

EÖTVÖS LORÁND TUDOMÁNYEGYETEM

TERMÉSZETTUDOMÁNYI KAR

SZAKDOLGOZAT

Mészáros Zoltán

2004



EÖTVÖS LORÁND TUDOMÁNYEGYETEM
TERMÉSZETTUDOMÁNYI KAR
INFORMATIKA TANSZÉKCSOPORT

**ÉPÜLET VEZÉRLÉSE
(INTELLHOUSE)**
Szakdolgozat

Témavezető: Szabadhegyi Csaba
ELTE Informatikai
Szakmódszertani Csoport

Készítette: Mészáros Zoltán
Programozó matematikus
szak

Budapest
2004

TARTALOM

BEVEZETÉS

A feladat rövid leírása, motivációja	7
Gazdaságosság	8
Integráció	8
Biztonságtechnika	9
Dolgozat témája és eredménye	10
Dolgozat felépítése	10
A feladat megoldása során felhasznált eszközök	10

I. FELHASZNÁLÓI DOKUMENTÁCIÓ

TERVEZŐ FELHASZNÁLÓI DOKUMENTÁCIÓ	13
---	----

I. Installálás

Szükséges hardver konfiguráció	13
Szükséges szoftver konfiguráció	13
Installálás	13

II. Menüszerkezet

1. <i>File</i> menü	14
1.1. <i>Új</i> menüpont	14
1.1.1. <i>Új épület</i> menü	15
1.1.2. <i>Új épületegység</i> menü	16
1.2. <i>Megnyitás</i> menü	16
1.3. <i>Mentés</i> menü	17
1.4. <i>Mentés másként</i> menü	17
1.5. <i>Bezárás</i> menü	17
1.6. <i>Kilépés</i> menü	18
2. <i>Beállítások</i> menü	18
3. <i>Objektumok</i>	18
4. <i>Berendezések</i>	19
5. <i>Monitor</i>	20
6. <i>Help</i> menü	20

III. A program használata

1. Program indítása	21
2. Rajzolás	22
2.1. <i>Nyíl</i>	22
2.2. <i>Kijelölés gomb</i>	23
2.3. <i>Vonal</i> gomb	24
2.4. <i>Poligon</i> gomb	25
2.5. <i>Téglalap</i> gomb	26
2.6. <i>Ellipszis</i> gomb	26
2.7. <i>Ajtó</i> gomb	27
2.8. <i>Ablak</i> gomb	28
2.9. <i>Törlés</i> gomb	29
2.10. <i>Mozgatás</i> gomb	29
2.11. <i>Forgatás</i>	31
2.12. <i>Méretezés</i> gomb	32
3. Objektumok	33
3.1. Objektum létrehozása	33
3.2. Objektumok listázása	34
4. Berendezések	35
4.1. Berendezés létrehozása	35
4.2. Berendezések listázása	36

TESZTELŐ FELHASZNÁLÓI DOKUMENTÁCIÓ38

I. Installálás

Szükséges hardver konfiguráció	38
Szükséges szoftver konfiguráció	38
Installálás	38

II. A program használata

Program indítása	39
Kapcsolat megnyitása, lezárása	40

III. Menüszerkezet

1. Áttekintés menüpont	41
2. Hőmérők menüpont	42

3. <i>Riasztók</i> menüpont	43
4. <i>Tűzjelzők</i> menüpont	43
5. <i>Fűtőtestek</i> menüpont	44
6. <i>Más berendezések</i> menüpont	45
7. <i>Beállítások</i> menüpont	46
8. <i>Output buffer</i> menüpont	47
9. <i>Help</i> menüpont	48
10. <i>Kilépés</i> menüpont	48
VEZÉRLŐ FELHASZNÁLÓI DOKUMENTÁCIÓ	50

I. Installálás

Szükséges hardver konfiguráció	50
Szükséges szoftver konfiguráció	50
Installálás	50

II. Menüszerkezet

1. <i>File</i> menü	51
1.1. <i>Megnyitás</i> menüpont	51
1.2. <i>Mentés</i> menüpont	52
1.3. <i>Bezárás</i> menüpont	52
1.4. <i>Kapcsolat megnyitása</i> menüpont	52
1.5. <i>Kapcsolat bezárása</i> menüpont	53
1.6. <i>Kilépés</i> menüpont	53
2. <i>Beállítások</i> menü	53
2.1. <i>Hőmérő</i> menüpont	53
2.1.1. Eszközhöz kapcsolódó berendezések beállítása	54
2.2. <i>Riasztó</i> menüpont	55
2.3. <i>Tűzjelző</i> menüpont	56
2.4. <i>Fűtőtest</i> menüpont	57
2.5. <i>Egyéb berendezés</i> menüpont	58
2.6. <i>Vezérlőpult</i> menüpont	59
2.7. <i>Beállítások</i> menüpont	60
2.7.1. Általános beállítások menüpont	61
2.7.2. <i>Kapcsolat</i> menüpont	61
3. <i>Napló</i> menüpont	61
4. <i>Berendezések</i> menüpont	63

5. <i>Help</i> menüpont	63
6. Áttekintés gomb	64
7. Riasztási esemény és kezelése	65

II. FEJLESZTŐI DOKUMENTÁCIÓ

AZ INTELLHOUSE PROGRAMCSOMAG FEJLESZTŐI DOKUMENTÁCIÓJA

Feladat leírása	67
Feladathoz kapcsolódó fogalmak	67
A programcsomagban használt Unit-ok	68
Adatszerkezetek	73
Globális konstansok, változók	73
Globális eljárások, függvények	74
Típus- és osztályleírások	78

Adatállományok fizikai rendszerterve

<i>Alaprajz.db</i>	85
<i>Objektum.db</i>	86
<i>Berendezes.db</i>	86
<i>Homero.db</i>	88
<i>Futotest.db</i>	88
<i>Riaszto.db</i>	89
<i>Tuzjelzo.db</i>	89
<i>Egyeb.db</i>	90
<i>Kapcsolat.db</i>	91
Adatállományok kapcsolata	92

Egy épület felépítésének adatszerkezeti megvalósítása

Épület szerkezeti felépítése	93
Műveletek az épületláncon	95
Új épület létrehozása	95
Új alakzat hozzáfűzése az épülethez	95
Alakzatlánc megszüntetése	96
Épület megszüntetése	96

Az adatkapcsolati réteg feladata és működése

Keretek képzése	97
Adatkapcsolati protokoll	98

Simplex megáll- és vár protokoll	99
A programban használt protokoll	99
A kommunikációban használt konstans változók	100
A kommunikációban használt típusok	100
Globálisan elérhető függvény	105
Üzenet kódolása, dekódolása	
Kódolás	106
Dekódolás	107
Kódolás-dekódolás folyamata	108
TERVEZŐ FEJLESZTŐI DOKUMENTÁCIÓ111	
Specifikáció	111
A program főbb funkciói	111
A programban használt UNIT-ok	111
Globális eljárások, függvények	113
Típus és osztályleírások	113
Műveletek az alakzatokon	114
Mozgatás	114
Forgatás egy pont körül	115
Méretezés	117
Törlés	118
TESZTELŐ FEJLESZTŐI DOKUMENTÁCIÓ120	
Specifikáció	120
A program főbb funkciói	120
A programban használt UNIT-ok	120
Globális konstansok, változók	121
Globális eljárások, függvények	122
Típus és osztályleírások	125
VEZÉRLŐ FEJLESZTŐI DOKUMENTÁCIÓ128	
Specifikáció	128
A program főbb funkciói	128
A programban használt UNIT-ok	128
Globális konstansok, változók	131
Típus és osztályleírások	131

Összefoglaló

Tesztelés	133
Fejlesztési lehetőségek	133
Értékelés	133

Melléklet

Naplófájl nyomtatási példája	135
------------------------------------	-----

Függelék

A programcsomagban használt típusok	140
---	-----

BEVEZETÉS

A feladat rövid leírása, motivációja

Széleskörben és egyre erőteljesebben jelentkezik az az igény, hogy a nagyobb épületeket, ingatlanokat olyan épület-felügyeleti és vezérlő rendszerrel lássák el, mely lehetővé teszi a különböző épületautomatizáláshoz kapcsolódó funkciók integrálását. Így például egy modern irodaház számtalan különböző funkciójú felügyelő és vezérlőrendszerrel rendelkezhet, melyek általában minden többségükben biztos, hogy - külön rendszert alkotnak. (Pl.: tűzjelző, betörésjelző, beléptető, világítás vezérlés, klíma vezérlés, stb.)

A különálló rendszerek egyszerre hordozzák magukban a kiépítés költségeinek növekedését, a felügyelet problematikáját, a karbantartás és üzemeltetés nehézségeit. Egységbefoglalásuk csak és kizárolag erős kompromisszumokkal lehetséges, távirányításuk, lekérdezésük komoly műszaki problémákat és kompatibilitási kérdéseket vet föl.

Mai világunkban fontos szerepet kapnak a biztonsági rendszerek is, melyek adott környezetet, objektumot, történést felügyelnek. Az ember úgy tarthat felügyelet alatt bizonyos dolgokat, ha folyamatosan a helyszínen tartózkodik és ezzel bizonyosodik meg arról, hogy minden folyamat rendben működik. Ez valamilyen szinten működik is, például ha csak néhány berendezés működését kell szemmel tartani, vagy az eszközök viszonylag kis helyen vannak elhelyezve, amit az ember még figyelemmel tud kísérni. Viszont, ha a kényelmi szempontokat tekintjük, vagy a rendszer túl nagy és bonyolult ahhoz, hogy ezt egy ember képes legyen koordinálni, akkor szükség van olyan eszközre és berendezésre, amely ezeket a megfigyelési és irányítási feladatokat átveszi tölünk. Egy másik szempont lehet az automatizálás alkalmazására, hogy az emberi erőforrások jobb kihasználása érdekében a megfigyelni kívánt területre érzékelőket helyezünk el és ezek segítségével kísérjük figyelemmel az eseményeket.

Így a PC-k rohamos elterjedésével, és az információk egyre könnyebb hozzáférésével fokozatosan növekedett az igény az intelligens, PC-ről vezérelhető épület-felügyeleti rendszerek iránt.

A következő fejezetekben részletesebb összefoglaló olvasható a már említett fontosabb feladatokról és elvárásokról.

Gazdaságosság

Az „intelligens épület” hasznosabban üzemel, mint a „nem intelligens épület”. Gazdasági szempontból ennek két vetülete van. Egyszerűbb minél több szolgáltatást kínál egy létesítmény, annál értékesebb lesz a felhasználók számára, másrészről az átgondolt és felügyelt üzemeltetési folyamatokkal költségek takaríthatók meg.

Például:

- Jelentős fűtési, hűtési energia takarítható meg a különböző helyiségek összehangolt üzemeltetésével (pl. tárgyalótermek, előadótermek szabályozásai dinamikusan, a termek előre ütemezett illetve tényleges foglaltságaihoz alakíthatók).
- Komplex létesítményeknél a rendszer kiépítésével csökkenhető a kezelői személyzet létszáma illetve alkalmazható olyan helyeken is, ahol kezelői személyzet használata nem célszerű (pl. lakott területektől távol eső rendszerek esetében).
- Vézhelyzet esetén az előre felkészített rendszer igen gyors és hatékony reakcióival (pl. tűz esetén) emberi életeket és jelentős vagyontárgyakat menthet meg.
- A megnövelt kényelem, komfort lehetővé teszi az épületben élők, dolgozók hatékonyabb munkavégzését, kellemesebb közérzetet biztosít.
- A számos értéknövelő szolgáltatás között szerepelhet vagyontárgyakat (tűz- és behatolásjelzés, beléptetés, zárt láncú televízió-hálózat...).

Integráció

Az épület-üzemeltetési feladatok általában nem igazodnak az egyes alrendszerök korlátozásaihoz ezért ilyenkor több rendszer összehangolt működésére van szükség. Az integráció megvalósítható központosított illetve elosztott intelligenciával.

Központosított intelligencia

A már üzembe helyezett, de különálló rendszerek összehangolása legegyszerűbben központosítással érhető el. Az épületen belüli rendszereket központi felügyelet alá szervezzük, amelynek adatbázisa az összes, üzemeltetési szempontból lényeges adatot tartalmazza. A központ biztosítja a rendszerek közötti együttműködést; minden információ ide fut be. Az integráció egy olyan egységes, egyszerűsített kommunikációs protokollon alapul, amely lehetővé teszi a különböző gyártmányú és szerepű alrendszerök összeillesztését. Az ily módon integrált alrendszerök eseményei, információi közös rendszeridő szerint rögzítődnek, rekonstruálható sorrendiséggel.

Elosztott intelligencia

A ma legkorszerűbbnek számító technológiáknál a funkcionális határok nem csak az épület alrendszeri között (tűzjelző-, klímarendszer, világítás stb.), hanem már az egyes rendszerelemek szintjén elmosódnak. Az épületen belül egy ún. nyílt vezérlőrendszert hoznak létre, mely magába foglalja az összes, üzemeltetés szempontjából fontos készüléket, berendezést. minden rendszerelem azonos "nyelvet beszélve", közvetlenül is kommunikálhat egymással. A nyílt vezérlés leegyszerűsíti az integrált rendszer szerkezetét, hiszen az egyes elemek eleve beépített tulajdonsága az egymással való együttműködési képesség, az ún. "interoperabilitás". A vezérlést sem egyetlen központi egység látja el, ezért a rendszer "elosztott intelligenciával" rendelkezik.

A gyakorlatban vegyesen alkalmazzák az alrendserek egységes felügyelet alá integrálását és a nyílt vezérlőrendszereket.

Biztonságtechnika

A biztonságos üzemelés, valamint a személy- és vagyonvédelem a létesítmények működésének, szolgáltatásainak egyik legfontosabb összetevője. Komplexen kezelendő területről van szó, hiszen a biztonsági feladatok általában több alrendszer együttes működésével valósulnak meg, sőt adott esetben a biztonsági rendszerek határain is túlmutatnak.

Mi történhet például, ha egy irodaházban tűz üt ki?

Tűzriasztás esetén a tűzjelző rendszer nem csak az automatikus oltásvezérlést indítja be vagy a bent tartózkodókat figyelmezteti a tűzre – pl. a hangosbemondó segítségével –, hanem átjelzést küld a klímarendszernek és az épületgépészeti, melynek következtében egy sor egyéb rendszer léphet működésbe. Automatikusan lekapcsolónak a klímavezérlő szekrények és leáll a levegő befúvás, így megtagadja a füst átterjedését más területekre, illetve kinyílnak a füstelvezető zsaluk. Beindul a füstmentesítő ventillátor a lépcsőházban, hogy túlnyomással biztosítsa a menekülési útvonal füstmentességét. Áramtalanodik az épület – vagy annak egy zónája – és beindul a vészvilágítás. A szerverek mentési folyamatot kezdenek, hogy ha végképp megszűnik az áram, ne történjen adatvesztés. A beléptető rendszer evakuációs riportot készít, és bizonyos ajtókon lehetővé teszi az azonosító kártya nélküli áthaladást, miközben a videórendszer ellenőrzi az épület kiürítését.

A vagyonvédelmi és életvédelmi berendezések egy része nehezen kezelhető profittermelő befektetésként. Ezek a rendszerek arra valók, hogy az esetleges veszteséget minimalizálják. Jellemzően ilyen a tűzjelző rendszer, melynek telepítését esetenként hatósági előírás követeli meg.

Más részük munkaerő megtakarítást eredményezhet: a behatolás-jelzővel csökkenhető például az őrző-védő személyzet létszáma. A biztonságtechnikai felszereltség érvényesíthető a létesítmény bérleti díjaiban is, így az üzemeltetés szempontjából nemcsak szükséges, de jó befektetés is egyben.

Dolgozat témája és eredménye

A dolgozat témájának egy olyan programrendszer létrehozását tűztem ki céljául, amelynek fő feladata egy épület, épületegyüttes elektromos berendezéseinek működtetése, irányítása és felügyelete, emberi tényező minimalizálása mellett.

A dolgozat eredménye egy két programból álló programrendszer és egy kifejezetten a tesztelésre készített program. A programrendszer tervező egységevel egy épület egyszerűsített alaprajzát tudjuk megrajzolni és az irányítani kívánt berendezéseket elhelyezni. A vezérlő programmal a már megrajzolt épület berendezéseit tudjuk irányítani és a felügyeletet ellátni. A tesztelő program a berendezések működését szimulálja, amely egy másik számítógépen fut és I/O porton keresztül kommunikál a vezérlő programmal.

Dolgozat felépítése

A dolgozat két nagy fejezetből épül fel.

Az első részben a három program: a Tervező, a Tesztelő és a Vezérlő felhasználói dokumentációja olvasható, melyek a programok használatát tartalmazzák és mutatják be.

A második rész a fejlesztői dokumentációkat tartalmazza, melynek elején a három program mindegyikére vonatkozó adatszerkezetek, típusok, eljárások és algoritmusok leírása szerepel. A további fejezetek az egyes programok fejlesztői dokumentációt tartalmazzák.

A feladat megoldása során felhasznált eszközök

A dolgozat minden programegysége a Delphi 5 Professional fejlesztő eszközzel készült, Windows alapú operációs rendszeren. A fejlesztés egy Celeron 533-as, 256 MB RAM-mal rendelkező számítógépen történt.

A teszteléshez egy Pentium I 100 MHz-es, 16 MB RAM-mal rendelkező laptop számítógépet használtam, ami az épület berendezéseit és érzékelőit szimulálta.

I.

**FELHASZNÁLÓI
DOKUMENTÁCIÓ**

**TERVEZŐ
FELHASZNÁLÓI DOKUMENTÁCIÓ**

I. INSTALLÁLÁS

1. Szükséges hardver konfiguráció

- PC 100 MHz
- CD-ROM
- 32 MB RAM
- 10 MB szabad lemezterület

2. Szükséges szoftver környezet

- Microsoft Win9x/ME/2000/NT

3. Installálás

- Ha a számítógép alkalmas a CD lemez automatikus indítására:
A telepítőlemez behelyezése után a telepítőprogram automatikusan elindul.
- Ha a telepítést a CD-ről indítjuk:
A CD lemez gyökér könyvtárában indítsuk el az *Install.exe* programot.

Az *Install.exe* program indulása után válasszuk a ’Tervezőprogram telepítése’ funkciót és kövessük a program utasításait.

II. MENÜSZERKEZET

Az egyes menüpontokat elérhetjük az ALT és a menü nevében az aláhúzott betű billentyűjének együttes lenyomásával.

Pl.: a File menü megnyitása: ALT+F.



1. ábra

A főmenü:

1. *File*: az alaprajzzal kapcsolatos műveleteket lehet elvégezni.
2. *Beállítások*: a program működésével kapcsolatos beállítások
3. *Objektumok*: objektumok listázása
4. *Berendezések*: berendezések listázása
5. *Monitor*: az épület szerkezetfája
6. *Help*: segítség a program használatához

1. A *File* menü

Az alaprajzzal kapcsolatos műveleteket lehet elvégezni.

A *File* menüben az alábbi menüpontok közül lehet választani:

- *Új* menüpont: új alaprajz készítése.
- *Megnyitás* menüpont: meglévő épület betöltése.
- *Mentés* menüpont: épület mentése.
- *Mentés másként* menüpont: épület mentése más néven.
- *Bezárás* menüpont: épületfájl bezárása.
- *Kilépés* menüpont: kilépés a programból.

1.1. *Új* menüpont

Az *Új* menüpontot kiválasztva, két újabb menüpont közül választhatunk:

- *Új épület* menüpont
- *Új épületegység* menüpont

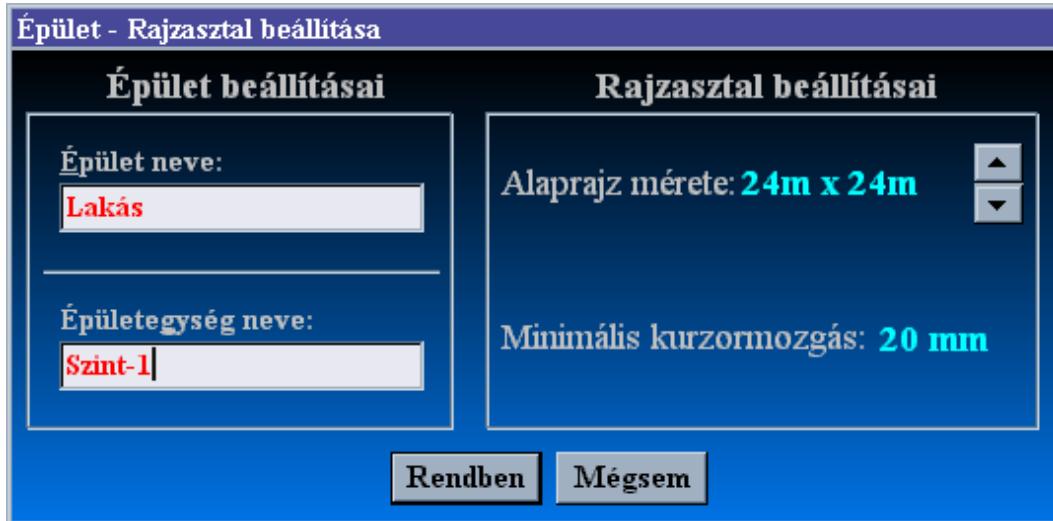
Az *Új épület* menüben egy új épület alaprajzát készíthetjük el.

Az *Új épületegység* menüben egy már meglévő és megnyitott épület alaprajzához tudunk újabb épületegységet hozzáadni.

Egy épületegység lehet például: melléképület, emeletszint, kert, stb....

1.1.1. Új épület menü

Az *Új épület* menüpont választása esetén a megjelenő ablakban meg kell adnunk az épületre vonatkozó adatokat.



