az eszközök vagy a pénztőke. Az információ azért is nélkülözhetetlen, mivel a többi erőforrást aktivizálni, koordinálni kell. Ehhez az szükséges, hogy a rájuk vonatkozó információk eljussanak a vállalat tulajdonosaihoz, működtetőihez (CHIKÁN, 2005; SÁNTÁNÉ et.al., 2008).

A döntéshozatal fejlesztése értelmes döntések meghozatalára szolgál, de az értelmesség definíciója gyakran nem pontos. A döntéshozatalt tanulmányozók gyakran ingadoznak az olyan definíciók között, amelyek a folyamat, illetve az eredmény oldaláról közelítik meg az értelmességet és nem képesek kielégítően feloldani azokat a nehézségeket sem, amelyek abból fakadnak, hogy a jó eredmény mögött bizonyos alkuk húzódnak meg (MARCH, 2000).

### **4.5.1. A döntés**

A döntés egy probléma, kielégítetlen állapot megoldására irányul. A döntéshozatal egy folyamat, mely során a probléma észlelése után több cselekvési alternatívát képzünk és azok közül választunk egy bizonyos célállapot elérése érdekében. A probléma megoldása akkor következik be, ha a döntéshozó számára a jelenlegi állapot megegyezik a kívánatos állapottal. A probléma észlelés szubjektív, mivel azonos helyzetben lévő emberek különbözőképpen észlelhetik az adott állapotot (ZOLTAYNÉ, 2005; SÁNTÁNÉ, 2008; ELSTER, 1986).

A döntéshozatal többféle helyzetértékelésen alapulhat.

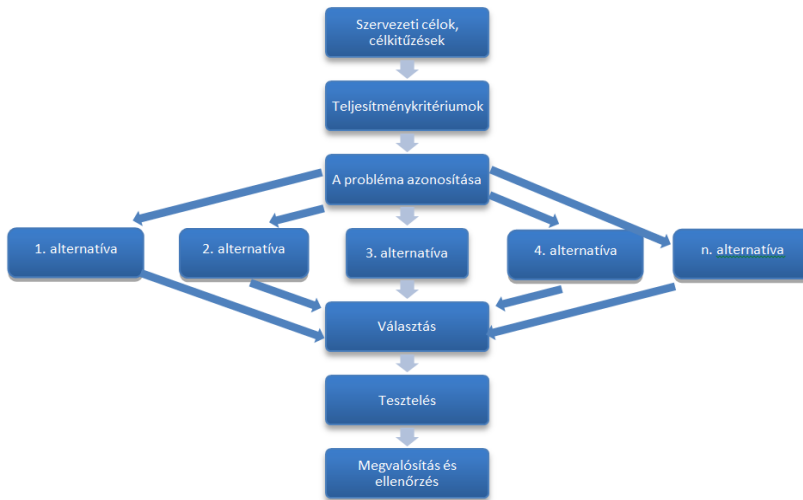


Optimális döntés esetében a döntéshozó teljesen informált, objektív, ismeri az egyes cselekvési változatok kimenetelét. Ebben az esetben érhető el a legjobb cselekvési változat kiválasztása. Herbert Simon 1950-ben tudományosan igazolta, hogy a valós életben az ember nem képes objektív döntést hozni, elménk korlátozott mértékű információ befogadására képes. Ez alapján bevezette a korlátozott racionalitás (principle of bounded rationality) elvét. „A komplex problémák megfogalmazásában és megoldásában a valóságos helyzetekben előforduló problémák nagyságához viszonyítva az emberi elme kapacitása nagyon kicsi ahhoz, hogy az objektív racionalitás alapján oldhassuk meg ezeket a problémákat” (cit. SIMON 1982; ZOLTAYNÉ, 2002).

A korlátozott racionalitás elvét továbbfejlesztve Herbert Simon kidolgozta a kielégítő döntés alapjait. Ilyenkor a döntéshozó nem az optimális állapotra törekszik, hanem megelégszik egy ahhoz közeli „kielégítő” megoldással (SIMON, 1983; ZOLTAYNÉ, 2002; SÁNTÁNÉ, 2008).

#### **4.5.2. A döntéshozatal szakaszai**

A döntéshozatal folyamata több, logikailag összefüggő lépésből áll. A menedzsment könyvekben többnyire a normatív modellen alapuló döntéshozatali fázisokat találjuk meg (RICHARDSON, 1992; ROÓZ, 2006; RUE, 1992). A jól strukturált problémák esetében alkalmazható a módszer. Az 4. ábra a modell lényegét mutatja, azt hogy ideális esetben hogyan kellene a döntéseket meghozni. Feltétel, hogy a döntéshozónak pontosan megfogalmazott céljai legyenek, vagyis tudnia kell mit akar elérni.



#### 4. ábra: A döntéshozás szakaszai, normatív modell

Saját szerkesztés Zoltayné (2002) alapján

A modellt olyan esetekben érdemes használni, amikor átlátható, egyszerű problémával állunk szemben.

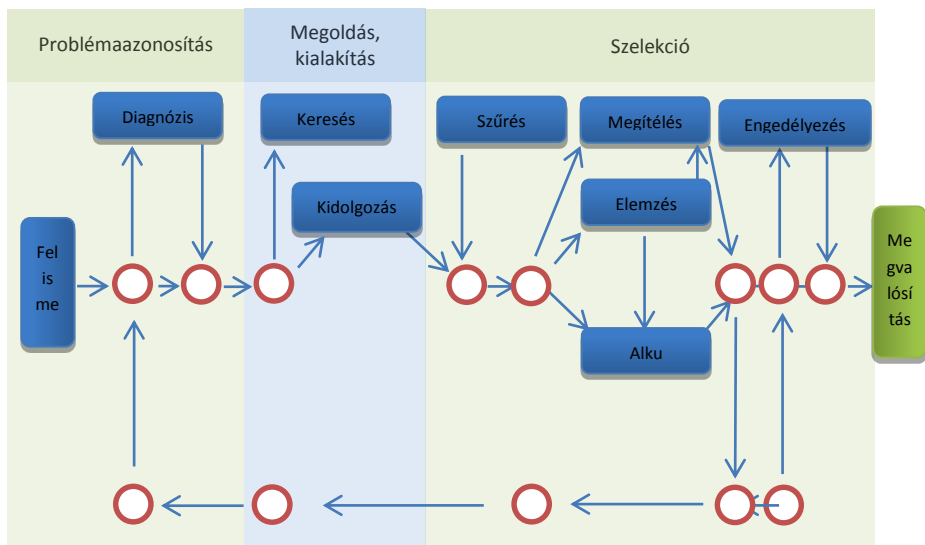
A szervezet céljaihoz igazítva teljesítmény-kritériumok készíthetők és lehetőség van ezen kritériumok rangsorolására. Ezek a kritériumok meghatározzák és egyben indikátorai is annak, hogy a kitűzött célok mennyire valósulnak meg. Méri, hogy a célként kitűzött állapot és a jelenlegi helyzet között van-e eltérés. Amennyiben ez az eltérés fennáll, azonosítva van a probléma.

A probléma meghatározását az alternatívák meghatározása követi, melyek közelebb vihetnek a megoldáshoz. Lehetséges akciók, megoldások számos módon nyerhetőek, például tapasztalat, mások véleménye vagy különböző kreatív technikák által. A cél megfelelő számú alternatíva képzése, ugyanis ezek közül ki kell választani egyet, amelyik a legjobban megfelel a kitűzött célnak.

A kitűzött célt ezután tesztelni kell, ha ez pozitívan zárult, akkor a tesztelést követi a megvalósítás, majd az ellenőrzés szakasza.

A modell valós környezetbe történő ültetésénél több probléma is felvetődik. A vezetők többnyire félig strukturált vagy strukturálatlan problémákkal találkoznak. A problémát sokszor nem a pontos célokkal paritásban lévő megoldás okozza, hanem az alternatívákkal felmerülő eltérő kockázatvállalás.

Mintzberg és munkatársai 25 stratégiai döntés empirikus vizsgálata alapján azonosították a döntési folyamat alapelemeit és készítették el a döntéshozás folyamatára vonatkozó modelljüket (5. ábra).



### 5. ábra: Döntéshozatal szakaszai, nem programozott döntési helyzetben

Saját szerkesztés Zoltayné (2002) alapján

Úgy találták, hogy a probléma felismerése és a megoldást jelentő akció kiválasztása között eltérő folyamatok zajlódhatnak le.

A folyamat több fázisból állhat.

*Első fázis a probléma azonosítása (felismerése).* Természetesen nem feltétlenül játszik szerepet minden lépés egy adott döntés esetében. A probléma felismerését követően történik meg a probléma pontosítása és megfogalmazása. Sokszor nem világosak a célok, nincs kész megoldási javaslat.

*A döntési folyamat második fázisa a megoldás kialakítása.* Abban az esetben, ha van az adott problémára kész javaslat, megoldás, akkor ez a lépés kihagyható. A megoldás kialakítása háromféleképpen mehet végbe. Megpróbálunk olyan alternatívát találni, amelyik kész megoldást jelent a problémánkra. Esetleg teljesen új megoldást dolgozunk ki vagy egy már meglévő megoldást tovább javítva jutunk el a probléma megoldásához. Ebben a fázisban van lekötve a szervezet legtöbb erőforrása. A feladatot a vállalat két tevékenységen keresztül végzi. Általában arra nincs esély, hogy az összes megoldást nyújtó alternatívát megtaláljuk, fontos hogy a keresési tevékenységet milyen területen és mennyi ideig végezzük.

Többnyire új megoldást az újszerű, egyedi, eddig fel nem merült probléma esetén kell keresni.

*A harmadik fázis a szelekció.* A döntési folyamat során sok esetben az ezt megelőző szakaszok több időt és erőforrást kötnek le, mint a szelekció. Továbbá az előző lépések sok esetben nagyobb jelentőségűek, mivel ha a probléma felismerése és diagnosztizálása során hibák lépnek fel, az befolyásolja az egész döntési folyamatot. Szűrésre abban az esetben kerül sor, ha túl sok alternatíva készült, emiatt nem tudjuk őket összehasonlítani.

*Negyedik fázis az értékelés - választás,* ami a döntési folyamat kihagyhatatlan szakasza. Az értékelés-választás háromféleképpen mehet végbe:

- *Megítélés:* Ilyenkor a döntéshozó az alternatívák mélyebb vizsgálata nélkül választ.

- Alku: Csoportos döntéshozás során kerülhet erre sor. Az érintett döntéshozók nem értenek egyet, konfliktushelyzetbe kerülnek.
- Elemzés: A racionális döntéshozó a céljainak megfelelően meghatározza és súlyozza a választás kritériumait, értékeli az alternatívák következményeit a kritériumok szerint és az alternatívák összehasonlítása után dönt.

Gyakori, hogy a döntéshozás itt nem zárul le, miután kiválasztásra került a megfelelő alternatíva, szükség van engedélyeztetésre, ami a döntési folyamat újabb szakaszát (fázisát) jelenti. A vállalat menedzsmentje által kidolgozott vagy kiválasztott stratégiai döntést a vállalat tulajdonosai engedélyezik (ZOLTAYNÉ, 2002).

#### **4.5.3. Döntések csoportosítása**

A döntések eredményei sokszor nem közvetlenül jelentkeznek, hanem közvetve, idővel válnak láthatóvá. A rövid távot a hosszú táv magában foglalja. Sok döntésnek, amely rövid távon előnyösen járul hozzá a megoldáshoz, hosszú távon negatív kihatása lehet, és fordítva. Sőt, a preferenciák és identitások az idővel változnak, részben a cselekvések miatt (MARCH, 2000). A mai szervezetekre nagy hatással van az őket körülvevő környezet, például a piac. Számos változásra szinte azonnal kell reagálni, ehhez gyors, logikai döntéseket kell hozni. A kielégítő, olykor optimális döntéshez elengedhetetlen, hogy a megfelelő mennyiségű és minőségű információ gyorsan álljon rendelkezésre. Ahhoz, hogy egy cég jövőjét kedvezően befolyásoló döntéseket tudjunk hozni, legtöbbször a múltban lezajlott történéseket kell elemezni és fel kell ismerni milyen lényeges irányzatok, tendenciák érvényesülnek (ADRIAANS et.al., 2002).

A döntések típusait a szakirodalom eltérő módon határozza meg. Egyik felosztás:

- strukturált,
- félig strukturált,
- nem strukturált.

A probléma típusa alapján, a döntéseket az alábbi csoportra bonthatjuk.

- jól strukturált
- rosszul, illetve félig strukturált (HERBERT, 1970; ZOLTAYNÉ, 2002; COOKE et.al., 1991)

Más szerzők kiegészítik az előző felsorolást egy újabb csoportal:

- Nem definiálható (JAMES et.al., 1992; CIT. TURBAN 2005).

Strukturált döntésként határozták meg azokat a kiszámítható, ismétlődő helyzeteket, ahol a megoldás módja jól ismert. Ilyenkor zárt problémát tételeznek fel. A kívánatos állapotot jellemző cél egyértelműen definiálható. A cél eléréséhez vezető utak ismertek, csupán választani kell közülük. A problémát nem a „mit”, inkább a „hogyan” eldöntése jellemzi. A jól strukturált problémák, a hozzá kapcsolódó döntések jól algoritmizálhatóak, jól programozhatóak (CIT. TURBAN 2005).

Rosszul strukturált döntés az a döntési helyzet, amelyben a döntéshozónak a probléma definiálásakor ítéletet, értékelést, feltételezést kell alkalmaznia. Ezzel nyílt problémák esetében találkozunk. Nem rutinszerű, azonban általában fontos döntések. Nem állnak rendelkezésre egyetemlegesen elfogadott technikák. Nem egyértelmű, hogy mit akarunk elérni (ZOLTAYNÉ, 2002; COOKE et.al., 1991).

Nem strukturált döntés esetében nem ismertek eléggé a kitűzhető állapotok, sem a műveletek. Szokás nem programozható problémának is nevezni (CIT. TURBAN 2005).

Turban megkülönbözteti a nem definiálható problémát, ami alatt a nem formalizálható problémateret érti, mivel túl nagy területet ölel át. Ezen problémák megoldására a jelenleg használatos módszerek nem alkalmasak (CIT. TURBAN 2005).

A nehézséget az okozza, hogy a gyakorlatban felmerülő kérdések, problémák többnyire nem strukturáltak, így nem alkalmazhatók a bevált eljárások.

Vezetői alapfeladat, hogy jó döntéseket kell hozni. Ezért egy jó vezető támogató információs rendszer közvetlenül a rosszul strukturált problémákat orvosolja, megfelelő algoritmusokkal és széles információs bázissal (DOBAY, 1997).

A döntések eltérő problématerületet ölelhetnek fel és más megközelítést igényelnek.

Szakirodalmak alapján megkülönböztetünk:

- operatív vezetői döntést,
- stratégiai vezetői döntést (EDVARD et.al., 1999; COOKE, 1991).

Egyes szerzők megkülönböztetnek

- taktikai vezetői döntést (KALLMAN et.al., 1983; LAUDON 2000; DONELLY 1992) is.

Egy operatív döntéshez valóban kell egy részletes - rendezett, formailag megfelelő, de mégis teljes - lista bizonyos tételekről. A döntéshozatal algoritmizálható, szabályai érthetőek, számítógépre programozhatók. Mindezekhez a döntéshozatalhoz szükséges az üzleti tevékenység operatív adatainak létrehozása, menedzselése. Operatív szinten az adatokat nem a vezetés céljára hozzuk létre, de a vezetéshez szükséges információk alapjául szolgálnak (EDVARD et.al., 1999).

Stratégiai döntéshez viszont magas fokon feldolgozott trendszámítási adatokra, grafikonokra van szükség. Az ilyen döntéseknél feldolgozott információk sokszor hiányosan definiáltak, többnyire nem ismétlődőek.

A hosszú távú, strukturálatlan döntések tartoznak ide (DOBAY, 1997; COOKE, 1991).

A vállalatokon belül különböző döntéshozói szinteket különböztetünk meg.

A döntéshozók tartós és kölcsönös kapcsolatban állnak a vállalkozással. Az úgynevezett stakeholder-elmélet a vállalat belső érintettjeit három csoportba sorolja (FREEMAN, 1984)

- tulajdonosok,
- menedzserek,
- alkalmazottak.

A felső vezetői döntések feladata a vállalat stratégiai céljainak, tevékenységének meghatározása. A feladatok nem rutin jellegűek, nem gépesíthetőek, viszont a számítógépes támogatás jelen van.

A középszintű vezetői döntések feladata a felső vezetők, tulajdonosok által kitűzött stratégiának megfelelően kialakított tevékenységek szervezése. A döntések időszakosak, de rendszeresen ismétlődnek.

Az alsószintű vezetői döntések az operatív irányítást, ellenőrzést ölelik fel. Rövid távú célok elérése érdekében allokálják a szervezet erőforrásait. A szükséges adatok és információk köre jellemzően jól definiált és jól strukturált (FREEMAN, 1984; SÁNTÁNÉ et.al., 2008).



## **4.6. Adat és információ**

### **4.6.1. Adat és információ értelmezése**

Az információ a rendszerek működésének, fenntartásának nélkülözhetetlen eleme. Az adatokból kiválasztással és/vagy feldolgozással származtatott értesülés, amit egy döntés meghozatalában, egy folyamat vagy cselekvés irányításában használunk fel. Lehetséges, hogy további műveletekhez inputként, mint adatot kezelünk és őrzünk meg. Az információ fogalmát a fentiek alapján az alábbiak szerint értelmezzük:

Az információ olyan tájékozódás, közlés, adat, ismeret, hír, amely a címzett által értelmezhető, és amelynek célja a bizonytalanság csökkentése, a lehetséges alternatívák közötti döntés elősegítése.

Az információnak különösen nagy jelentősége van a vezetők döntési tevékenységében (tervezés, szervezés, igazgatás, operatív irányítás), amelyhez az összegyűjtött, tárolt adatok szükséges csoportjának kiválasztására és feldolgozására épülő releváns információkra van szükség. A döntéshozatali információk a döntési folyamat fontos erőforrásai, amelyek az irányítási információrendszer alapját képezik.

Az adat olyan szimbólum vagy jelsorozat, amely esetleges későbbi felhasználás céljából, működő környezetben végbemenő változások, meglévő állapotok egyes jellemzőit továbbításra és megőrzésre alkalmas formában rögzíti (RAFFAI, 2003). A gazdasági szervezeteknek mindig szükségük van az információra, ez az információfüggőség működésük valamennyi területére kihat (BERDE et. al., 2003). Székely és munkatársai kutatásaik során az információt a hatékonyság szempontjából az ökológiai, humán és a tőkével egyenrangú, sajátos tulajdonságokkal bíró tényezőként definiálják (SZÉKELY et. al., 2008).

Az adatok információvá való konvertálása alapkövetelmény egy szervezet számára a helyes döntések meghozatalához (HANNON, 2005). Ugyanaz a nyers adat más céllal, más környezetbe helyezve természetesen eltérő információkat hordozhat. Az információs rendszereknek ez a transzformációs funkciója magába foglalja az adatok gyűjtését, tárolását, feldolgozását és továbbítását, mindezt jó esetben az információt hordozó adatok használójának, irányítási szempontból a döntéshozóknak az igényei szerint (CHIKÁN, 2005). Az információ minősége kiemelkedően fontos a döntéshozatal során. Szakirodalmi források alapján hat – információminőséget befolyásoló - kritériumot lehet meghatározni. Eszerint az információ legyen releváns, időszerű, pontos, ellenőrizhető, teljes és könnyen érthető (KACSUKNÉ, 2007).

#### **4.6.2. Az információ szerepe a döntéshozás folyamatában**

A vezetői döntések meghozatalához sok információ szükséges, a vezetőknek azonban feldolgozott információkra van szüksége. Az információnak időben kell rendelkezésre állni a szükséges döntések meghozatalához. A különböző szakterületeknél az ott dolgozók részére csak a fontos (és ne mindenféle) információk álljanak rendelkezésre. A döntéstámogató rendszerek alapvető erőforrásként adatokat, információkat használnak fel annak érdekében, hogy feladatainak végzéséhez hasznos információkat szolgáltatassanak. Fő céljuk tehát az információ-előállítás, vagyis olyan célorientált üzenetek létrehozása, amelyek a felhasználó számára újdonságot jelentenek, bizonytalanságot szüntetnek meg és feladataik, döntéseik teljesítésében segítséget nyújtanak (SÁNTÁNÉ et.al., 2008).

Az információ olyan mértékben szövi át a vállalkozások működését, hogy a vállalkozást információ-feldolgozó szervezetként is fel lehet fogni. Az információ kezelésére szakosodott alrendszer a vállalkozás

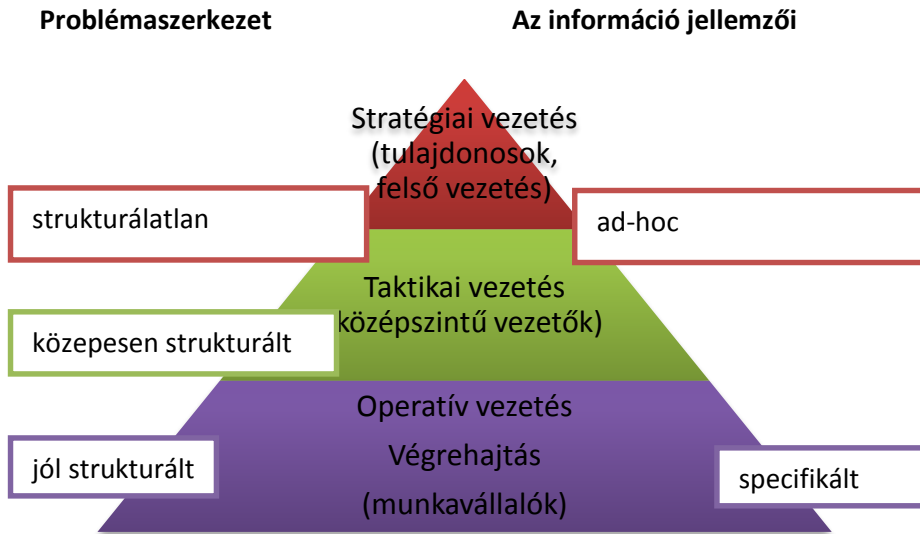
minden érintettjével (fogyasztók, szállítók, versenytársak, vállalati termékek, adatok a gazdaság egészéről stb.) Az összegyűjtött adatok a feldolgozás során alakulnak információvá, amikor is a feldolgozáson átesve érdemben használhatóvá válnak a döntéshozó számára, aki értelmezi az információt és ennek alapján döntéseket hoz (CHIKÁN, 2005). A mezőgazdaság átalakult rendszerében olyan információs rendszerek kialakítása a célravezető, amelyek alacsony költségek mellett lehetővé teszik a gazdasági tevékenységek széles körű áttekintését, a legfontosabb gazdasági folyamatok rögzítését, kiértékelését, a döntés-előkészítést és a tervezést (SZÉKELY et. al. 2008).

Az információkat négy, széles kiterjedésű tartomány keretei között lehet - legalábbis potenciálisan - felhasználni:

- eredményjelzőként;
- figyelemfelkeltőként;
- a rendszer struktúrájának és -dinamikájának visszatükrözőjeként;
- a rendszer mindenkori, aktuális állapotát jelző paraméterek meghatározójaként.

Egyetlen statisztikai adat vagy a statisztikai adatok halmaza különböző időszakokban mind a négy említett célra felhasználható. Másrészt, bizonyos típusú adatok különösen megfelelhetnek az ilyen alkalmazások valamelyikének (SIMON, 1977).

Sziray az információigényt szorosan szervezeti hierarchia szintjei mellett ábrázolja (6. ábra). Az eltérő szintek és problémátípusok más-más igényt támasztanak az információval szemben. Felső vezetői szinten, ahol strukturálatlan problémákkal találkozunk és annak megoldásával küzd a vezető (tulajdonos), jellemzően ad-hoc (alkalomszerű), összegzett (aggregált), előre tekintő, külső, széles átfogású információkra van szükség.



#### 6. ábra: Információ jellemzői az egyes vezetői szintek és problémaszerkezetek esetében

Forrás: Saját szerkesztés, SZIRAY (2006) alapján

Az operatív döntéseknél a munkavállalóknak előre specifikált, ütemezett, részletes, gyakori, hátra tekintő, belső, szűk területre szóló információkra van szükségük. A taktikai vezetés számára az információ jellemzői közbenső jelleget mutatnak a két szélső helyzetbeli jellemzők között (SZIRAY, 2006).

#### 4.7. Döntéstámogató rendszerek

A számítógépes információrendszerek egyik fontos válfaját alkotják a döntéstámogató rendszerek (Decision Support Systems DSS). A meghatározásuk nehéz feladat. A folyamatosan fejlődő hardware és software ipar számtalan új lehetőséget nyújtott és nyújt ma is. A technikai fejlődéssel párhuzamosan e tudományterület is folyamatos változáson, fejlődésen megy keresztül.

A számítógépes információrendszerek között kiemelkedőek a döntéstámogató rendszerek. Döntést kell hozni az élet számos területén, mely rövidebb vagy hosszabb távon kihat további életünkre, vezetőként pedig mások életére is. A döntéseket adatokra, információkra alapozzuk, amelyeket felhasználva választunk az egyes lehetőségek között (KŐ, 1997).

A hazai és nemzetközi szakirodalomban számos szerző rávilágított, hogy a számítógépes döntéstámogató rendszerek nagyjából egyidősek a személyi számítógépek, illetve a többfeladatos operációs rendszerek megjelenésével. A döntéstámogató rendszerek fejlődése az elmúlt 50 évben több elkülöníthető szakaszra bontható.

A kutatók már az 1960-as években kezdték tanulmányozni a számítógépes modellek által támogatott döntéshozatal és tervezés lehetőségeit (RAYMOND, 1966; URBAN, 1967).