2. ábra

- **Épület neve:** Az új épületnek nevet kell adni. Ezen a néven lesz nyilvántartva és automatikusan ezen a néven fogja a program elmenteni.
- **Épületegység neve:** Az első épületegység neve. Ezen az egységen kezdjük el a rajzolást.
- **Alaprajz mérete:** Megadjuk, hogy mekkora területen fekszik az épület.
- **Minimális kurzormozgás:** Azt adja meg, hogy az egérkurzor mozgatása a valóságban mekkora távolságnak felel meg milliméterben.

Terület	Érték (mm)
6m x 6m	5 mm
12m x 12m	10 mm
24m x 24m	20 mm
30m x 30m	25 mm
60m x 60m	50 mm
120m x 120m	100 mm
150m x 150m	125 mm
180m x 180m	150 mm
240m x 240m	200 mm
270m x 270m	225 mm
300m x 300m	250 mm

3. ábra

1.1.2. Új épületegység menü

A már meglévő épületünket tudjuk bővíteni újabb épületegységekkel.

A megjelenő ablakban csak az új épületegység nevének bevitelére van lehetőségünk (4. ábra). Jóváhagyás csak abban az esetben történik, ha még nincs olyan név, mint amilyet szeretnénk felvenni.

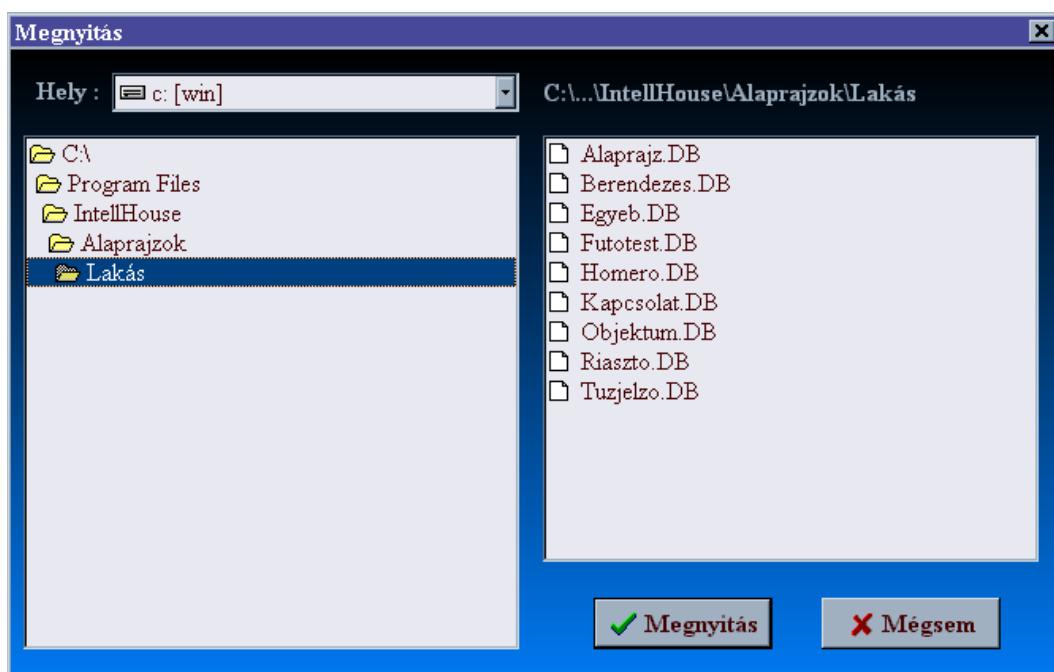


4. ábra

1.2. Megnyitás menü

Ha már létezik olyan épület amit már korábban létrehoztunk, akkor ebben a menüpontban tudjuk betölteni.

Betöltéshez csak ki kell választani a megfelelő könyvtárat, amelyikben a betöltendő épület található. Ahogy a könyvtárak között lépkedünk, a program automatikusan megvizsgálja, hogy a megfelelő adatbázisok megtalálhatók-e az adott könyvtárban.



5. ábra

- Ha megtalálhatók, akkor a *Megnyitás* gomb aktívvá válik (5. ábra). A *Megnyitás* gombra kattintva az épület alaprajza és a hozzá tartozó adatbázisok (berendezések, objektumok, stb....) betöltődnek.

- Ha egy könyvtárat kiválasztva, a *Megnyitás* gomb nem lesz aktív, akkor biztosak lehetünk abban, hogy a könyvtár nem tartalmazza a megfelelő fájlokat.

1.3. *Mentés* menü

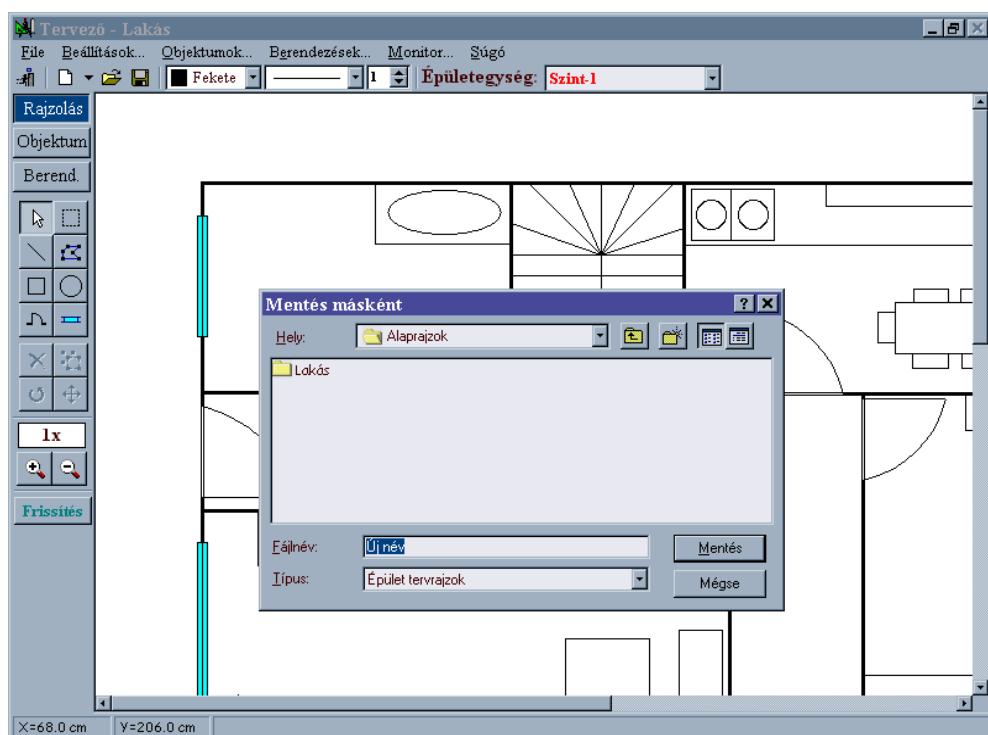
Egy épület alaprajzát, -berendezéseit, -objektumait tudjuk elmenteni.

A mentés minden az ‘Alaprajzok’ könyvtárba történik, azon belül pedig az épület nevével megegyező alkönyvtárba. Ha az alkönyvtár nem létezik, akkor az automatikusan létrejön.

A mentés során a korábbi verzió felülíródik. Ha azt szeretnénk, hogy megmaradjon a korábbi változat, akkor használjuk a *Mentés másként* menüpontot.

1.4. *Mentés másként* menü

Egy épület alaprajzát, -berendezéseit, -objektumait tudjuk elmenteni az alapértelmezett névtől eltérő néven vagy eltérő helyre. A mentés során meg kell adni azt a könyvtárat ahová menteni szeretnénk az épület adatait.



6. ábra

1.5. *Bezárás* menü

A bezárás menü választása esetén, a megnyitott épülethez tartozó összes adatbázis bezárásra kerül. Ha volt változtatás, akkor a bezárás előtt mentésre is van lehetőség.

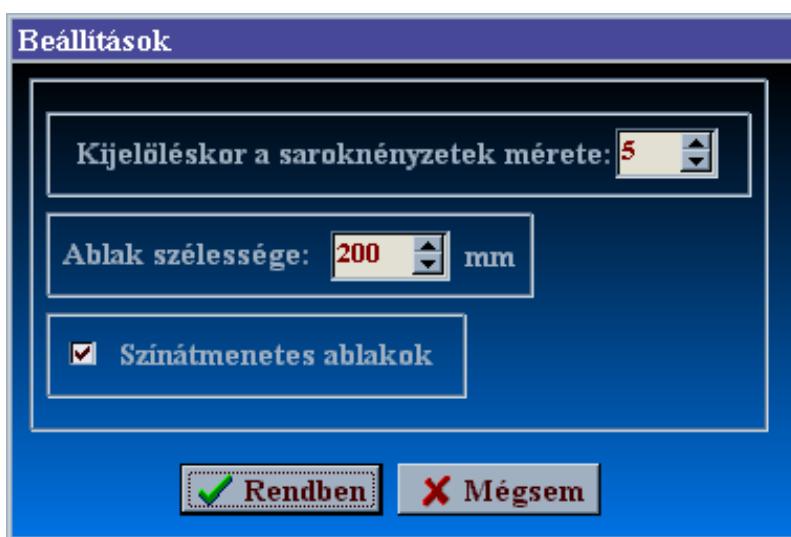
1.6. Kilépés menü

Kilépés a programból.

Ha az épülethez tartozó adatbázisokat módosítottuk, pl.: változott a rajz, új berendezést adtunk az épülethez, stb...., akkor a program rákérdez, hogy mentjük-e a változásokat. Válaszoljunk Igen-nel, ha menteni szeretnénk a változásokat. Ha Nem-mel válaszolunk, akkor a legutóbbi mentett változatot tartja meg, és az azóta történt változtatásokat elveti.

2. Beállítások menü

A program működésével kapcsolatos beállítások.



7. ábra

- **Saroknégyzetek mérete:** Kijelöléskor, ha méretezni szeretnénk az alakzatokat, akkor a sarkokban megjelenő négyzetek méretét állíthatjuk be.
 - Minimum érték: 2
 - Maximum érték: 10
- **Ablak szélessége:** Az ablakok szélességét állíthatjuk be.
 - Minimum érték: 5 mm
 - Maximum érték: 1000 mm
- **Színátmenetes ablakok:** Ha ki van jelölve, akkor aktív és minden ablakra jellemző. Ha nincs kijelölve, akkor a Windows alapértelmezett ablakszíne a háttér kitöltési színe.

3. Objektumok menü

A menüpontban az épületben található objektumokról kapunk egy áttekintő listát, melyben egy objektumról fel vannak sorolva, hogy mi az objektum neve, melyik épületegységben található, és mennyi berendezés tartozik hozzá (8. ábra).

Objektumok listája		
Objektum neve	Épületegység neve	Berendezések száma
Vendégszoba	Szint-2	1
Dolgozószoba	Szint-2	3
Fürdőszoba	Szint-1	2
Konyha	Szint-1	3
Nappali	Szint-1	4

Törles Bezárás

8. ábra

Itt lehetőség van az egyes objektumok törlésére is.

Objektum törlése esetén, ha az tartalmaz berendezéseket is, akkor az egyes berendezések helye automatikusan ‘Egyikhez sem’ lesz. Tehát a berendezések nem törlődnek. Ha mégis törölni szeretnénk őket, akkor azt a *Berendezések* menüpontban tehetjük meg.

4. Berendezések menü

A menüpontban az épületben található berendezésekről kapunk egy áttekintő táblázatot, melyben egy berendezésről fel vannak sorolva, hogy mi a berendezés azonosítója (ID), neve, típusa, melyik épületegységben található, és melyik objektumhoz tartozik (9. ábra).

Itt lehetőség van az egyes berendezések törlésére is.

Berendezések listája				
ID	Berendezés neve	Típusa	Épületegység neve	Objektum neve
14	Vendég-tűzjelző	Tűzjelző	Szint-2	Vendégszoba
13	Dolgozó-riasztó	Riasztó	Szint-2	Dolgozószoba
12	Dolgozó-radiátor	Fűtőtest	Szint-2	Dolgozószoba
11	Dolgozó-hőmérő	Hőmérő	Szint-2	Dolgozószoba
10	Térvilágítás	Egyéb	Szint-1	Egyikhez sem
9	Nappali-radiátor	Fűtőtest	Szint-1	Nappali
8	Nappali-tűzjelző	Tűzjelző	Szint-1	Nappali
7	Nappali-riasztó	Riasztó	Szint-1	Nappali
6	Nappali-hőmérő	Hőmérő	Szint-1	Nappali
5	Konyha-radiátor	Fűtőtest	Szint-1	Konyha
4	Konyha-tűzjelző	Tűzjelző	Szint-1	Konyha
3	Konyha-hőmérő	Hőmérő	Szint-1	Konyha
2	Fürdő-radiátor	Fűtőtest	Szint-1	Fürdőszoba
1	Fürdő-hőmérő	Hőmérő	Szint-1	Fürdőszoba

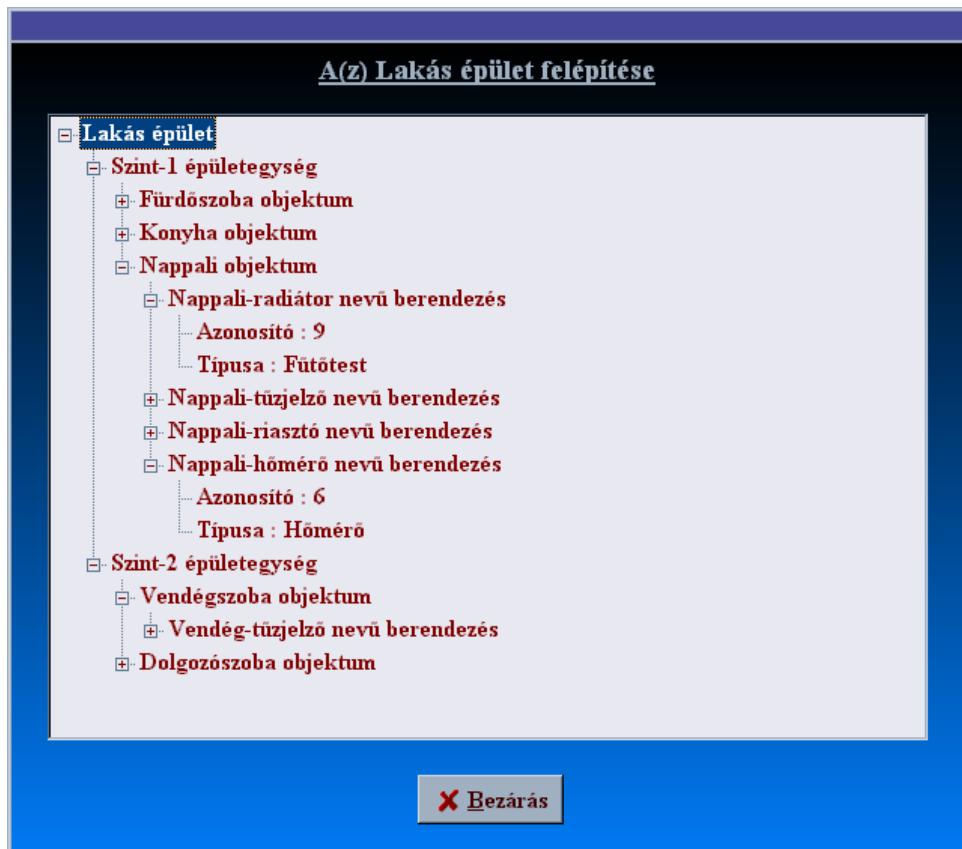
Törles Bezárás

9. ábra

5. Monitor menü

Ebben a menüben az épület felépítését tekinthetjük át.

Az épület szerkezete könyvtárstruktúra szerűen van ábrázolva, ahol a gyökérnek az épület neve felel meg (10. ábra). A struktúrában egyre beljebb haladva eljutunk egészen az épületben található berendezésekig.



10. ábra

A struktúra 5 szintet tartalmazhat:

- 1. szint: épület neve (gyökér)
- 2. szint: épületegységek
- 3. szint: objektumok
- 4. szint: berendezések
- 5. szint: berendezések adatai

6. Help menü

Segítséget nyújt a program kezelésében.

A help az F1 billentyű lenyomásával azonnal elérhető.

III. PROGRAM HASZNÁLATA

1. Program indítása

A program a *Tervezo.exe* fájl betöltésével indítható.

Indítás után a program megróbálja megkeresni és beállítani a számítógépen alkalmazható legjobb színmélységet és felbontást az alábbi listából:

- 1024x768 - 32 bit
- 1024x768 - 16 bit
- 1024x768 - 8 bit
- 800x600 - 32 bit
- 800x600 - 16 bit
- 800x600 - 8 bit
- 640x480 - 32 bit
- 640x480 - 16 bit
- 640x 480 - 8 bit

Ha nem sikerül egyiket sem beállítani, akkor a program nem indul el.



11. ábra

A felbontás beállítása után, egy megjelenő dialógus ablakban választhatunk, hogy új épületet szeretnénk-e kezdeni vagy egy már korábban létrehozott épület alaprajzát és berendezéseit szeretnénk megnyitni (11. ábra).

2. Rajzolás

A *rajzolás* funkciórészben egy épülethez tartozó épületegységet tudunk megrajzolni.

Ahhoz, hogy rajzolni tudjunk, a bal oldalon található funkció gombok közül ki kell választani a ‘Rajzolás’ gombot.

A ‘Rajzolás’ gomb aktív, ha a gomb színe kék.



12. ábra

A Rajzolás-ból elérhető gombok:

- *nyíl*: alaphelyzet, ekkor nem lehet a rajzon változtatni
- *kijelölés*: az alaprajz egy területét lehet kijelölni
- *vonal*: vonal rajzolása
- *poligon*: poligon rajzolása
- *téglalap*: négyzet, téglalap rajzolása
- *ellipszis*: kör, ellipszis rajzolása
- *ajtó*: ajtó rajzolása
- *ablak*: ablak rajzolása
- *törlés*: a kijelölt elemek törlése
- *mozgatás*: a kijelölt elemek mozgatása
- *forgatás*: a kijelölt elemek forgatása
- *méretezés*: a kijelölt elemek átméretezése

2.1. Nyíl gomb

Ha a gomb le van nyomva, akkor a rajzon semmilyen változtatást nem tudunk végrehajtani.

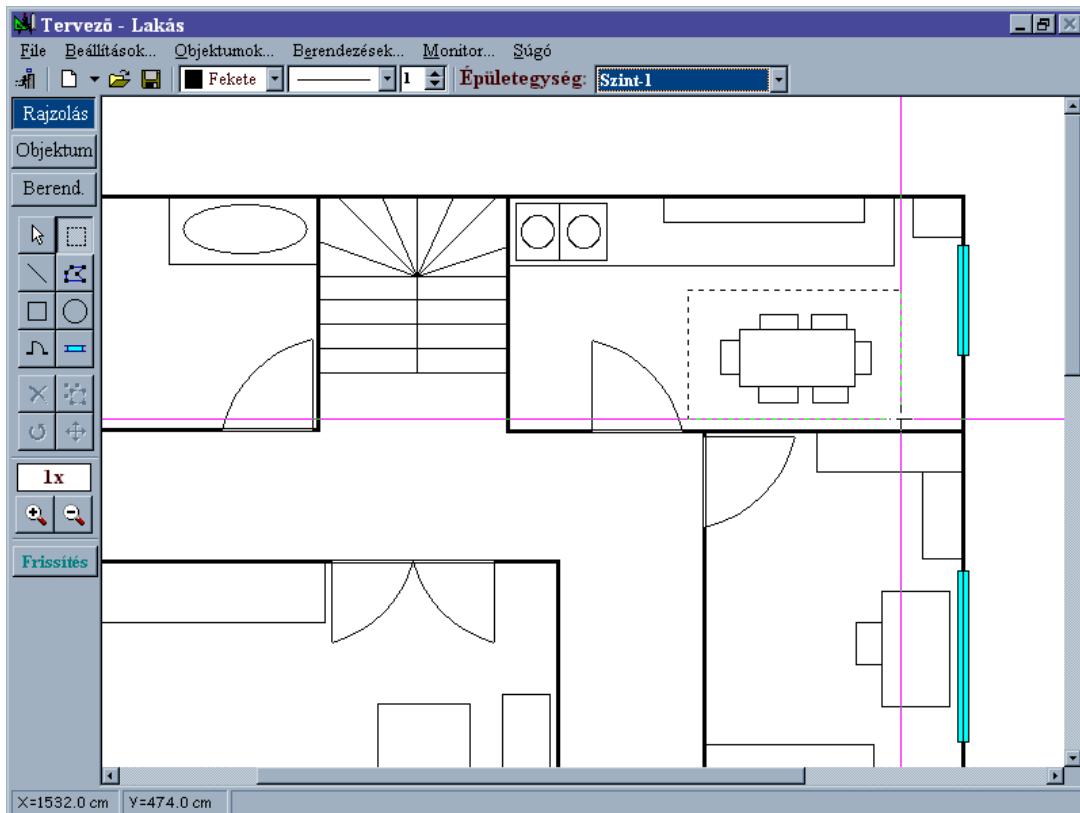
Tipp: Kiválasztott állapotban, ha a rajz felett lenyomjuk az egér bal gombját és folyamatos nyomvatartás mellett mozgatjuk az egeret, akkor az alaprajzot tudjuk lapozni, mozgatni. Ekkor az egérkursor formája egy kéz. Ha felengedjük a gombot, akkor az egérkursor is visszaáll az eredeti állapotába.

2.2. Kijelölés gomb

A rajzon egy terület kijelölésére szolgál.

Ha bizonyos rajzelemeken módosításokat szeretnénk elvégezni, akkor használjuk a *kijelölés* funkciót. A kijelölés minden egy téglalapot határoz meg.

Használata: Vigyük az egér a kijelölni kívánt terület bal felső sarkához, majd az egér bal gombjának folyamatos nyomvatartása mellett, jelöljük ki a területet (13. ábra). Ha körbekerítettük a területet, engedjük fel a gombot.



13. ábra

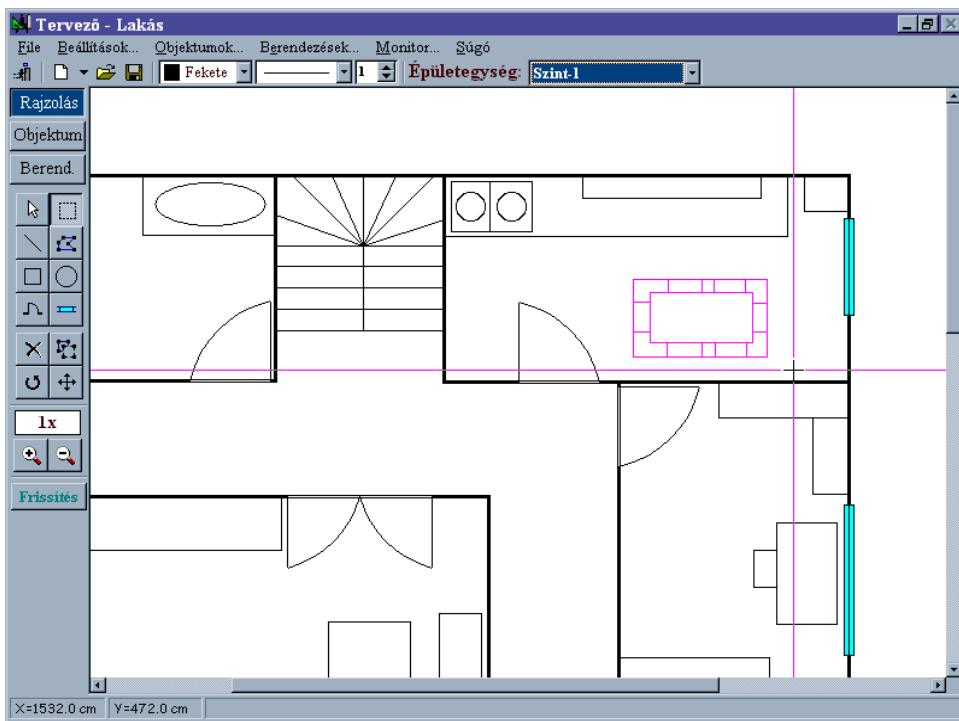
Ekkor azok az alakzatok - amelyeket a kijelölés során meghatározott téglalap körbekerített – kitüntetett színnel (rózsaszín) lesznek újra kirajzolva (14. ábra).

A kijelölést megszűntethetjük, ha a rajz felett bárhol lenyomjuk az egér jobb gombját.

A jobb gomb használatával lehet az egyes műveletekből is visszalépni.

Műveletek a kijelölt terüettel:

- *törlés:* a kijelölt elemek törlése
- *mozgatás:* a kijelölt elemek mozgatása
- *forgatás:* a kijelölt elemek forgatása
- *méretezés:* a kijelölt elemek átméretezése

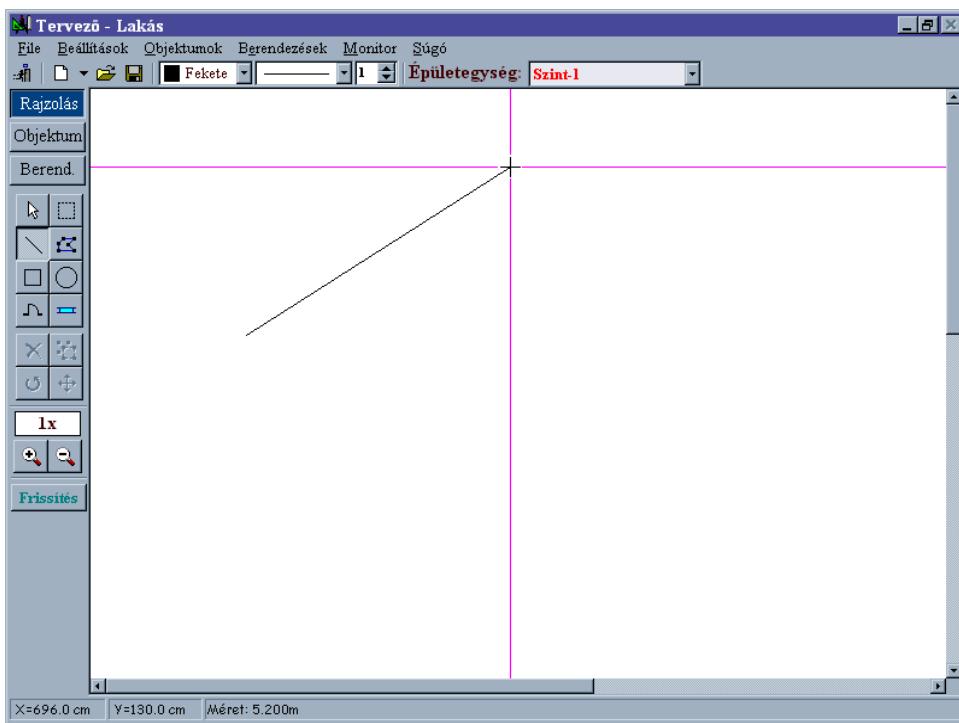


14. ábra

2.3. Vonal gomb

A rajzlapon egy vonalat húz.

Használata: Vigyük az egeret oda, ahol a vonalat kezdeni szeretnénk. A bal gomb folyamatos nyomvattartása mellett mozgassuk az egeret a kívánt pozícióba, majd engedjük fel a gombot (15. ábra).



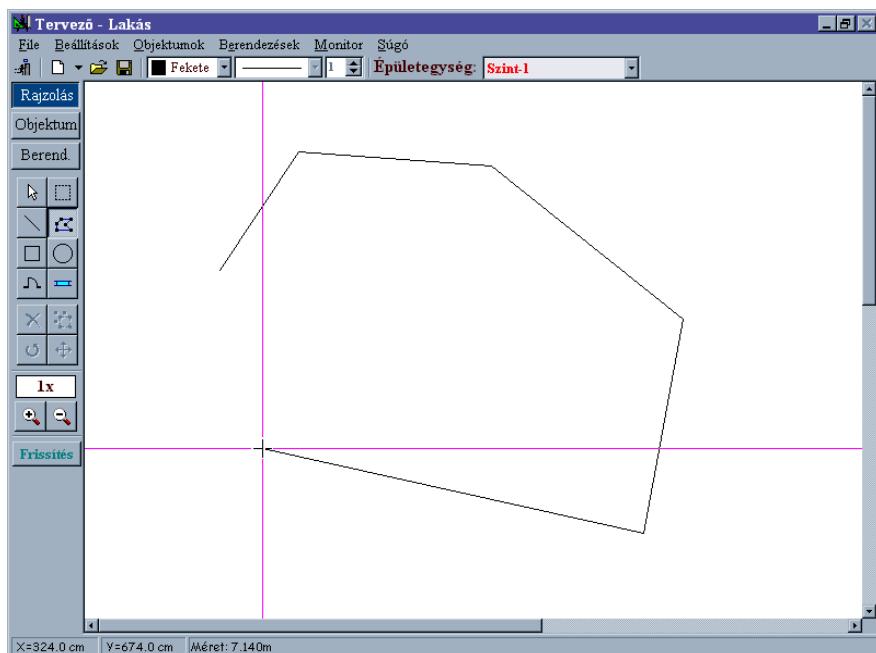
15. ábra

A felengedés után a vonal rögzítődik, és eltárolódik a kezdő, -végző koordinátája, a színe, típusa, és vastagsága.
Rajzolás közben, alul a státszbáron látni lehet a vonalzó aktuális pozíóját és a vonal hosszát méterben.

2.4. Poligon gomb

Ezt a rajzolási módot akkor válasszuk, ha több egymáshoz kapcsolódó szakaszt, vonalat szeretnénk rajzolni pl.: fal rajzolása.

Használata: Vigyük az egeret oda, ahol a poligont kezdeni szeretnénk. Kattintsunk a rajzon egyszer az egér bal gombjával, majd mozgassuk az egeret a következő koordinátához és ismét kattintsunk egyet. Ezt addig folytassuk, amíg meg nem rajzoltuk a kívánt alakzatot (16. ábra).



16. ábra

Ha végeztünk, akkor az egér jobb gombjával fejezzük be a rajzolást. Ekkor a program megkérdezi, hogy zárja-e az alakzatot.

Két lehetőségünk van:

- ha az alakzatot be szeretnénk zární, vagyis az első és az utolsó vonalat még össze szeretnénk kötni, akkor válaszoljunk *Igen*-nel. Ekkor egy sokszöget kapunk.
- ha nem szeretnénk összekötni az első és az utolsó vonalat, akkor válaszoljunk *Nem*-mel.

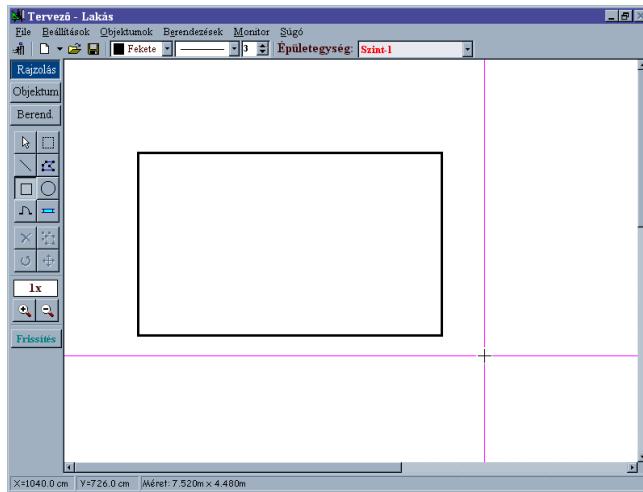
Ha végeztünk a poligon rajzolásával, akkor a program a továbbiakban a poligon szakaszait vonalakként kezeli.

Rajzolás közben, alul a státszbáron látni lehet a vonalzó aktuális pozíóját és a szakasz hosszát méterben.

2.5. Téglalap gomb

A rajzlapra egy téglalapot (négyzetet) rajzol.

Használata: Vigyük az egeret oda, ahol a téglalapot kezdeni szeretnénk. A bal gomb folyamatos nyomvatartása mellett mozgassuk az egeret a kívánt pozícióba, majd engedjük fel a gombot (17. ábra).



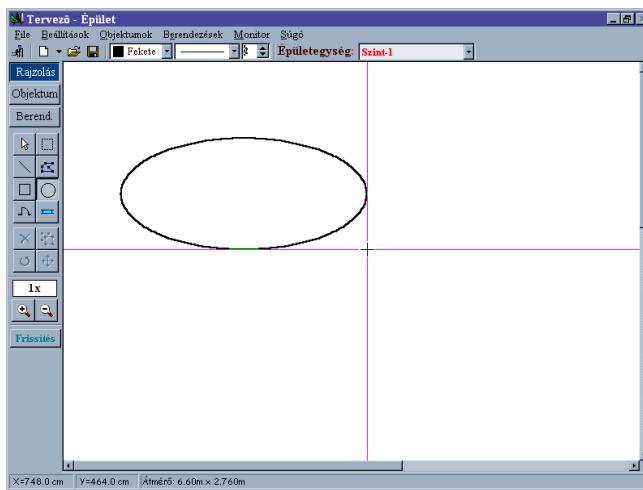
17. ábra

Rajzolás közben, alul a státuszbáron látni lehet a vonalzó aktuális pozíóját és a téglatest méretét méterben.

2.6. Ellipszis gomb

A rajzlapra egy ellipszist (kört) rajzol.

Használata: Vigyük az egeret oda, ahol az ellipszist kezdeni szeretnénk. Ez a pont az ellipszis köré rajzolható téglalap bel-felső koordinátája. A bal gomb folyamatos nyomvatartása mellett mozgassuk az egeret a kívánt pozícióba, majd engedjük fel a gombot (18. ábra). Ez a pont az ellipszis köré rajzolható téglalap jobb-alsó koordinátája.



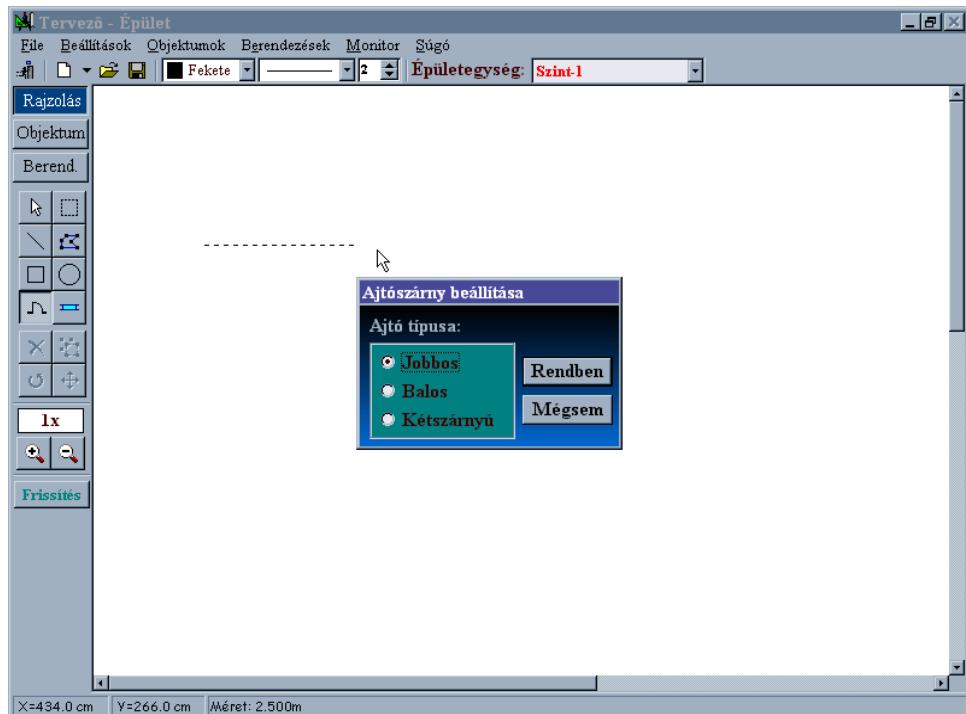
18. ábra

Rajzolás közben, alul a státuszbáron látni lehet a vonalzó aktuális pozíóját és az ellipszis méretét méterben.

2.7. Ajtó gomb

Különböző típusú ajtókat tudunk készíteni.

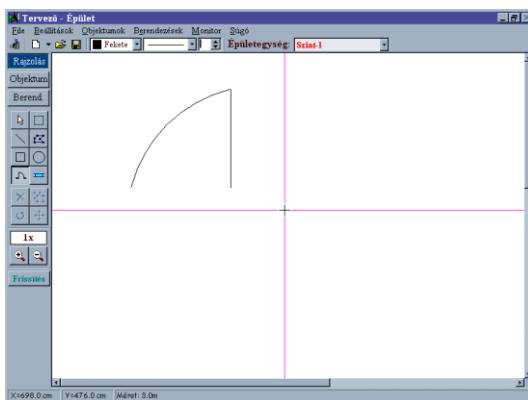
Használata: Vigyük az egeret oda, ahol az ajtót kezdeni szeretnénk. A bal gomb folyamatos nyomvatartása mellett mozgassuk az egeret a kívánt pozícióba, majd engedjük fel a gombot. (19. ábra)



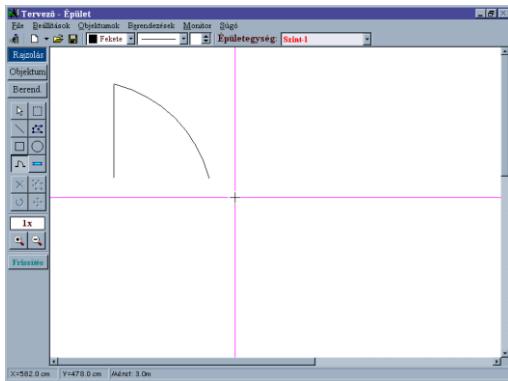
19. ábra

Három fajta ajtót lehet készíteni:

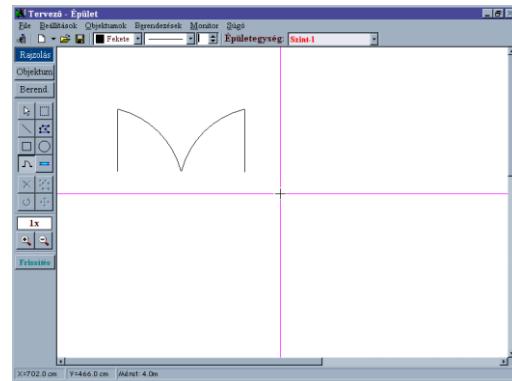
- *jobbos*: az ajtó szárnya jobbra nyílik (20. ábra)
- *balos*: az ajtó szárnya balra nyílik (21. ábra)
- *kétszárnyú*: dupla szárnyú ajtó (22. ábra)



20. ábra



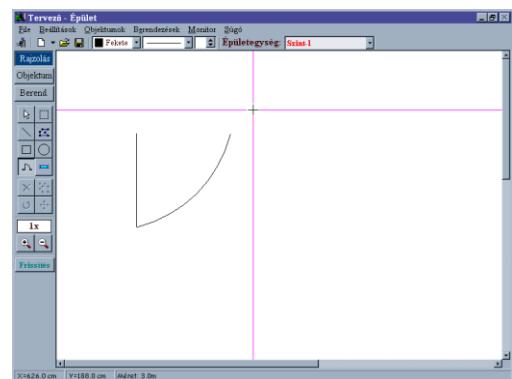
21. ábra



22. ábra

Tipp: Ha az ajtót balról jobbra rajzoljuk, akkor az ajtó szárnya felfelé mutat, ha viszont az ajtót jobbról balra húzzuk, akkor az ajtósárny lefelé fog mutatni (23. ábra).

23. ábra

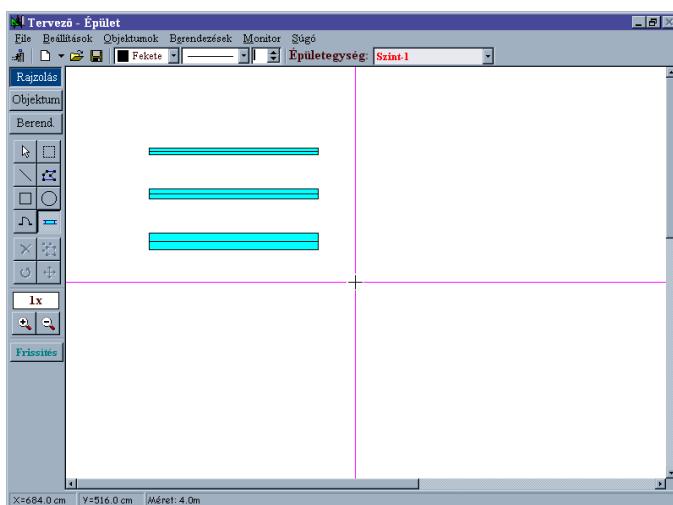


Rajzolás közben, alul a státuszbáron látni lehet a vonalzó aktuális pozícióját és az ajtó méretét méterben.

2.8. Ablak gomb

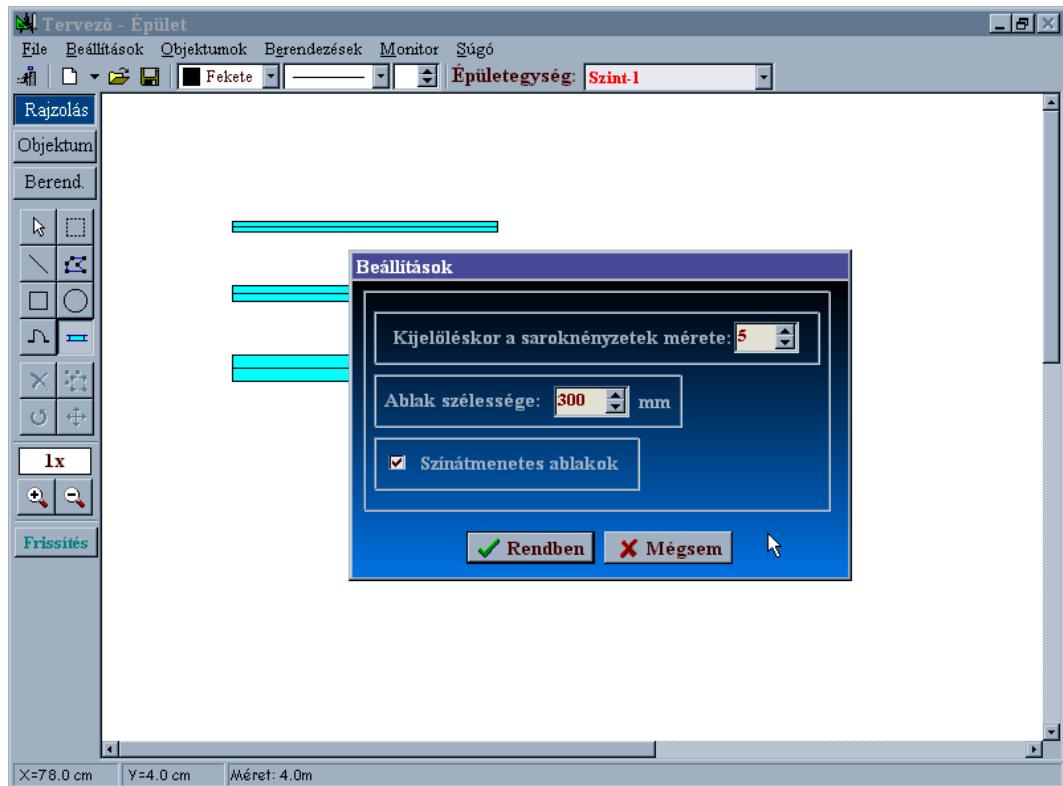
A rajzlapra egy ablakot rajzol.