Ebben az időszakban több disszertáció és tanulmány is született. 1971-ben elsőként Gorry és Scott Morton használták a döntéstámogató rendszer fogalmát. Azzal érveltek, hogy a vezetői információs rendszerek a strukturált döntésekre fókuszálnak és azt a javaslatot tették, hogy a félig strukturált és strukturálatlan döntéseket támogató információs rendszerekre a „döntéstámogató rendszer” kifejezést kell használni (GORRY, 1971; SCOTT, 1971).

A DSS fejlődése több szakaszra osztható, amit (SÁNTÁNÉ et.al., 2008) által készített táblázattal (6. táblázat) szemléltetnek. Más szerzőknél is hasonló felosztás található (KŐ, 1997; LOUIS, 1992; SZABÓ, 2001).

## 6. táblázat: A döntéstámogató rendszerek fejlődése

Idő	Rendszer	Tipikus funkciók
1960 -	Tranzakció feldolgozó rendszer (TPS, Transaction Processing System)	Tranzakciók feldolgozása időszakosan, illetve folytonosan.
1970 -	VIR, Vezetői információrendszer (MIS, Management Information Systems)	Jelentésgenerálás: on-line ellenőrzés, valós idejű lekérdezések
	Irodai rendszerek (OAS, Office Automation Systems) Integrált (intelligens irodai rendszerek)	Irodai tevékenységek teljes körű integrálása (berendezések, hálózatok, folyamatok stb.)
1980 -	DTR, Döntéstámogató rendszer (DSS, Decision Support System)	Szimulációs modellezés: on-line valós idejű tervezés, strukturált döntéshozatal automatizálása
	Csoportos döntéstámogató rendszer (GDSS, Group Decision Support System)	Általában a csoportmunka támogatása
	Ismeretalapú rendszer (KBS, Knowledge-Based System) Szakértő rendszer (ES: Expert System)	SZR: a szakma gyakorlatának explicit modellezése, komplex döntéseknél tanácsadás
	Neuronhálózatok (NN, Neural Networks)	Korábban nem ismert összefüggések feltárása
1990 -	FVIR, Felsővezetői információrendszer (EIS, Executive Information Systems)	Flexibilis funkciók, felhasználóbarát felület
	OLAP, On-line elemző feldolgozás (OLAP, On-Line Analytical Processing)	Codd 12 szabálya
	TMR, Tudásmenedzsment rendszer (KMS, Knowledge Management System)	A szervezeti tudásvagyon mint tőke dokumentálása, szétosztása, kiaknázása
2000 -	Üzleti intelligencia (BI, Business Intelligence)	A szervezet minden tagjához, minden döntési helyzetben eljutatják a releváns adatokat, információt, szervezeti tudást

Forrás: SÁNTÁNÉ et.al., 2008.

Az elmúlt évtized során mind a döntéstámogató eszközök, mind az informatikai környezet hatalmas fejlődésen mentek keresztül. A PC-k

tekintetében elterjedtek az erősödő processzorok, a növekvő memória és az egyre nagyobb háttértárak. A helyi hálózatok nemcsak a felső vezetők, hanem a középvezetők és több alsóbb szinteken dolgozó munkatársak számára is megfelelő erőforrásokat kínálnak.

A jelenleg használt szoftverek mindegyike besorolható az alábbi csoport valamelyikébe:

*Tranzakciófeldolgozó rendszer* (Transaction Processing System, TPS) egy adott szervezet rutinszerűen üzemeltetett, ismétlődő, alapvető tranzakcióit dolgozza fel. (Tranzakción olyan akciót értünk, amely bizonyos műveleteket hajt végre egy vagy több erőforráson – ilyen műveletek például olvasás, írás, törlés, módosítás egy adatbázisban.)

*Vezetői információs rendszer*, VIR (Management Information System, MIS). Ezek olyan üzleti információrendszerek, amelyek egy szervezet számára információt képesek nyújtani annak tervező-, szervező- és kontrolling tevékenységeinek múltbeli, jelenlegi és jövőbeli helyzetéről. Egy VIR az adott szervezet egyes funkcionális területeinek (például gyártás, könyvelés, pénzügy) megfelelő alrendszerekből áll, amely alrendszerek tranzakciófeldolgozó, működésellenőrző, menedzsmentellenőrző és stratégiai tervező részekből tevődik össze.

*Irodai rendszer* (Office Automation Systems, OAS). Ezek az irodai tevékenységek teljes körű ellátását nyújtó, a dokumentum feldolgozást és archiválást, a munkafolyamat-szabályozást, a csoportmunka és a kommunikáció támogatását biztosító, integrált rendszerek. A későbbiekben ezeket felváltották az ún. integrált irodai rendszerek.

Az 1980-as évekre tehető a „valódi” *döntéstámogató rendszerek*, DTR-ek (Decision Support Systems, DSS) megjelenése. Csak támogatják, de nem helyettesítik a vezetőket a döntéshozatal során. Általában

féligstruktúrált problémák megoldásában segítenek, ahol emberi megfontolásra, megítélésre is szükség van.

*Csoportos döntéstámogató rendszer*, (Group Decision Support systems, GDSS). Ezek a döntéshozók együtt dolgozó csoportjának támogatása céljából olyan specifikus igényeket is ki tudnak elégíteni, mint az adatok és az információk szétküldése a csoporttagoknak, szavazások lebonyolítása, grafikus összegzése, ötletek gyűjtése.

*Az ismeretalapú rendszerek* (Knowledge-Based System, KBS) a tárgyterületi szakértő heurisztikus ismereteinek felhasználásával következtetéseket tudnak levonni, sőt következtetéseikhez magyarázatot is képesek fűzni. A program egyéb komponenseitől elkülönített ismeretbázisban elhelyezett szakterületi ismeretek feladatmegoldásra történő aktivizálását e rendszerekben egy ún. következtető gép irányítja.

*Szakértői rendszerek* SZR-ek (Expert Systems, ES) a tárgyköri szakértő ismereteinek felhasználásával magas szintű teljesítményt nyújtanak egy szűk problémakör kezelésében. Olyan tudásintenzív problémák esetében lehet szakértő rendszereket kidolgozni, amelyekhez nem lehet normatív modellt adni, és amelyeknél a szakismereteket egy ember több éves tanulással és gyakorlással tudja csak elsajátítani. A szakértői rendszerekkel megoldható problémák tehát olyanok, hogy valaki, pontosabban a szakértő tudja, hogyan lehetne azokat megoldani, és a szakértő ismereteit tartalmazó rendszer által kidolgozott megoldást a döntéshozó konzultációs segítségként veheti igénybe.



*Neuronhálózatok* (Neuron Networks, NN) az emberi agy működését modellező, egyszerű processzorokból (mint mesterséges neuronokból) felépülő, párhuzamos működésű architektúrák. Egyszerű átmeneti függvényt megvalósító processzoraik változtatható súlytényezőkkel ellátott összeköttetések hálózatán keresztül kommunikálnak.

*Felsővezetői információrendszereket* FVIR-eket (Executive Information System, EIS). Ezek a felsőszintű és további vezetők számára felhasználóbarát módon segítik a szervezet működésének áttekintését – akár alkalmi igények kielégítését is biztosítva. Egy új adattárolási és kezelési megoldást nyújtó adattárház biztosítja a lekérdezések rugalmas kielégítését, a több szempontú elemzést.

A rendszer nagyon könnyen kezelhető, betanulást gyakorlatilag nem igényel, szemléletesen mutatja be a kritikus tényezőket, a várható tendenciákat. Az EIS adatigényét a TPS rendszerek szolgáltatják, folyamatosan figyeli és kiszűri a szervezeti működés szempontjából kritikus információkat.

Gyorsan áttekinthető, egyszerűen kezelhető és segíti a vezetőket a számukra lényeges kérdések megválaszolásában. Első lépésként ezeket a rendszereket érdemes használni olyan helyeken, ahol még az informatika használata elsősorban a tranzakció feldolgozásban merül ki és ezt szeretnék fejleszteni.

Nagy előrelépést jelentene, ha a vezetői döntéshozatal segítségével a tervadatok különböző változatai megadása után olyan terv-tény elemzést lehetne végezni, ami a változó körülményekhez való alkalmazkodását segíti a vállalatnak. Kutatásaim ered

*On-line elemző feldolgozó rendszerei* (On-Line Analytical Processing, OLAP) olyan felsővezetői információrendszerek, amelyek számos új szolgáltatással bővültek, és az adattárházak által nyújtott lehetőségek teljes kiaknázását biztosítják.

*Üzleti intelligencia rendszerek* (Business Intelligence Systems, BI Systems). Az üzleti intelligencia összefoglalja mindazon fogalmakat és módszereket, amelyek célja, hogy feljavítsák a tényalapú rendszerek fölött működő üzleti döntéshozatalt – vagyis hogy az üzleti kérdésekre minden rendelkezésre álló adat, információ felhasználásával, minél értelmesebb válaszokat lehessen adni (RADFORD, 1978; FRADA, 2008; ÖZGÜR, 1998; DAVID, 2009; KERR, 1999; RAFFAI, 2006; JANAKIRAMAN, 2006; SZŰCS, 2002; JUHÁSZ; 2011 RAFFAI, 2006).

## 5. ANYAG ÉS MÓDSZER

---

A disszertációban feldolgozott primer adatok Magyarország teljes területéről származnak. A vizsgálatok során **a minta kiválasztása azokra az önálló jogi személyiséggel rendelkező cégekre terjedt ki, melyek 50 főnél több alkalmazottat foglalkoztattak**, továbbá a cégbírósági bejegyzésben a tevékenységi körök között szerepelt a 0141-es TEAOR kód, mely alatt a "Tejhasznú szarvasmarha tenyésztése" tevékenységet értjük. E szakágazatba tartozik:

- a tejhasznú szarvasmarha és bivaly tenyésztése,
- a nyerstej termelése.

A minta kialakítása a Complex Céginfó adatbázisát alapul véve, rétegzett mintavétel történt.

A felmérés részét képezte az ÚMFT keretében meghirdetett **GOP-1.1.1-09/1-2009-0007-es K+F pályázat**nak, amelyben a korábban megfogalmazott célkitűzésekben a Magyarországon működő nyerstej termeléssel és tejhasznú szarvasmarha tenyésztéssel foglalkozó **cégek felmérése szerepel információ-technológiai szemszögből, kiemelten a termeléssel összefüggő vezetői döntéshozás és tervezés támogatottságára**. (1. melléklet) A téma komplexitása a primer kutatás módszereként a kvalitatív terepkutatást indokolta, melynek megoldásaként **mélyinterjúk felmérésekre került sor**.

Az interjút megelőzte egy interjúvázlat kialakítása, ami a projekt célkitűzéseire igazodott (2. melléklet).

## 5.1. Mélyinterjú kutatás

A **vizsgálható minta** kialakítása után e-mail és telefon segítségével történt a kapcsolatfelvétel a mintában szereplő vállalatokkal. Egyeztetés után minden olyan céget sikerült személyesen felkeresni, amely jelezte részvételi szándékát. Ez **összesen 20 céget jelent**. Az interjúalanyok száma vállalatonként eltérő. A cél az volt, hogy minél több terület szakemberét sikerüljön bevonni, a problémákat a vállalati hierarchia több eltérő szintjéről megközelíteni. Minden alkalommal **az ügyvezetés, a könyvelés, a növénytermesztés, az állattenyésztés és az informatika területén kompetens személyek megkérdezésére is sor került**.

**Az interjúk egy előre kidolgozott interjúvázlat (2. melléklet) témái mentén haladtak**, erről hangfelvételek készültek a későbbi elemzés megalapozása céljából.

Az interjúba bevont 20 cég közül 15-nek az információi kerültek be a végső kiértékelésbe. Előfordult, hogy időközben felhagytak a tejtermeléssel, csupán a cégadatbázisban szerepelt a tevékenység.

A 2. mellékletben látható interjúvázlat témacsoportjai és azok altémái szerint felosztva került sor az elemzésekre. Ebben segítséget jelentettek az interjú során készült jegyzetek, valamint a rögzített hanganyagok. A disszertáció nem tartalmazza a teljes kiértékelést, csupán a témához közvetlenül kapcsolódó területek eredményeit.

Az egyes interjúk összehasonlítása kulcsszavak egységesítésével és kódolásával történt. Ezen **adatok kiértékelésére táblázatkezelő programmal, valamint SPSS v.19-es statisztikai szoftver segítségével került sor**.

## 5.2. Szekunder kutatás a vizsgálatba vont vállalatokról

A vizsgálandó minta kialakítása a fejezet elején leírt feltételek szerint történt. A minta bemutatásához a Complex Céginfó mérlegadatai szolgáltak. Az **adatok általános statisztikai elemzése az MS Office Excel program segítségével történt.**

## 5.3. Szoftverfejlesztés

**A kutatások egyik célja a vezetői döntéshozással kapcsolatban felmerülő informatikai hiányok, problémák feltárása és ezek kiküszöböléséhez olyan megoldás bemutatása, amely az érintettek számára elérhető.** Ezért - gazdasági informatikusként végzett tanulmányaimat felhasználva - a kor követelményeinek megfelelő programozói technológiák segítségével szoftvert fejlesztettem. A szoftver forráskódját nem teszem zárttá, a dolgozatomban mellékletként szerepeltetem. Ezzel szeretnék hozzájárulni a magyarországi mezőgazdasági vállalatok döntéseinek megalapozásához. Mivel szabadon használható, másolható, terjeszthető és módosítható, így specializáltabb vállalati igényeket is ki tud szolgálni.

A fejlesztés 4 területet érint:

- tervezés,
- adatbázis kialakítása,
- programírás,
- a tesztelés, hibák javítása.

A **tervezés** a szoftverfejlesztésben a világon legelterjedtebb és szabványosított UML 2 (Unified Modeling Language) programnyelv használatával történt. Az UML jelentése Egységes Modellező Nyelv, mely **egy általános célú vizuális modellező nyelv és arra**

**használható, hogy specifikáljuk, szemléltessük, megtervezzük és dokumentáljuk egy szoftver architektúráját.** A kiinduláskor előállított úgynevezett tervezési diagramok lényegesen különböznek az implementáláskor szöveges formában előállított programozási forráskódtól.

Ez a technológia **nem más, mint egy tervezési nyelv, ami egy szoftver rendszer minél megalapozottabb kidolgozásának, elkészítésének folyamatát szolgálja.** Az így elkészült tervezési diagramok alapján válik lehetővé a forráskód megírása és a futtatható szoftver elkészítése, ami a fejlesztési folyamat végső célja.

Az UML nyelv támogatja a modern objektum-orientált programozási nyelvek szemléletmódját, tervezési filozófiáját.

**Adatbázis-struktúra kialakításánál** szempont volt a bővíthetőség, a többfelhasználós lehetőség, valamint az, hogy elterjedt technológia legyen. A szempontrendszer által előállt lehetőségeket korlátozta a rendelkezésre álló ismeretanyag és tapasztalat. Így esett a **választás a Microsoft által fejlesztett MS-SQL relációs adatbáziskezelő szerverre.** Ahhoz, hogy a program eltérő relációs adatbázis kezelőkkel is együtt tudjon dolgozni, ezért került beiktatásra a LINQ elnevezésű nyelv, amely egy egységes programozási modell, bármilyen adatforrásra nézve. A LINQ **lehetőséget biztosít arra, hogy egységes módon kérdezzük le és módosítsuk az adatokat,** függetlenül az adatbázistól. Egy új adat absztrakciós réteg a tényleges adatok és az adatokkal dolgozó szoftver között.

Adatbázisszerverként az MS-SQL Server Express szolgált.

**A programíráskor** a programozási nyelvet tekintve szintén **az objektumorientált szemlélet került előtérbe, az alkalmazott C# nyelv választására,** amely a Microsoft által a .NET keretrendszer részeként kifejlesztett objektumorientált programozási nyelv. Fejlesztéséhez alapot szolgáltatottak a C++ és a Java nyelvek.

A szoftverfejlesztő környezetként a Microsoft Visual Studio 2010-et használtam.

**A programírás három szakaszban történt: 1. csatlakozás az adatbázissal, 2. üzleti logika leprogramozása, 3. felhasználói felület elkészítése a funkciók kialakítása során.**

**Ezt követte a kész szoftver tesztelése valós körülmények között, valós felhasználókkal, majd az "éles" rendszer telepítése, beüzemelése.**

## 6. EREDMÉNYEK ÉS ÉRTÉKELÉSÜK

### 6.1. Mélyinterjúk kutatás bevont vállalkozások főbb jellemzői

Táblázat készült a KSH adatai alapján a TEAOR 0141-es ("Tejhasznú szarvasmarha tenyésztése") tevékenységet folytató cégekről. Ez a táblázat kiegészült az interjúba bevont vállalatok számával, az egyes osztályközökben– melyek az alkalmazotti létszámok alapján kerültek felosztásra - feltüntetésre kerültek az interjúba bevont vállalatok száma. (7. táblázat)

#### 7. táblázat: TEAOR-0141 tevékenységet végző vállalkozások száma (2008-2010)

Me.: vállalkozások száma

	1-4 fő	5-9 fő	10-19 fő	20-49 fő	50-249 fő	250- fő	Összesen
2008	314	53	57	111	78	6	619
2009	227	40	50	112	79	6	514
2010	169	38	44	107 (100 %)	75 (100 %)	4 (100 %)	437
Vizsgálatba bevont cégek				1	13	1	15
Vizsgálatba bevont cégek	0%	0%	0%	1%	17%	25%	

Forrás: KSH, saját szerkesztés

A 7. táblázatban 2008-tól 2010-ig szerepelnek azok a cégek, melyek a vizsgálatok tárgyát is képező tevékenységet végezték. A 2010-es évben, amikor a mélyinterjúk vizsgálata zajlott, összesen 437 ilyen vállalat volt. A táblázat 4. sorában szerepelnek az interjúba bevont cégek, alatta a



bevont cégek számának aránya a maximálisan bevonható cégek számához.

A KSH által közölt létszám-kategóriánkénti megoszlás alapján 13 vállalat esetében az alkalmazotti létszám az 50 és 249 létszám kategóriába esett, ami ebben a kategóriában a Magyarországon bejegyzett tejhasznú szarvasmarha tenyésztéssel foglalkozó cégek 17%-át adja.

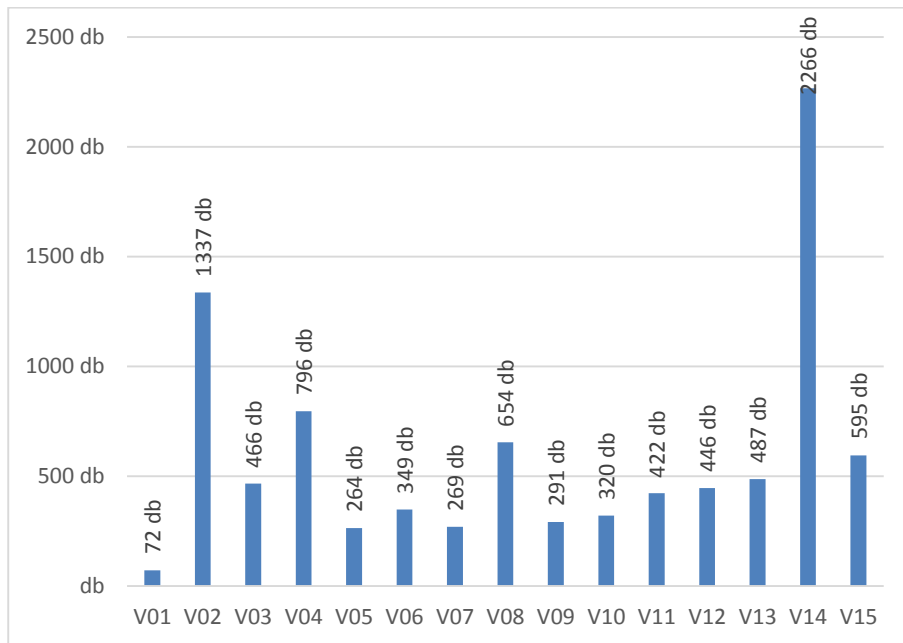
Egy vállalat esetében fordult elő a 369 fős alkalmazotti létszám, ami 25%-át adja a sokaságnak.

Ez az elemszám biztosítani tudja a vizsgálatok reprezentativitását.

Az alábbi felsorolás tartalmazza név szerint a vállalkozásokat és a vizsgálat további részében használt kódot:

- V01: GAK Nonprofit Közhasznú Kft.
- V02: Tedej Zrt.
- V03: Solum Zrt.
- V04: Bicskei Mezőgazdasági Termelő és Szolgáltató Zrt.
- V05: Claessens Kft.
- V06: Duna Gyöngye 2000 Mezőgazdasági Zrt.
- V07: Felsőánai Agrár Kft.
- V08: Középtiszai Mezőgazdasági Zrt.
- V09: Hajdúböszörményi Mezőgazdasági Zrt.
- V10: Jászapáti 2000 Mezőgazdasági Zrt.
- V11: Pély-Tiszatáj Agrár Zrt.
- V12: Petőfi Mezőgazdasági Szövetkezet
- V13: Canko 2000 Kft.
- V14: Agroprodukt Zrt.
- V15: Gödöllői Tangazdaság Zrt.

Az interjú során felmérésre kerültek a vállalkozásnál fejt tehénlétszám adatok. Az oszlopdiagram alján szereplő kódok (V01-V15) a vizsgált tehenészetek kódszámjai.



### 7. ábra: Fejt tehénlétszám adatok

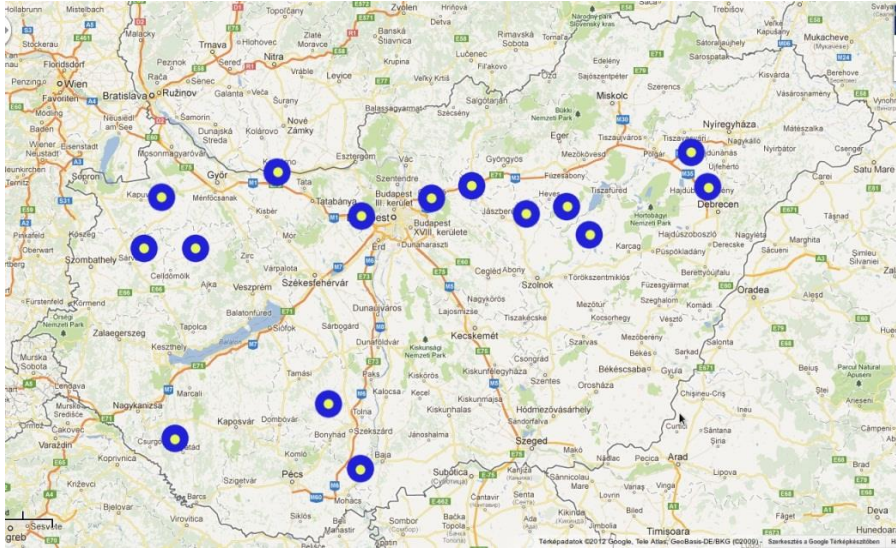
Forrás: Saját vizsgálat

A 7. ábra jól szemlélteti, 2010-ben az általunk vizsgált szarvasmarha telepek fejt tehén állományát, kiemelendő a V14-es vállalat, ahol több mint 2200 db tehenet fejtek átlagosan.

A válaszadók megismerése érdekében bemutatásra kerül:

- a területi elhelyezkedés,
- a vállalkozási forma,
- az árbevétel 2008-2011-es időszakban,
- az eszközérték 2008-2011-es időszakban,
- a megkérdezett személyek munkaköre.

Összesen 15 vállalatot sikerült a mélyinterjúba bevinni, ezek országos elhelyezkedését a 8. ábra mutatja.

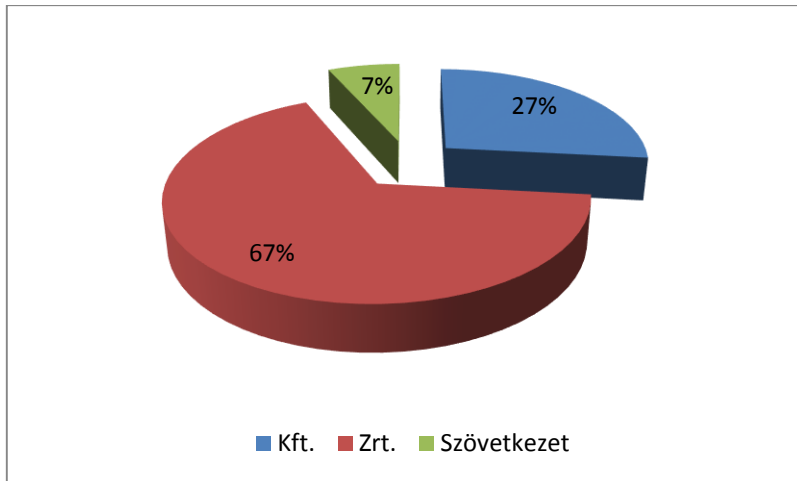


### 8. ábra: A vizsgált minta területi elhelyezkedése

Forrás: Saját vizsgálat

Ahogy az a térképen látható, a felmérés összesen 12 megyét és 6 régiót érintett. Nagyon eltérő adottságú és különböző méretű vállalkozások vettek részt a felmérésben.

A 9. ábra a vizsgált gazdaságok vállalászási formák szerinti megoszlását mutatja.



### 9. ábra: Vizsgált minta megoszlása vállalkozási forma szerint

Forrás: Saját vizsgálat

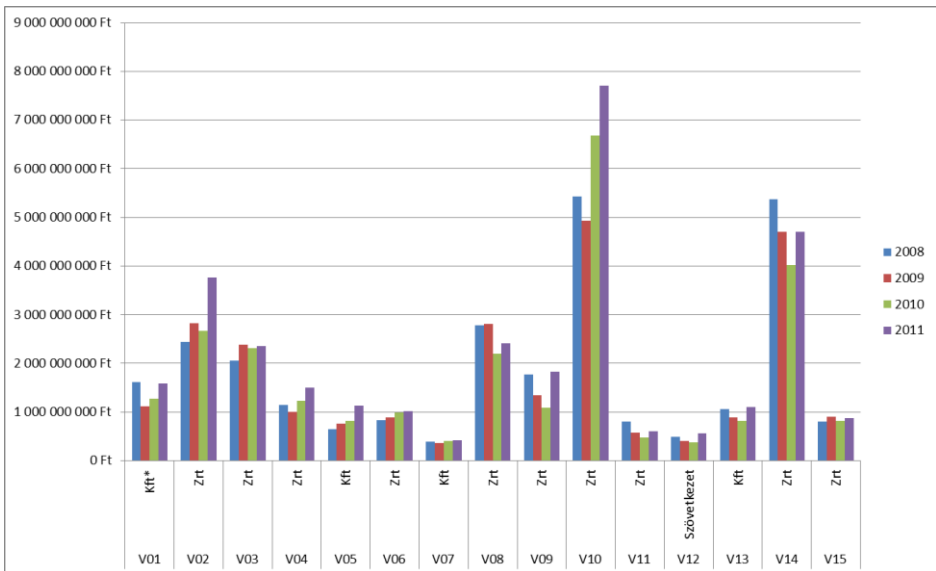
A Zrt.-k magas aránya a minta kialakításánál figyelembe vett paramétereknek a következménye.

A 10. ábra bemutatja a felmérésbe vont vállalkozások árbevételét a 2008-tól 2011-ig terjedő időszakban. A 2010-es évben – a mélyinterjú időszakában- 7 vállalat rendelkezett nettó 1 Mrd. forintnál kisebb árbevétellel, 3 vállalat árbevétele az 1-2 Mrd. közé esett, 4 vállalat árbevétele 2-5 Mrd. között helyezkedett el, 1 esetben az árbevétel meghaladta a 6 Mrd. forintot.

Az árbevétel változása 2008-tól 7 esetben kismértékű (átlagosan évi 10 százaléknál nagyobb) csökkenést mutat. Ez a 7 vállalat számolt be visszaesésről, 8 esetben árbevétel – növekedés történt.

A vállalatok eszközállomány értékének átlagos változását tekintve 4 cég esetében történt csökkenés a 2008-2011-es időszakban, mind a 4 esetben a csökkenés mértéke kisebb volt 10 százaléknál. 11 vállalkozásnál volt eszközérték növekedés, 10 cégnél ez a növekedés meghaladta a 10 százalékot.

Az eszközértékben csakúgy, mint az árbevétel esetében tág intervallum alakult ki a mintán belül. Félmilliárdos alsó határtól több mint tízmilliárdos felső határig oszlott meg a minta, 87%-ban nem érte el az ötmilliárdot.



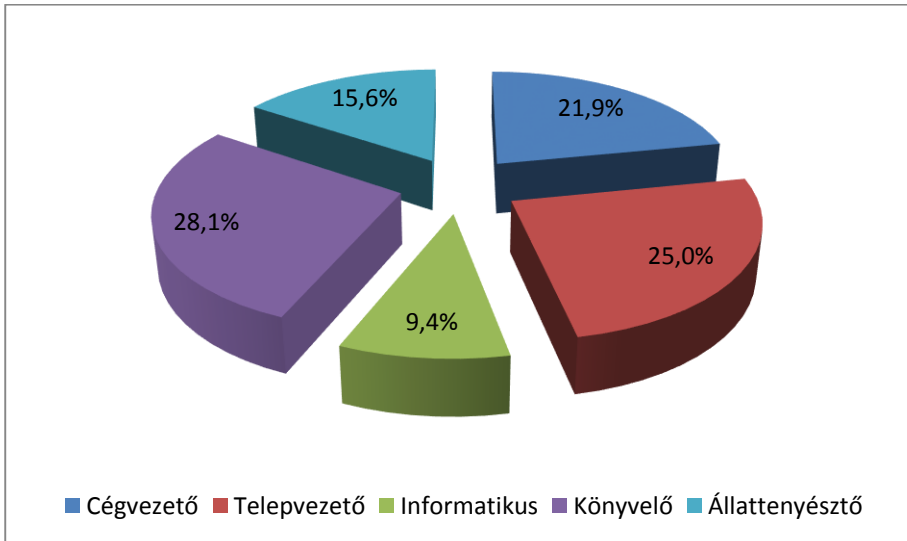
## 10. ábra: Az árbevétel alakulása 2008-2011 között vizsgált vállalkozásonként

Forrás: Saját vizsgálat

A 15 cégnél készített interjúkban összesen 32 személy vett részt. Mivel a problémák több irányból való megközelítése volt a cél, ezért az interjúban több terület szakemberének a megkérdezésére törekedtünk. Az interjúba bevont személyek munkaköri megoszlását a 11. ábra szemlélteti, ami azt mutatja, hogy az ügyvezetés, a könyvelés, a

\*2 Nonprofit Közhasznú Kft.

növénytermesztés, az állattenyésztés és informatika területéről egyaránt részt vettek a felmérésben.



### 11. ábra: A válaszadók munkaköri megoszlása

Forrás: Saját vizsgálat

A válaszadók megoszlásával kapcsolatos alapvető célkitűzés sikerült, egy szakterület sem képviseli magát dominánsan, kellően széles bázisú és sokrétű munkaköri véleményeket sikerült rögzíteni.

A vezetés két szintjét, a cégvezetőket (21,9%) és telepvezetőket (25,0%) külön kezeltük, mivel a vezetés eltérő szintjei eltérő információkat és döntéseket képviselnek.

### 6.2. Mélyinterjú kiértékelése

Az interjúk után esszé formában összefoglaló készült a válaszokról. Ehhez segítséget nyújtottak az interjú során készült írásos feljegyzések valamint a hangfelvételek.

Az interjú során feltüntetésre került:

- a vállalat neve,
- az interjú helyszíne,
- az interjúkészítés ideje,
- a résztvevők,
- a témakörönként a kérdésekre adott válaszok.

### 8. táblázat: A gazdaságokban alkalmazott szoftverek

Vállalkozás	MS Excel	MS Access	Riska	Talp	Boumatic/Alpro/Westfal ia
V01	Igen		Igen	Igen	Igen
V02			Igen		Igen
V03			Igen		Igen
V04			Igen		Igen
V05			Igen		Igen
V06			Igen		Igen
V07					
V08			Igen		
V09	Igen		Igen		
V10			Igen		Igen
V11	Igen		Igen		
V12		Igen			Igen
V13			Igen		Igen
V14			Igen		Igen
V15	Igen		Igen		Igen
<b>Összesen:</b>	<b>4</b>	<b>1</b>	<b>13</b>	<b>1</b>	<b>11</b>

Forrás: Saját vizsgálat

A 15 tejtermelő cégnél előforduló szoftvereket a 8. táblázat tartalmazza. A kérdés elsődlegesen az adott szoftver meglétére irányult, a szoftverek alkalmazási lehetőségeiről, gyakoriságáról és fontosabb jellemzőiről ezt követően lesz szó.

Alkalmazott szoftverek áttekintését tekintve egy esetben, a V01 vállalkozás esetében az Állattenyésztési Teljesítményvizsgáló Kft. által fejlesztett TALP (Telepi Adatgyűjtő és Lekérdező Program) programot használják, egyelőre tesztelés céljából, más funkcionalitásokkal rendelkező szoftverek mellett.

Négy gazdaságban használnak kiegészítésként az állattörzsadatok és állományváltozás kezelésére Microsoft Excel táblázatkezelő programot.

A megkérdezettek közül az egyetlen szövetkezet esetében Microsoft Access adatbáziskezelő programmal dolgoznak, ami a Boumatic fejőházi rendszerrel van kapcsolatban, a termelési adatok importálása megoldott. A fejlesztést a szövetkezet állattenyésztője végezte, teljesen a saját igényeikhez igazítva. Az adatokat két számítógépről, hálózatba kötve is el tudják érni, a megkérdezéskor 28 féle lekérdezést tudtak vele készíteni. Alkalmas állat-törzsadatok, tenyésztési adatok, termelési adatok, állategészségügyi kezelések, gyógyszerek nyilvántartására.

Fejőházi rendszer számítógépes adatgyűjtő moduljával 11 gazdaságban dolgoznak, egy-egy esetben Alpro, valamint Westfalia, továbbá 9 esetben a Boumatic fejőházi rendszer állt rendelkezésre.

A vizsgált 15 vállalkozás közül 13-ban alkalmazzák a Systo Kft. által szolgáltatott Riska szoftvert.

A fellelhető szoftverek három kategóriára sorolhatók. Az első kategóriába az egyszerű, irodai programcsomag részeként meglévő táblázat és adatbázis-kezelők, a második kategóriába a fejőházi rendszerrel érkező telepirányító, nyilvántartó szoftverek (Bou-Matic, Alpro, Westfalia), harmadik kategóriába a fejőházi rendszerek adatait



felhasználó (TALP, Riska) szoftverek tartoznak. A cégméret nem befolyásolja a telepírányítási rendszerek alkalmazását az 50 főnél többet foglalkoztató vállalkozások esetében.

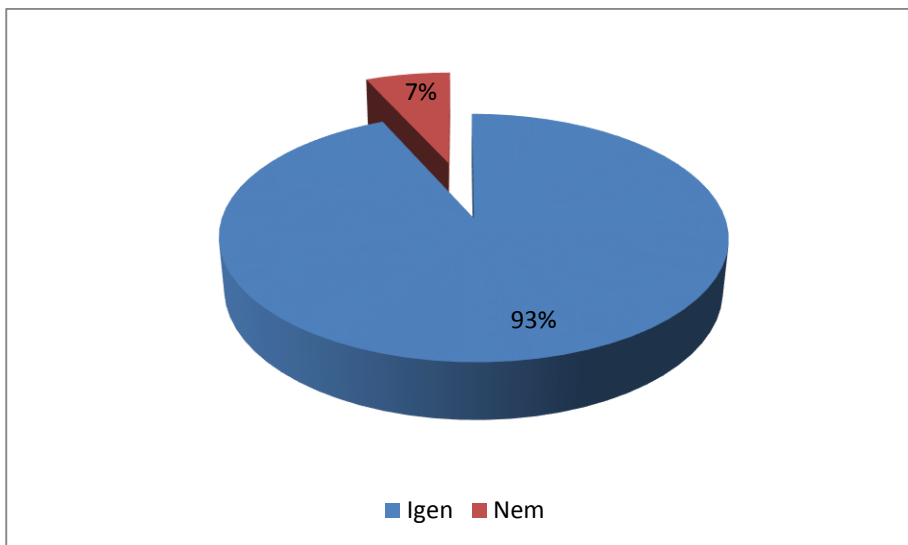
A fejőházi rendszerek kategóriába tartozó három alkalmazott szoftver (Alpro, Westfalia, Bou-Matic fejőházi rendszerek) fordult elő a vizsgált cégeknél. E rendszerek jellemzője, hogy adatgyűjtő processzonnal rendelkeznek. A Bou-Matic esetében redundáns működés kialakítása is megoldott. Működtetéséhez személyi számítógépre nincs szükség, viszont az adatok elemzéséhez ajánlott PC-n futtatható gyártó által készített telepírányító szoftver. Utóbbi szoftver azonban csak egy esetben fordult elő. A gyakorlat azt mutatja, hogy külön telepírányítási alkalmazással rendszerbe exportálják az adatokat, a nyilvántartás és elemzés ott történik.

Telepírányító szoftverek közé tartoznak a Riska és TALP rendszerek. Képesek többek között nyilvántartani:

- az állat törzsadatokat,
- a tenyésztési adatokat,
- a termelési adatokat,
- az állategészségügyi adatokat,
- külön modulként tőgyegészségügyi adatokat, egyedenként,
- a gyógyszer készletváltozást,
- az ENAR adatokat,
- a termékenyítő anyagokat,
- a takarmányozási naplót.

Egy eset kivételével – ahol manuális adatbevitel történik - interfészen keresztül kialakított kapcsolatot használtak a fejőházakkal.

A mélyinterjú során vizsgáltuk a termelési- és gazdálkodási informatikai rendszerek kapcsolatát, azok esetleges hiányosságait.



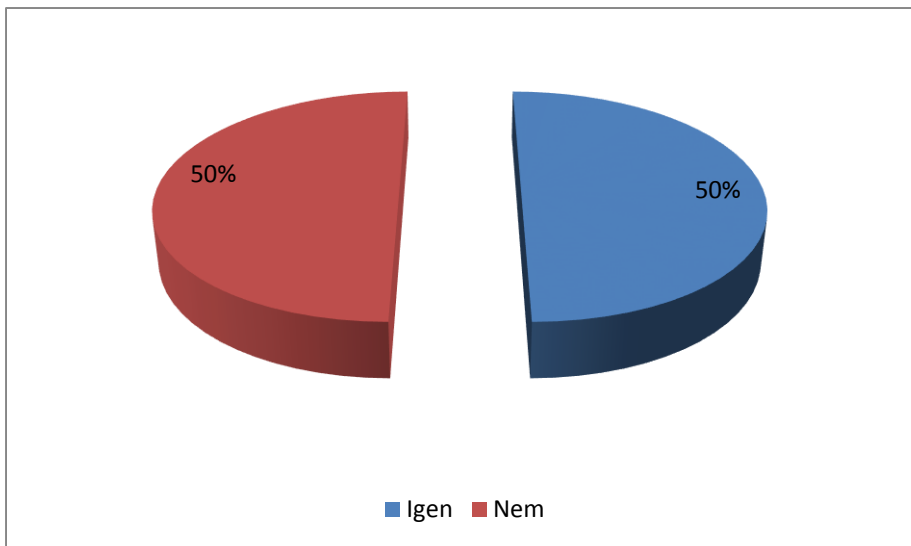
## 12. ábra: Kapcsolat a főkönyvi rendszerrel

Forrás. Saját vizsgálat

A válaszadók 93%-ban egyetértettek abban, hogy a rendelkezésre álló és használt fejtáblái, valamint telepírányító rendszerek nincsenek közvetlen kapcsolatban a könyveléssel (12. ábra). Az adatok áramoltatása manuálisan történik, a szoftverekből nyomtatott listák alapján kézzel viszik fel az adatokat a könyvelésbe. A gazdasági események duplán kerülnek rögzítésre, az adatrögzítés során fellépő időszükséglet és kockázat megduplázódik. A megkérdezettek - munkakörükötől függetlenül - kiemelték mindkét problémát. Az adatok lassan áramlanak az elkülönült rendszerek között, emiatt nem rendelkeznek naprakész információkkal.

Az interjúk során általános elvárásként merült fel az, hogy a telepírányítási szoftverek legyenek kapcsolatban a készlet, valamint a főkönyvi szoftverekkel.

A 15 vállalkozásból 14-ben, ahol legalább fejőházi vagy telepírányítási rendszer rendelkezésre áll, 7-ben vélekedtek úgy, hogy a napi működés során nyert adatok nem támogatják a döntéshozót. Az adatok nem olyan struktúrában jelennek meg számukra, amely kiszolgálná az igényeiket a döntéshozás során. Ez a probléma a telepvezető és/vagy állattenyésztő döntéseire vonatkozik.



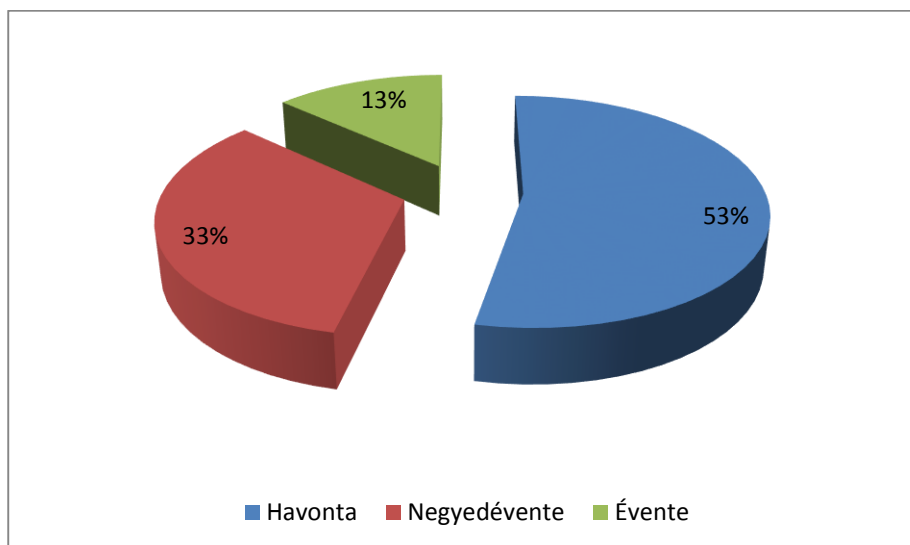
### 13. ábra: Döntéshozatal támogatottsága

Forrás: Saját vizsgálat

A válaszadók 50 százaléka arról számolt be, hogy nem tudja teljesen kihasználni a program-nyújtotta lehetőségeket, funkciókat. Az operatív döntésekhez nem használják a számítógépet annak ellenére, hogy rajta vannak azok az adatok és információk, melyekkel meg tudnák alapozni a döntéseiket (13. ábra). Számítógép helyett a papír alapú nyilvántartásokat használják. Négy esetben elhangzott, hogy a programok kezelése bonyolult, túl sok funkcionalitással rendelkezik, olyanokkal is, melyekre egyáltalán nincs szükségük.

Az interjúk során az is kiderült, hogy a felhasználók nem rendelkeznek elegendő tudással a számítógép-, és telepírányítási rendszerek használatához és hiányzik az a vezetői ráhatás is, amely ösztönözné a telepvezetőt döntéseinek számítógépes támogatására.

Az interjú-kérdések egy másik része annak feltárását célozta, hogy a vállalkozások vezetői milyen informatikai támogatást vesznek igénybe döntéseik meghozatalához. Ehhez a kérdéskörhöz a tervezés, elemzés, döntéshozás témakörökkel kapcsolatos információk tartoznak.



#### 14. ábra: Terv-tény elemzés gyakorisága

Forrás: Saját vizsgálat

A tervezés, elemzés és ellenőrzés információs gyakorlata eltéréseket mutat a vizsgált vállalkozásoknál. A vizsgálat arra irányult, hogy milyen koncepciót követnek a vezetők a tervekészítésnél, milyen felületen készítik a tervet, továbbá milyen igények merülnek fel ezzel kapcsolatban. Az eredményeket a 14. ábra tartalmazza.

A vállalatok mindegyike legalább éves tervet készít, 13%-a a megkérdezett cégeknek úgy nyilatkozott, hogy nem alkalmaznak negyedéves, vagy havi tervadatokat. Az elkészült éves terv ellenőrzése is évente egyszer történik meg.

A válaszadók 33%-a az éves tervet negyedévekre bontja, ezekből tervadatokat készít. A tervadatok elemzése és ellenőrzése szintén negyedévente történik. A vállalkozások több, mint fele (53%) havi bontásban készít tervet, melynek ellenőrzését is havonta végzik. Előfordult, hogy a tényadatokat nem a tervadathoz viszonyították, hanem az előző év azonos időszakához.

Az interjú során végig kiemelt problémaként kezelték a vezetők a tervezés, döntéstámogatás témakört. A terv-tényadatok összevetése segíti őket az esetleges változásokhoz való gyors alkalmazkodásban. Igaz, a tervkészítés és elemzés minden cég esetében sok munkaidőt vesz el mind az ágazatvezetőktől, mind a cégvezetőtől.

A témakörrel kapcsolatban kérdésként szerepelt, hogy milyen módon történik a tervezés, a terv-tényadatok összevetése, ellenőrzése. A válaszokat a 9. táblázat szemlélteti.

## 9. táblázat: A vezetői döntések megalapozása

Vállalkozás	MS Excel	Papír alapon
V01	Igen	Igen
V02	Igen	
V03	Igen	
V04		Igen
V05	Igen	
V06	Igen	
V07	Igen	Igen
V08	Igen	
V09	Igen	Igen
V10	Igen	
V11	Igen	Igen
V12		Igen
V13	Igen	
V14	Igen	Igen
V15	Igen	Igen
<b>Összesen:</b>	<b>13</b>	<b>8</b>

Forrás: Saját vizsgálat

Szinte minden megkérdezett vállalkozás használ táblázatkezelőt (Excel), mely egy irodai programcsomag (MS Office) része. 86%-ban használják az Excel-t, melybe a terv- és tényadatok rögzítése manuálisan történik, gépelési hiba előfordul az adatok bevitelkor. A vállalkozások 53%-a papíron készíti el a terveit. Meglepő volt hallani egy vezérigazgatótól, hogy a tervet "kockás papíron" készíti és havonta vezeti a tényadatokat.

A vizsgált cégek egyike sem használt a tervezés-elemzés és döntéstámogatás céljából készített szoftvert.

A cégvezetők egyöntetűen igennel válaszoltak arra a kérdésre, hogy szükséges-e döntéseiket számítógéppel támogatni. Egy képzeletbeli szoftverrel kapcsolatos véleményeket a 10. táblázat tartalmazza.

**10. táblázat: Egy döntéstámogató rendszertől elvárható igények a tervezésben**

Vállalkozás	Automatizált	Hatékony ellenőrzés	Tervadatok korrigálása módosítása
V01	Igen	Igen	Igen
V02	Igen	Igen	
V03	Igen	Igen	Igen
V04	Igen		
V05	Igen	Igen	
V06	Igen	Igen	
V07	Igen	Igen	
V08	Igen	Igen	Igen
V09	Igen	Igen	
V10	Igen		
V11	Igen	Igen	
V12	Igen	Igen	Igen
V13	Igen	Igen	Igen
V14	Igen	Igen	Igen
V15	Igen	Igen	Igen
<b>Összesen:</b>	<b>15</b>	<b>13</b>	<b>7</b>

Forrás: Saját vizsgálat

Minden cégvezető szeretné kiszűrni, de legalább minimalizálni a hibás adatok rögzítését, így olyan rendszert tudnának elképzelni, amely a tényadatokat a főkönyvi rendszerrel összehangolva gyűjti. Ez egyfajta automatizálást jelent a szoftverben, elkerülvén a dupla, vagy tripla manuális adatrögzítést.

A megkérdezett vezetők 86%-a azt várja az elképzelt szoftvertől, hogy hatékonyabbá váljon az ellenőrzés, vagyis a számára fontos területek, analitikus mutatószámok, költségadatok feletti kontrollt átlátható módon tudja kezelni.

A válaszadók 47%-ban azt várják a szoftvertől, hogy év közben a gazdasági és egyéb környezeti változásokra reagálva, segítségével módosítani tudják a tervadatokat.

### **6.3. Adatbázis kialakítása a szarvasmarha ágazattal kapcsolatos döntések támogatásához**

A mélyinterjúk során jelentkező igények és a vizsgált döntési területek megmutatták a szoftverfejlesztés szükséges irányát, alapvető kritériumait. Világossá vált, hogy a szarvasmarhatartó telepek vezetői számára nincs olyan program, mely segítené őket a tervezés, ellenőrzés és döntéstámogatás folyamatában. Fontosnak látszott továbbá az is, hogy a vizsgálatokból ne csak egy elméleti modell kialakítására kerüljön sor, hanem egy, a gyakorlatban letesztelt szoftver készüljön el, amelynek a forráskódját nyílttá téve annak használata is lehetővé váljon. A mélyinterjúk felmérés során számos egyedi, cégspecifikus igény fogalmazódott meg az egyes eltérő vezetői látásmódokból, vezetői stílusokból kifolyólag. A fejlesztés során mindenképpen gyakorlati szakemberek, cégvezetők, állattenyésztők bevonására volt szükség. Végül két cég vett részt a fejlesztésben, akikkel az egész folyamat során sikerült munkakapcsolatot biztosítani. Ennek az



együtműködésnek az eredményeként a tervezés, az adatbázis kialakítás, a programírás és a program kipróbálása terén valósult meg az együtműködés a szóban forgó cégekkel.

A tervezésnél az interjúk tapasztalatai alapján, a szoftver funkcionalitásával kapcsolatban tervezési és döntéstámogató feladatokat kellett biztosítani.

A számítástechnika oldaláról az alkalmazók követelményeit és lehetőségeit figyelembe véve a legkorszerűbb eszközök és technológiák biztosítása a feladat. A kialakítandó rendszert alkalmassá kellett tenni a hálózati működésre és az adatok automatikus importálására egyaránt.

A célkitűzésekben szereplő második pont alapján felmérésre kerültek azok a mutatószámok, amelyeket a felsővezetői információs rendszerben a döntéshozók kritikusnak véltek, mivel ezeknek a súlya jelentősen befolyásolja a döntéseiket. A közép- és felső vezetőkkel sikerült rögzíteni azokat az adatcsoportokat, amelyeket egy adatbázis kialakításával a tervezés során mindenképpen használni szeretnének és viszonyítási pontként kell, hogy szolgáljanak a tényadatokkal való összevetésben. Négy fő csoport került kialakításra:

- tehenészet,
- növendékállomány,
- hízóállomány,
- borjúállomány.

A négy terület kezelése négy különálló fül segítségével történt. Az egyes területeken belül az alábbi önálló pontok kerültek kialakításra:

**Tehenészet:**

- takarmányozási nap,
- átlagléltszám (db),
- főtermék (liter),
- értékesített tej (liter),
- értékesített tej/tehén (l/tehén),
- takarmányozásra felhasznált tej (liter),
- szaporulat (db),
- szaporulat (kg),
- fajlagos tej (l/tehén),
- tehénne korosbítás (db),
- tehénne korosbítás (kg),
- tehén elhullás (db),
- tej értékesítés árbevétele (forint),
- fajlagos tej árbevétel (Ft/l),
- tehén értékesítés (db),
- tehén értékesítés (kg),
- tehén kényszervágás (db),
- tehén kényszervágás (kg),
- tehén értékesítés árbevétele (Ft),
- tehén átlagár (Ft/kg),
- tehén átlagár (Ft/db),
- átlagsúly(kg/db),
- abrakfelhasználás (kg),
- fajlagos abrakfelhasználás (kg/l).