Használata: Vigyük az egeret oda, ahol az ablakot kezdeni szeretnénk. A bal gomb folyamatos nyomvataartása mellett mozgassuk az egeret a kívánt pozícióba, majd engedjük fel a gombot (24. ábra).



24. ábra

Tipp: Az ablak szélességének módosítására a *Beállítások* menüben van lehetőség (25. ábra). Az ablak szélességét mm-ben kell megadni.



25. ábra

Rajzolás közben, alul a státuszbáron látni lehet a vonalzó aktuális pozícióját és az ablak méretét méterben.

2.9. Törlés gomb

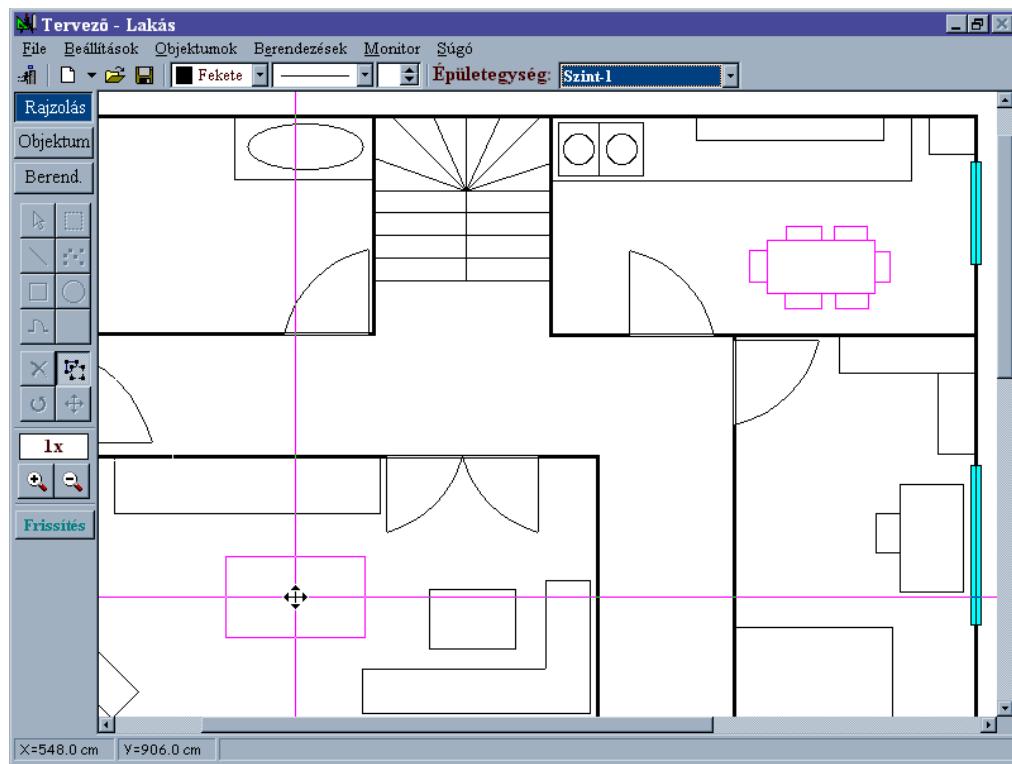
A kijelölt alakzat(ok) törlése az alaprajzról.

Használata: Jelöljük ki a törlni kívánt területet. Ha ez megtörtént, akkor már csak a törlés gombra kell kattintani és a kijelölt alakzatok törlődnek.

2.10. Mozgatás gomb

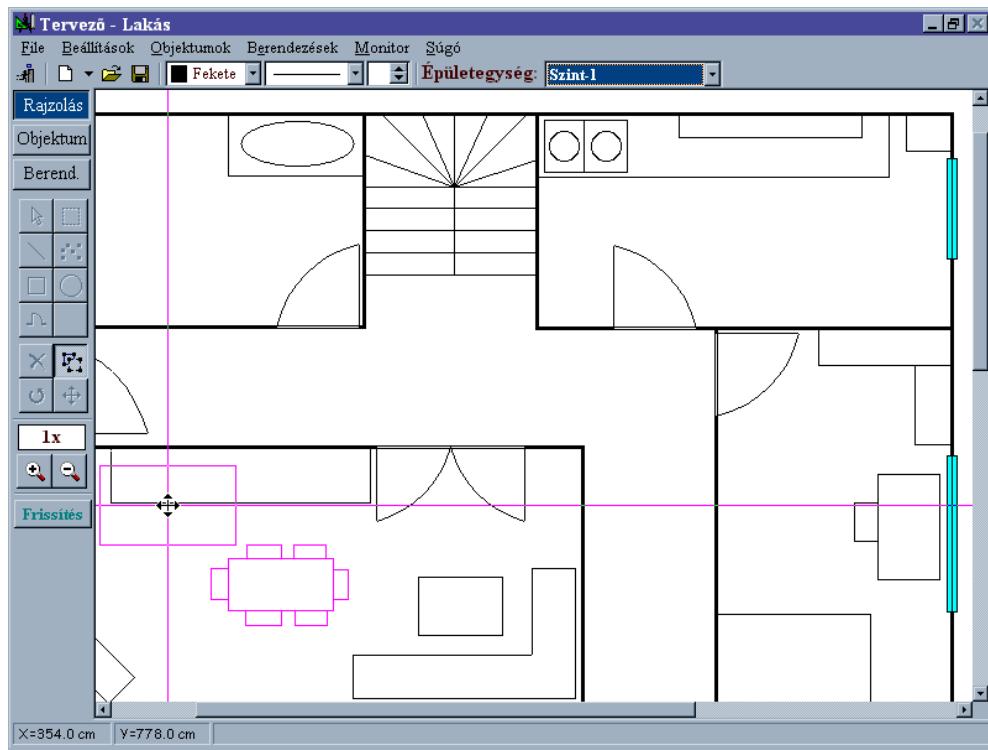
A kijelölt alakzat(ok) mozgatása az alaprajzon.

Használata: Jelöljük ki a mozgatni kívánt területet. Ha ez megtörtént, akkor az áthelyezés gombra kattintva a kijelölt területet a kívánt helyre mozgathatjuk és az egér bal gombjával az alakzatokat lerakhatjuk (26. ábra).



26. ábra

Ha ez megfelel, akkor az új helyet a jobb gombbal véglegesíthetjük, ha nem, akkor folytathatjuk a mozgatást (27. ábra).

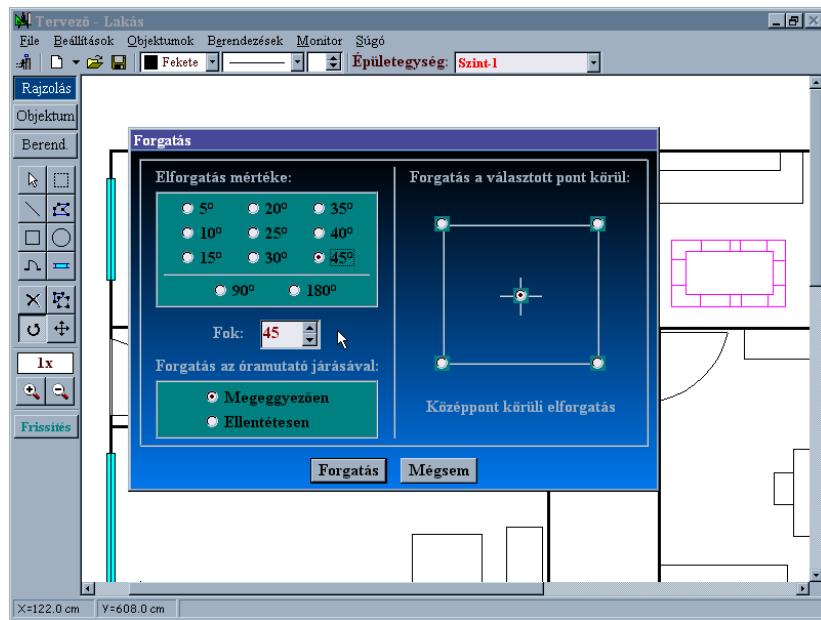


27. ábra

2.11. Forgasztás gomb

A kijelölt alakzat(ok) forgatása az alaprajzon.

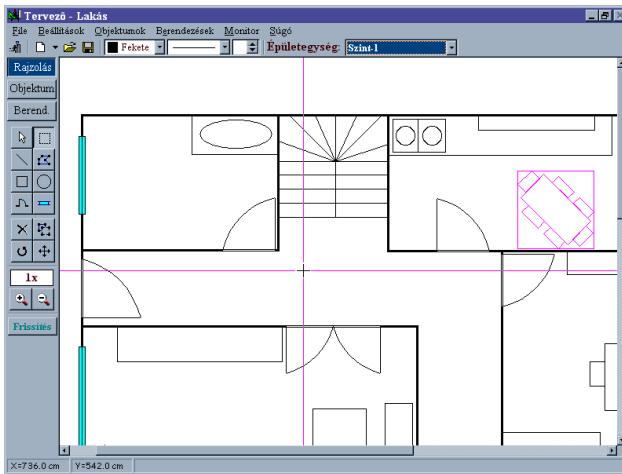
Használata: Jelöljük ki a forgatni kívánt területet. Ha ez megtörtént, akkor a forgatás gombra kattintva megjelenik a forgatás dialógus ablak, ahol beállíthatjuk a forgatás paramétereit (28. ábra).



28. ábra

Itt lehetőség van a forgatás szögének beállítására, a forgatási irány és típusának kiválasztására.

- **forgatás szögének beállítása:** megadhatunk tetszőleges szöget, de választhatunk a felsorolt szögek közül is.
- **forgatási irány kiválasztása:** választhatunk, hogy az óramutató járásával megegyező vagy ellentétes forgatást szeretnénk.
- **forgatás típusának kiválasztása:** megadhatjuk, hogy a kijelölt téglalap középpontja körül vagy valamelyik sarka körül szeretnénk forgatni.

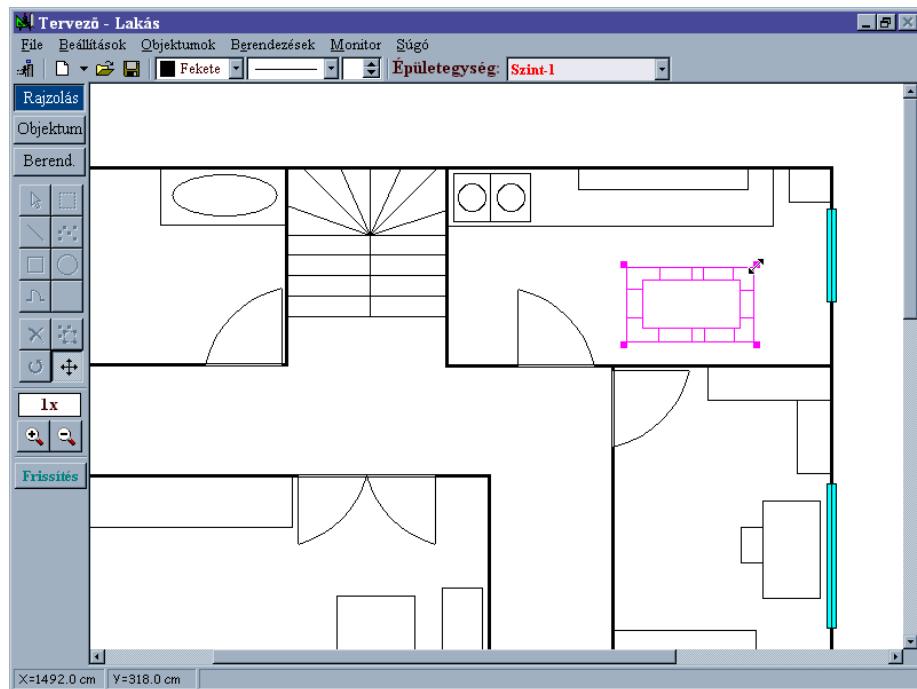


29. ábra

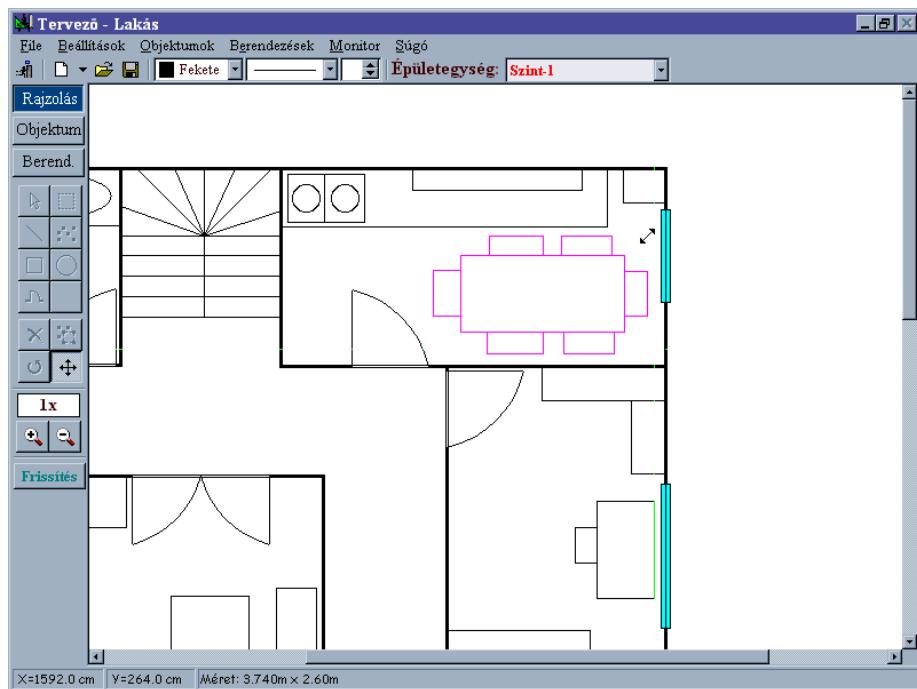
2.12. Méretezés gomb

A kijelölt alakzat(ok) méretezése az alaprajzon.

Használata: Jelöljük ki a méretezni kívánt területet. Ha ez megtörtént, akkor a méretezés gombra kattintva elvégezhetjük az alakzat(ok) méretezését (30-31. ábra).



30. ábra



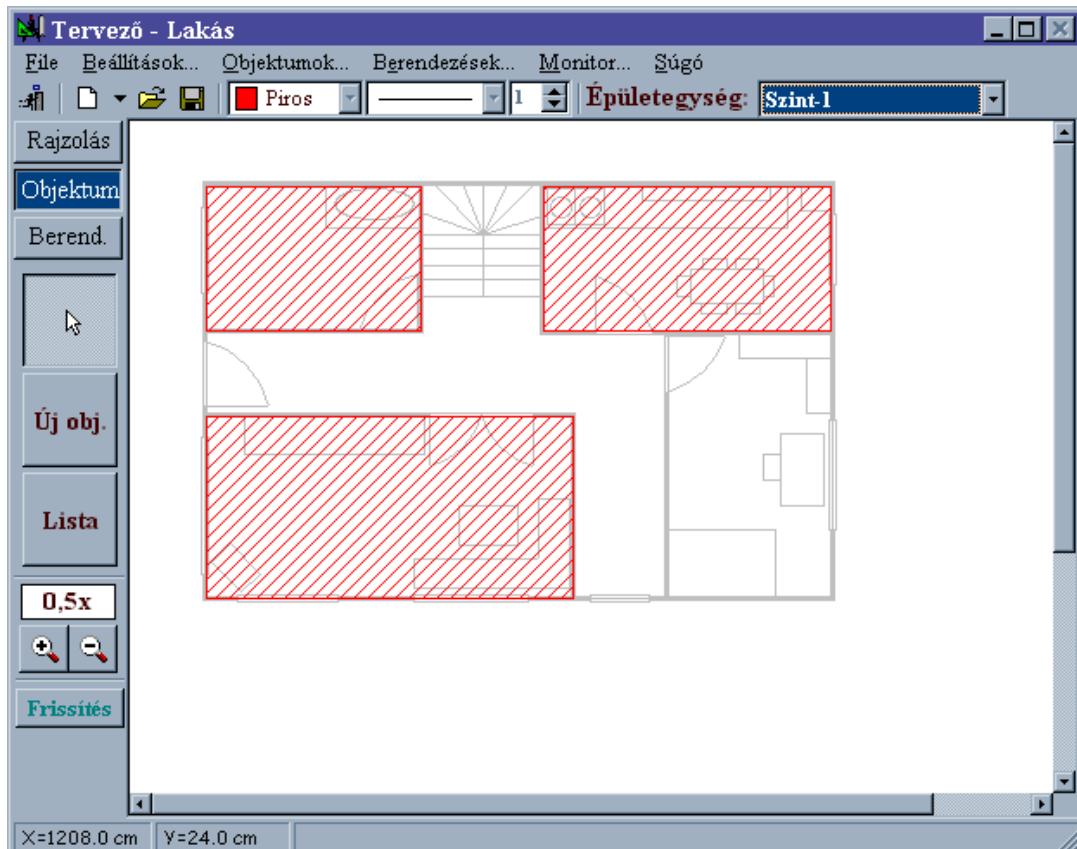
31. ábra

3. Objektumok

A *objektumok* funkció részben:

- új objektumokat tudunk létrehozni,
- a meglévőket tudjuk kilistázni, törölni.

Az ‘Objektum’ gomb aktív, ha a gomb színe kék.

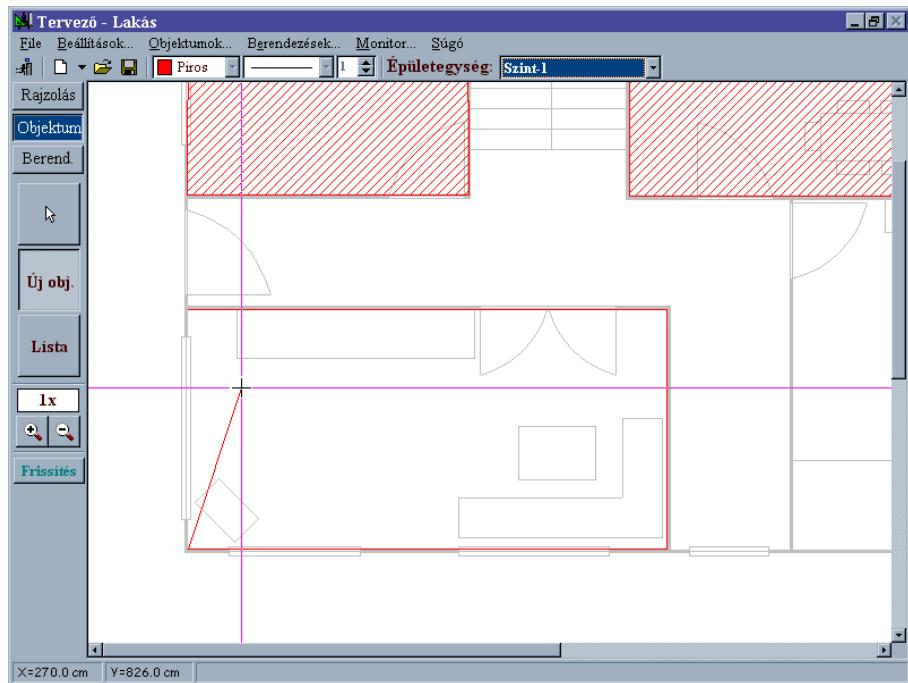


32. ábra

3.1. Objektum létrehozása

Új objektum létrehozásához válasszuk ki az *Új* gombot. Ekkor az egérkursor formája kereszt lesz, ami jelzi, hogy rajzolni tudunk az alaprajzra. A rajzolás menete megegyezik a poligon rajzolás menetével, egészen a rajzolás befejezéséig, amikor is a jobbgomb lenyomásakor az alakzat automatikusan záródik (33. ábra). Vagyis az objektum egyes pontjait az egér balgombjának lenyomásával tudjuk rögzíteni, az objektum rajzolását pedig az egér jobbgombjának lenyomásával tudjuk befejezni.

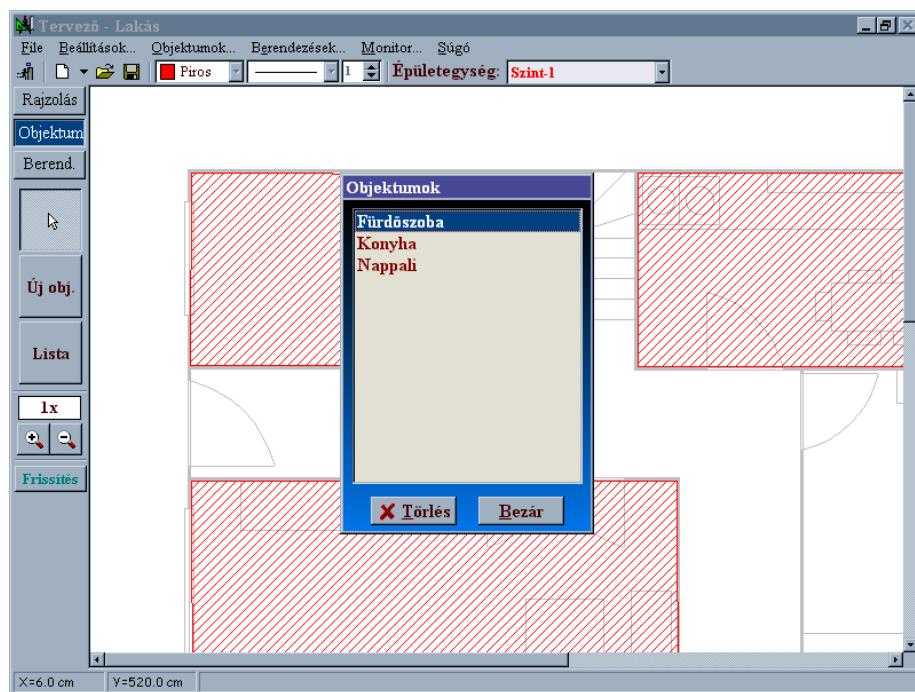
A rajzolás végeztével meg kell adnunk az objektum nevét, amelynek egyedinek kell lennie.