**Növendékállomány:**

- takarmányozási nap,
- átlagléltszám (db),
- súlygyarapodás (kg),
- fajlagos súlygyarapodás (kg/takarmányozási nap),

- tenyésztésbe állítás (db),
- tenyésztésbe állítás (kg),
- értékesítés vágásra (db),
- értékesítés vágásra (kg),
- árbevétel (forint),
- növendék átlagár (Ft/kg),
- növendék elhullás (db),
- növendék elhullás (kg),
- abrakfelhasználás (100 kg),
- fajlagos abrakfelhasználás (kg/kg).

**Hízóállomány:**

- takarmányozási nap,
- átlaglétszám (db),
- súlygyarapodás (kg),
- fajlagos súlygyarapodás (kg/takarmányozási nap),
- értékesítés vágásra (db),
- értékesítés vágásra (kg),
- árbevétel (forint),
- bika átlagár (Ft/kg),
- bika elhullás (db),
- bika elhullás (kg),
- abrak felhasználás (100 kg),
- fajlagos abrak felhasználás (kg/kg).

**Borjúállomány:**

- takarmányozási nap,
- átlaglétszám (db),
- súlygyarapodás (kg),
- fajlagos súlygyarapodás (kg/takarmányozási nap),
- értékesítés vágásra (db),
- értékesítés vágásra (kg),

- árbevétel (forint),
- borjú átlagár (Ft/kg),
- borjú elhullás (db),
- borjú elhullás (kg),
- borjú elhullás (%),
- abrakfelhasználás (100 kg),
- fajlagos abrakfelhasználás (kg/kg).

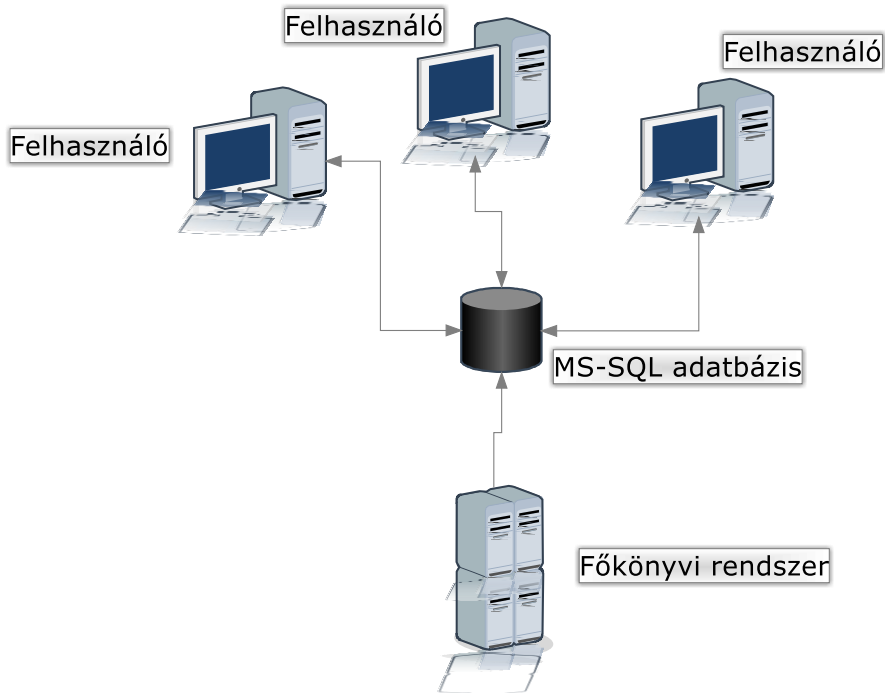
A felsorolt sarokpontok kerültek előtérbe az interjúk során. Az egyes mezők számolt adatok, kiszámításuk a program feladata.

**Az interjú alapján, valamint a konzultációk során is a havi terv-tény összehasonlítás látszott célravezetőnek, ezért a program kialakítása ennek megfelelően történt.** Havi egységként kerülnek bevitelre a tervadatok és az összehasonlítás is havi bontásban történik. Már a tervezés során fontos volt, hogy a szoftver ne tartalmazzon sok funkcionalitást, inkább a gyakorlati szakemberek igényeire fókuszáljon, hogy a felesleges funkciók miatt ne alakuljon ki egy átláthatatlan rendszer.

**Tervezéskor szem előtt kellett tartani a könnyű kezelhetőséget, amit az interjúalanyok külön is kértek. A vezetők az éves, negyedéves és havi tervadatokat többségében táblázatkezelő segítségével készítették.** A felhasználói felület kialakításakor elsődleges szempont volt, hogy a táblázatos megjelenítés könnyen kezelhető és áttekinthető legyen a felhasználók számára.

**Az interjúk eredményei alapján szintén alapkövetelmény volt, hogy ne egy újabb, független szoftver kerüljön be a vállalkozás gyakorlatába, amelybe a főkönyvi adatokat kell felvinni, manuálisan rögzíteni.** Ennek kivédésére egy egyirányú interfész kialakítására volt szükség, kivédve ezzel a rögzítés során keletkező újabb hibalehetőséget, egyidejűleg munkaidőt lehet megtakarítani a folyamat automatizálása által.

A követelményeknek megfelelő rendszer struktúráját, a 19. ábra szemlélteti.



### 15. ábra: Az új rendszerstruktúra

Forrás: Saját szerkesztés

Amint az a 15. ábrán látható, a főkönyvi rendszer és az adatbázis között egyirányú-, míg a felhasználók és az adatbázis között kétirányú a kapcsolat. Ezzel a megoldással megvalósítható az a gyakorlat, hogy a vállalaton belül a vállalati hierarchia csúcsán lévő személy határozza meg a következő évi tervadatokat. Akár rögzítheti is, majd a könyvelés a főkönyvi rendszerből átküldi a tényadatokat az adatbázisba. Terv-tény elemzést pedig a vállalati hierarchia egyéb szintjein is nyomon tudják követni.

A vállalkozást körülvevő környezet folyamatosan változik, nem várt események következhetnek be. A megváltozott körülmények új döntési helyzetet generálnak, a vezető számára fontos az új helyzet gyors felismerése. Ennek érdekében, egy **figyelmeztető funkció beépítésével történt, mely a terv-tény adatok megadott eltérése esetében jelez a felhasználó számára.**

Azon okból kifolyólag, hogy az elkészített szoftver gyakorlati használata eltérő vezetői szemléletek mellett is megoldható legyen, **a rendszert úgy kellett kialakítani, hogy az bővíthető legyen, képes legyen egyedi, cégspecifikus vezetői igényeket is kielégíteni.** Ezért a főkönyv teljes adattartománya bekerült az adatbázis szerverbe, azon további adatmanipulációk végezhetők.

Az adatbázis kialakításánál is szempont volt a kor követelményeinek megfelelő technológia megválasztása. Az adatbázis szerver kialakítására azért volt szükség, mivel nagy mennyiségű adattal kellett dolgozni, akár több évre visszamenőleg, mindezt nagy adatbiztonság mellett. Olyan relációs adatbázisról van szó, melyben a reláció nem más, mint egy kétdimenziós tábla, a tábla soraiban tárolt adatokkal. A relációs adatbázis pedig ezen relációk összessége. A reláció soraiban történik a logikailag összetartozó adatok tárolása, az oszlopokban pedig az azonos mennyiségre vonatkozó adatok jelennek meg.

Az adatbázis-tervezés több lépésből tevődött össze. Először az adatbázisban leképezendő rendszer elemzésére, ezután a tárolandó adatok körének kijelölésére, majd a relációk közötti kapcsolatok vizsgálatára került sor. A logikai adatbázis modell, a vállalkozások hardver és szoftver rendszereinek figyelembe vételével készült el. Az adatbázis végső kialakítása MS-SQL Server Express-en történt.

**Fontos, hogy eltérő adatbázis szerver igény felmerülése esetén ne kelljen a forráskódot átírni.** Ezért a programkódba a LINQ nyelv beépítésére került sor.

A főkönyvi rendszer felől érkező adatok CSV (Comma-Separated Values), ami egy vesszővel tagolt szöveges fájl, amelyben rendszerint a vessző (,) választja el egymástól a szöveg mezőit. A 11. táblázatban egy részlet látható az importálandó CSV file-ből.

```
2;64211;Szarvasmarhatenyesztesi agazat;55105;Önk.nyugdíj.p.fiz.tagdíjhozzáj.;101046;0;0
2;64211;Szarvasmarhatenyesztesi agazat;561;Nyugdíjbizt. és eü.járulék;381196;0;0
2;64211;Szarvasmarhatenyesztesi agazat;564;Szakképzési hjárulás;29009;0;0
2;64211;Szarvasmarhatenyesztesi agazat;566;Start járulék;1024;0;0
2;64211;Szarvasmarhatenyesztesi agazat;571;Tervszerinti értékcsökk.leír.;0;0;0
2;64211;Szarvasmarhatenyesztesi agazat;6911;Javitóműhely ktsgeinek átvez.;28800;0;0
2;64211;Szarvasmarhatenyesztesi agazat;69311;Traktorüzem ktsgeinek átvezet.;485610;0;0
2;64211;Szarvasmarhatenyesztesi agazat;69312;Traktorüzem tak.gazd.ktsgátvz.;97560;0;0
2;64211;Szarvasmarhatenyesztesi agazat;69313;Traktorüzem szmarha ktsg.átvez.;0;0;0
2;64211;Szarvasmarhatenyesztesi agazat;69384;Rakodógépek ktsgeinek átvezet.;440700;0;0
2;64211;Szarvasmarhatenyesztesi agazat;69386;Gréder ktsg.átvezetése;0;0;0
2;64211;Szarvasmarhatenyesztesi agazat;69387;Manitou ktsg.átvezetése;0;0;0
2;64211;Szarvasmarhatenyesztesi agazat;694211;Szarvasmarha ált.;0;0;0
2;71201;Tehenészet költségei;511101;Vill.energia felhasználás;807366;0;0
2;71201;Tehenészet költségei;511103;Növényvédőszer felhasználás;0;0;0
2;71201;Tehenészet költségei;511104;Ip.eredetű készlet felhaszn.;2408707,65;106233;0
2;71201;Tehenészet költségei;511107;Gáz felhasználás;0;0;0
2;71201;Tehenészet költségei;511108;Benzin felhasználás;0;0;0
2;71201;Tehenészet költségei;511109;Kenőolaj felhasználás;12775;5;0
2;71201;Tehenészet költségei;511111;Alkatrész;815848,99;418;0
2;71201;Tehenészet költségei;511112;Egyéb anyag;600554,25;569,5;0
2;71201;Tehenészet költségei;511113;Fogyóeszközök felhaszn.;58428;826;0
2;71201;Tehenészet költségei;511114;Építőanyag felhasználása;6103;1,25;0
2;71201;Tehenészet költségei;511115;Állatgyógyszer felhasználás;0;0;0
2;71201;Tehenészet költségei;51111520;Fejőház gyógyszerfelhasználás;1292176,73;7830;0
2;71201;Tehenészet költségei;51111521;Ellető gyógyszer felhasználás;246385,77;7149;0
2;71201;Tehenészet költségei;51111522;Tehenészet fogadó gyógyszerfh.;0;0;0
2;71201;Tehenészet költségei;51111523;Itatásos borju gyógyszerfh.ket;0;0;0
2;71201;Tehenészet költségei;511116;Állatgyógyászati segédanyag;3500;200;0
2;71201;Tehenészet költségei;5112011;Vásárolt széna felhasználás;0;0;0
2;71201;Tehenészet költségei;5112012;Vásárolt szalma;596025,15;90300;0
2;71201;Tehenészet költségei;511203;Vás.sperma felhasználás;837122,99;204;0
2;71201;Tehenészet költségei;5131;Nyomtatvány irodaszer;0;0;0
2;71201;Tehenészet költségei;523;Javítás, karbantartási költ.;66500;0;0
2;71201;Tehenészet költségei;5293;Foglalk.egügyi szolg.;0;0;0
2;71201;Tehenészet költségei;5294;Szakértői pályázati díjak;0;0;0
2;71201;Tehenészet költségei;5295;Tagsági és rendszerdíj;0;0;0
2;71201;Tehenészet költségei;5296;Állategészségügy ellátás;243680;0;0
2;71201;Tehenészet költségei;5299;Egyéb igénybevett szolgáltatás;223015;0;0
2;71201;Tehenészet költségei;531;Hatósági iq. szolg.díj,illeték;7350;0;0
```

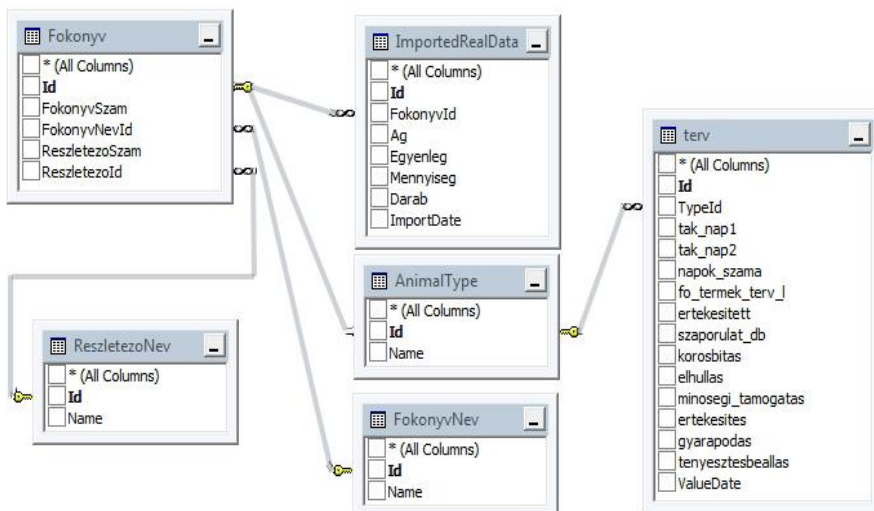
## 11. táblázat: Importálandó CSV file (részlet)

Forrás: Saját vizsgálat

A főkönyvi rendszerből az alábbi felosztásban érkeznek az adatok, balról jobbra haladva:

- az ágazat kódja,
- a főkönyvi kód,
- a főkönyvnev,
- a részletező számla kódja,
- a részletező számlanév,
- az egyenleg,
- a mennyiség,
- a darabszám.

Az adatbázis-táblák kialakításánál ezek adták az alapot. Ezután kerültek kialakításra a tervadatok táblái és segédtáblái. Az elsődleges kulcsok és táblák összekötése után kialakított SQL relációs kapcsolatot a 16. ábra szemlélteti.



## 16. ábra: SQL táblakapcsolatok

Forrás. Saját szerkesztés



Az adatok importálása során - az azonosíthatóság miatt - még egy adattag felvétele szükséges. Ez az importálás dátuma (ImportDate), ami az ImportedRealData táblán belül helyezkedik el:

```
<?xmlversion="1.0"encoding="utf-8" ?>
<configuration>
<configSections>
</configSections>
<connectionStrings>
<addname="Mezogazdasag.DAL.Properties.Settings.MezogazdasagConnectionString"
connectionString="Data Source=TESCHNERGERGELY\SQLEXPRESS;Initial
Catalog=Mezogazdasag;Integrated Security=True;User Instance=False;Context
Connection=False"
providerName="System.Data.SqlClient" />
</connectionStrings>
</configuration>
```

A rekordok azonosításához egy „id” mező bevezetésére került sor. Az adatbázisban a felhasználó által megadott tervadatok és a főkönyvi rendszerből importált adatok tárolása történik. Az alkalmazási felületen keresztül ezen adatokat felhasználva az üzleti logika alapján, (tehát számításokból) kapott adatok jelennek meg. Ez utóbbiak nem kerülnek tárolásra, csak a futásidőben léteznek. Az importálás havonta egyszer történik, mindig az előző hónap főkönyvi adattartalmával. Szükség esetén megoldható a gyakoribb, akár heti vagy kéthetenkénti adatimportálás is, egyedi kialakítással.

Miután a kutatásba, illetve a fejlesztésbe bevont két vállalkozás azonos könyvelőszoftverrel rendelkezik, így ugyanazon interfész segítségével lehet az adatokat mozgatni a két rendszer között.

## 6.4. Tervezési és döntéstámogató információs rendszer kialakítása

A programíráshoz a készülő programot –a működés alapján – célszerű volt több rétegre, projektre bontotani. Ezek az alábbiak:

- **Business Logic:** az itt található forráskódok tartalmazzák a számításokat, az egyszerű szorzás-osztás műveleteket, (például fajlagos abrakszükséglet) stb.
- **Data Access Layer:** ez a réteg biztosítja az adatbázissal való kapcsolatot. Az egyes adatbázis-műveleteknek itt történik a deklarálása a terv és a tényadatokra egyaránt.
- **Import Layer:** a főkönyvi adatok importálása ebben a rétegben valósul meg.
- **Models:** ebben a rétegben találhatóak azok a forráskódok, melyek a felhasználó számára megjelenítendő adatokat szolgáltatják (a terv, a tény és a százalékos adatokat is beleértve).
- **User Interface:** Ez a réteg felel a felhasználó felületért. Az itt lévő file-okban van elhelyezve a program színekódjai, a funkciógombok, a cellák stb.

**A program mérete nem indokolta volna ezt a fajta bonyolult felosztást, viszont ezzel a módszerrel könnyebbé válik a további fejlesztés. Lehetőséget ad bármilyen irányú, speciális igényeket kielégítő bővítésre.** A végső felhasználóknak az adatbázis elérése más lesz, mint a tesztgépen. Ennek megoldásához csak a Data Access Layer-ben leírt csatlakozás deklarációt kell módosítani. Ha módosítást kell végrehajtani a meglévő adatokon, akkor ezt elég a Business Logicban elvégezni, majd a Models rétegben érvényesíteni az új metódusokat.

Az is előfordulhat, hogy a felhasználói felületen kell módosítani, például azért, mert az adott szarvasmarha telepen jelentősen eltérő felbontású képernyőket használnak. Az ehhez történő optimalizálás szintén elvégezhető az User Interface módosításával.

A mélyinterjúk során is jelentkeztek egyedi igények, ezzel a megoldással - kisebb munkabefektetés árán - ezeket ki lehet elégíteni.

**A programkészítés során a vezetés támogatásának biztosítása prioritás volt. A vezetés alapvető feladata a szervezet céljait segítő döntések meghozatala.** Az érték-előállítási folyamat során koncentrálni az erőforrásokra, mérni, értékelni, tervezni és elemezni azokat. A programozás, valamint később a felhasználói felület kialakítása is ezeket a tényezőket figyelembe véve történt.

Terv-tény ellenőrző v1.0

Tehénésztés | Növendék állomány | Hízó állomány | Borjú állomány

Főbizonyi importálás

2012. január

Mentés

Módosítás

Képek

Hónap napjai: 31

Tervadatok	Elérés (%)	Tényadatok
Takarmányozási napok száma:	0.43	17934
Átlagos létszám (db):	0.43	579
Főtermék (l):	6.34	472815
Értékesített tej (l):	6.26	451665
Értékesített tej/tehen (l/db):	5.85	781
Takarmányozásra fordított tej (l):	8.05	21150
Szaporulat (db):	0.00	63
Szaporulat (kg):	0.00	2268
Fajlagos tej (l/db):	5.93	81729
Tehénnek korosbítás (db):	0.00	23
Tehénnek korosbítás (kg):	0.00	13340
Tehén elullás (db):	100	0
Tehén értékesítés (db):	10.00	11
Tehén értékesítés (kg):	20.54	6750
Tehén értékesítés árbevétele (Ft):	27.52	2008800
Tej értékesítés árbevétele (Ft):	6.32	42481947

Tehén kényszervágás (db):	2
Tehén kényszervágás (kg):	1200
Tehén átlagár (Ft/kg):	297.60
Tehén átlagár (Ft/kg):	182618.20
Tehén átlag súly (kg/db):	613.60
Abrok felhasználás (kg):	181510
Fajlagos abrok felhasználás (kg/l):	0.38

17. ábra: Az új rendszer felhasználói felülete

Forrás: Saját fejlesztés

A 17. ábra a tehenészettel összefüggő terv- és tényadatokat tartalmazza a program futása során, valós főkönyvi adatokkal.

A program működtetése a következőképpen történik:

- bal oldalon találhatóak meg a felhasználói funkciógombok, melyekkel utasítások adhatók, valamint egy listaelem, melyben az év hónapjait tudja kiválasztani a felhasználó;
- a főablak bal oldalán helyezkednek el a tervadatok. A felhasználónak a sárga cellák tartalmát kell megadnia, a további adatok számolt értékek;
- jobb oldalon a főkönyvi rendszerből importált adatokból kiszámolt értékek szerepelnek;
- ezek egy része összehasonlítható az igényként felmerült tervadatokkal. Az itt nem szereplő adatok a főképernyő jobb szélső oszlopában kaptak helyet.

A terv és tényadatok elemzése során fontos a döntéstámogatás biztosítása oly módon, hogy a vezető igényeihez igazodva a **terv és tényadatok eltérése megjeleníthető legyen**. Ez oly módon valósul meg, hogy a százalékos eltérések osztályozott értékei eltérő színjelölést kaptak:

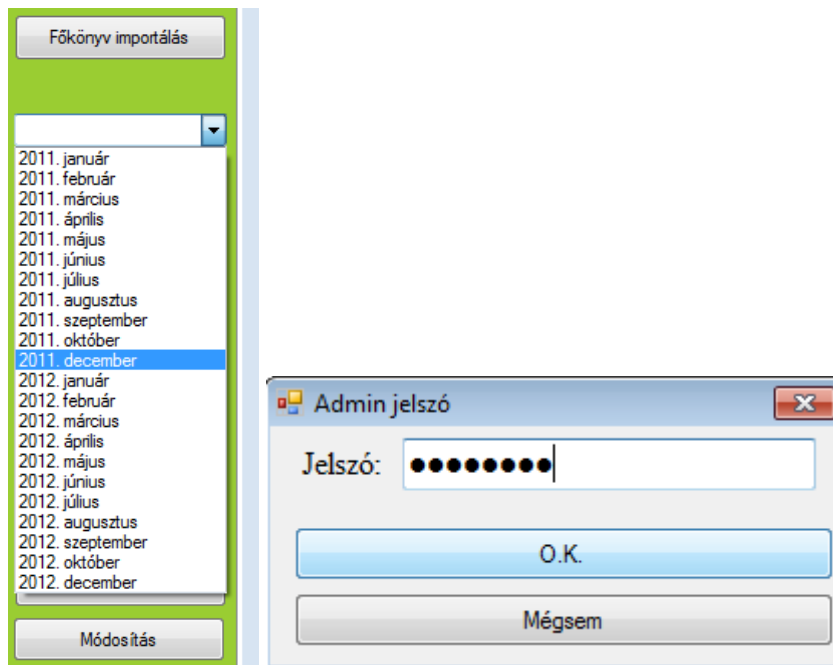
- 0-5%: az eltérés nincs külön színnel jelölve;
- 5-10%: narancssárga számok jelzik az ilyen mértékű eltérést;
- 10%<: piros színű számok jelzik az eltérést.

Az eltérés irányával kapcsolatban nem igényelték a vezetők a színbeli megkülönböztetést. A figyelemfelhívást tartották fontosnak, mivel ha az adott területen nagyobb eltérés van a tervezettnél, az **további vizsgálatot igényel**. A vezetői igényeknek megfelelően a figyelemfelhívást szolgáló százalék-intervallumok szintén változtathatók.

**A mélyinterjúk során az is kiderült, hogy a vezetők a tárgyévet megelőzően alakítják ki éves tervszámaikat, majd ezeket az év**

**tizenkét hónapjára lebontják.** A gyakorlathoz igazodóan ezeket a tervszámokat a programban a 18. ábrán látható módon kezelheti a felhasználó: a listából kiválasztja a kívánt hónapot, majd a megadott – sárgával jelölt - cellákba felviszi az adatokat. **Esetleges hiba vagy a piaci környezet változása esetén a tervadatokon lehetséges módosítani.** Mivel a rendszer automatikusan elvégzi a terv-tény adatok összehasonlítását, így az új, megváltozott helyzetnek megfelelő döntéshozatalat támogatja.

Módosítás esetén a hibák elkerülése érdekében jelszó felhasználása szükséges.



**18. ábra: A főkönyvi importálás sémája**

Forrás. Saját fejlesztés

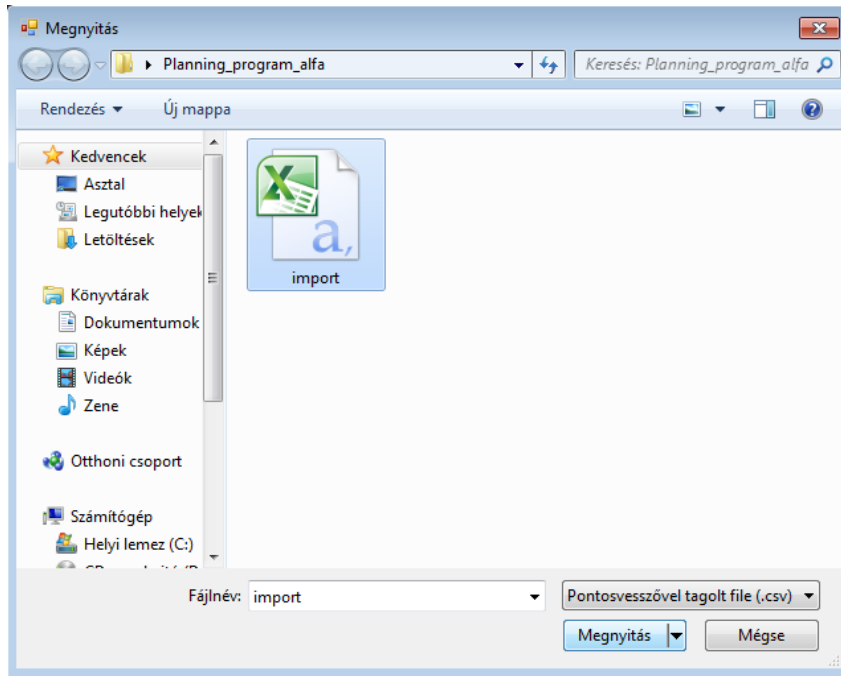
Amennyiben **a felhasználó külön akarja megjelentetni az adott hónaphoz tartozó tervadatokat**, lehetősége van ezek rögzítésére, majd eltárolhatja azokat az adatbázisban. A program telepítése során lehetőség van arra is, hogy a szoftver használata előtti terv és tényadatok közvetlenül az adatbázisba kerüljenek.

Az éves tervszámok kialakításához meg kell oldani az előző időszaki adatok összesítését is. A rendszerben rögzített, előző évre vonatkozó havi terv és tényadatok összege segíti a tárgyév tervszámainak kialakítását. Ugyanez az összesítés teszi lehetővé, hogy a felhasználó év végén a vállalat teljes évi teljesítményét értékelni tudja.

A rendszerstruktúra kialakítása során arra kellett figyelni, hogy miután a tervadatok rögzítése és adatbázisba történő mentése megtörtént, ezek az adatok a vállalat bármelyik felhasználója számára hozzáférhetők. Ezután következhet a főkönyvi rendszerből a könyvelési tényadatok importálása. A számítógépes rendszer elvégzi a további számításokat, amiket a felhasználó rendelkezésére bocsát.

A 19. ábrán az előzőekben leírt folyamat számítógépes megjelenítése látható. A pontosvesszővel tagolt CSV file (10. táblázat) beolvasását követően a program feltölti a tényadatok celláit. A terv és tényadatok közötti eltérések azonnal leolvashatóvá válnak a felhasználói felület középső oszlopában.

**A vezető az importálás után azonnal látja azokat a kiemelt területeket, melyek a saját maga által meghatározott határértékeket átlépték.**



### 19. ábra: Importálandó file kiválasztása

Forrás: Saját fejlesztés

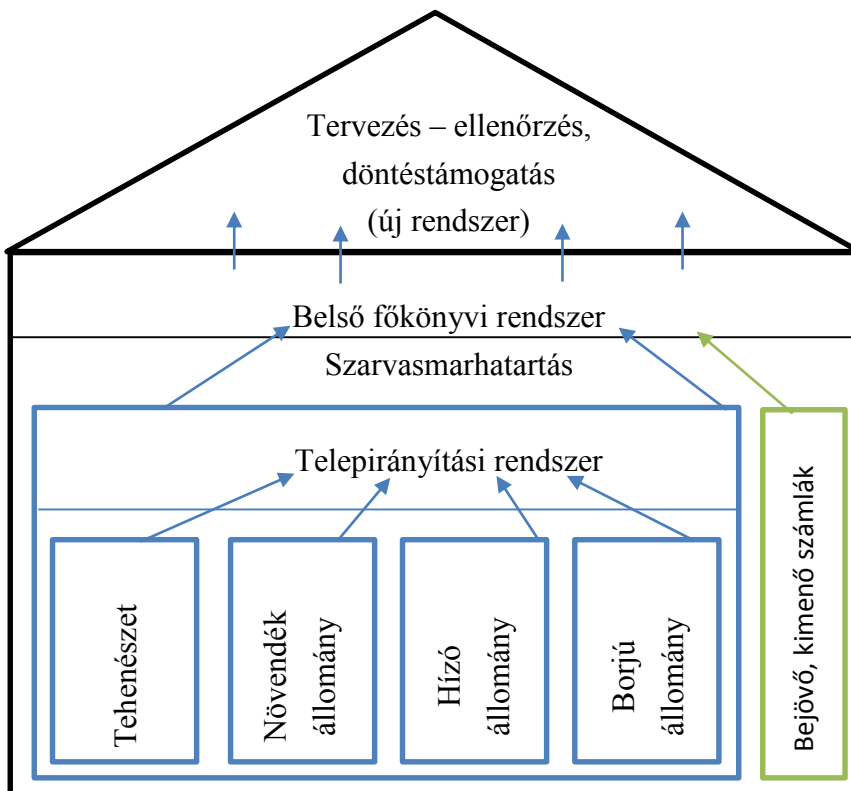
A rendszer használata - a betanulás után – viszonylag egyszerű, a felhasználónak nem kell a főkönyvi rendszer adataiból hosszas számolásokat végezni. A Business Logic-ban **meghatározott és letesztelt képletek közvetlenül a főkönyv adataiból dolgoznak, nincs dupla manuális adatrögzítés. E két tényező kombinációja lehetővé teszi a pontos adatok megjelenítését a döntéshozó számára.**

További fejlesztési igények jelentkeztek, többek között a grafikonos ábrázolás, az előző év azonos hónapjához való viszonyítás, és elemzés. Ezek a fejlesztések a következő időszak feladatát képezik.

**A program használata hozzásegíti a tejtermelő tehenészetek döntéshozóit a problémák felismeréséhez és döntéseik megalapozásához.**



A mélyinterjúk kutatás alapján készültek bonyolultabb, nagy informatikai beruházást igénylő vállalatirányítási rendszerek. Ezek használata azonban csak nagyvállalati szinten célszerű, mert a nagyteljesítményű szoftver alkalmazásához drágább informatikai berendezések szükségesek, melyek komoly létszámú szakértő személyzetet is igényelnek. Ezért volt a cél egy egyszerű, könnyen kezelhető, felsővezetői információs rendszer létrehozása, mely akár a kis és közepes vállalkozások vezetői számára is döntéstámogató funkciót tud ellátni.



**20. ábra:** Az új rendszer elhelyezkedése és információ forrásai a szarvasmarhatartó telep működésében

Forrás: Saját szerkesztés

A kialakított információs rendszer ellenőrzési funkciója a már meglévő rendszerek által gyűjtött információkon alapszik. A külső és belső információforrásokra egyaránt szükség van a tervszámok kialakításához. A vezető számára kialakított tervező rendszer strukturális felépítéséből adódóan fejleszthető, bővíthető. Széleskörű információforráson alapszik, valamint az általunk kialakított rendszer a használatban lévő információs rendszerekre épül, ami azt bizonyítja, hogy a szorosan együttműködő rendszerek kialakítása és felhasználása a gyakorlatban megvalósítható.

## 7. KÖVETKEZTETÉSEK, JAVASLATOK

---

A szarvasmarhatartó mezőgazdasági vállalkozások korszerű döntéshozatali rendszerének kialakításához **először a jelenlegi helyzetet kellett felmérni**. Az interjúk során az önkéntesen közreműködő vállalkozások közép- és felsővezetői szolgáltatottak információt a kutatáshoz.

A megkérdezettek aktivitása a felmérés során jónak mondható, azonban több vállalkozás elutasította a közreműködést, így egy 15 gazdaságból álló minta alapján lehetett az elemzéseket elvégezni és ez szolgált alapul a döntéstámogató informatikai rendszer kidolgozásához.

1. A felmérés során megállapításra került, hogy **a vizsgált vállalkozások által használt informatikai rendszerek elsősorban az állattenyésztés irányításához és a könyveléshez nyújtanak segítséget**. Ezek a rendszerek általában egymástól függetlenek és szigetszerűen működnek. Az is kiderült, hogy **nem tekinthetők komplex döntéstámogató rendszernek**, csupán egy-egy részterület információigényét képesek kielégíteni.

2. Vizsgálataink középpontjában a tejelő szarvasmarhatartás adatainak elemzése állt, így **kiemelt figyelmet fordítottunk az állattenyésztésben használt különböző szoftverek közül a tejtermeléshez kapcsolódó informatikai eszközökre**.

Ezek alapján a tejtermeléshez köthető szoftverek az alábbi három kategóriába sorolhatók:

- irodai programcsomag részeként használt adatbázis- és táblázatkezelő programok,
- fejóházi rendszerek szoftverei,
- a fejóházból érkező adatokat is kezelő telepírányítási rendszerek.

A felsorolt szoftverekből nyert információk segítik, segíthetik a döntéshozót a termelés során keletkező problémák megoldásában.

3. Említést érdemel, hogy a szoftverek által biztosított aggregált információk nem képezik részét minden döntéselőszítésnek. A vizsgálataink során a vállalatoknál azalkalmazott informatikai rendszerek felmérése volt a cél. Így csupán azt vizsgáltuk, hogy az adott cég hány darab szoftvert használ, de az alkalmazott információs rendszerek funkcionalitásának kihasználtsági szintjét nem állt módunkban pontosan meghatározni. **A kapott eredmények így is azt mutatják, hogy a szoftverek kihasználtsága messze nem teljes**, de néhány funkciót (pl. szaporodásbiológiai kezelések, állománycsoportok kezelése) kivétel nélkül alkalmaztak a vizsgált gazdaságok.

4. Az interjúk során szerzett tapasztalatok alapján megállapítható, hogy a szoftverek adta lehetőségek nincsenek teljesen kihasználva. **Az általuk biztosított adatok nem adnak releváns információkat a döntéshozók számára, ezáltal nem alkalmasak rövid-, és hosszútávú tervek készítéséhez és ellenőrzéséhez.**

E megállapítás főbb okait a kapott eredmények alapján két oldalról érdemes megközelíteni. Az említett szoftverek kihasználtságának alacsony szintje egyrészt az alkalmazotti, másrészt a vezetői motivációban kereshető. Eszerint:

Az alkalmazottak egy részénél hiányosak az informatikai ismeretek, ami alapvetően gátolta a meglévő szoftverek teljes kihasználását.

**Tapasztalataink szerint hiányzik a megfelelő belső motiváció is az említett szoftverek pontos és naprakész használatához**, mert az alkalmazottak gyakran nem látják a hosszú távú adatsorokban rejlő döntéstámogató lehetőségeket, főleg az ismerethiányból következően. **Ezért fontos az általános felhasználói számítógépismereteken túl az irányítást támogató informatikai rendszerek és az egyéb vállalati információs rendszerek megismerése.**

A vezetők részéről szintén fontos az informatikai-szakmai ismeretek elsajátítása. Sok esetben tapasztalható volt a nem megfelelő vezetői hozzáállás a rendelkezésre álló szoftverek használatával kapcsolatban. A napi irányítás során az alkalmazottak nem voltak rákényszerítve a szoftverek használatára. **A hiányos informatikai ismereteik miatt nem látták a szoftverek által biztosított aggregált információk adta lehetőségeket.**

5. A rendelkezésre álló döntéstámogató információs rendszerekkel kapcsolatban számos probléma merült fel. Ezek közül az egyik legjelentősebb a szoftverfejlesztő cégek oldaláról kínált rendszerek piacán megfigyelhető széles terméksortiment. **A különböző szoftverek** által biztosított adatkezelési eljárások és funkciók egyedileg jól használhatók, de **nagy hátrányuk, hogy egymással nem kompatibilisek.** Ebből következően a vizsgált cégek által megvásárolt rendszerek - a kommunikációjuk során - eltérő interfészeket használnak, így nem képesek egymással kommunikálni.

A kompatibilitási problémák ellenére a vállalatok használják a különböző szoftverek által tárolt adatokat. Ezt általában manuális adatrögzítéssel oldják meg, növelve ezzel a felhasznált munkaidőt és az adathibák esélyét. **Az információs rendszerek bevezetése és használata ma még nem alapvető része a tejtermelő vállalkozások közép és hosszú távú stratégiájának.** Előfordul, hogy egy új fejőházi technológia bevezetésével és beruházásával kapcsolatos döntésnek nem képezi részét az új fejőházi technológiához kapcsolt információs rendszer integrálása a vállalat már meglévő informatikai rendszerébe. Pedig a vállalkozások piaci alkalmazkodását javítaná, ha a közép- és hosszútávú terveikben prioritást adnának a döntéstámogatásnak és az ehhez szükséges információs rendszereknek is.

6. A felmérés rávilágított arra is, hogy a **közép és felsővezetők számára nem kielégítő információkat szolgáltatnak a jelenleg**

**használt rendszerek**, mert a tervezéshez nem hordoznak elegendő információt. Jelentős volt azoknak a vezetőknek a száma, akik úgy vélekedtek a meglévő rendszerekről, hogy azok nem segítik a termeléssel összefüggő döntések meghozatalát, **mivel a kapott aggregált információk nem cég- és vezetőspezifikusak, így nem illeszkednek egyértelműen a döntés-előkészítő munkálatokba.**

A felmerült problémákra **a jövőben megoldást adhat, ha a szoftverfejlesztő cégek együttműködve a felsőoktatási és kutató intézményekkel, - szem előtt tartva a vállalati igényeket – a jelenleginél „vállalatközelibb”, a döntéstámogató informatikai rendszereket alaposabban kihasználó programcsomagokat hoznának létre.**

7. A vizsgált tejtermelő vállalkozásoknál **igényként merült fel egy olyan tervezést és ellenőrzést végző információs rendszer létrehozása, amely segíti az operatív döntéshozást, a problémák felismerését és jobban illeszkedik a cégek irányítási profiljába.**

Az is kiderült, hogy az elmúlt évtizedben számos kutatás és fejlesztés irányult erre a részterületre, de ennek ellenére a gyakorlat nem hasznosította a már meglévő fejlesztéseket. **Célszerű lenne a már rendelkezésre álló fejlesztéseket szélesebb szakmai közönséghez eljuttatni, a gyakorlatban több helyen kipróbálni és véleményeztetni.** A tudományos folyóiratokban publikált eredmények mellett fontos lenne a gyakorlat által ismert és olvasott szaklapokban is közölni a kapott eredményeket közérthető formában, így várhatóan növekedne a döntéstámogató rendszereket alkalmazók köre. Ez ágazati szinten jobb tervezhetőséget, kiszámíthatóbb és előkészítettebb döntéshozást eredményezhetne.

8. **Jelenleg** a vizsgált tejtermelő vállalkozások döntéseinek előkészítésében **még mindig döntő szerepet kap a papíralapú kalkuláció, illetve néhány helyen az egyszerű táblázatkezelő rendszerek használatával készítenek terveket.** Ezek távlati terveket

és komplex számításokat nem tartalmaznak, csak a tervszámokat. Ezek alkalmazása jelentős időt vesz igénybe és nagy hibalehetőséggel terhelt.

**9. A vizsgálatok eredményei alapján elkészült egy döntéstámogató információs rendszer a tejtermelő vállalkozások számára,** ami tartalmazza mindazon elemeket, melyek a kutatások során igényként jelentkeztek a vizsgált gazdaságoknál. **Az elkészített szoftver - reményeink szerint – jobban segíteni fogja a vezetők döntéseinek megalapozását.**

Az eddigi tapasztalatok alapján az a következtetés vonható le, hogy a kifejlesztett szoftver által kalkulált mutatószámok megfelelőnek bizonyulnak az operatív döntések során, így hozzájárulnak a vezetők döntéseinek megalapozásához, valamint a tervezési folyamatok ésszerűsítéséhez.

**További előnyt jelent, hogy az egymástól független tranzakciós rendszerekből generált listák helyett az aggregált, egy felületen feltüntetett adatok jelennek meg, jobban segítve a vállalatok irányítását, azok hatékonyabb működését.**

A fentiekben leírt megállapítások lehetőséget adnak arra, hogy a Hipotézisek c. fejezetben megfogalmazott feltételezésekre válaszolni tudjunk:

**Első tézis (T1):** A felmérések alapján igazolódott az a hipotézis (H1), mely szerint a tejtermelő gazdaságokban található számítástechnikai eszközállomány eléggé heterogén és a használt szoftverek nem alkalmasak egy telep teljeskörű, komplex irányítására.

**Második tézis (T2):** A második hipotézisben (H2) szereplő megállapítást csak részben sikerült igazolni, mivel az alkalmazott információs rendszerek funkcionalitásának kihasználtsági szintjét nem lehetett pontosan megállapítani. Annyi azonban nagy biztonsággal állítható, hogy a szoftverek kapacitásának egy jelentős része nem hasznosul.

**Harmadik tézis (T3):** Teljes egészében igazolódott a harmadik hipotézisben (H3) megfogalmazott állítás, mivel a vizsgált gazdaságokban (telepeken) az irányítás egy része ma is manuálisan történik, ezért a döntés-előkészítés és –támogatás gyakran szubjektív elemekkel terhelt.

**Negyedik tézis (T4):** Igazolódott a negyedik hipotézis (H4) is. Eszerint a tejtermelő vállalatok döntési folyamatait nehezíti (esetenként lehetetlenné teszi) a különféle szoftverek közötti inkompatibilitás.

**Ötödik tézis (T5):** A felmérések egyértelműen igazolták azt, hogy a tejtermelő gazdaságok menedzsmentje részéről olyan döntéstámogató informatikai rendszerre lenne igény, amelyik önállóan, a meglévő adatbázisra épülve ad lehetőséget egy konzisztens tervezési rendszer működtetésére.

**Hatodik tézis (T6):** A vizsgálatok eredményei lehetővé tették egy olyan döntéstámogató információs rendszer kialakítását, amelyik új lehetőségeket kínál a vezetői döntések megalapozásához, mindennek előtt a tervezési folyamatok komplexebbé tételéhez. Ez azt jelenti, hogy megvalósult az a feltételezés, amelyik a hatodik hipotézisben (H6) fogalmazódott meg.



## 8. ÚJ ÉS ÚJSZERŰ TUDOMÁNYOS EREDMÉNYEK

---

A PhD tanulmányok folyamán elvégzett szekunder vizsgálatokból ugyanúgy, mint a primer kutatások során alkalmazott kérdőíves felmérésekből és a szakmai mélyinterjúkból nyert eredmények összevetéséből egybehangzóan az alábbi megállapítások tehetők:

1. A vizsgálatok eredményei alapján megállapítható, hogy a kis- és közepes méretű **tejtermelő vállalkozások** – objektív okok miatt – **nem alkalmasak arra, hogy jelenleg a piacon hozzáférhető komplex döntéstámogató informatikai rendszereket hatékonyan működtessenek.** Még a viszonylag **nagyobb méretű, korszerű számítástechnikai eszközökkel ellátott gazdaságok** is csak **részben használják ki** a rendelkezésükre álló **információs rendszerek-nyújtotta lehetőségeiket.**
2. A kutatási eredmények felhasználásával **készült, tervezést és ellenőrzést segítő döntéstámogató információs szoftver** – a már korábban kialakított tranzakciós feldolgozó rendszerekhez kapcsolódva – jobban illeszthető a vállalat egyéb információs rendszereihez, interaktív kapcsolatot biztosítva a vállalat főkönyvi rendszerével.
3. A **kialakított döntéstámogató információs rendszer elsődleges alkalmazási területe az ágazati tervezés.** Strukturális felépítéséből adódóan **lehetőséget kínál a rendszer bővítésére, illetve továbbfejlesztésére.** A gyakorlatban kipróbált új szoftver segítségével lehetőség van az éves tervadatokhoz képest bekövetkező változások (eltérések) folyamatos nyomonkövetésére (képi megjelenítésére) és a változásokat (eltéréseket) generáló tényezők feltárására.

## 9. ÖSSZEFOGLALÁS

---

Kutatásaink és a szakirodalom alapján megállapítható, hogy a döntéstámogató rendszerek vizsgálata időszerű és folyamatos fejlesztés tárgyát kell, hogy képezze. Az informatikai eszközök térhódítása nem állt meg, a személyi számítógépek mellett számos új adatot és információt kezelni képes eszköz jelent meg a mezőgazdasági, ideértve a tejtermelő vállalkozások munkájában. Ezzel együtt a vezetőhöz áramló információ mennyisége is exponenciálisan nő, a vállalkozás céljait szem előtt tartva a releváns adatokat és információkat szűrnie kell.

Vizsgálatunk a magyarországi tejtermelő vállalkozások döntéstámogató rendszereire terjedt ki.

A dolgozat negyedik fejezetében bemutatásra kerültek a legelterjedtebb rendszerek, szem előtt tartva a gyakorlati alkalmazás során szerzett tapasztalatokat, továbbá a kutatást megelőző eredményeket és történeti folyamatokat amelyek megalapozzák a jelenlegi fejlesztési lehetőségeket.

A hatodik fejezetben kifejtettük a disszertáció célkitűzéséhez igazodva, hogy a megkérdezett cégek milyen információs rendszerekkel rendelkeznek és használnak, valamint a vezetők a döntéshozás során milyen adat és információforrást alkalmaznak. A mélyinterjúk kutatás során számos információt nyertünk a vezetők szokásairól, elvárásairól és azokról az igényekről, melyeket egy új döntéstámogató rendszerrel szemben támasztanak.

A kapott eredmények alapján kifejlesztésre került egy olyan vezetőket támogató adatbázis és erre épülő információs rendszer, mely a tejtermelő vállalkozásokkal kapcsolatos döntések meghozatalában segítségül lehet. A kutatás nem zárul le, a kialakított rendszerre további

fejlesztések várnak, annak érdekében, hogy a vezetői döntéshozatalt még hatékonyabban tudja támogatni.

## Summary

---

Based on our researches and the literature we find that the analysis of decision-supporting systems should be the subject of timely and continuous development. The spread of information technology devices has not stopped; in addition to personal computers, numerous other new data and information managing devices have appeared in the work of agricultural businesses, including dairy companies. At the same time, the amount of information flowing to the manager is growing exponentially; therefore, the relevant data and information should be filtered keeping in mind the objectives of the company.

Our study has covered the decision-supporting systems of the Hungarian dairy businesses.

In Chapter 4 of this thesis the most wide-spread systems are presented, keeping in view the experiences gained during implementation in praxis, as well as the results achieved prior to the research and the historical processes that serve as basis for present development opportunities. In line with the target of the thesis, in Chapter 6 we explained what information systems the surveyed companies own and use, and what data and information sources the managers are using in their decision-making. During the in-depth interviews, we have gained a lot of information about the managers' habits, expectations and their needs for a new decision-supporting system.

Based on the received results, a management-supporting information system has been developed which can help make decisions regarding the dairy businesses. The research has not been completed yet; the created system is expected to be the subject of further developments to support the managers' decision-making even more effectively.

## 10. KÖSZÖNETNYILVÁNÍTÁS

---

Köszönetet szeretném kifejezni Dr. Csatai Rózsa egyetemi docens és Dr. Tenk Antal professor emeritus témavezetőimnek, akik iránymutató javaslataikkal segítettek a kutatómunkámat, a témaválasztással kezdődően a dolgozat végleges összeállításáig.

Köszönöm a Nyugat-magyarországi Egyetem Mezőgazdaság- és Élelmiszertudományi Kar, Újhelyi Imre Állattudományi Doktori Iskola vezetőinek és tagjainak, hogy kutatási témámat befogadták és a kutatáshoz szükséges szellemi támogatást biztosították.

Köszönöm mindazon tejtermelő vállalkozás vezetőinek és alkalmazottainak a segítő munkját, akik részt vettek a kutatásban.

Köszönöm a munkatársaimnak, a Gazdaságtudományi Intézet dolgozóinak, hogy közvetlenül vagy közvetve hozzájárultak a disszertációm elkészüléséhez, külön köszönet Dr. Hegyi Judit Intézetigazgató asszonynak, hogy az Intézetben minden feltételt biztosított a kutatás zavartalan lebonyolításához.

Végül, de nem utolsó sorban hálás vagyok feleségemnek és családomnak, akik támogattak, bíztak bennem és megértő türelemmel viseltettek irántam a Ph.D. értekezés elkészítése során.

## 11. IRODALOMJEGYZÉK

---

1. ÁT KFT. (2009): TALP kezelési útmutató, ver.:1.1.0.528, Gödöllő, 22-172. pp.
2. B. Richardson – R. Richardson (1992): Business Planning an approach to strategic management. Pitman Publishing, London, 2-5. pp.
3. Balogh V. (2008): Az élelmiszerfogyasztás tendenciáinak vizsgálata Magyarországon. Animal welfare, etológia és tartástechnológia, Gödöllő, 500-501.pp.)
4. Bartos A. (1967): Optimális döntés tejelési pótabrak összeállításához. Gazdálkodás, 1967, XI. évfolyam, 6. szám, Budapest, 21-27.pp.
5. Berde Cs. (2003): Menedzsment a mezőgazdaságban. Debrecen. Campus Kiadó. 237 p.
6. Blaskó B. (2011): World Importance and present tendencies of dairy sector. Applied Studies in Agribusiness and Commerce, Volume 05, Numbers 3-4, Budapest, 119-121.pp.
7. C. Edwards – J. Ward – A. Bytheway (1999): Az információs rendszerek alapjai. Panem kiadó, Budapest, 25-26.pp.
8. Chikán A. (2005): Vállalatgazdaságtan. Közgazdasági és Jogi Könyvkiadó – AULA, Budapest, 243-244 pp.
9. Citálás Turban 2005, Sátnáné, Döntéstámogató rendszerk 32 oldal
10. Csáki Cs. - Mátrai Z. (1980): Számítógépes irányítási rendszer kialakítása a bábolnai mezőgazdasági kombinátban. Gazdálkodás, XXIV. évfolyam, 7. szám, Budapest, 27-35.pp.
11. Csete L. - Nyújtó F. (1981): A számítástechnika mezőgazdasági alkalmazásának szakemberigénye. Gazdálkodás, XXV. évfolyam, 6. szám, Budapest, 69-72.pp.