33. ábra

3.2. Objektumok listázása

A létező objektumokról nézhetünk meg egy listát ebben a menüpontban. A lista természetesen csak az éppen aktuális épületegységhez tartozó objektumokat tartalmazza (34. ábra).



34. ábra

Itt lehetőség van az egyes objektumok törlésére is.

4. Berendezések



35.ábra

A program öt különböző berendezéstípust tartalmaz:

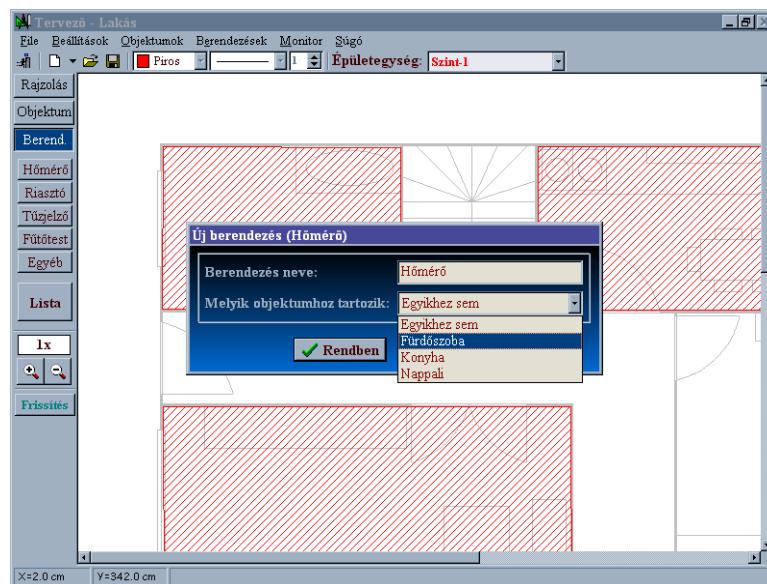
- hőmérő
- riasztó
- tűzjelző
- fűtőtest
- általános berendezés

Ezeknek a berendezéseknek a részletes leírása megtalálható a Vezérlő program felhasználói dokumentációjában, ugyanis a Tervező programban nincs lehetőség az egyes berendezések tulajdonságainak beállítására.

Ebben a programrészben új berendezéseket vihetünk be a rendszerbe, listázhatjuk őket és törölhetünk onnan.

4.1. Berendezés létrehozása

A menüből válasszuk ki, hogy milyen típusú berendezést szeretnénk hozzákapcsolni a rendszerhez.



36. ábra

- **Berendezés neve:** adjuk meg az új berendezés nevét.
- **Melyik objektumhoz tartozik:** a legördülő listából válasszuk ki, hogy melyik objektumhoz szeretnénk hozzákapcsolni a berendezést.

4.2. Berendezések listázása

A menüpontban az épületben található berendezésekről kapunk egy áttekintő táblázatot (lásd: Berendezések menüpont alatt).

**TESZTELŐ
FELHASZNÁLÓI DOKUMENTÁCIÓ**

I. INSTALLÁLÁS

1. Szükséges hardver konfiguráció

- PC 100 MHz
- CD-ROM
- 16 MB RAM
- 10 MB szabad lemezterület

2. Szükséges szoftver környezet

- Microsoft Win9x/ME/2000/NT

3. Installálás

- Ha a számítógép alkalmas a CD lemez automatikus indítására:
A telepítőlemez behelyezése után a telepítőprogram automatikusan elindul.
- Ha a telepítést a CD-ről indítjuk:
A CD lemez Setup könyvtárában indítsuk el a *Setup.exe* programot.

Az *Install.exe* program indulása után válasszuk a 'Tesztelőprogram telepítése' funkciót és kövessük a program utasításait.

II. PROGRAM HASZNÁLATA

Program indítása

A program a *Tesztelo.exe* fájl betöltésével indítható.

Indítás után a program megpróbálja megkeresni és beállítani a számítógépen alkalmazható legjobb színmélységet és felbontást az alábbi listából:

- 1024x768 - 32 bit
- 1024x768 - 16 bit
- 1024x768 - 8 bit
- 800x600 - 32 bit
- 800x600 - 16 bit
- 800x600 - 8 bit
- 640x480 - 32 bit
- 640x480 - 16 bit
- 640x 480 - 8 bit

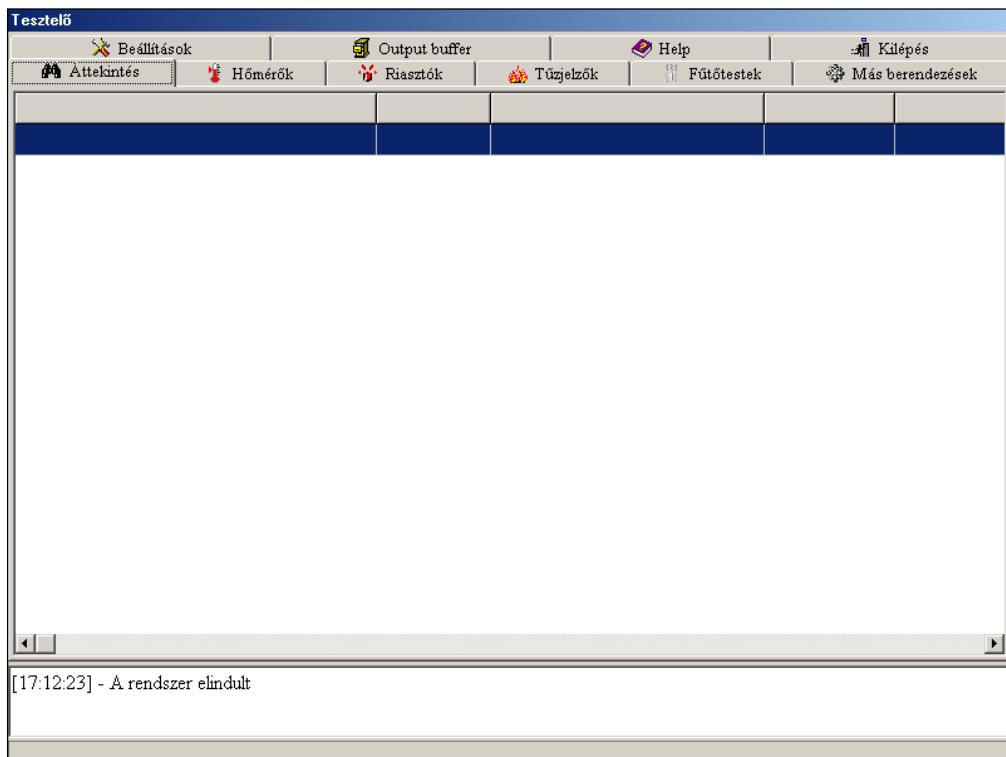
Ha nem sikerül egyiket sem beállítani, akkor a program nem indul el.

A képernyő beállítása után, ha van alapértelmezett épület beállítva, akkor a program megpróbálja betölteni (12. ábra)



12. ábra

Ha a betöltés során bármilyen hiba lép fel, akkor a betöltés megszakad és a program „üresen” indul el (13. ábra), vagyis nekünk kell majd gondoskodni az épület betöltéséről.



Ebben a programban egyik berendezésnek sem lehet a paraméterein változtatni, mert ez a program úgy tekinthető, mintha az épület fizikai megvalósítása lenne. Természetesen a hőmérőknek a mérési eredményeit tudjuk befolyásolni és az egyes riasztók, hőmérők, stb... állapotait is.

Kapcsolat megnyitása, lezárása

Válassza ki a *Beállítások* fület. Az oldal jobb oldalán található *Kommunikáció* panel alsó részén található a *Csatlakozás/Kapcsolat bontása* gomb. A gomb lenyomására a kapcsolat létrejön.

Figyelni kell a port helyes beállításaira. Mind a két programban ugyanazok a paraméterek legyenek beállítva.

III. MENÜSZERKEZET

Az egyes menüpontokat elérhetjük, ha a kívánt fülre kattintunk.



1. ábra

A főmenü:

1. *Áttekintés* : az összes berendezést tartalmazza és itt lehet figyelemmel kísérni a folyamatokat.
2. *Hőmérők* : a hőmérő berendezések működésébe lehet beavatkozni.
3. *Riasztók* : a riasztó berendezések működésébe lehet beavatkozni.
4. *Tűzjelzők* : a tűzjelző berendezések működésébe lehet beavatkozni.
5. *Fűtőtestek* : a fűtőtest berendezések működésébe lehet beavatkozni.
6. *Más berendezések* : más berendezések működésébe lehet beavatkozni.
7. *Beállítások* : a program működésével kapcsolatos beállításokat lehet elvégezni.
8. *Output buffer* : a kommunikációs buffer tartalmát lehet figyelemmel kísérni.
9. *Help* : segítség a programmal kapcsolatban.
10. *Kilépés* : kilépés a programból.

1. Áttekintés menüpont

Ebben a menüpontban tudjuk az összes berendezés állapotát és állapotváltozását figyelemmel kísérni (2. ábra).

Beállítások	Output buffer	Help	Kilépés		
Áttekintés	Hőmérők	Riasztók	Tűzjelzők	Fűtőtestek	Más berendezések
Berendezés neve	Azonosító	Állapot	Aktuális érték	Beállított érték	
Vendég-tűzjelző	14	Normál	-	-	
Dolgozó-riasztó	13	Normál	-	-	
Dolgozó-radiátor	12	Ki	-	-	
Dolgozó-hőmérő	11	-	17 °C	18 °C	
Térvilágítás	10	Ki	-	-	
Nappali-radiátor	9	Ki	-	-	
Nappali-tűzjelző	8	Normál	-	-	
Nappali-riasztó	7	Normál	-	-	
Nappali-hőmérő	6	-	25 °C	23 °C	
Konyha-radiátor	5	Ki	-	-	
Konyha-tűzjelző	4	Normál	-	-	
Konyha-hőmérő	3	-	27 °C	18 °C	
Fürdő-radiátor	2	Ki	-	-	
Fürdő-hőmérő	1	-	20 °C	21 °C	

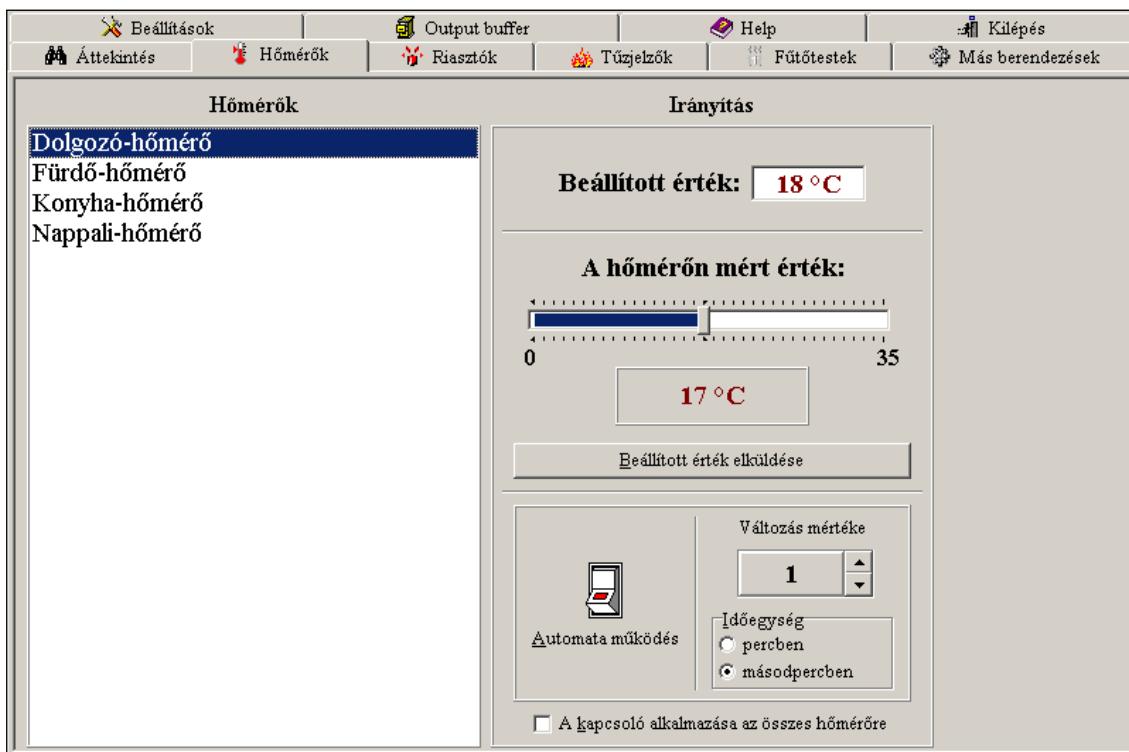
2. ábra

A táblázatról leolvasható:

- berendezés neve,
- azonosítója,
- a berendezés állapota (Ki, Be, Normál, Riasztás)
- hőmérő esetén az aktuális hőmérséklet érték
- hőmérő esetén a beállított hőmérséklet érték

2. Hőmérők menüpont

Ebben a menüpontban szimulálhatunk egy vagy több hőmérőt, amelyek képesek a hőmérséklet változását figyelemmel kísérni és ezt az értéket továbbítani. Fontos, hogy itt nem tudjuk az egyes hőmérők paramétereit állítani, csak a hőmérséklet értékeit változtathatjuk (3. ábra).



3. ábra

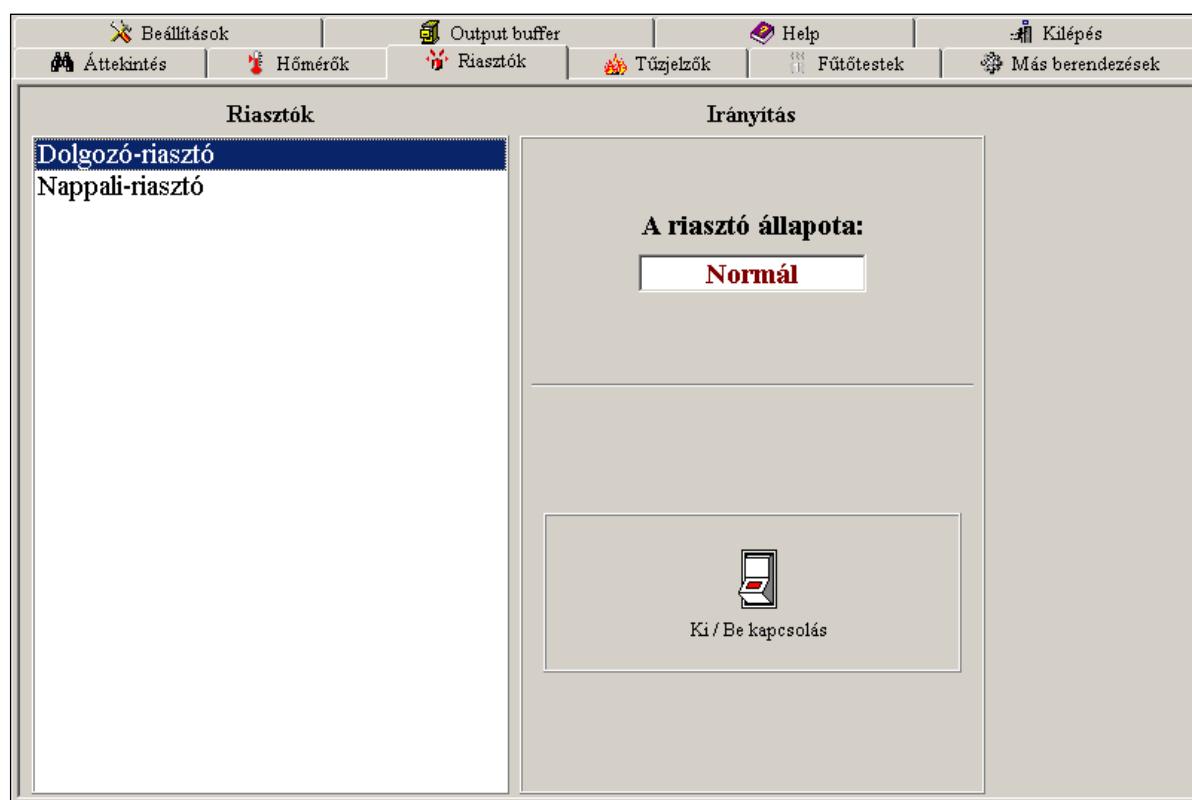
A bal oldalon látható listából választhatjuk ki, hogy melyik hőmérőt szeretnénk szimulálni. A jobb oldali panelen, a listából kiválasztott hőmérő tulajdonságait láthatjuk.

- **Beállított érték:** ezt a hőmérséklet értéket állítottuk be a *Vezérlő* programban.
- **A hőmérőn mért érték:** a hőmérő által mért környezeti hőmérséklet.
- **Beállított érték elküldése:** a mért érték elküldése a *Vezérlő* programnak.
- **Automata működés:** a program szimulálja a hőmérséklet változását a beállított időközönként.

- **Változás mértéke:** időegységet lehet beállítani.
 - Perc esetén: 0 – 15 időegység
 - Másodperc esetén: 0 – 59 időegység
- **Időegység:** a hőmérsékleti értékek változásának gyakoriságát állíthatjuk.
- **A kapcsoló alkalmazása az összes hőmérőre:** ha ki van jelölve, és az automata üzemmód is be van kapcsolva, akkor a változás a beállított időpontban az összes hőmérőre alkalmazásra kerül.

3. Riasztók menüpont

A menüpontban kezelhetjük egy riasztó állapotát. A bal oldalon látható listából választhatjuk ki, hogy melyik riasztót szeretnénk szimulálni. A jobb oldalon található kapcsolóval pedig állíthatjuk a berendezés állapotát. A kapcsoló bekapcsolt állása esetén 'A riasztó állapota' Riasztást jelez, kikapcsolt állapotban (alapállás) Normál értéket mutat.

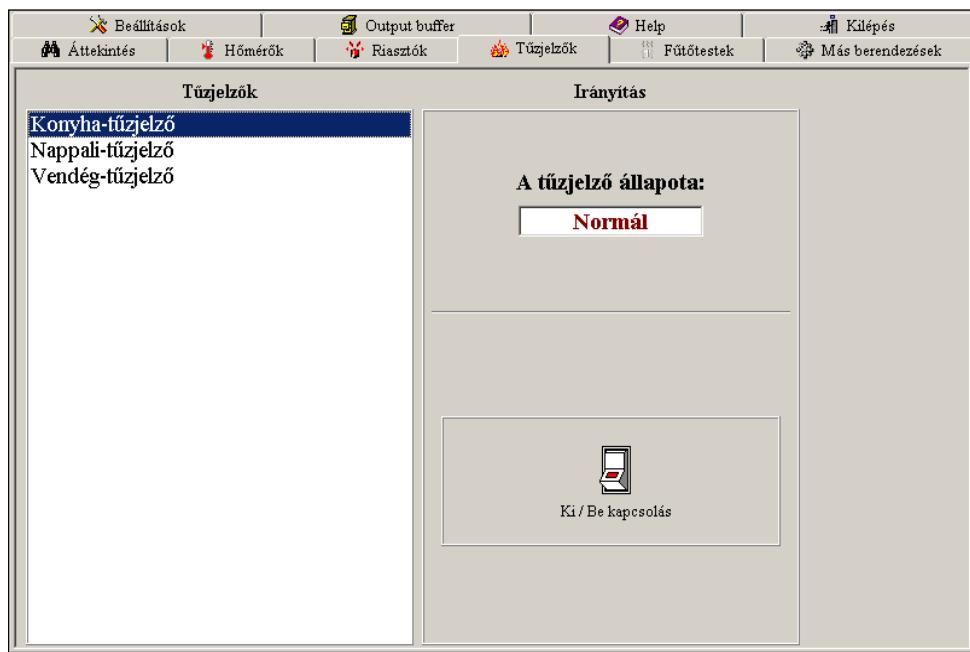


4. ábra

4. Tűzjelzők menüpont

A menüpontban kezelhetjük egy tűzjelző állapotát. A bal oldalon látható listából választhatjuk ki, hogy melyik tűzjelzőt szeretnénk szimulálni. A jobb oldalon található kapcsolóval pedig állíthatjuk a berendezés állapotát. A kapcsoló

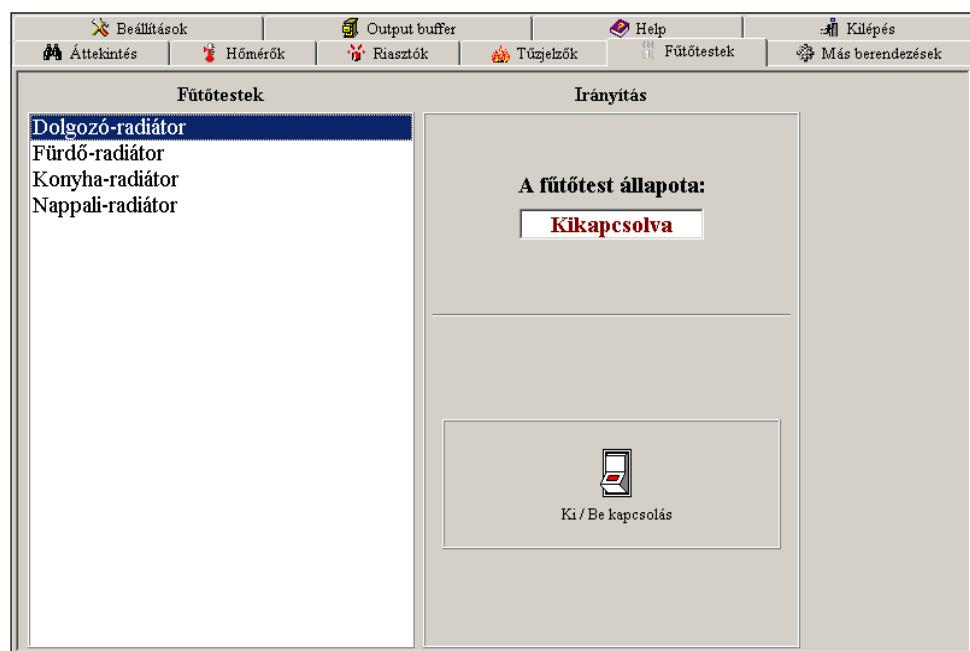
bekapcsolt állása esetén ’A tűzjelző állapota’ Tűzjelzést jelez, kikapcsolt állapotban (alapállás) Normál értéket mutat.



5. ábra

5. Fűtőtestek menüpont

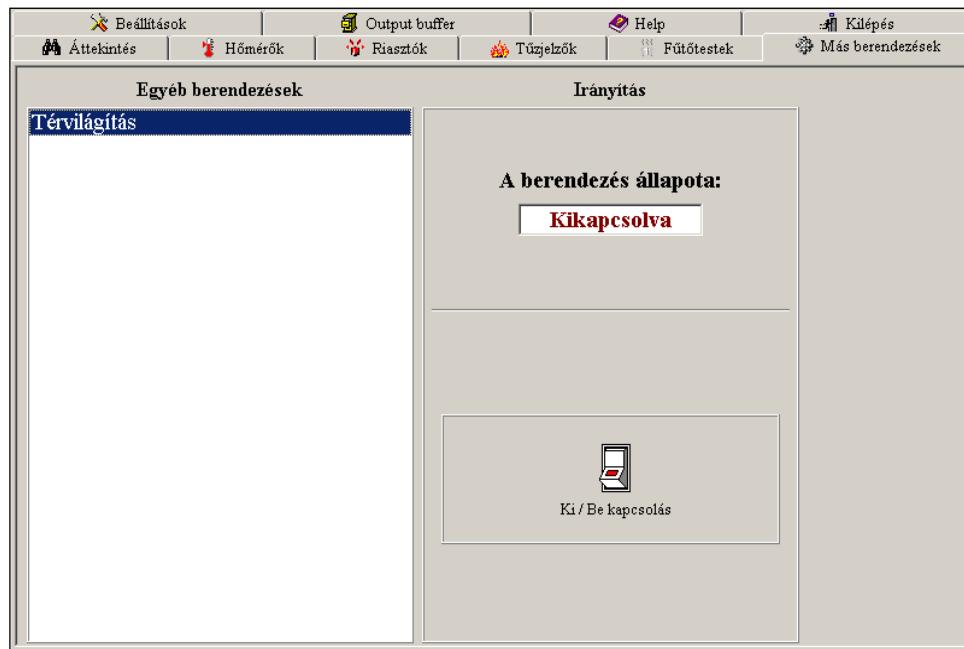
A menüpontban kezelhetjük egy fűtőtest állapotát. A bal oldalon látható listából választhatjuk ki, hogy melyik fűtőtestet szeretnénk szimulálni. A jobb oldalon található kapcsolóval pedig állíthatjuk a berendezés állapotát. A kapcsoló bekapcsolt állása esetén ’A fűtőtest állapota’ Bekapcsolva értéket mutat, kikapcsolt állapotban (alapállás) Kikapcsolva értéket mutat.



6. ábra

6. Más berendezések menüpont

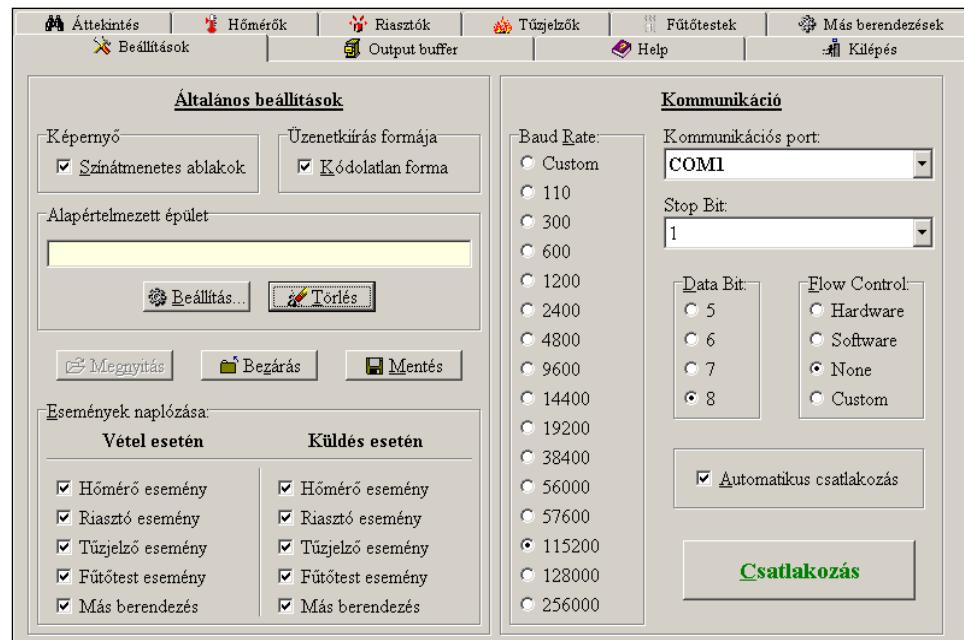
A menüpontban kezelhetjük egyéb berendezések állapotát. A bal oldalon látható listából választhatjuk ki, hogy melyik berendezést szeretnénk szimulálni. A jobb oldalon található kapcsolóval pedig állíthatjuk a berendezés állapotát. A kapcsoló bekapcsolt állása esetén *'A berendezés állapota'* Bekapcsolva értéket mutat, kikapcsolt állapotban (alapállás) Kikapcsolva értéket mutat.



7. ábra

7. Beállítások menüpont

A program működésével kapcsolatos beállítások.



8. ábra

- **Színátmenetes ablakok:** Ha ki van jelölve, akkor aktív és minden ablakra jellemző. Ha nincs kijelölve, akkor a Windows alapértelmezett ablakszíne a háttér kitöltési színe.
- **Kódolatlan forma:** Ha ki van jelölve, akkor aktív és az adatátvitel kiírása szöveges formában történik (9. ábra), különben a naplózás abban a formában történik, mint ahogy az adatok az adatcsatornára kerülnek (10. ábra).

```
[15:59:02] - Kapcsolat létrejött.
Elkuldve: #2#14#0#
(11) Dolgozó-hőmérő beállítása.
Elkuldve: #2#13#0#
Elkuldve: #2#12#0#
(6) Nappali-hőmérő beállítása.
Elkuldve: #2#11#17#
(3) Konyha-hőmérő beállítása.
Elkuldve: #2#10#0#
(1) Fürdő-hőmérő beállítása.
Elkuldve: #2#9#0#
[15:59:04] - (12) Dolgozó-radiátor állapota: Be
Elkuldve: #2#8#0#
Elkuldve: #2#7#0#
Elkuldve: #2#6#25#
Elkuldve: #2#5#0#
Elkuldve: #2#4#0#
Elkuldve: #2#3#27#
```

9. ábra

```
[16:00:41] - Kapcsolat létrejött.
Elkuldve: Tűzjelző(ID:14,Érték:Ki)
(11) Dolgozó-hőmérő beállítása.
Elkuldve: Riasztó(ID:13,Érték:Ki)
Elkuldve: Fütőtest(ID:12,Érték:Ki)
(6) Nappali-hőmérő beállítása.
Elkuldve: Hőmérő(ID:11,Érték:17)
(3) Konyha-hőmérő beállítása.
Elkuldve: Berendezés(ID:10,Érték:Ki)
(1) Fürdő-hőmérő beállítása.
Elkuldve: Fütőtest(ID:9,Érték:Ki)
[16:00:42] - (12) Dolgozó-radiátor állapota: Be
Elkuldve: Tűzjelző(ID:8,Érték:Ki)
Elkuldve: Riasztó(ID:7,Érték:Ki)
Elkuldve: Hőmérő(ID:6,Érték:25)
Elkuldve: Fütőtest(ID:5,Érték:Ki)
Elkuldve: Tűzjelző(ID:4,Érték:Ki)
```

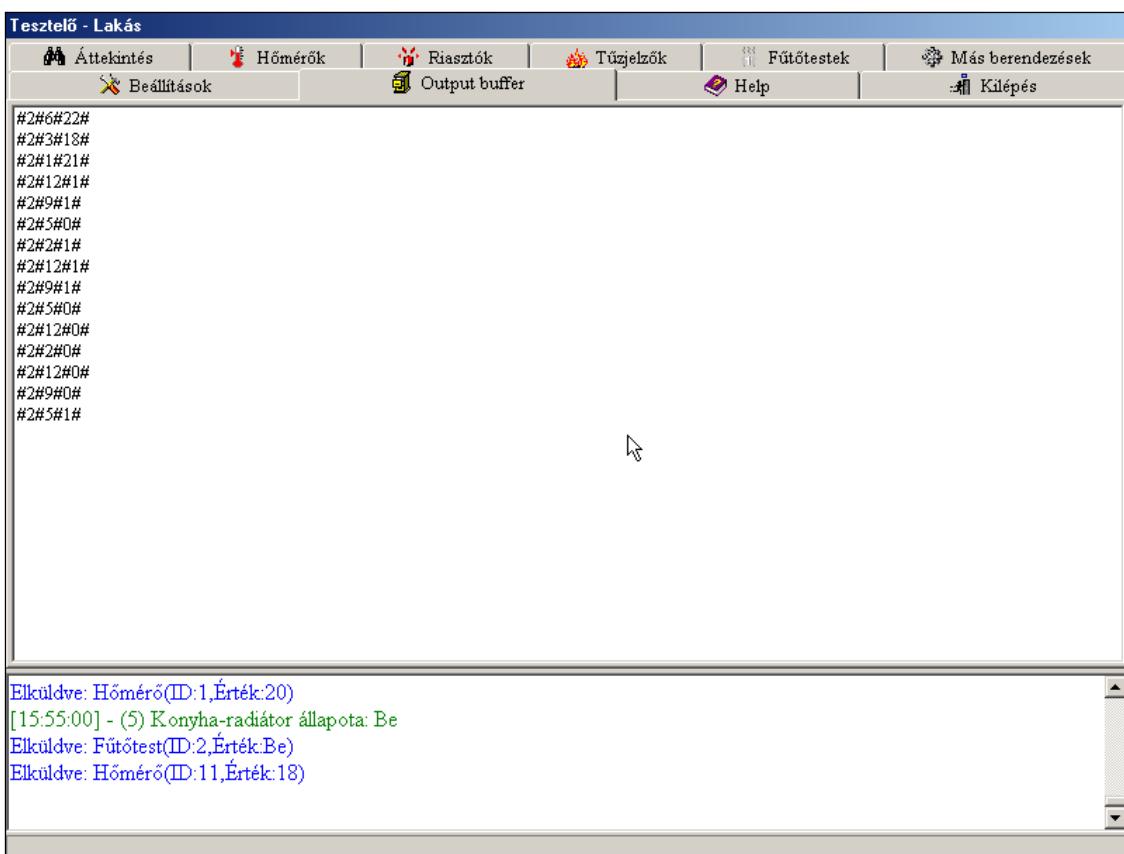
10. ábra

- **Alapértelmezett épület:** Lehetőség van arra, hogy kijelöljünk egy épületet, amely minden programindításkor automatikusan betöltődik.
- **Beállítás:** Itt tudjuk az épületet megkeresni és kiválasztani.
- **Törlés:** Az inputsort törölhetjük. Ha üres az inputsor, akkor nincs alapértelmezett épület beállítva.
- **Megnyitás:** Egy tetszőleges épület berendezéseit lehet betölteni.
- **Bezárás:** Ha van megnyitva épület állományunk, akkor a gomb segítségével bezárhatjuk.
- **Mentés:** Beállítások elmentése.

- **Események naplózása:** Beállíthatjuk, hogy történjen-e naplózás vétel vagy küldés esetén, külön-külön az egyes berendezéstípusokra nézve.
- **Baud Rate:** A kommunikációs csatorna átviteli sebességét lehet beállítani.
- **Kommunikációs port:** Kiválasztható, hogy melyik csatornán szeretnénk kommunikálni a vezérlő programmal.
- **Stop Bit:** Beállíthatjuk a Stop bit hosszát. Ez biztosítja a vevőnek, hogy felkészüljön a következő csomag küldésére.
- **Data Bit:** Beállíthatjuk, hogy egyszerre mennyi adatbit kerüljön elküldésre.
- **Flow Control:** Megvédi a számítógépet a feldolgozási sebességénél gyorsabb adatvételtől.
- **Automatikus csatlakozás:** Ha ki van jelölve, akkor aktív és a program indításakor, ha van alapértelmezett épület beállítva, akkor megnyitja a kommunikációs csatornát.
- **Csatlakozás/Kapcsolat bontás:** A beállított kommunikációs csatorna megnyitása, illetve bezárása.

8. Output buffer menüpont

A kifelé menő kommunikációs folyamatot kíséri figyelemmel. Az értékei pontosan megegyeznek a porton kimenő üzenetekkel. Ez a forma már csak kódolt lehet.



11. ábra

A listában a még elküldésre nem került üzenetek sorakoznak. A lista olyan ütemben változik, ahogy az átvitel történik. Ennek megfigyeléséhez állítsuk be a hőmérőknél az *Automata működés-t* és jelöljük be az *A kapcsoló alkalmazása az összes hőmérőre* kiválasztó dobozt. Így folyamatos működést kapunk eredményül és megfigyelhetők a lista változásai.

9. Help menüpont

Segítséget nyújt a program kezelésében.

A help az F1 billentyű lenyomásával azonnal elérhető.

10. Kilépés menüpont

A program bezárása. Ha a kapcsolat aktív a kilépéskor, akkor előbb a port lezáródik és csak aztán záródik be a program.

**VEZÉRLŐ
FELHASZNÁLÓI DOKUMENTÁCIÓ**

I. INSTALLÁLÁS

1. Szükséges hardver konfiguráció

- PC 100 MHz
- CD-ROM
- 16 MB RAM
- 10 MB szabad lemezterület

2. Szükséges szoftver környezet

- Microsoft Win9x/ME/2000/NT

3. Installálás

- Ha a számítógép alkalmas a CD lemez automatikus indítására:
A telepítőlemez behelyezése után a telepítőprogram automatikusan elindul.
- Ha a telepítést a CD-ről indítjuk:
A CD lemez Setup könyvtárában indítsuk el a *Setup.exe* programot.

Az *Install.exe* program indulása után válasszuk a '*Vezérlőprogram telepítése*' funkciót és kövessük a program utasításait.

II. MENÜSZERKEZET

Az egyes menüpontokat elérhetjük az ALT és a menü nevében az aláhúzott betű billentyűjének együttes lenyomásával.

Pl.: a File menü megnyitása: ALT+F.



1. ábra

A főmenü:

1. *File* : a programmal kapcsolatos műveleteket lehet elvégezni.
2. *Beállítások* : a program működésével kapcsolatos beállítások
3. *Napló* : eseménynapló kezelése, megtekintése
4. *Berendezések* : berendezések listázása
5. *Súgó* : segítség a program használatához

1. A *File* menü

A programmal kapcsolatos műveleteket lehet elvégezni.

A *File* menüben az alábbi menüpontok közül lehet választani:

- *Megnyitás* menüpont: meglévő épület betöltése.
- *Mentés* menüpont: épület mentése.
- *Bezárás* menüpont: épületfájl bezárása.
- *Kapcsolat megnyitása*: kommunikációs csatorna létrehozása
- *Kapcsolat bezárása*: kommunikációs csatorna bezárása
- *Kilépés* menüpont: kilépés a programból.

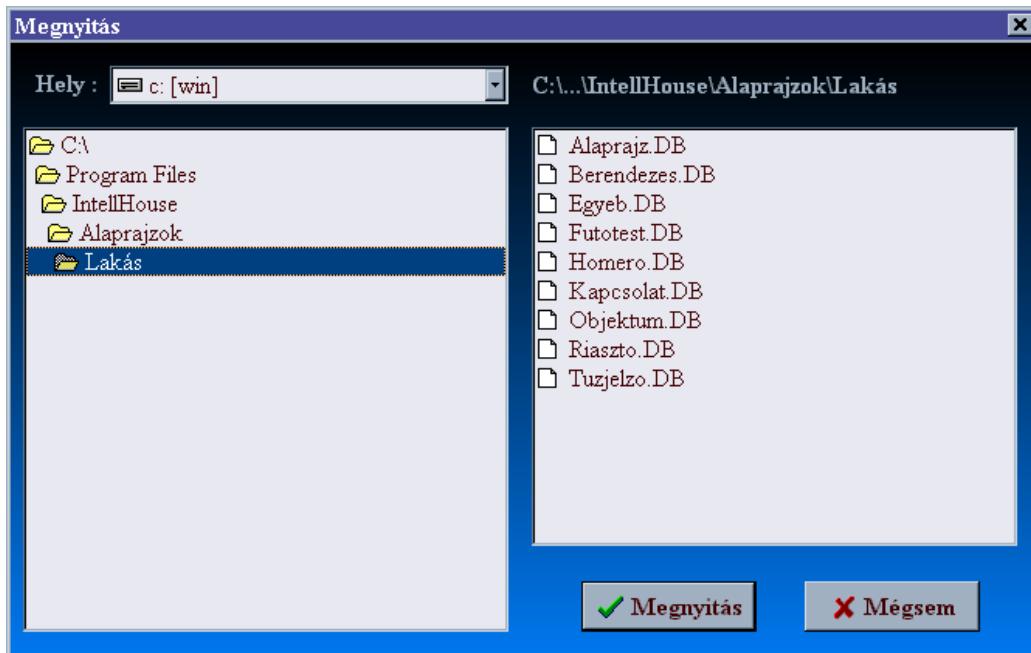
1.1. *Megnyitás* menüpont

Ha már létezik olyan épület amit korábban a Tervező programmal korábban létrehoztunk, akkor ebben a menüpontban tudjuk betölteni.

Betöltéshez csak ki kell választani a megfelelő könyvtárat, amelyikben a betöltendő épület található. Ahogy a könyvtárak között lépkedünk, a program automatikusan megvizsgálja, hogy a megfelelő adatbázisok megtalálhatók-e az adott könyvtárban.

- Ha megtalálhatók, akkor a *Megnyitás* gomb aktívvá válik (2. ábra). A *Megnyitás* gombra kattintva az épület alaprajza és a hozzá tartozó adatbázisok (berendezések, objektumok, stb....) betöltődnek.

- Ha egy könyvtárat kiválasztva, a *Megnyitás* gomb nem lesz aktív, akkor biztosak lehetünk abban, hogy a könyvtár nem tartalmazza a megfelelő fájlokat.



2. ábra

1.2. *Mentés* menüpont

Egy épület berendezéseit, -objektumait tudjuk elmenteni.

A mentés minden az ‘Alaprajzok’ könyvtárba történik, azon belül pedig az épület nevével megegyező alkönyvtárba. Ha az alkönyvtár nem létezik, akkor az automatikusan létrejön.

A mentés során a korábbi verzió felülíródik.

1.3. *Bezárás* menüpont

A bezárás menü választása esetén, a megnyitott épülethez tartozó összes adatbázis bezárásra kerül. Ha volt változtatás, akkor a bezárás előtt mentésre is van lehetőség.

1.4. *Kapcsolat megnyitása* menüpont

Kommunikációs kapcsolat létrehozása a *Tesztelő* programmal, amely a soros adatkábelrel összekötött másik számítógépen fut.

A kapcsolat létrehozására 5 másodperc áll rendelkezésre. Ha ez idő alatt nem sikerül kapcsolatot teremteni a másik számítógéppel, akkor a soros port lezáródik. Az újabb próbálkozásra csak ezután van lehetőség. Miután a kapcsolat létrejön a meghatározott időn belül, akkor a program lekérdezi a berendezések adatait és frissíti az adatokat.

1.5. *Kapcsolat bezárása* menüpont

Kommunikációs kapcsolat lezárása a *Tesztelő* programmal, amely a soros adatkábellel összekötött másik számítógépen fut.

1.6. *Kilépés* menüpont

Kilépés a programból.

Ha az épülethez tartozó adatbázisok módosultak, pl.: változott egy berendezés tulajdonsága, paramétere, stb...., akkor a program kérdezi, hogy mentjük-e a változásokat. Válaszolunk Igen-nel, ha menteni szeretnénk a változásokat. Ha Nem-mel válaszolunk, akkor a legutóbbi mentett változatot tartja meg, és az azóta történt változtatásokat elveti.

2. Beállítások menü

A program és az egyes berendezések működésével kapcsolatos beállítások.