12. Csete L. (1965): Mezőgazdasági számítások matematikai módszerekkel a mezőgazdaságban. Gazdálkodás, IX. évfolyam, 3. szám, Budapest, 89-90.pp.
13. DeLaval (2006): Programleírás, Box 39, SE-147 21 Tumba, SWEDEN, 13-123. pp.
14. D. Naranjo-Gil (2009): Management information systems and strategic performances: The role of top ream composition. International Journal of Information Management, Volume 29, Issue 3, 105-109. pp.
15. D.V. Kerr - R.T.Cowan - J. Chaseling (1999): DairyPRO-a knowledge-based decision suport system for strategic planning on sub-tropical dairy farms. I. System description, Agricultural Systems, Volume 59, Issue 3, 245-255.pp.
16. Dobay P. (1997): Vállalati információmenedzsment. Nemzeti tankönyvkiadó, Budapest, 87-88. pp.
17. E. A. Kallman – L. Reinharth (1983): Information Systems for planning and decision making. Van Nostrand Reinhold Company, New York, 129.p.
18. E. Babbie (2001): A társadalomtudományi kutatás gyakorlata. Balassi Kiadó, 2001, 316-319.pp.
19. Egri E. (2012): Élőállat és hús XV.évfolyam 4.szám, Agrárgazdasági Kutató Intézet, 5.p
20. Egri E. (2012): Élőállat és Hús, XV. évfolyam, 7. szám, Agrárgazdasági Kutató Intézet, 4.p.
21. Egri E. (2012): Élőállat és hús. Agrárpiaci jelentések , XV. Évf. 2.szám, 4.p.
22. Egri E. (2012): Élőállat és hús. Agrárpiaci jelentések –XV.évf. 13.szám, 5.p.
23. Egri E. (2012): Tej és tejtermékek. Agrárpiaci jelentések –XV. Évf. 1 szám, 5.p.

24. Ertsey I. (1978): Operációkutatás és számítástechnika a mezőgazdaságban. *Gazdálkodás*, XXII. évfolyam, 12. szám, Budapest, 65-67.pp.
25. F. Burstein - Clyde W. Holsapple (2008): *Handbook on Decision Support System*. Springer-Verlag, Berlin, 2008, 170-260 pp.
26. Fehér J. (1991): A számítógép a Baksai „Ezüstkalász” termelészövetkezetben. *Gazdálkodás*, XXXV. 12.43-44.pp.
27. Gorry A. – S. Morton (1971): A Framework for Information Systems. *Sloan Management Review*, 13, 1, 56-79.pp.
28. Hannon N. J. (2005): Making Data the Center of Your Information System. *Strategic Finance*. 87(4), pp. 55-61.)
29. Hegyi J.-Teschner G. (2010): Helyzetfelmérés a mezőgazdasági vállalkozások állattenyésztő tevékenységének informatikai hátteréről 364.p.
30. Herbert A. Simon (1977): *The New Science of management decision*. Prentice-Hall, Inc. Englewood Cliffs, 114. p.
31. Herbert A. Simon (1983): *Alternative vision of rationality*. Standford University Press, Standford, 7-34.pp.
32. Herdon M (2004): Információ az agrárgazdaságban. *Gazdálkodás*. XLVIII.1. 1-13 p.
33. Herdon M. (2004): Információtechnológia az agrárgazdaságban. *Gazdálkodás*, XLVII. évfolyam, 1. szám, Budapest, 6-7.pp.
34. HFTE-NÉBIH-ÁT Kft.(2012): *Tenyészetek, telepek megyei rangsora standard laktációs tejtermelésük alapján 2011.01.01.-2011.12.31.*
35. J. Elster (1986): *Rational Choice*. New York University Press, 1-11.pp.

36. J. H. Donelly - J. L. Gibson – J. M. Ivancevich (1992):  
Fundamentals of Management. USA, 1992, 581.p.
37. James G. March (2000): Bevezetés a döntéshozatalba. Panem  
Kiadó, Budapest, 230 p.
38. James G. March (2000): Bevezetés a döntéshozatalba. Panem  
Kiadó, Budapest, 235.p.
39. James H. - James L.-John M.(1992): Fundamentals of  
Management. Library of Congress, USA, 111.p.
40. Juhász S. (2011): Vállalati információs rendszerek műszaki  
alapjai. SZAK Kiadó, 416. p.)
41. K. C. Laudon – J. P. Laudon (2000): Management Information  
Systems, organization and technology in the networked  
enterprise. Prentice Hall, New Jersey, 112.p.
42. K.J. Radford (1978): Information Systems for strategic  
Decision. Reston Publishing Company, Reston, Virginia, 94-  
98.pp.
43. Kacsukné B. L. - Kiss T. (2007): Bevezetés az üzleti  
informatikába. 106. oldal, Akadémiai kiadó, Budapest)
44. Kiss E. - Székely Cs. (1975): A lineáris programozás vállalati  
alkalmazásáról. Gazdálkodás, XIX. évfolyam, 5. szám.,  
Budapest, 55-56.pp.
45. Klátyik J. (1969): Tervezés és információ a  
mezőgazdaságban.Gazdálkodás, XIII. évfolyam, 3. szám, 59-  
60.pp.
46. Kovács A. - Timon B. - Nagy J. (1967): Hálódigram eljárás  
alkalmazása a mezőgazdasági üzemszervezésben. Gazdálkodás,  
XI. évfolyam, 7. szám, Budapest, 1-11.pp.
47. Kovács A. (1969): A hálótervezési módszerek alapjai.  
Gazdálkodás, XIII. évfolyam, 10. szám, Budapest 9-21.pp.



48. Kovács K. (2004): Tejtermelő tehenészeti telepek számítógépes irányítása és ökonómiai vonatkozásai, Debreceni Egyetem, Debrecen, 18. p.
49. Kovácsné K. M. - Szomolányi F. (1986): A mikroelektronika alkalmazhatósága a mezőgazdasági vállalatok tevékenységében, Gazdálkodás, XXX. évfolyam, 9. szám, Budapest, 36-39.pp.
50. Kő A. - Lovrics L. (1997): Információ menedzsment, 1997, Aula kiadó, Budapest 427-428.pp.
51. L. W. Rue – L. L. Byars (1992): Management: Skills and application. Library of Congress Cataloging in Publication Data, USA, 55.p.
52. Louis E. Boone - David L. Kurtz (1992): Management. Library of Congress Cataloging in Publication Data, 490-495.pp.
53. Megyeri F. (1965): Az optimális programozás. Gazdálkodás, IX. évfolyam, 5. szám, Budapest, 94-95.pp.
54. Mester L. (1989): Ügyviteli számítógépes rendszerek bevezetésének lehetőségei. Gazdálkodás, 1989, XXXIII. évfolyam, 8. szám, Budapest, 72-74.pp.
55. Mészáros S. (1981): Számítógépek alkalmazása a mezőgazdaságban. Gazdálkodás, XXV. évfolyam, 10. szám, Budapest, 65-67.pp.
56. Mészáros S. (1981): Alapismeretek az operációkutatáshoz in Operációkutatási módszerek, Mezőgazdasági Kiadó, Budapest, 9-10.pp.
57. Monostori E. (1982): A számítástechnika alkalmazása az állami gazdaságokban, Gazdálkodás, XXVI. évfolyam, 1. szám, Budapest, 49-52.pp.
58. MVH Közlemény (2011): (177/2011. (XII.19.) számú MVH Közlemény). 1. számú melléklete, 1.p.

59. Nagy A. (2009):Jelentés a 111. Mezőgazdasági Piacok Közös Szervezésével Foglalkozó Irányító Bizottság – Állati termékek üléséről. FÖLDMŰVELÉSÜGYI ÉS VIDÉKFEJLESZTÉSI MINISZTERIUM AGRÁRPIACI FŐOSZTÁLY, 2.p.
60. Nyárs L. - Papp G. (2002): Az állati eredetű termékek feldolgozásának versenyhelyzete. Agrárgazdasági tanulmányok, 2002. 7. szám, Budapest, 35. p, 68-70.pp.
61. Nyitrai I. - Tóth I. (1982): A számítógépes készlet- és anyaggazdálkodás tapasztalatai egy mezőgazdasági nagyüzemben. Gazdálkodás, XXVI. évfolyam, 2. szám, Budapest, 17-20.pp.
62. Oláh E. (2011): Tájékoztató a tejtermelés és tejpiac helyzetéről, MOSZ, Budapest, 7.p.)
63. Ö. Ulusoy (1998): Transaction processing in distributed active real –time database systems. Journal of Systems and Software, Volue 42, 247-262.pp.
64. P. Adriaans, D. Zantinge (2002): Adatbányászat., Panem Könyvkiadó, 35.p.
65. Patkós I. (2007): A hazai tejtermelő tehenészetekben alkalmazott tartás- és üzemeltetés technológiák értékelése. Gazdálkodás 2007.5.51.évf. 1.p.
66. Pinhas G. - Gonda L. (2008): The role of information technology and data collection systems, in the improvement of milk production. Animal welfare, ethology and housing systems, Volume 4 Issue 2 Különszám, Gödöllő, 64-67.pp.
67. Pintér J. (1974): A több periódusú lineáris programozás módszerének alkalmazása egy mezőgazdasági vállalat középtávú tervezésében. Gazdálkodás, XVIII. évfolyam, 12. szám, Budapest, 19.p.

68. Popp J. (2010): Az állati eredetű termékek piaci kilátásai, Budapest, 26.p.
69. R. Artmann (1997): Sensor systems for milking robots. Computers and electronics in agriculture, 17. szám, 20-36. pp.
70. R. E. Freeman (1984): Strategic Management. Pitman Publishing, London, 8-11.pp.
71. Raffai M. (2003): Információrendszerek fejlesztése és menedzselése. Novadat Kiadó, 15-16.pp.
72. Raffai M. (2006): Az információ szerep, hatás, menedzsment. Palatia Nyomda és Kiadó, Budapest, 104-108. pp.
73. Raffai M. (2006): Az információ. Palatia Nyomda és Kiadó, 106 p.
74. Raymond, R.C. (1966): Use of the Time-sharing Computer in Business Planning and Budgeting. Management Science, 12, 8, 363-381.pp.
75. Roóz J. (2006): A menedzsment alapjai. Perfekt Zrt, Budapest, 85-86.pp.
76. S. Cooke, N. Slack (1991): Making Management Decision. Prentice Hall, 21-27.pp.
77. S. Devir, E. Maltz, J.H.M. Metz (1997): Strategic management planning and implementation at the milking robot dairy farm. Computers and electronics in agriculture, 17. szám, 95-110.pp.)
78. S. Herbert (1970): The New Science of Management, Management Decision Making, Penguin Books, 13-16.pp.
79. S. Morton (1971): Management Decision Systems; Computer-based suport for decision making. PhD. Dissertation, Graduate School of Business Administration, Harvard University
80. Salamon L. - Kettinger A. - Hegyi J. (2008): Tejtermelő gazdaságok vizsgálata a Nyugat-Dunántúlon. Animal welfare, etológia és tartástechnológia, Gödöllő, 565.p.

81. Sántáné-T. E. - Bíró M. - Gábor A. - Kő A. - Lovrics L. (2008):  
Döntéstámogató rendszerek. Panem Könyvkiadó, Budapest,  
2008, 13. p., 45-55.pp.
82. Systo Kft. (2008): Riska(c) telepirányítás, 12-102. pp.
83. Szabó A. (2001): Döntéstámogató rendszerek hazai  
alkalmazásai. Diplomamunka, Informatikusmérnöki és  
villamosmérnöki fakultás, Győr, 16. p.
84. Székely Cs. - Györök B. - Kovács A. - Pető I. - Szalay Zs. G.  
(2008): Az információ szerepe a hatékonyság növelésében.  
Hatékonyság a mezőgazdaságban, Agroinform Kiadó, 257-  
258.pp.
85. Székely Cs. (1986): A számítástechnika alkalmazása a  
mezőgazdasági vállalatok irányításának és tervezésének  
fejlesztésében. Gazdálkodás, XXX. évfolyam, 4. szám, 18-  
22pp.
86. Székely Cs. (2004): Versenyképes családi modellgazdaság.  
Agroinform kiadó, Gödöllő, 125-145.pp.
87. Székely Cs. - Györök B. - Kovács A. (2007): Kutatási  
zárójelentés, „Szimulációs modell fejlesztése a tejelő  
szarvasmarha ágazat döntéseinek támogatására” című tematikus  
OTKA pályázathoz, 20. p.
88. Szelényi E. (1987): A szarvasmarha-tenyésztési ágazat vállalati  
irányítási modellje. Gazdálkodás, XXXI. évfolyam, 9. szám,  
Budapest, 13-15.pp.
89. Széles Gy. (1993): Az állattenyésztés feszültségdijainak  
gazdasági megközelítése. Gazdálkodás, 37. évf. 5. sz. 1-14. pp.)
90. Szilágyi R. (2006): Mobil eszközökre épített internet  
alkalmazási lehetőségek és igények a mezőgazdaságban. Phd  
dolgozat, Debrecen, 94.p.

91. Sziray J. - Gaul G. (2007): Vállalati információs rendszerek I. Universitas-Győr Kht. , Győr, 54-55.pp.
92. Szűcs G. (2002): Döntéelmélet informatikai vonatkozásai. Alma Mater, BME GTK Információ és Tudásmenedzsment Tanszék, KFKI Számítástechnikai Csoport, Budapesti Műszaki egyetem kiadó, 119. p.
93. Szűcs I. - Pupos T. - Blaskó B. - Cégla B. - Kiss I. - Kovács K. - Lapis M. – Madai H. - Nagy A. Sz. - Szöllősi L. - Nábrádi A. (2010): Állattenyésztési ágazatok ökonómiája, MsC jegyzet, Debrecen, Keszthely, 2010, 140.p.
94. Takátsy T. (2000): Elektronika, mérés, mérnöki pontosság. Mezőgazdasági Szaktudás Kiadó Budapest
95. Teschner G. – CSATAI R. – GOMBKÖTŐ N. (2014): Operative decision-supporting information technology system for the companies of the cattle sector, Acta Ováriensis, MEGJELENÉS ALATT
96. Tóth J. (1964): Takarmányadagok összeállítása a lineáris programozás módszerével. Gazdálkodás, VIII. évfolyam, 1. szám, Budapest, 35-48.pp
97. Tóth J. (1985): A számítástechnika alkalmazásának eredményei. Gazdálkodás, XXIX. évfolyam, 8. szám, Budapest, 67-68.pp.
98. Tóth T. (1986): A számítástechnika helyzete a mezőgazdaságban. Gazdálkodás, XXX. évfolyam, 11. szám, Budapest, 55-56.pp.
99. Urban, G.L. (1967): SPRINTER: A Tool for New Products Decision Makers. Industrial Management Review, 8, 2, Spring, 43-54.pp.)

100. Vojtela T. - Bak J. - Fenyvesi L. (2011): Tehénészeti-irányítás informatikai nézőpontból, *Animal welfare, ethology and housing systems*, Volume 7, Issue 4, Gödöllő, 193-195. pp.
101. V.S. Janakiraman - K. Sarukesi (2006): *Decision Support Systems*. Prentice Hall of India, 19-24.pp.
102. Vágó Sz. (2005): A magyarországi tejpiac várható alakulása. *Gazdálkodás*, 2005.4.XLIX.évf. 18.p.
103. Xiong Ben-hai - LUO Qing-yao - Lü Jian-qiang - YANG Liang (2008): Study of Digital Management System of Milking Process on Large-Sized Dairy Farm, *Agricultural Sciences in China* ,8. szám, 1022-1028.pp.
104. Zoltayné P. Z. - Kindler J. kéziratának felhasználásával (2002): *Döntésmélet*. Alinea Kiadó, Budapest, 41.p.
105. URL 1: <http://www.tcd.ie/iis/policycoherence/eu-agricultural-policy-reform/dairy-case-study.php> Letöltve 2011.06.12.
106. URL 2: <http://fogyasztovedelem.info/node/145> Letöltve: 2012.06.01.
107. URL 3: <http://riska.hu> Letöltve: 2014.03.02.
108. <http://ec.europa.eu/eurostat> Letöltve: 2011.06.18.
109. <http://faostat.fao.org> Letöltve: 2013.02.05.
110. <http://www.ksh.hu> Letöltve: 2011.07.05.

## 12. Mellékletek

---

### 2. Melléklet: A mélyinterjú során használt vázlat

- Állat törzsadatok
  - Szakmai terület informatikai lefedettsége
  - Milyen állatokkal rendelkeznek?
  - Milyen korcsoportokra osztják az állatokat?
  - Milyen rendszert használnak az állomány nyilvántartására?
  - Hiányosságok az informatikai lefedettségben
  - Felmerülő igények
  - Elvárások az integrált rendszertől
  - Az adott terület informatikai kiszolgálásának fontossága
- Állattenyésztés és állatorvosi folyamatok
  - Szakmai terület informatikai lefedettsége
  - Hiányosságok az informatikai lefedettségben
  - Felmerülő igények
  - Elvárások az integrált rendszertől
  - Az adott terület informatikai kiszolgálásának fontossága
- Földnyilvántartási adatok
  - Szakmai terület informatikai lefedettsége
  - Milyen módon kezelik a földnyilvántartási adatokat?
  - Hogyan tartják nyilván a táblákat?
  - Hiányosságok az informatikai lefedettségben Felmerülő igények
  - Elvárások az integrált rendszertől
  - Az adott terület informatikai kiszolgálásának fontossága
- Hivatalos beszámolók
  - Szakmai terület informatikai lefedettsége

- Hiányosságok az informatikai lefedettségben
  - Felmerülő igények
  - Elvárások az integrált rendszertől
  - Az adott terület informatikai kiszolgálásának fontossága
- Jogosultságkezelés
  - Szakmai terület informatikai lefedettsége
  - Hiányosságok az informatikai lefedettségben
  - Felmerülő igények
  - Elvárások az integrált rendszertől
  - Az adott terület informatikai kiszolgálásának fontossága
- Kereskedelem
  - Szakmai terület informatikai lefedettsége
  - Hiányosságok az informatikai lefedettségben
  - Felmerülő igények
  - Elvárások az integrált rendszertől
  - Az adott terület informatikai kiszolgálásának fontossága
- Készletgazdálkodás
  - Szakmai terület informatikai lefedettsége
  - Hiányosságok az informatikai lefedettségben
  - Felmerülő igények
  - Elvárások az integrált rendszertől
  - Az adott terület informatikai kiszolgálásának fontossága
- Munkaerő
  - Szakmai terület informatikai lefedettsége
  - Hiányosságok az informatikai lefedettségben
  - Felmerülő igények
  - Elvárások az integrált rendszertől
  - Az adott terület informatikai kiszolgálásának fontossága
- Növénytermesztés
  - Szakmai terület informatikai lefedettsége



- Hiányosságok az informatikai lefedettségben
  - Felmerülő igények
  - Elvárások az integrált rendszertől
  - Az adott terület informatikai kiszolgáltatásának fontossága
- Specifikus hardverkapcsolatok
  - Szakmai terület informatikai lefedettsége
  - Hiányosságok az informatikai lefedettségben
  - Felmerülő igények
  - Elvárások az integrált rendszertől
  - Az adott terület informatikai kiszolgáltatásának fontossága
- Térinformatikai alrendszer
  - Szakmai terület informatikai lefedettsége
  - Hiányosságok az informatikai lefedettségben
  - Felmerülő igények
  - Elvárások az integrált rendszertől
  - Az adott terület informatikai kiszolgáltatásának fontossága
- Termelési adatok
  - Szakmai terület informatikai lefedettsége
  - Hiányosságok az informatikai lefedettségben
  - Felmerülő igények
  - Elvárások az integrált rendszertől
  - Az adott terület informatikai kiszolgáltatásának fontossága
- Termelőeszközök
  - Szakmai terület informatikai lefedettsége
  - Hiányosságok az informatikai lefedettségben
  - Felmerülő igények
  - Elvárások az integrált rendszertől
  - Az adott terület informatikai kiszolgáltatásának fontossága
- Tervezés, elemzés, döntéstámogatás
  - Szakmai terület informatikai lefedettsége

- Hiányosságok az informatikai lefedettségben
- Felmerülő igények
- Elvárások az integrált rendszertől
- Az adott terület informatikai kiszolgálásának fontossága

## 4. Melléklet:

### 4.1. Business Logic Layer

#### 4.1.1 CalculateValues.cs

```
using System;
using System.Collections.Generic;
using System.Text;

namespace Mezogazdasag.BL
{
    public class CalculateValues
    {
        public double AverageCount(double DevideBy, double value)
        {
            if (DevideBy > 0)
                return value / DevideBy;
            else
                return 0;
        }
        public double Minus(double val1, double val2)
        {
            return val2 - val1;
        }
        public double Multiply(double val1, double val2)
        {
            return val1 * val2;
        }
        public double sulygyarapodas(double val1, double
        devideBy)
        {
            if (devideBy > 0)
                return (val1 / devideBy) * 1000;
            else
                return 0;
        }
    }
}
```

```
publicdouble DiePercent(int elhullasdb, int
szaporulatdb)
    {
if (szaporulatdb > 0)
return (Convert.ToDouble(elhullasdb) /
Convert.ToDouble(szaporulatdb)) * 100;
else
return 0;
    }
publicdouble FajlagosAbrak(double AbrakFelhaszn, double
Sulygyarapodaskg)
    {
if (Sulygyarapodaskg > 0)
return (AbrakFelhaszn / Sulygyarapodaskg) * 100;
else
return 0;
    }
}
```

#### 4.1.2.PercentLogic.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace Mezogazdasag.BL
{
    publicclassPercentLogic
    {
        publicdouble CalculatePercent(double real, double
        devideByPlan)
        {
            if (devideByPlan != 0)
            {
                //return (value1 / devideBy) * 100;
                return (real * 100) / devideByPlan;
            }
            else
            return 0;
        }
    }
}
```

## 4.2. Data Access Layer

### 4.2.1. app.conf

```
<?xmlversion="1.0"encoding="utf-8" ?>
<configuration>
<configSections>
</configSections>
<connectionStrings>
<addname="Mezogazdasag.DAL.Properties.Settings.MezogazdasagConnectionString"
connectionString="Data
Source=TESCHNERGERGELY\SQLEXPRESS;Initial
Catalog=Mezogazdasag;Integrated Security=True;User
Instance=False;Context Connection=False"
providerName="System.Data.SqlClient" />
</connectionStrings>
</configuration>
```

#### 4.2.2. FokonyvProvider.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using Mezogazdasag.Models.Providers;
namespace Mezogazdasag.DAL
{
    publicstaticclass FokonyvProvider
    {
        publicstatic Fokonyv GetFokonyv(string FokonyvId, string
        ReszletezoId)
        {
            using (var dc =
            new MezogazdasagDataContext(ConfigProvider.ConnectionString))
                {var item = dc.GetTable<Fokonyv>().Where(f
                => f.FokonyvSzam == FokonyvId && f.ReszletezoSzam ==
                ReszletezoId).FirstOrDefault();
            return item;
                }
        }

        publicstatic Models.Models.Fokonyv GetFokonyvById(int
        id)
        {using (var dc =
        new MezogazdasagDataContext(ConfigProvider.ConnectionString))
            {var item = dc.GetTable<Fokonyv>().Where(f
            => f.Id == id).FirstOrDefault();
            return new Models.Models.Fokonyv {
            FokonyvNev=item.FokonyvNev.Name, ReszletezoNev =
            item.ReszletezoNev.Name,
            ReszletezoId=item.ReszletezoSzam.ToString(),
            FokonyvId=item.FokonyvSzam.ToString() };
            }
        }
    }
}

```

### 4.2.3. ImportedDataProvider.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using Mezogazdasag.Models.Providers;
using Mezogazdasag.BL;

namespace Mezogazdasag.DAL
{
    publicstaticclass ImportedDataProvider
    {
        #region insert methods
        publicstaticvoid AddEntity(ImportedRealData _entity)
        {
            using (var dc =
                newMezogazdasagDataContext(ConfigProvider.ConnectionString))
            {

                dc.GetTable<ImportedRealData>().InsertOnSubmit(_entity);
                dc.SubmitChanges();
            }
        }

        publicstaticvoid AddRealData(int fokonyvId, double
            egyenleg, double mennyiseg, int datab, int ag, DateTime?
            date)
        {
            using (var dc =
                newMezogazdasagDataContext(ConfigProvider.ConnectionString))
            {

                dc.GetTable<ImportedRealData>().InsertOnSubmit(CreateEnt
                    ity(fokonyvId, egyenleg, mennyiseg, datab, ag, date));
                dc.SubmitChanges();
            }
        }
    }
    #endregion
}

```



```

        #region select methods
publicstatic Models.Models.ImportedData
GetImportedRealDataByDate(DateTime _date, int _type)
    {
var convertedModel = new Models.Models.ImportedData();

using (var dc =
newMezogazdasagDataContext(ConfigProvider.ConnectionString))
    {
var items = dc.GetTable<ImportedRealData>().Where(r =>
r.ImportDate == _date).ToList();
//BL.AdditionalLogicForConversion conversionLogic = new
AdditionalLogicForConversion();

        convertedModel.typeId = _type;

foreach (var item in items)
    {
        #region tehenészet
if (_type == 1)
    {
if (item.FokonyvId == 191)
    {
        convertedModel.FoTermeK =
Convert.ToDouble(item.Mennyiseg);
    }

elseif (item.FokonyvId == 220 || item.FokonyvId == 221)
    {
        convertedModel.Ertekesített
+= item.Darab;
    }
elseif (item.FokonyvId == 190)
    {
        convertedModel.SzaporulatDb
= item.Darab;
        convertedModel.SzaporulatKg
= Convert.ToDouble(item.Mennyiseg);
    }

```