A *Beállítások* menüben az alábbi menüpontok közül lehet választani:

- *Hőmérő*: egy hőmérő berendezés tulajdonságainak beállítása.
- *Riasztó*: egy riasztó berendezés tulajdonságainak beállítása.
- *Tűzjelző*: egy tűzjelző berendezés tulajdonságainak beállítása.
- *Fűtőtest*: egy fűtőtest berendezés tulajdonságainak beállítása.
- *Egyéb*: más berendezés tulajdonságainak beállítása.
- *Berendezés lista*: berendezések gyors elérése irányítópulton keresztül.
- *Beállítások*: program működésével kapcsolatos beállítások.

2.1. *Hőmérő* menüpont

Egy hőmérő tulajdonságainak beállítása.

Egy hőmérő feladata az, hogy ha hőmérsékletváltozást érzékel, akkor az új értéket jelezze a programnak, amely az itt beállított adatok alapján eldönti, hogy a hozzá rendelt berendezésekkel mi a teendő.

Egy hőmérőnek beállíthatjuk a méréstartományát (min, max), melyen belül a hőmérő érzékel a változásokat. Ezen kívül megadjuk azt az értéket, amelyet ha elér a hőmérséklet, akkor a program a megadott berendezéseket ki, illetve –be kapcsolja. A beállítást segíti a hőmérséklet értékeit tartalmazó csúszka.

Ha elfogadjuk a beállításokat, akkor a soros porton keresztül frissítésre kerül a *Tesztelő* program beállításai is (3. ábra).



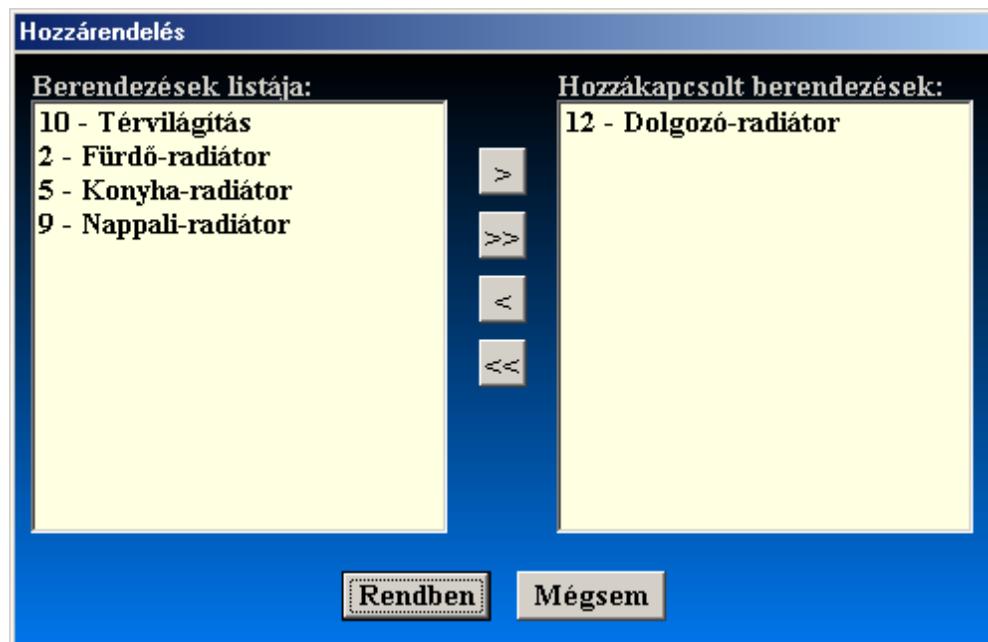
3. ábra

- **Hőmérő:** Az ablak bal felső sarkában található listából kiválaszthatjuk, hogy melyik hőmérő adatait szeretnénk beállítani.
- **Naplázás....:** Ha ki van jelölve, akkor minden hőmérsékletváltozás naplázásra kerül.
- **Kapcsolódó berendezések:** A hőmérőhöz hozzárendelt berendezések listája.
- **Beállítás:** A hőmérőhöz hozzárendelt berendezések listáját lehet karbantartani (ld.: 2.1.1. Eszközhöz kapcsolódó berendezések beállítása).
- **Méréshatár:** A hőmérőn mérhető minimális és maximális hőmérséklet értéket lehet módosítani.
- **Beállított érték:** Ha a hőmérséklet eléri ezt az értéket, akkor a program ki vagy –be kapcsolja a berendezéseket.
- **Rendben:** A beállított értékek elfogadása és az ablak bezárasa.
- **Alkalmaz:** A beállított értékek elfogadása, de az ablak nyitva marad.
- **Kilépés:** Az ablak bezárasa. A módosítások nem kerülnek elfogadásra.

2.1.1. Eszközhöz kapcsolódó berendezések beállítása

Egy eszközhöz hozzárendelt berendezések listáján lehet változtatni (4. ábra):

- hozzáadás,
- módosítás,
- törlés.



4. ábra

- **Berendezések listája:** Az épületben található olyan berendezések listája, amelyeket vezérelni lehet.
- **Hozzákapcsolt berendezések:** Azoknak a berendezéseknek a listája, amelyeket hozzárendelünk a hőmérőhöz.

A két listát a köztük található négy gombbal módosíthatjuk.

- egy berendezés hozzárendelése a hőmérőhöz
- az összes berendezés hozzárendelése a hőmérőhöz
- egy berendezés elvétele a hozzárendelt listából
- az összes berendezés elvétele a hozzárendelt listából.

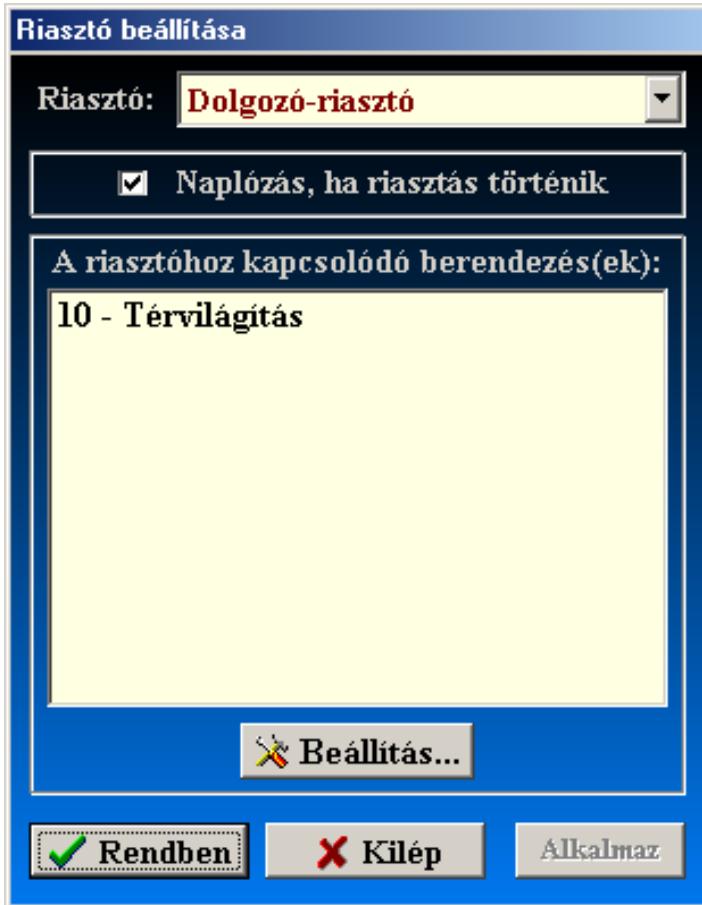
2.2. Riasztó menüpont

Egy riasztó tulajdonságainak beállítása.

Egy riasztó feladata az, hogy ha egy aktiválását kiváltó esemény történik, akkor azt az eseményt jelezze a programnak, amely az itt beállított adatok alapján eldönti, hogy a hozzá rendelt berendezésekkel mi a teendő. (5. ábra)

- **Riasztó:** Az ablak bal felső sarkában található listából kiválaszthatjuk, hogy melyik riasztó adatait szeretnénk beállítani.
- **Naplázás....:** Ha ki van jelölve, akkor minden riasztási esemény naplázásra kerül.
- **Kapcsolódó berendezések:** A riasztóhoz hozzárendelt berendezések listája.
- **Beállítás:** A riasztóhoz hozzárendelt berendezések listáját lehet karbantartani (ld.: 2.1.1. Eszközhöz kapcsolódó berendezések beállítása).

- **Rendben:** A beállított értékek elfogadása és az ablak bezárása.
- **Alkalmaz:** A beállított értékek elfogadása, de az ablak nyitva marad.
- **Kilépés:** Az ablak bezárása. A módosítások nem kerülnek elfogadásra.



5. ábra

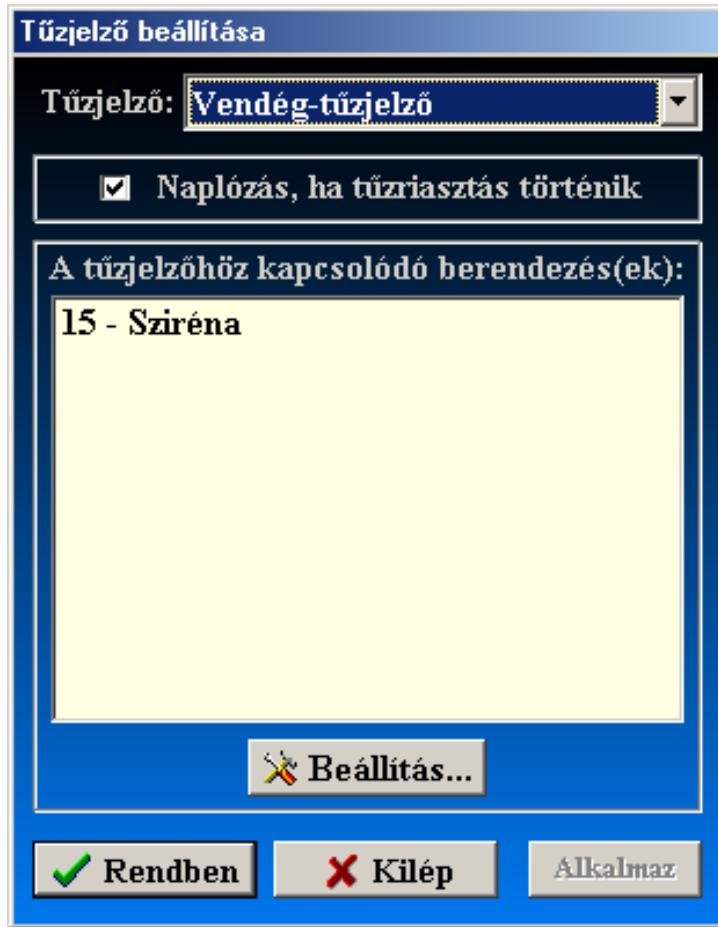
2.3. Tűzjelző menüpont

Egy tűzjelző tulajdonságainak beállítása.

Egy tűzjelző feladata az, hogy ha egy aktiválását kiváltó esemény történik (tűzeset, füstérzékelés, stb...), akkor azt az eseményt jelezze a programnak, amely az itt beállított adatok alapján eldönti, hogy a hozzá rendelt berendezésekkel mi a teendő. (6. ábra)

- **Tűzjelző:** Az ablak bal felső sarkában található listából kiválaszthatjuk, hogy melyik tűzjelző adatait szeretnénk beállítani.
- **Naplázás...:** Ha ki van jelölve, akkor minden tűzjelző esemény naplázásra kerül.
- **Kapcsolódó berendezések:** A tűzjelzőhöz hozzárendelt berendezések listája.
- **Beállítás:** A tűzjelzőhöz hozzárendelt berendezések listáját lehet karbantartani (ld.: 2.1.1. Eszközhöz kapcsolódó berendezések beállítása).
- **Rendben:** A beállított értékek elfogadása és az ablak bezárása.

- **Alkalmaz:** A beállított értékek elfogadása, de az ablak nyitva marad.
- **Kilépés:** Az ablak bezárása. A módosítások nem kerülnek elfogadásra.



6. ábra

2.4. Fűtőtest menüpont

Egy fűtőtest működési tulajdonságainak beállítása.

Változtatás esetén az új beállítások csak akkor lépnek életbe, amikor az ablakot bezártuk. (7. ábra)

- **Fűtőtest:** Az ablak bal felső sarkában található listából kiválaszthatjuk, hogy melyik fűtőtest adatait szeretnénk beállítani.
- **Naplózás...:** Ha ki van jelölve, akkor minden fűtőtest esemény naplózásra kerül.
- **Automata vezérlés:** Ha be van kapcsolva, akkor a működést a program vezérli. Tehát a berendezés ki-be kapcsolása távműködtetéssel történik.
- **Direkt kapcsolás:** A berendezés kézi ki-be kapcsolása. A kapcsolót állását csak akkor lehet módosítani, ha az Automata vezérlés ki van kapcsolva. Tehát ezzel a kapcsolóval a berendezést közvetlenül - a számítógép nélkül – tudjuk vezérelni.

- **Rendben:** A beállított értékek elfogadása és az ablak bezárása.
- **Alkalmaz:** A beállított értékek elfogadása, de az ablak nyitva marad.
- **Kilépés:** Az ablak bezárása. A módosítások nem kerülnek elfogadásra.



7. ábra

2.5. Egyéb berendezés menüpont

Egy tetszőleges berendezés működési tulajdonságainak beállítása. Olyan berendezést kezelhetünk itt, amelynek működési állapota legalább két állású (ki-be kapcsolható). Változtatás esetén az új beállítások csak akkor lépnek életbe, amikor az ablakot bezártuk. (7. ábra)

- **Egyéb:** Az ablak bal felső sarkában található listából kiválaszthatjuk, hogy melyik berendezés adatait szeretnénk beállítani.
- **Naplózás...:** Ha ki van jelölve, akkor a berendezés minden eseménye naplózásra kerül.
- **Automata vezérlés:** Ha be van kapcsolva, akkor a működést a program vezérli. Tehát a berendezés ki-be kapcsolása távműködtetéssel történik.
- **Direkt kapcsolás:** A berendezés kézi ki-be kapcsolása. A kapcsolót állását csak akkor lehet módosítani, ha az Automata vezérlés ki van kapcsolva. Tehát ezzel a kapcsolóval a berendezést közvetlenül - a számítógép nélkül – tudjuk vezérelni.
- **Rendben:** A beállított értékek elfogadása és az ablak bezárása.
- **Alkalmaz:** A beállított értékek elfogadása, de az ablak nyitva marad.

- **Kilépés:** Az ablak bezárása. A módosítások nem kerülnek elfogadásra.



8. ábra

2.6. Vezérlőpult menüpont

Berendezés	Tulajdonság
Vendég-tűzelző	
Dolgozó-riasztó	
Dolgozó-radiátor	
Dolgozó-hőmérő	
Térvilágítás	
Nappali-radiátor	
Nappali-tűzelző	
Nappali-riasztó	
Nappali-hőmérő	
Konyha-radiátor	
Konyha-tűzelző	
Konyha-hőmérő	
Fürdő-radiátor	
Fürdő-hőmérő	

Berendezés	Tulajdonság
Tulajdonság	Érték
ID	12
Neve	Dolgozó-radiátor
Típusa	Fűtőtest
Egység neve	Szint-2
Objektum neve	Dolgozsoba

9. ábra

10. ábra

Ebből a vezérlőpultból tudjuk az összes berendezést irányítani. Felfoghatjuk úgy is, mint egy távirányítót (9. ábra). A panelen kiválasztjuk azt a berendezést, amelyet irányítani szeretnénk, majd átkapcsolunk a *Tulajdonságok* gombra (10. ábra). Az átkapcsolást egy dupla kattintással is elérhetjük, ha az aktív berendezésen kattintunk.

A tulajdonságok között szerepelnek:

- *ID*: berendezés azonosítója
- *Neve*: berendezés neve
- *Tipusa*: berendezés fajtája
- *Egység neve*: a berendezés melyik épületegységben található
- *Objektum neve*: a berendezés melyik objektumban található

A berendezés típusától függően többféle kezelőfelületet kaphatunk.

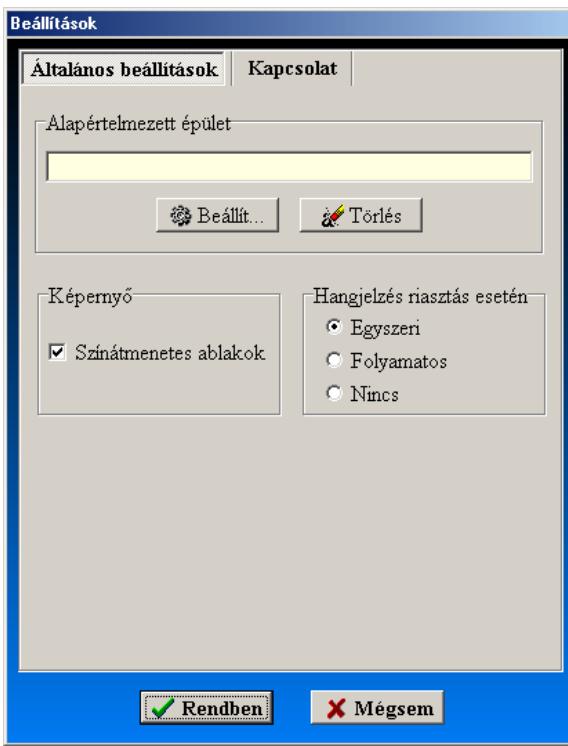
A 10. ábrán egy fűtőtest tulajdonságai láthatóak, melynek a működésébe közvetlenül is beavatkozhatunk. Egy hőmérő esetén nyomon tudjuk követni a hőmérsékletváltozást, egy riasztó vagy tűzjelző esetében pedig a jelzőberendezés állapotát.

2.7. Beállítások menüpont

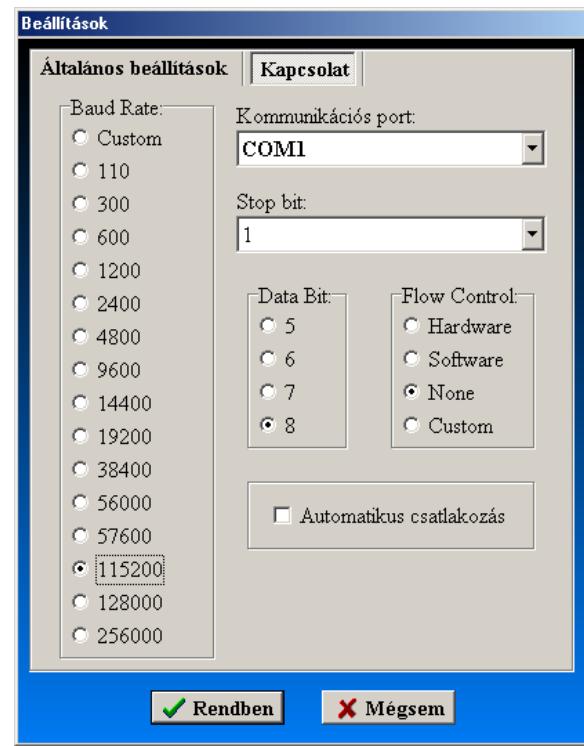
A program működésével kapcsolatos beállítások.

Az ablak két részből áll:

- általános beállítások (11. ábra)
- kommunikációs beállítások (12. ábra)



11.ábra



12. ábra

2.7.1. Általános beállítások menüpont

- **Alapértelmezett épület:** Lehetőség van arra, hogy kijelöljünk egy épületet, amely minden programindításkor automatikusan betöltődik.
- **Beállít:** Itt tudjuk az épületet megkeresni és kiválasztani.
- **Törlés:** Az inputsort törölhetjük. Ha üres az inputsor, akkor nincs alapértelmezett épület beállítva.
- **Színátmenetes ablakok:** Ha ki van jelölve, akkor aktív és minden ablakra jellemző. Ha nincs kijelölve, akkor a Windows alapértelmezett ablakszíne a háttér kitöltési színe.
- **Hangjelzés riasztás esetén:** Megadhatjuk, hogy riasztás esetén milyen legyen a figyelmeztető hangjelzés típusa. Ez lehet
 - Egyszeri: egyszeri figyelmeztetés
 - Folyamatos: folyamatos hangjelzés
 - Nincs: ebben az esetben nincs hangjelzés

2.7.2. Kapcsolat menüpont

- **Baud Rate:** A kommunikációs csatorna átviteli sebességét lehet beállítani.
- **Kommunikációs port:** Kiválasztható, hogy melyik csatornán szeretnénk kommunikálni a vezérlő programmal.
- **Stop Bit:** Beállíthatjuk a Stop bit hosszát. Ez biztosítja a vevőnek, hogy felkészüljön a következő csomag küldésére.
- **Data Bit:** Beállíthatjuk, hogy egyszerre mennyi adatbit kerüljön elküldésre.
- **Flow Control:** Megvédi a számítógépet a feldolgozási sebességénél gyorsabb adatvételtől.
- **Automatikus csatlakozás:** Ha ki van jelölve, akkor aktív és a program indításakor, ha van alapértelmezett épület beállítva, akkor megnyitja a kommunikációs csatornát.

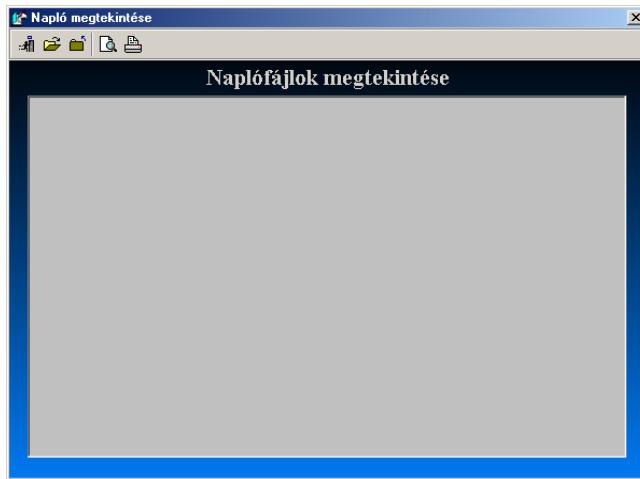
3. Napló menüpont

A berendezések működésének nyomon követését teszi lehetővé (13. ábra). Az egyes berendezések minden állapotváltozása rögzítésre kerül, hogy később a történések rekonstruálhatók legyenek. A naplózási opció, minden berendezés esetén a saját beállítási ablakában ki- és bekapcsolható, igény szerint.