```

elseif (item.FokonyvId == 205)
    {
        convertedModel.KorosbitasDb
= item.Darab;
        convertedModel.Korosbitas =
Convert.ToDouble(item.Mennyiseg);
    }

elseif (item.FokonyvId == 233 || item.FokonyvId == 234)
    {
        convertedModel.ErtekesitesDb
+= item.Darab;
        convertedModel.ErtekesitesKg
+= Convert.ToDouble(item.Mennyiseg);

if (item.FokonyvId == 234)
    {

convertedModel.TehenKenyszerDb = item.Darab;

convertedModel.TehenKenyszerKg =
Convert.ToDouble(item.Mennyiseg);
    }
}

/*else if (item.FokonyvId == 234)
    {

}*/

elseif (item.FokonyvId == 236)
    {
        convertedModel.TakNapTehen =
Convert.ToDouble(item.Egyenleg);
    }

elseif (item.FokonyvId == 237)
    {
        convertedModel.TakNapBika =
Convert.ToDouble(item.Egyenleg);

```

```

    }

elseif (item.FokonyvId == 238)
    {

convertedModel.AbrakFelhasznalas =
Convert.ToDouble(item.Mennyiseg);
    }

if (item.Fokonyv.FokonyvSzam == "9612")
    {

convertedModel.TehenErtArbevetel +=
Convert.ToDouble(Math.Abs(item.Egyenleg));
    }

if (item.Fokonyv.FokonyvSzam == "91201" &&
item.Fokonyv.ReszletezoSzam != "902")
    {
        convertedModel.Arbevetel +=
Convert.ToDouble(item.Egyenleg);
    }

//if (item.Fokonyv.FokonyvSzam == "71201" &&
item.Fokonyv.ReszletezoSzam == "1")
//{
//convertedModel.TakNapTehen =
Convert.ToDouble(item.Egyenleg);
//}
    }
#endregion

#region növendék
elseif (_type==2)
    {
if (item.FokonyvId == 200)
    {

```

```
convertedModel.SulyGyarapodasKg =
Convert.ToDouble(item.Mennyiseg);
    }

elseif (item.FokonyvId == 205)
    {

convertedModel.TenyesztesbeAllitasDb = item.Darab;

convertedModel.TenyesztesbeAllasKg =
Convert.ToDouble(item.Mennyiseg);
    }

elseif (item.FokonyvId == 209)
    {
        convertedModel.ElhullasDb =
item.Darab;
        convertedModel.ElhullasKg =
Convert.ToDouble(item.Mennyiseg);
    }

//245
elseif (item.FokonyvId == 225)
    {
        convertedModel.ErtekesitesDb
= item.Darab;
        convertedModel.ErtekesitesKg
= Convert.ToDouble(item.Mennyiseg);
        convertedModel.Arbevetel =
Convert.ToDouble(Math.Abs(item.Egyenleg));
    }

elseif (item.FokonyvId == 239)
        convertedModel.TakNapTehen =
Convert.ToInt32(item.Egyenleg);

elseif (item.FokonyvId == 240)

convertedModel.AbrakFelhasznalas =
Convert.ToDouble(item.Mennyiseg);
```

```

    }
    #endregion

    #region hízó
elseif (_type == 3)
    {
    if (item.FokonyvId == 212)
        {

        convertedModel.SulyGyarapodasKg =
        Convert.ToDouble(item.Mennyiseg);
        }

    if (item.Fokonyv.FokonyvSzam == "91204" ||
        item.Fokonyv.FokonyvSzam == "912041")
        {
            convertedModel.ErtekesitesDb
            += item.Darab;
            convertedModel.ErtekesitesKg
            += Convert.ToDouble(item.Mennyiseg);
            convertedModel.Arbevetel +=
            Convert.ToDouble(item.Egyenleg);
        }

    elseif (item.FokonyvId == 246)
        {
            convertedModel.ElhullasDb =
            item.Darab;
            convertedModel.ElhullasKg =
            Convert.ToDouble(item.Mennyiseg);
        }

    elseif (item.FokonyvId == 241)
        convertedModel.TakNapTehen =
        Convert.ToDouble(item.Egyenleg);

    elseif (item.FokonyvId == 242)

```

```

convertedModel.AbrakFelhasznalas =
Convert.ToDouble(item.Mennyiseg);

        }
        #endregion

        #region borjú
elseif (_type == 4)
{
if (item.FokonyvId == 190)
{
convertedModel.SzaporulatDb
= item.Darab;
convertedModel.SzaporulatKg
= Convert.ToDouble(item.Mennyiseg);
}

elseif (item.FokonyvId == 247)
{

convertedModel.SulyGyarapodasKg =
Convert.ToDouble(item.Mennyiseg);
}

elseif (item.FokonyvId == 248)
{
convertedModel.ErtekesitesDb
= item.Darab;
convertedModel.ErtekesitesKg
= Convert.ToDouble(item.Mennyiseg);
convertedModel.Arbevetel =
Convert.ToDouble(item.Egyenleg);
}

elseif (item.FokonyvId == 197)
{
convertedModel.ElhullasDb =
item.Darab;

```

```

        convertedModel.Elhullaskg =
Convert.ToDouble(item.Mennyiseg);
    }

elseif (item.FokonyvId == 243)
        convertedModel.TakNapTehen =
Convert.ToDouble(item.Egyenleg);

elseif (item.FokonyvId == 194)

convertedModel.SulyGyarapodaskg =
Convert.ToDouble(item.Mennyiseg);
    }
    #endregion
}

}

return convertedModel;
}
#endregion
#region helpers
privatestaticImportedRealData CreateEntity(int
fokonyvId, double egyenleg, double mennyiseg, int datab,
int ag, DateTime? date)
{
returnnewImportedRealData { Ag = ag, Darab = datab,
Egyenleg = Convert.ToDecimal(egyenleg), FokonyvId =
fokonyvId, Mennyiseg = Convert.ToDecimal(mennyiseg),
ImportDate = (date==null) ?
newDateTime(DateTime.Today.Year, DateTime.Today.Month,
1) : date.Value};
}

#endregion
}
}

```

#### 4.2.4. Settings.cs

```
namespace Mezogazdasag.DAL.Properties {

    internalsealedpartialclass Settings {

        public Settings() {
            // // To add event handlers for saving and changing
            // settings, uncomment the lines below:
            //
            // this.SettingChanging +=
            this.SettingChangingEventHandler;
            //
            // this.SettingsSaving +=
            this.SettingsSavingEventHandler;
            //
        }

        private void SettingChangingEventHandler(object sender,
            System.Configuration.SettingChangingEventArgs e) {
            // Add code to handle the SettingChangingEvent event
            // here.
        }

        private void SettingsSavingEventHandler(object sender,
            System.ComponentModel.CancelEventArgs e) {
            // Add code to handle the SettingsSaving event here.
        }
    }
}
```



#### 4.2.5. TervProvider.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using Mezogazdasag.Models.Providers;
using Mezogazdasag.Models.Models;
using System.Data.Linq;

namespace Mezogazdasag.DAL
{
    publicstaticclass TervProvider
    {
        publicstatic List<Plan> GetAll()
        {
            var list = newList<Plan>();
            using (var dc =
                newMezogazdasagDataContext(ConfigProvider.ConnectionString))
            {
                var items = dc.GetTable<terv>();

                foreach (var item in items)
                {
                    list.Add(newPlan { Id = item.Id,
                        Type = item.AnimalType.Name });
                }
            }

            return list;
        }

        publicstatic Plan GetById(int id)
        {
            using (var dc =
                newMezogazdasagDataContext(ConfigProvider.ConnectionString))
            {
                var item = dc.GetTable<terv>().Where(t => t.Id ==
                    id).FirstOrDefault();
            }
        }
    }
}
```

```

return new Plan { Id = item.Id, Type =
item.AnimalType.Name };
    }
}

public static void SaveAPlan(Models.Models.Plan _entity,
int _type)
{
var entity = ConvertToDalEntity(new terv(), _entity,
_type);

using (var dc =
new MezogazdasagDataContext(ConfigProvider.ConnectionString))
{
dc.GetTable<terv>().InsertOnSubmit(entity);

dc.SubmitChanges();
}

public static bool isExist(int type, DateTime date)
{
using (var dc =
new MezogazdasagDataContext(ConfigProvider.ConnectionString))
{
var item = dc.GetTable<terv>().Where(p => p.TypeId == type
&& p.ValueDate == date).FirstOrDefault();

if (item != null)
return true;
else
return false;
}
}
}

```

```

publicstatic Models.Models.Plan GetByDate(DateTime date,
int type)
    {
using (var dc =
newMezogazdasagDataContext(ConfigProvider.ConnectionString))
    {
var item = dc.GetTable<terv>().Where(t => t.ValueDate ==
date && t.TypeId==type).FirstOrDefault();

if (item != null)

returnnewPlan
        {
            TakNapTehen =
Convert.ToInt32(item.tak_nap1),
            TakNapBika =
Convert.ToInt32(item.tak_nap2),
            Nap = item.napok_szama.Value,
            SzaporulatDb =
item.szaporulat_db.Value,
            FoTermek =
Convert.ToDouble(item.fo_termek_terv_1),
            ElhullasDb =
item.elhullas.Value,
            MinosegiTamogatasFt =
Convert.ToDouble(item.minosegi_tamogatas),
            ErtekesitesDb =
item.ertekesites.Value,
            Ertekesitett =
item.ertekesitett.Value,
            KorosbitasDb =
item.korosbitas.Value,
            ValueDate = date,
            Type =
item.AnimalType.Name.ToString(),
            typeId = item.TypeId,
            TenyesztesbeAllitasDb =
item.tenyesztesbeallas.Value,

```

```

                SulyGyarapodasKg =
Convert.ToDouble(item.gyarapodas),
                Id = item.Id
            };

else
returnnull;

        }
    }

publicstaticvoid UpdateById(Models.Models.Plan _entity)
    {
using (var dc =
newMezogazdasagDataContext(ConfigProvider.ConnectionStri
ng))
    {
var e = dc.GetTable<terv>().Where(t => t.Id ==
_entity.Id).FirstOrDefault();

var temp = ConvertToDalEntityt(e, _entity,
_entity.typeId);

//dc.GetTable<terv>().InsertOnSubmit(temp);
        dc.SubmitChanges();
    }
}

#region helper functiony
privatestaticterv ConvertToDalEntityt(terv e,
Models.Models.Plan _entity, int _type)
    {
        e.elhullas = _entity.ElhullasDb;
        e.fo_termek_terv_l =
Convert.ToInt32(_entity.FoTermek);
        e.korosbitas = _entity.KorosbitasDb;
        e.napok_szama = _entity.Nap;
        e.tak_nap1 =
_entity.TakNapTehen.ToString();
    }
}

```

```
        e.tak_nap2 =
_entity.TakNapBika.ToString();
        e.minosegi_tamogatas =
Convert.ToInt32(_entity.MinosegiTamogatasFt);
        e.szaporulat_db = _entity.SzaporulatDb;
        e.ertekesitett =
Convert.ToInt32(_entity.Ertekesitett);
        e.ertekesites = _entity.ErtekesitesDb;
        e.TypeId = _type;
        e.ValueDate = _entity.ValueDate;
        e.gyarapodas =
Convert.ToInt32(_entity.SulyGyarapodasKg);
        e.tenyesztesbeallas =
_entity.TenyesztesbeAllitasDb;
        e.Id = _entity.Id;

return e;
    }

    #endregion
}
}
```

## 4.3. Import Layer

### 4.3.1.ImportEngine.cs

```

using System;
using System.Collections.Generic;
using System.Text;
using Mezogazdasag.Models.Providers;
using Mezogazdasag.Models.Models;
using Mezogazdasag.DAL;
namespace Mezogazdasag.Import
{publicclass ImportEngine
    {
        #region Constructors
public ImportEngine()
    { }
public ImportEngine(string _file)
    {
        LoadFile(_file);
    }
        #endregion
publicvoid LoadFile(string _file)
    {FileProvider.LoadImportFile(_file);}
publicint[] DoImport(DateTime? date)
    {int[] exportedNumber = newint[2];
var ImprotList = RealDataProvider.GetList();
foreach (var item in ImprotList)
    {var FokonyEntity =
FokonyvProvider.GetFokonyv(item.Fokonyv, item.Reszl_sz);
if (FokonyEntity != null)
    {
DAL.ImportedDataProvider.AddRealData(FokonyEntity.Id,
item.Egyenleg, item.Mennyiseg, item.Darab, item.Ag,
date);}}
return exportedNumber;
    }
}
}

```

## 4.4. Models Layer

### 4.4.1. Fokony.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Mezogazdasag.Models.Models
{
    public class Fokonyv
    {
        public string FokonyvNev { get; set; }
        public string ReszletezoNev { get; set; }
        public string FokonyvId { get; set; }
        public string ReszletezoId { get; set; }
    }
}
```

#### 4.4.2. ImportedData.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Mezogazdasag.Models.Models
{
    public class ImportedData : Plan
    {
        public int TehenKenyszerDb { get; set; }
        public double TehenKenyszerKg { get; set; }

        public double TehenErtArbevetel { get; set; }

        public double AbrakFelhasznalas { get; set; }

        private double _TejArbevetelft_1;

        public double TejArbevetelft_1
        {
            get { return Math.Abs(_TejArbevetelft_1); }
            set { _TejArbevetelft_1 = value; }
        }

        private double _AtlagarTehenKg;
        public double AtlagarTehenKg
        {
            get { return Math.Abs(_AtlagarTehenKg); }
            set { _AtlagarTehenKg = value; }
        }

        private double _AtlagarTehenDb;
        public double AtlagarTehenDb
        {
            get { return Math.Abs(_AtlagarTehenDb); }
            set { _AtlagarTehenDb = value; }
        }
    }
}
```



```

publicdouble AtlagarSuly { get; set; }
publicdouble FajlagosAbrak { get; set; }

//ovveride int
publicnewdouble TakNapTehen { get; set; }
//override int
publicnewdouble TakNapBika { get; set; }
publicoverrideint Nap { get; set; }
publicoverridedouble Ertekesitett { get; set; }
publicoverridedouble FoTermek { get; set; }

privatedouble ar;

publicoverridedouble Arbevetel
    {
get { returnMath.Abs(ar); }
set { ar = value; }
    }

publicoverridedouble ErtekesitesKg { get; set; }
publicoverrideint ErtekesitesDb { get; set; }

publicoverridedouble SulyGyarapodasKg { get; set; }

privateint _TenyesztesbeAllitasDb;
publicoverrideint TenyesztesbeAllitasDb
    {
get { returnMath.Abs(_TenyesztesbeAllitasDb); }
set { _TenyesztesbeAllitasDb = value; }
    }

privatedouble _TenyesztesbeAllasKg;
publicoverridedouble TenyesztesbeAllasKg
    {
get { returnMath.Abs(_TenyesztesbeAllasKg); }
set { _TenyesztesbeAllasKg = value; }
    }

```

```

publicoverrideint ElhullasDb { get; set; }
publicoverridedouble ElhullasKg { get; set; }
publicoverrideint SzaporulatDb { get; set; }
publicoverridedouble SzaporulatKg { get; set; }
publicoverridedouble SulyGyarapodasGrTaknap { get; set;
}
publicoverridedouble AtlagAr { get; set; }
publicoverridedouble ElhullasSzazalek { get; set; }

publicoverridevoid Recalculate()
    {
if (typeId == 1)
    {
        AtlagarTehenKg =
calculateValues.AverageCount(ErtekesitesKg,
TehenErtArbevetel);
        FajlagosAbrak =
calculateValues.AverageCount(FoTermek,
AbrakFelhasznalas);
        AtlagosTehenLetszam =
calculateValues.AverageCount(Nap, TakNapTehen);
        ErtekesitettTejTehen =
calculateValues.AverageCount(AtlagosTehenLetszam,
Ertekesitett);
        FajlagosTej =
calculateValues.AverageCount(AtlagosTehenLetszam,
FoTermek);
        AtlagosBikaLetszam =
calculateValues.AverageCount(Nap, TakNapBika);
        TakarmanyTej =
calculateValues.Minus(FoTermek, Ertekesitett);
        TejArbevetelft_1 =
calculateValues.AverageCount(Ertekesitett, Arbevetel);
        AtlagarTehenDb =
calculateValues.AverageCount(ErtekesitesDb,
TehenErtArbevetel);
        AtlagarSuly =
calculateValues.AverageCount(ErtekesitesDb,
ErtekesitesKg);
    }
}

```

```
elseif (typeId == 2)
    {
        AtlagosTehenLetszam =
calculateValues.AverageCount(Nap, TakNapTehen);
        SulyGyarapodasGrTaknap =
calculateValues.sulygyarapodas(SulyGyarapodasKg,
TakNapTehen);
        AtlagAr =
calculateValues.AverageCount(ErtekesitesKg, Arbevetel);
        Elhullaskg =
calculateValues.Multiply(ElhullasDb, 350);
        FajlagosAbrak =
calculateValues.FajlagosAbrak(AbrakFelhasznalas,
SulyGyarapodasKg);
    }

elseif (typeId == 3)
    {
        AtlagosTehenLetszam =
calculateValues.AverageCount(Nap, TakNapTehen);
        SulyGyarapodasGrTaknap =
calculateValues.sulygyarapodas(SulyGyarapodasKg,
TakNapTehen);
        AtlagAr =
calculateValues.AverageCount(ErtekesitesKg, Arbevetel);
        FajlagosAbrak =
calculateValues.FajlagosAbrak(AbrakFelhasznalas,
SulyGyarapodasKg);
    }

elseif (typeId == 4)
    {
        AtlagosTehenLetszam =
calculateValues.AverageCount(Nap, TakNapTehen);
        SulyGyarapodasGrTaknap =
calculateValues.sulygyarapodas(SulyGyarapodasKg,
TakNapTehen);
        AtlagAr =
calculateValues.AverageCount(ErtekesitesKg, Arbevetel);
        ElhullasSzazalek =
calculateValues.DiePercent(ElhullasDb, SzaporulatDb);
```

```
                FajlagosAbrak =  
calculateValues.FajlagosAbrak(AbrakFelhasznalas,  
SulyGyarapodasKg);  
            }  
        }  
    }
```

### 4.4.3. Percent.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using Mezogazdasag.BL;

namespace Mezogazdasag.Models.Models
{
    publicclass Percent
    {
        publicstring TakNapTehen { get; set; }
        publicstring TakNapBika { get; set; }
        publicstring AtlagTehenLetszam { get; set; }
        publicstring AtlagBikaLetszam { get; set; }
        publicstring Fotermek { get; set; }
        publicstring ErtekesítettTej { get; set; }
        publicstring ErtekesítettTejTehen { get; set; }
        publicstring TakarmanyozasraFordítottTejL { get; set; }
        publicstring SzaporulatDb { get; set; }
        publicstring SzaporulatKg { get; set; }
        publicstring FajlagosTej { get; set; }
        publicstring TehenneKorosbitasDb { get; set; }
        publicstring TehenneKorosbitasKg { get; set; }
        publicstring ElhullasDb { get; set; }
        publicstring Arbevetel { get; set; }
        publicstring MinosegiTamogatas { get; set; }
        publicstring ErtekesítettTehenDb { get; set; }
        publicstring ErtekesítettTehenKg { get; set; }

        publicstring Sulygyarapodas { get; set; }
        publicstring TenyesztesbeAllitasDb { get; set; }
        publicstring TenyesztesbeAllitasKg { get; set; }
        publicstring ErtekesitesVagasraDb { get; set; }
        publicstring ErtekesitesVagasraKg { get; set; }
        publicstring ElhullasKg { get; set; }
        publicstring AtlagosNovendekLetszam { get; set; }
        publicstring SulygyarapodasGrTakNap { get; set; }
        publicstring Atlagar { get; set; }
        publicstring ElhullasSzazalek { get; set; }
    }
}

```

```
privatePlan p;  
privateImportedData r;  
privateint type;  
privatePercentLogic pLogic;  
  
public Percent(Plan _p, ImportedData _r, int _type)  
    {  
        pLogic = newPercentLogic();  
        p = _p;  
        r = _r;  
        type = _type;  
    }  
  
publicvoid ProcessValues()  
    {  
    if (type == 1)  
        {  
            TakNapTehen =  
pLogic.CalculatePercent(r.TakNapTehen,  
p.TakNapTehen).ToString("0.00");  
            TakNapBika =  
pLogic.CalculatePercent(r.TakNapBika,  
p.TakNapBika).ToString("0.00");  
            AtlagTehenLetszam =  
pLogic.CalculatePercent(r.AtlagosTehenLetszam,  
p.AtlagosTehenLetszam).ToString("0.00");  
            AtlagBikaLetszam =  
pLogic.CalculatePercent(r.AtlagosBikaLetszam,  
p.AtlagosBikaLetszam).ToString("0.00");  
            FotermeK =  
pLogic.CalculatePercent(r.FoTermek,  
p.FoTermek).ToString("0.00");  
            ErtekesitettTej =  
pLogic.CalculatePercent(r.Ertekesitett,  
p.Ertekesitett).ToString("0.00");  
            ErtekesitettTejTehen =  
pLogic.CalculatePercent(r.ErtekesitettTejTehen,  
p.ErtekesitettTejTehen).ToString("0.00");  
        }  
    }  
}
```

```

        TakarmanyozasraForditottTejL =
pLogic.CalculatePercent(r.TakarmanyTej,
p.TakarmanyTej).ToString("0.00");
        SzaporulatDb =
pLogic.CalculatePercent(r.SzaporulatDb,
p.SzaporulatDb).ToString("0.00");
        SzaporulatKg =
pLogic.CalculatePercent(r.SzaporulatKg,
p.SzaporulatKg).ToString("0.00");
        FajlagosTej =
pLogic.CalculatePercent(r.FajlagosTej,
p.FajlagosTej).ToString("0.00");
        TehenneKorosbitasDb =
pLogic.CalculatePercent(r.KorosbitasDb,
p.KorosbitasDb).ToString("0.00");
        TehenneKorosbitasKg =
pLogic.CalculatePercent(r.Korosbitas,
p.Korosbitas).ToString("0.00");
        ElhullasDb =
pLogic.CalculatePercent(r.ElhullasDb,
p.ElhullasDb).ToString("0.00");
        Arbevetel =
pLogic.CalculatePercent(r.Arbevetel,
p.Arbevetel).ToString("0.00");
        MinosegiTamogatas =
pLogic.CalculatePercent(r.MinosegiTamogatasFt,
p.MinosegiTamogatasFt).ToString("0.00");
        ErtekesitettTehenDb =
pLogic.CalculatePercent(r.ErtekesitesDb,
p.ErtekesitesDb).ToString("0.00");
        ErtekesitettTehenKg =
pLogic.CalculatePercent(r.ErtekesitesKg,
p.ErtekesitesKg).ToString("0.00");
    }

elseif (type == 2)
{
    TakNapTehen =
pLogic.CalculatePercent(r.TakNapTehen,
p.TakNapTehen).ToString("0.00");

```

```

        Sulygyarapodas =
pLogic.CalculatePercent(r.SulyGyarapodasKg,
p.SulyGyarapodasKg).ToString("0.00");
        TenyesztesbeAllitasDb =
pLogic.CalculatePercent(r.TenyesztesbeAllitasDb,
p.TenyesztesbeAllitasDb).ToString("0.00");
        ErtekesitesVagasraDb =
pLogic.CalculatePercent(r.ErtekesitesDb,
p.ErtekesitesDb).ToString("0.00");
        ElhullasDb =
pLogic.CalculatePercent(r.ElhullasDb,
p.ElhullasDb).ToString("0.00");
        AtlagosNovendekLetszam =
pLogic.CalculatePercent(r.AtlagosTehenLetszam,
p.AtlagosTehenLetszam).ToString("0.00");
        SulygyarapodasGrTakNap =
pLogic.CalculatePercent(r.SulyGyarapodasGrTaknap,
p.SulyGyarapodasGrTaknap).ToString("0.00");
        TenyesztesbeAllitasKg =
pLogic.CalculatePercent(r.TenyesztesbeAllasKg,
p.TenyesztesbeAllasKg).ToString("0.00");
        ErtekesitesVagasraKg =
pLogic.CalculatePercent(r.ErtekesitesKg,
p.ErtekesitesKg).ToString("0.00");
        Arbevetel =
pLogic.CalculatePercent(r.Arbevetel,
p.Arbevetel).ToString("0.00");
        Atlagar =
pLogic.CalculatePercent(r.AtlagAr,
p.AtlagAr).ToString("0.00");
        ElhullasKg =
pLogic.CalculatePercent(r.ElhullasKg,
p.ElhullasKg).ToString("0.00");
    }

elseif (type == 3)
    {
        TakNapTehen =
pLogic.CalculatePercent(r.TakNapTehen,
p.TakNapTehen).ToString("0.00");

```



```

        Sulygyarapodas =
pLogic.CalculatePercent(r.SulyGyarapodasKg,
p.SulyGyarapodasKg).ToString("0.00");
        ErtekesitesVagasraDb =
pLogic.CalculatePercent(r.ErtekesitesDb,
p.ErtekesitesDb).ToString("0.00");
        ElhullasDb =
pLogic.CalculatePercent(r.ElhullasDb,
p.ElhullasDb).ToString("0.00");
        AtlagosNovendekLetszam =
pLogic.CalculatePercent(r.AtlagosTehenLetszam,
p.AtlagosTehenLetszam).ToString("0.00");
        SulygyarapodasGrTakNap =
pLogic.CalculatePercent(r.SulyGyarapodasGrTaknap,
p.SulyGyarapodasGrTaknap).ToString("0.00");
        ErtekesitesVagasraKg =
pLogic.CalculatePercent(r.ErtekesitesKg,
p.ErtekesitesKg).ToString("0.00");
        Arbevetel =
pLogic.CalculatePercent(r.Arbevetel,
p.Arbevetel).ToString("0.00");
        Atlagar =
pLogic.CalculatePercent(r.Atlagar,
p.Atlagar).ToString("0.00");
        Elhullaskg =
pLogic.CalculatePercent(r.Elhullaskg,
p.Elhullaskg).ToString("0.00");
    }

elseif (type == 4)
    {
        TakNapTehen =
pLogic.CalculatePercent(r.TakNapTehen,
p.TakNapTehen).ToString("0.00");
        Sulygyarapodas =
pLogic.CalculatePercent(r.SulyGyarapodasKg,
p.SulyGyarapodasKg).ToString("0.00");
        SzaporulatDb =
pLogic.CalculatePercent(r.SzaporulatDb,
p.SzaporulatDb).ToString("0.00");
    }

```

```

        ErtekesitesVagasraDb =
pLogic.CalculatePercent(r.ErtekesitesDb,
p.ErtekesitesDb).ToString("0.00");
        ElhullasDb =
pLogic.CalculatePercent(r.ElhullasDb,
p.ElhullasDb).ToString("0.00");
        AtlagTehenLetszam =
pLogic.CalculatePercent(r.AtlagosTehenLetszam,
p.AtlagosTehenLetszam).ToString("0.00");
        SulygyarapodasGrTaknap =
pLogic.CalculatePercent(r.SulyGyarapodasGrTaknap,
p.SulyGyarapodasGrTaknap).ToString("0.00");
        SzaporulatKg =
pLogic.CalculatePercent(r.SzaporulatKg,
p.SzaporulatKg).ToString("0.00");
        ErtekesitesVagasraKg =
pLogic.CalculatePercent(r.ErtekesitesKg,
p.ErtekesitesKg).ToString("0.00");
        Arbevetel =
pLogic.CalculatePercent(r.Arbevetel,
p.Arbevetel).ToString("0.00");
        Atlagar =
pLogic.CalculatePercent(r.Atlagar,
p.Atlagar).ToString("0.00");
        ElhullasKg =
pLogic.CalculatePercent(r.ElhullasKg,
p.ElhullasKg).ToString("0.00");
        ElhullasSzazalek =
pLogic.CalculatePercent(r.ElhullasSzazalek,
p.ElhullasSzazalek).ToString("0.00");
    }
}
}
}

```

#### 4.4.4. Plan.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using Mezogazdasag.BL;

namespace Mezogazdasag.Models.Models
{
    publicclass Plan
    {
        protectedCalculateValues calculateValues;

        #region private members

        protectedint _TakNapTehen=0;
        protectedint _TakNapBika = 0;
        protectedint _Nap = 0;
        protecteddouble _Ertekesitett = 0;
        protecteddouble _FoTermek = 0;
        protectedint _SzorulatDb = 0;
        protectedint _KorosbitasDb = 0;
        protecteddouble _SulyGyarapodaskg = 0;
        protectedint _TenyeszetsbeAllitasDb = 0;
        protectedint _ErtekesitesDb = 0;
        protectedint _ElhullasDb = 0;
        protecteddouble _Arbevetel=0;

        #endregion

        publicint Id { get; set; }
        publicstring Type { get; set; }
        publicint typeId { get; set; }
        publicDateTime ValueDate { get; set; }

        publicvirtualint TakNapTehen
        {
            get
            {
                return _TakNapTehen;
            }
        }
    }
}

```

```
    }  
  
set  
    {  
        _TakNapTehen = value;  
  
if (typeId == 1)  
    {  
        AtlagosTehenLetszam =  
calculateValues.AverageCount(_Nap, _TakNapTehen);  
        ErtekesitettTejTehen =  
calculateValues.AverageCount(AtlagosTehenLetszam,  
_Ertekesitett);  
        FajlagosTej =  
calculateValues.AverageCount(AtlagosTehenLetszam,  
_FoTermek);  
    }  
  
elseif (typeId == 2)  
    {  
        AtlagosTehenLetszam =  
calculateValues.AverageCount(_Nap, _TakNapTehen);  
        SulyGyarapodasGrTaknap =  
calculateValues.sulygyarapodas(_SulyGyarapodasKg,  
_TakNapTehen);  
    }  
  
elseif (typeId == 3)  
    {  
        AtlagosTehenLetszam =  
calculateValues.AverageCount(_Nap, _TakNapTehen);  
        SulyGyarapodasGrTaknap =  
calculateValues.sulygyarapodas(_SulyGyarapodasKg,  
_TakNapTehen);  
    }  
  
elseif (typeId == 4)  
    {  
        AtlagosTehenLetszam =  
calculateValues.AverageCount(_Nap, _TakNapTehen);  
    }  
}
```

```

        SulyGyarapodasGrTaknap =
        calculateValues.sulygyarapodas(_SulyGyarapodasKg,
        _TakNapTehen);
    }
}

publicvirtualint TakNapBika
{
get
    {
return _TakNapBika;
    }

set
    {
        _TakNapBika = value;

if (typeId == 1)
    {
        AtlagosBikaLetszam =
        calculateValues.AverageCount(_Nap, _TakNapBika);
    }
}

publicvirtualint Nap
{
get
    {
return _Nap;
    }

set
    {
        _Nap = value;

if (typeId == 1)
    {
        AtlagosTehenLetszam =
        calculateValues.AverageCount(_Nap, _TakNapTehen);
    }
}

```

```

        AtlagosBikaLetszam =
calculateValues.AverageCount(_Nap, _TakNapBika);
        ErtekesitettTejTehen =
calculateValues.AverageCount(AtlagosTehenLetszam,
        _Ertekesitett);
        FajlagosTej =
calculateValues.AverageCount(AtlagosTehenLetszam,
        _FoTermek);
    }

elseif (typeId == 2)
    {
        AtlagosTehenLetszam =
calculateValues.AverageCount(_Nap, _TakNapTehen);
    }

elseif (typeId == 3)
    {
        AtlagosTehenLetszam =
calculateValues.AverageCount(_Nap, _TakNapTehen);
    }

elseif (typeId == 4)
    {
        AtlagosTehenLetszam =
calculateValues.AverageCount(_Nap, _TakNapTehen);
    }
}

publicvirtualdouble FoTermek
{
get
    {
return _FoTermek;
    }

set
    {
        _FoTermek = value;
    }
}

```

```

if (typeId == 1)
    {
        TakarmanyTej =
calculateValues.Minus(_FoTermek, _Ertekesitett);
        FajlagosTej =
calculateValues.AverageCount(AtlagosTehenLetszam,
_FoTermek);
    }
}

publicvirtualdouble Ertekesitett
//liter
{
get
    {
return _Ertekesitett;
    }

set
    {
        _Ertekesitett = value;
    }

if (typeId == 1)
    {
        ErtekesitettTejTehen =
calculateValues.AverageCount(AtlagosTehenLetszam,
_Ertekesitett);
        TakarmanyTej =
calculateValues.Minus(_FoTermek, _Ertekesitett);
        Arbevetel =
calculateValues.Multiply(_Ertekesitett, 94);
    }
}

publicvirtualint SzaporulatDb
{
get

```

```

        {
return  _SzaporulatDb;
        }

set
        {
            _SzaporulatDb = value;

if (typeId == 1)
            {
                SzaporulatKg =
calculateValues.Multiply(_SzaporulatDb, 36);
            }

elseif (typeId == 4)
            {
                SzaporulatKg =
calculateValues.Multiply(_SzaporulatDb, 36);
            }
        }

public int KorosbitasDb
    {
get
        {
return Math.Abs(_KorosbitasDb);
        }

set
        {
            _KorosbitasDb = value;

if (typeId == 1)
            {
                Korosbitas =
calculateValues.Multiply(_KorosbitasDb, 580);
            }
        }
    }

```



```

publicvirtualint ElhullasDb
//Tehén elhullás, Növendék elhullás
    {
get
    {
return _ElhullasDb;
    }

set
    {
        _ElhullasDb = value;

if (typeId == 2)
    {
        ElhullasKg =
calculateValues.Multiply(_ElhullasDb, 350);
    }

elseif (typeId == 3)
    {
        ElhullasKg =
calculateValues.Multiply(_ElhullasDb, 250);
    }

elseif (typeId == 4)
    {
        ElhullasKg =
calculateValues.Multiply(_ElhullasDb, 50);
        ElhullasSzazalek =
calculateValues.DiePercent(_ElhullasDb, _SzaporulatDb);
    }
    }
}

publicdouble MinosegiTamogatasFt { get; set; }

publicvirtualint ErtekesitesDb
//Tehén értékesítés, Értékesítés vágásra
    {
get
    {

```

```
return _ErtekesitesDb;
    }

set
    {
        _ErtekesitesDb = value;

if (typeId == 1)
    {
        ErtekesitesKg =
calculateValues.Multiply(_Ertekesitett, 560);
        ErtekesitesFt =
calculateValues.Multiply(ErtekesitesKg, 260);
    }

if (typeId == 2)
    {
        ErtekesitesKg =
calculateValues.Multiply(_ErtekesitesDb, 505);
        Arbevétel =
calculateValues.Multiply(ErtekesitesKg, 260);
        AtlagAr =
calculateValues.AverageCount(ErtekesitesKg, Arbevétel);
    }

elseif (typeId == 3)
    {
        ErtekesitesKg =
calculateValues.Multiply(_ErtekesitesDb, 540);
        Arbevétel =
calculateValues.Multiply(ErtekesitesKg, 520);
        AtlagAr =
calculateValues.AverageCount(ErtekesitesKg, Arbevétel);
    }

elseif (typeId == 4)
    {
        ErtekesitesKg =
calculateValues.Multiply(_ErtekesitesDb, 1);
        Arbevétel =
calculateValues.Multiply(ErtekesitesKg, 1);
```

```

        AtlagAr =
calculateValues.AverageCount(ErtekesitesKg, Arbevetel);
    }
}

publicvirtualdouble SulyGyarapodasKg
{
get
    {
return _SulyGyarapodasKg;
    }

set
    {
        _SulyGyarapodasKg = value;

if (typeId == 2)
    {
        SulyGyarapodasGrTaknap =
calculateValues.sulygyarapodas(_SulyGyarapodasKg,
_TakNapTehen);
    }

elseif (typeId == 3)
    {
        SulyGyarapodasGrTaknap =
calculateValues.sulygyarapodas(_SulyGyarapodasKg,
_TakNapTehen);
    }

elseif (typeId == 4)
    {
        SulyGyarapodasGrTaknap =
calculateValues.sulygyarapodas(_SulyGyarapodasKg,
_TakNapTehen);
    }
}

publicvirtualint TenyesztesbeAllitasDb

```

```

        {
get
        {
return _TenyeszetsbeAllitasDb;
        }

set
        {
            _TenyeszetsbeAllitasDb = value;

if (typeId == 2)
        {
            TenyesztesbeAllasKg =
calculateValues.Multiply(_TenyeszetsbeAllitasDb, 580);
        }
        }
    }

//számolt mezők

publicdouble AtlagosTehenLetszam { get; set; } //Átlagos
növendék létszám, Átlagos hizó létszám, Átlagos borjú
létszám
publicdouble AtlagosBikaLetszam { get; set; }
publicdouble ErtekesítettTejTehen { get; set; }

privatedouble _TakarmanyTej ;
publicdouble TakarmanyTej
    {
get { returnMath.Abs(_TakarmanyTej); }
set { _TakarmanyTej = value; }
    }

publicvirtualdouble SzaporulatKg { get; set; }
//Borjúsaporulat, szaporulat
publicdouble FajlagosTej { get; set; }

privatedouble _Korosbitas;
publicdouble Korosbitas
    {
get { returnMath.Abs(_Korosbitas); }
    }

```

```

set { _Korosbitas = value; }
    }

publicvirtualdouble Arbevetel
    {
get { returnMath.Abs(_Arbevetel); }
set { _Arbevetel = value; }
    }

publicvirtualdouble ErtekesitesKg {get;set;}
//Értékesítés vágásra, Tehén értékesítés
publicdouble ErtekesitesFt { get; set; }
publicvirtualdouble SulyGyarapodasGrTaknap { get; set; }
publicvirtualdouble TenyesztesbeAllaskg { get; set; }
publicvirtualdouble AtlagAr { get; set; }
publicvirtualdouble Elhullaskg { get; set; }
publicvirtualdouble ElhullasSzazalek { get; set; }

public Plan()
    {
        calculateValues = newCalculateValues();
    }

publicCalculateValues GetInstance()
    {
return calculateValues;
    }

publicvirtualvoid Recalculate()
    {
if (typeId == 1)
    {
        AtlagosTehenLetszam =
calculateValues.AverageCount(_Nap, _TakNapTehen);
        ErtekesitettTejTehen =
calculateValues.AverageCount(AtlagosTehenLetszam,
_Ertekesitett);
        FajlagosTej =
calculateValues.AverageCount(AtlagosTehenLetszam,
_FoTermek);
    }
    }

```

```

        AtlagosBikaLetszam =
calculateValues.AverageCount(_Nap, _TakNapBika);
        TakarmanyTej =
calculateValues.Minus(_FoTermek, _Ertekesitett);
        Arbevetel =
calculateValues.Multiply(_Ertekesitett, 94);
        ErtekesitesKg =
calculateValues.Multiply(_ErtekesitesDb, 560);
        ErtekesitesFt =
calculateValues.Multiply(ErtekesitesKg, 260);
        SzaporulatKg =
calculateValues.Multiply(_SzaporulatDb, 36);
        Korosbitas =
calculateValues.Multiply(_KorosbitasDb, 580);
    }

elseif (typeId == 2)
    {
        AtlagosTehenLetszam =
calculateValues.AverageCount(_Nap, _TakNapTehen);
        SulyGyarapodasGrTaknap =
calculateValues.sulygyarapodas(_SulyGyarapodasKg,
_TakNapTehen);
        ElhullasKg =
calculateValues.Multiply(_ElhullasDb, 350);
        ErtekesitesKg =
calculateValues.Multiply(_ErtekesitesDb, 505);
        Arbevetel =
calculateValues.Multiply(ErtekesitesKg, 260);
        AtlagAr =
calculateValues.AverageCount(ErtekesitesKg, Arbevetel);
        TenyesztesbeAllasKg =
calculateValues.Multiply(_TenyeszetsbeAllitasDb, 580);
    }

elseif (typeId == 3)
    {
        AtlagosTehenLetszam =
calculateValues.AverageCount(_Nap, _TakNapTehen);

```



#### 4.4.5. RealData.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Mezogazdasag.Models.Models
{
    public class RealData
    {
        public int Ag { get; set; }
        public string Fokonyv { get; set; }
        public string FokonyNev { get; set; }
        public string Reszl_sz { get; set; }
        public string ReszletezoNev { get; set; }
        public double Egyenleg { get; set; }
        public double Mennyiseg { get; set; }
        public int Darab { get; set; }
    }
}
```



#### 4.4.6. ConfigProvider.cs

```
using System;
using System.Collections.Generic;
using System.Text;
using System.Configuration;

namespace Mezogazdasag.Models.Providers
{
    public static class ConfigProvider
    {
        public static string ConnectionString
        {
            get {
                return ConfigurationManager.ConnectionStrings["OnlineDb"]
                    .ConnectionString; }
        }

        public static string Password
        {
            get {
                return ConfigurationManager.AppSettings["Password"].ToString(); }
        }

        public static string[] ManualDataId
        {
            get
            {
                return ConfigurationManager.AppSettings["ManualDataIds"]
                    .Split(';'); }
        }
    }
}
```

#### 4.4.7. FileProvider.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.IO;

namespace Mezogazdasag.Models.Providers
{
    publicstaticclass FileProvider
    {
        publicstaticvoid LoadImportFile(string _path)
        {
            Providers.RealDataProvider.EmptyDataList();
            using (TextReader stream = newStreamReader(_path,
                Encoding.Default))
            {
                string line;

                while ((line = stream.ReadLine()) != null)
                {
                    Providers.RealDataProvider.AddEntity(line);
                }
            }
        }
    }
}
```

#### 4.4.8. RealDataProvider.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace Mezogazdasag.Models.Providers
{publicstaticclassRealDataProvider
    {privatestaticList<Models.RealData> realList =
newList<Models.RealData>();
publicstaticvoid AddEntity(string line)
    {string[] sLine = SplitString(line);
    realList.Add(new Models.RealData
    {
        Ag = Convert.ToInt32(sLine[0]),
        Fokonyv = sLine[1],
        FokonyNev = sLine[2],
        Reszl_sz = sLine[3],
        ReszletezoNev = sLine[4],
        Egyenleg = Convert.ToDouble(sLine[5]),
        Mennyiseg = Convert.ToDouble(sLine[6]),
        Darab = Convert.ToInt32(sLine[7])
    });}
publicstaticvoid EmptyDataList()
    {
    realList.Clear();
    }
publicstaticList<Models.RealData> GetList()
    {
return realList;
    }
    #region helper methods
privatestaticstring[] SplitString(string text)
    {return text.Split(';');
    }
    #endregion
}
}

```

## 4.5. User Interface Layer

### 4.5.1. ManualData.Designer.cs

```

namespace Mezogazdasag.UI
{
    partialclass ManualData
    {
        ///<summary>
        /// Required designer variable.
        ///</summary>
        private System.ComponentModel.IContainer components =
        null;

        ///<summary>
        /// Clean up any resources being used.
        ///</summary>
        ///<param name="disposing">true if managed resources
        should be disposed; otherwise, false.</param>
        protectedoverridevoid Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Windows Form Designer generated code

        ///<summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        ///</summary>
        privatevoid InitializeComponent()
        {
            this.textBox1 = new System.Windows.Forms.TextBox();
            this.textBox2 = new System.Windows.Forms.TextBox();
            this.textBox3 = new System.Windows.Forms.TextBox();
            this.label1 = new System.Windows.Forms.Label();
            this.label2 = new System.Windows.Forms.Label();
            this.label3 = new System.Windows.Forms.Label();
            this.label4 = new System.Windows.Forms.Label();
        }
    }
}

```

```
this.button1 = new System.Windows.Forms.Button();
this.SuspendLayout();
//
// textBox1
//
this.textBox1.Font = new System.Drawing.Font("Comic Sans
MS", 12F, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, ((byte)0));
this.textBox1.Location = new System.Drawing.Point(115,
136);
this.textBox1.Name = "textBox1";
this.textBox1.Size = new System.Drawing.Size(192, 30);
this.textBox1.TabIndex = 0;
//
// textBox2
//
this.textBox2.Font = new System.Drawing.Font("Comic Sans
MS", 12F, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, ((byte)0));
this.textBox2.Location = new System.Drawing.Point(115,
172);
this.textBox2.Name = "textBox2";
this.textBox2.Size = new System.Drawing.Size(192, 30);
this.textBox2.TabIndex = 1;
//
// textBox3
//
this.textBox3.Font = new System.Drawing.Font("Comic Sans
MS", 12F, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, ((byte)0));
this.textBox3.Location = new System.Drawing.Point(115,
208);
this.textBox3.Name = "textBox3";
this.textBox3.Size = new System.Drawing.Size(192, 30);
this.textBox3.TabIndex = 2;
//
// label1
//
this.label1.AutoSize = true;
this.label1.Font = new System.Drawing.Font("Comic Sans
MS", 12F, System.Drawing.FontStyle.Bold);
```

```
this.label1.Location = new System.Drawing.Point(12,
139);
this.label1.Name = "label1";
this.label1.Size = new System.Drawing.Size(84, 23);
this.label1.TabIndex = 3;
this.label1.Text = "Egyenleg:";
//
// label2
//
this.label2.AutoSize = true;
this.label2.Font = new System.Drawing.Font("Comic Sans
MS", 12F, System.Drawing.FontStyle.Bold);
this.label2.Location = new System.Drawing.Point(12,
175);
this.label2.Name = "label2";
this.label2.Size = new System.Drawing.Size(95, 23);
this.label2.TabIndex = 4;
this.label2.Text = "Mennyiség:";
//
// label3
//
this.label3.AutoSize = true;
this.label3.Font = new System.Drawing.Font("Comic Sans
MS", 12F, System.Drawing.FontStyle.Bold);
this.label3.Location = new System.Drawing.Point(12,
211);
this.label3.Name = "label3";
this.label3.Size = new System.Drawing.Size(65, 23);
this.label3.TabIndex = 5;
this.label3.Text = "Darab:";
this.label4.Font = new System.Drawing.Font("Comic Sans
MS", 12F, System.Drawing.FontStyle.Bold);
this.label4.Location = new System.Drawing.Point(17, 9);
this.label4.Name = "label4";
this.label4.Size = new System.Drawing.Size(290, 113);
this.label4.TabIndex = 6;
this.label4.TextAlign =
System.Drawing.ContentAlignment.MiddleCenter;
//
// button1
//
```

```

this.button1.Font = new System.Drawing.Font("Comic Sans
MS", 12F, System.Drawing.FontStyle.Bold);
this.button1.Location = new System.Drawing.Point(16,
250);
this.button1.Name = "button1";
this.button1.Size = new System.Drawing.Size(291, 28);
this.button1.TabIndex = 7;
this.button1.Text = "O.K.";
this.button1.UseVisualStyleBackColor = true;
this.button1.Click += new
System.EventHandler(this.button1_Click);
this.AutoScaleDimensions = new System.Drawing.SizeF(6F,
13F);
this.AutoScaleMode =
System.Windows.Forms.AutoScaleMode.Font;
this.ClientSize = new System.Drawing.Size(324, 286);
this.ControlBox = false;
this.Controls.Add(this.button1);
this.Controls.Add(this.label4);
this.Controls.Add(this.label3);
this.Controls.Add(this.label2);
this.Controls.Add(this.label1);
this.Controls.Add(this.textBox3);
this.Controls.Add(this.textBox2);
this.Controls.Add(this.textBox1);
this.FormBorderStyle =
System.Windows.Forms.FormBorderStyle.FixedSingle;
this.Name = "ManualData";
this.StartPosition =
System.Windows.Forms.FormStartPosition.CenterScreen;
this.Text = "Manuális adatok";
this.Load += new
System.EventHandler(this.ManualData_Load);
this.ResumeLayout(false);
this.PerformLayout();
    }
    #endregion
private System.Windows.Forms.TextBox textBox1;
private System.Windows.Forms.TextBox textBox2;
private System.Windows.Forms.TextBox textBox3;
private System.Windows.Forms.Label label1;

```

```
private System.Windows.Forms.Label label2;  
private System.Windows.Forms.Label label3;  
private System.Windows.Forms.Label label4;  
private System.Windows.Forms.Button button1;  
    }  
}
```



#### 4.5.2. Password.Designer.cs

```

namespace Mezogazdasag.UI
{
    partialclass Password
    {
        ///<summary>
        /// Required designer variable.
        ///</summary>
        private System.ComponentModel.IContainer components =
            null;

        ///<summary>
        /// Clean up any resources being used.
        ///</summary>
        ///<param name="disposing">true if managed resources
        should be disposed; otherwise, false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Windows Form Designer generated code

        ///<summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        ///</summary>
        private void InitializeComponent()
        {
            this.label1 = new System.Windows.Forms.Label();
            this.textBox1 = new System.Windows.Forms.TextBox();
            this.button1 = new System.Windows.Forms.Button();
            this.button2 = new System.Windows.Forms.Button();
            this.SuspendLayout();
            //
            // label1
            //
    }
}

```

```
this.label1.AutoSize = true;
this.label1.Font = new System.Drawing.Font("Times New
Roman", 12F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)0));
this.label1.Location = new System.Drawing.Point(12, 9);
this.label1.Name = "label1";
this.label1.Size = new System.Drawing.Size(48, 19);
this.label1.TabIndex = 0;
this.label1.Text = "Jelszó:";
//
// textBox1
//
this.textBox1.Font = new System.Drawing.Font("Times New
Roman", 12F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)0));
this.textBox1.Location = new System.Drawing.Point(66,
6);
this.textBox1.Name = "textBox1";
this.textBox1.Size = new System.Drawing.Size(206, 26);
this.textBox1.TabIndex = 1;
this.textBox1.UseSystemPasswordChar = true;
//
// button1
//
this.button1.Location = new System.Drawing.Point(12,
50);
this.button1.Name = "button1";
this.button1.Size = new System.Drawing.Size(269, 27);
this.button1.TabIndex = 2;
this.button1.Text = "O.K.";
this.button1.UseVisualStyleBackColor = true;
this.button1.Click += new
System.EventHandler(this.button1_Click);
//
// button2
//
this.button2.Location = new System.Drawing.Point(12,
83);
this.button2.Name = "button2";
this.button2.Size = new System.Drawing.Size(269, 27);
this.button2.TabIndex = 3;
```

```
this.button2.Text = "Mégsem";
this.button2.UseVisualStyleBackColor = true;
this.button2.Click += new
System.EventHandler(this.button2_Click);
//
// Password
//
this.AutoScaleDimensions = new System.Drawing.SizeF(6F,
13F);
this.AutoScaleMode =
System.Windows.Forms.AutoScaleMode.Font;
this.ClientSize = new System.Drawing.Size(284, 119);
this.Controls.Add(this.button2);
this.Controls.Add(this.button1);
this.Controls.Add(this.textBox1);
this.Controls.Add(this.label1);
this.FormBorderStyle =
System.Windows.Forms.FormBorderStyle.FixedSingle;
this.MaximizeBox = false;
this.MinimizeBox = false;
this.Name = "Password";
this.StartPosition =
System.Windows.Forms.FormStartPosition.CenterScreen;
this.Text = "Admin jelszó";
this.ResumeLayout(false);
this.PerformLayout();

    }

    #endregion

private System.Windows.Forms.Label label1;
private System.Windows.Forms.TextBox textBox1;
private System.Windows.Forms.Button button1;
private System.Windows.Forms.Button button2;
    }
}
```

### 4.5.3. Program.cs

```
using System;
using System.Collections.Generic;
using System.Windows.Forms;

namespace Mezogazdasag.UI
{
    staticclass Program
    {
        ///<summary>
        /// The main entry point for the application.
        ///</summary>
        [STAThread]
        staticvoid Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(newMezogazdasag());
        }
    }
}
```