Az ablakon található gombok:

- *Kilépés* gomb
- *Megnyitás* gomb
- *Bezárás* gomb

- *Nyomtatási kép* gomb
- *Nyomtatás* gomb



13. ábra

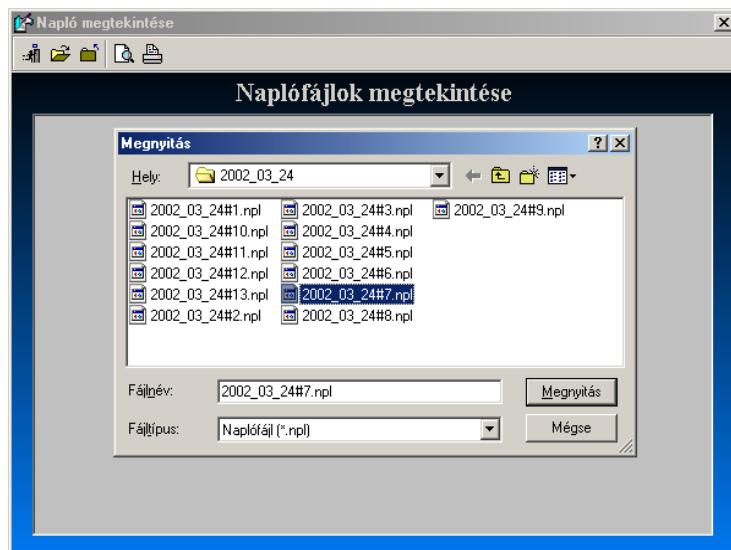
Egy nap több eseményfájlból is állhat, amit a 14. ábrán látható gombbal hozhatunk létre.



14. ábra

Gombok:

- *Kilépés* gomb
A napló ablak bezárása, visszatérés a főprogramba.
- *Megnyitás* gomb
Naplófájlok megnyitása (15. ábra).



15. ábra

A naplók az IntellHouse\Naplo könyvtárban találhatók. A könyvtáron belül az egyes napok ismét alkönyvtárakban találhatók, ahol az egyes naplófájlok találhatók.

- *Bezárás gomb*
A megnyitott fájl bezárása.
- *Nyomtatási kép gomb*
A nyomtatásra kiválasztott fájl nyomtatási képének megjelenítése.
- *Nyomtatás gomb*
A nyomtatásra kiválasztott fájl nyomtatása.

4. Berendezések menüpont

A menüpontban az épületben található berendezésekéről kapunk egy áttekintő táblázatot, melyben egy berendezésről fel vannak sorolva, hogy mi a berendezés azonosítója (ID), neve, típusa, melyik épületegységben található, és melyik objektumhoz tartozik (16. ábra).

Ha kiválasztunk egy berendezést és duplán kattintunk a soron, akkor az adott berendezéshez megjelenik a beállító ablak, ahol a berendezés tulajdonságait tudjuk megváltoztatni.

Berendezések listája				
ID	Berendezés neve	Típusa	Épületegység neve	Objektum neve
14	Vendég-tűzelző	Tűzelző	Szint-2	Vendégszoba
13	Dolgozó-riasztó	Riasztó	Szint-2	Dolgozószoba
12	Dolgozó-radiátor	Fűtőtest	Szint-2	Dolgozószoba
11	Dolgozó-hőméرő	Hőmérv	Szint-2	Dolgozószoba
10	Térvilágítás	Egyéb	Szint-1	Egyikhez sem
9	Nappali-radiátor	Fűtőtest	Szint-1	Nappali
8	Nappali-tűzelző	Tűzelző	Szint-1	Nappali
7	Nappali-riasztó	Riasztó	Szint-1	Nappali
6	Nappali-hőmérv	Hőmérv	Szint-1	Nappali
5	Konyha-radiátor	Fűtőtest	Szint-1	Konyha
4	Konyha-tűzelző	Tűzelző	Szint-1	Konyha
3	Konyha-hőmérv	Hőmérv	Szint-1	Konyha
2	Fürdő-radiátor	Fűtőtest	Szint-1	Fürdőszoba
1	Fürdő-hőmérv	Hőmérv	Szint-1	Fürdőszoba

 Kilepés

16. ábra

5. Help menüpont

Segítséget nyújt a program kezelésében.

A help az F1 billentyű lenyomásával azonnal elérhető.

6. Áttekintés gomb

Elő kapcsolat esetén, az épületben lévő berendezések működésének folyamatait lehet figyelemmel kísérni (17. ábra).

Berendezés neve	Azonosító	Vezérlés	Állapot	Aktuális érték	Beállított érték
Sziréna	15	Kézi	Ki		
Fürdő-hőmérő	1	Automata		21 °C	21 °C
Fürdő-radiátor	2	Automata	Ki		
Konyha-hőmérő	3	Automata		18 °C	18 °C
Konyha-tűjelző	4	Automata	Normál		
Konyha-radiátor	5	Automata	Ki		
Nappali-hőmérő	6	Automata		22 °C	23 °C
Nappali-riasztó	7	Automata	Normál		
Nappali-tűjelző	8	Automata	Normál		
Nappali-radiátor	9	Automata	Be		
Térvilágítás	10	Automata	Ki		
Dolgozó-hőmérő	11	Automata		18 °C	18 °C
Dolgozó-radiátor	12	Automata	Ki		
Dolgozó-riasztó	13	Automata	Normál		
Vendég-tűjelző	14	Automata	Normál		

17. ábra

Táblázat felépítése:

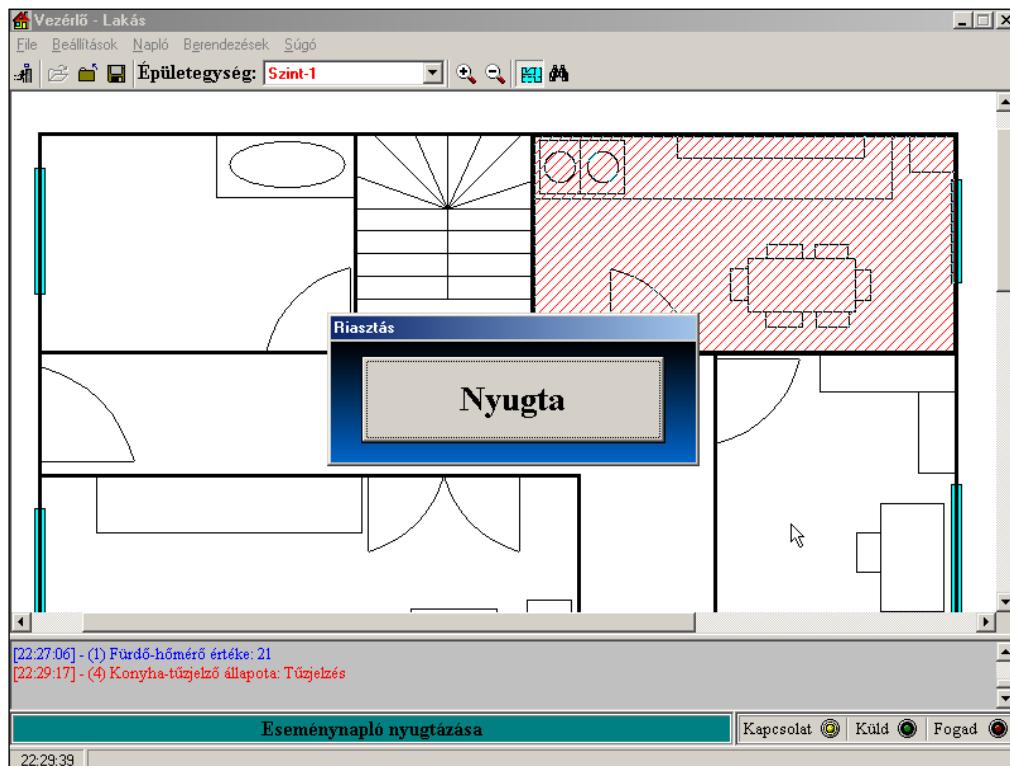
- *Berendezés neve*
A berendezés vagy érzékelő neve.
- *Azonosító*
Az egyértelmű azonosítás miatt, minden berendezésnek van egy száma.
- *Vezérlés*
Megmondja, hogy a berendezés automata vagy kézi vezérlés alatt áll.
- *Állapot*
A berendezés aktuális állapotát mondja meg.
- *Aktuális érték*
A hőmérőn jelenleg mért érték.
- *Beállított érték*
A hőmérőn beállított jelzési érték.

7. Riasztási esemény és kezelése

Tűzjelző vagy riasztó működésbe lépésekor egy riasztási esemény keletkezik, amely – ha be van állítva (2.7.1), akkor hangjelzés kíséretével – látható, hogy melyik helységben történt az esemény.

Riasztás esetén bárhol is legyen a vezérlés (másik épületegységen, áttekintés oldalon) a kép átugrik a riasztás helyét jelző alaprajzra. Ezzel egyidőben megjelenik egy nyugtázó gomb, amellyel a riasztás vételét lehet lenyugtázni. Nyugtázás esetén az érzékelő visszakapcsol alapállapotba.

Ha több helyről is érkezik riasztás, akkor a képernyő az utolsónak befutott riasztási helyet mutatja, de az alaprajzon minden esetben jelzsre kerülnek (piros vonalas rácsozás) a helyszínek. Nyugtáskor ebben az esetben minden érzékelő egyszerre kerül alapállapotba.



18. ábra

II.

FEJLESZTŐI

DOKUMENTÁCIÓ

AZ INTELLHOUSE PROGRAMCSOMAG FEJLESZTŐI DOKUMENTÁCIÓJA

Feladat leírása

Az IntellHouse programcsomag egy épület berendezéseinek és érzékelőinek a menedzselésére és vezérlésére készült.

A programcsomagon belül lehetőség van egy épület alaprajzának a grafikus megrajzolására, az alaprajzon különböző berendezések és érzékelők elhelyezésére valamint paramétereik beállítására. Ezeket a berendezéseket és érzékelőket tudjuk egy másik számítógép segítségével életre kelteni, szimulálni.

A szimuláció során az eseményeket naplózni kell tudni, hogy később a történeteket, eseményeket vissza lehessen nézni.

Ezek alapján a programcsomagnak három, egymástól jól elkülöníthető feladatot kell megoldania:

- Alaprajz megrajzolása és bevitelle a számítógépbe
- Berendezések, érzékelők vezérlése, állapotuk figyelemmel kísérése és az események naplózása
- Az irányító rendszer tesztelése, berendezések szimulálása

E három szempont alapján a feladatot is három különálló programmal oldottam meg, melyek fejlesztői dokumentációi a következő fejezetekben találhatóak meg. Tehát a programcsomag az alábbi három programot tartalmazza:

- *Tervezo.exe* : alaprajz megrajzolása, berendezések elhelyezése
- *Vezerlo.exe* : berendezések vezérlése, naplózás
- *Tesztelo.exe* : események, folyamatok szimulálása

Feladathoz kapcsolódó fogalmak

- Alakzat: Olyan geometriai alakzat, amely lehet egyszerű: vonal, téglalap, ellipszis vagy összetett: ajtó, ablak.
- Berendezés: Olyan eszköz, amely külső eseményeket tud fogadni vagy a számítógép parancsait hajtja végre. Ezek alapján megkülönböztetünk: érzékelőket és végberendezéseket.
- Épület: Olyan egység, amely egy vagy több építményt, létesítményt foglal magába. Épület lehet például egy lakás, családi ház, de épület lehet egy irodaház vagy intézmény is, amelyek egy vagy több épületegységből épülhetnek fel.

- **Épületegység**: Olyan épületrész, amely az épület egészéhez tartozik, de önálló egységeként szeretnénk kezelni. Például egy irodaházban az egyes szintek, vagy egy családi házban a melléképület vagy a kert.
- **Érzékelő**: Olyan berendezés, amely a külső világból jövő jeleket érzékelni tudja és ezt elektromos jellé alakítva továbbküldi a számítógépnek.
- **Objektum**: olyan kitüntetett terület, amely számunkra fontos elemeket tartalmaz. Például ez lehet egy szoba, folyosó, épületrész, stb.
- **Toll**: A tervező programban ezzel tudunk a rajzvászonra rajzolni. A tollnak három fontos tulajdonsága van, amit a programban meg is változtathatunk: színe, mérete (vagyis milyen vastagon fogjon), típusa (folytonos, szaggatott).
- **Végberendezés**: olyan berendezés, amely csak parancsot tud végrehajtani és visszajelzést csak a saját állapotáról tud adni. Pl.: elektromos ajtó-, ablak, fűtőtest, világítás, stb.

A programcsomagban használt UNIT-ok:

A felsorolásra kerülő unit-ok a programcsomag egészére vonatkoznak. A fájlnév mellett zárójelben fel van tüntetve, hogy melyik program(ok)-nál kerül felhasználásra. minden unit-hoz tartozik egy rövid leírás a funkciójáról.

- ***AboutUnit.pas* (Tervező)**
A készítőről és a programról ad információt.
- ***AboutUnit.pas* (Vezérlő)**
A készítőről és a programról ad információt.
- ***AjtoUnit.pas* (Tervező)**
Az ajtó form eljárásait, függvényeit tartalmazza. Form-hoz tartozó unit.
- ***BeallitUnit.pas* (Tervező)**
Beállítás form eljárásait, függvényeit tartalmazza. Form-hoz tartozó unit.
- ***BeallitUnit.pas* (Vezérlő)**
Beállítás form eljárásait, függvényeit tartalmazza. Form-hoz tartozó unit.
- ***BerendDlgUnit.pas* (Vezérlő)**
A berendezéseket megjelenítő form eljárásait, függvényeit tartalmazza. Form-hoz tartozó unit.
- ***BerendezesekLancUnit.pas* (Tervező, Tesztelő, Vezérlő)**
Berendezéseket tartalmazó lánc típusának eljárásai, függvényei.
- ***BerendInspUnit.pas* (Vezérlő)**
A berendezések irányítópaneljének eljárásai, függvényei.
- ***BerendTablaUnit.pas* (Tesztelő, Vezérlő)**
A berendezéseket tartalmazó adatbázisok eljárásai, függvényei.
- ***Beeper.pas* (Vezérlő)**
A rendszerhangszóró vezérlése. Más által írt unit.

- *ComPort* komponens (Tesztelő, Vezérlő)
Soros port vezérlése. Más által írt komponens. Használat előtt telepíteni kell.
- *DataModulUnit.pas* (Tesztelő, Vezérlő)
Adattáblákat tartalmazó unit.
- *DlgUnit.pas* (Vezérlő)
Az berendezések láncait tartalmazó típus.
- *EgyebDlgUnit.pas* (Vezérlő)
Az egyéb berendezések form eljárásait, függvényeit tartalmazza. Form-hoz tartozó unit.
- *EpuletLancUnit.pas* (Tervező, Tesztelő, Vezérlő)
Épület szerkezetét tartalmazó láncok.
- *FileTablaUnit.pas* (Tervező, Tesztelő, Vezérlő)
Adatbázisokkal kapcsolatok eljárások, függvények.
- *ForgatUnit.pas* (Tervező)
Forgatás form eljárásait, függvényeit tartalmazza. Form-hoz tartozó unit.
- *FormFestoUnit.pas* (Tervező, Tesztelő, Vezérlő)
Egy form átfestésének eljárásai.
- *FrmNyomtatUnit.pas* (Vezérlő)
Eseménynapló nyomtatási riportja.
- *FutotestDlgUnit.pas* (Vezérlő)
A fűtőtest berendezések form eljárásait, függvényeit tartalmazza. Form-hoz tartozó unit.
- *Gomb.pas* (Tesztelő, Vezérlő)
Kapcsoló komponens. Részben más által írt komponens. Használat előtt telepíteni kell.
- *GradientLabel* komponens (Tervező, Vezérlő)
Színátmenetes label. Más által írt komponens. Használat előtt telepíteni kell.
- *HangThreadUnit.pas* (Vezérlő)
Figyelmeztető hangjelzést vezérlő szál eljárásai, függvényei.
- *HomeroBeallUnit.pas* (Vezérlő)
Egy hőmérő berendezés beállításának formja.
- *HomeroLancUnit.pas* (Tesztelő, Vezérlő)
Hőmérőket tartalmazó lánc eljárásai, függvényei.
- *HozzaadUnit.pas* (Vezérlő)
Berendezések egymáshoz való rendelése form eljárásait, függvényeit tartalmazza. Form-hoz tartozó unit.
- *InditUnit.pas* (Tesztelő, Vezérlő)
Az InditasDlg form eljárásai. Épület betöltésekor hívódik meg. Form-hoz tartozó unit.
- *InstanceHandler.pas* (Tervező, Tesztelő, Vezérlő)
A program egy példányban való futtatását ellenőrzi. Más által írt komponens.

- *KapcsBerendLancUnit.pas* (Tesztelő, Vezérlő)
Az egymáshoz tartozó berendezéseket tartalmazó lánc eljárásai, függvényei.
- *KeretUnit.pas* (Tervező)
Kijelölő keret eljárásai, függvényei.
- *KijelolNegyzetUnit.pas* (Tervező)
A kijelölő négyzet eljárásai, függvényei.
- *KodolasUnit.pas* (Tesztelő, Vezérlő)
Üzenetek kódolása és dekódolása.
- *KuldThreadUnit.pas* (Tesztelő, Vezérlő)
Kommunikációhoz szükséges eljárások, függvények. Az üzenetek küldéséért felelős szál eljárásai, függvényei.
- *MegnyitasUnit.pas* (Tesztelő, Vezérlő)
Megnyitás form eljárásait, függvényeit tartalmazza. Form-hoz tartozó unit.
- *MeretFormUnit.pas* (Tervező)
Rajzasztal beállítása form eljárásait, függvényeit tartalmazza. Form-hoz tartozó unit.
- *NaploDlgUnit.pas* (Vezérlő)
Naplózó form. Form-hoz tartozó unit.
- *NyugtaDlgUnit.pas* (Vezérlő)
Riasztást nyugtázó form. Form-hoz tartozó unit.
- *ObjektumokDlgUnit.pas* (Tervező)
Objektumok megjelenítése form. Form-hoz tartozó unit.
- *ObjektumUnit.pas* (Tervező, Tesztelő, Vezérlő)
Objektum lánc eljárásai, függvényei.
- *ObjListaUnit.pas* (Tervező)
Objektumok listája form. Form-hoz tartozó unit.
- *ObjNevFormUnit.pas* (Tervező)
Új objektum nevének felvitele form. Form-hoz tartozó unit.
- *PenUnit.pas* (Tervező, Tesztelő, Vezérlő)
A tollal kapcsolatos eljárások.
- *PontLancUnit.pas* (Tesztelő, Vezérlő)
Pontokat tartalmazó lánc eljárásai, függvényei.
- *RajzolasUnit.pas* (Tervező, Tesztelő, Vezérlő)
Rajzolással kapcsolatos eljárások, függvények.
- *ResolutionUnit.pas* (Tervező, Tesztelő, Vezérlő)
Monitor felbontásának beállítása.
- *RiasztoBeallUnit.pas* (Vezérlő)
Egy riasztó beállításának formja. Form-hoz tartozó unit.
- *RiasztoLancUnit.pas* (Tesztelő, Vezérlő)
Riasztókat tartalmazó lánc eljárásai, függvényei.

- *TavolsagUnit.pas* (Tervező)
Két pont távolságát kiszámító eljárások, függvények.
- *TervezoUnit.pas* (Tervező)
A Tervező program főformjának unit-ja. Form-hoz tartozó unit.
- *TeszteloUnit.pas* (Tesztelő)
A Tesztelő program főformjának unit-ja. Form-hoz tartozó unit.
- *TuzjelzoBeallUnit.pas* (Vezérlő)
Egy tűzjelző beállításának formja. Form-hoz tartozó unit.
- *TuzjelzoLancUnit.pas* (Tesztelő, Vezérlő)
Tűzjelzőket tartalmazó lánc eljárásai, függvényei.
- *UjBerendezesUnit.pas* (Tervező)
Új berendezés hozzáadása form. Form-hoz tartozó unit.
- *UjEpEgysegDlgUnit.pas* (Tervező)
Új épületegység felvétele form. Form-hoz tartozó unit.
- *UjUnit.pas* (Tervező)
Indulási form. Form-hoz tartozó unit.
- *UtilUnit.pas* (Tervező, Tesztelő, Vezérlő)
Általános eljárások, függvények.
- *VegberendDlgUnit.pas* (Vezérlő)
Egy végberendezés beállításának formja. Form-hoz tartozó unit.
- *VegBerendLancUnit.pas* (Tesztelő, Vezérlő)
Végberendezéseket tartalmazó lánc eljárásai, függvényei.
- *VezerloUnit.pas* (Vezérlő)
A Vezérlő program főformjának unit-ja. Form-hoz tartozó unit.
- *ZoomUnit.pas* (Tervező, Tesztelő, Vezérlő)
Nagyítás-kicsinyítés függvényei.

Táblázatos formában

Unit neve	Tervező	Tesztelő	Vezérlő
AboutUnit.pas	X		
AboutUnit.pas			X
AjtoUnit.pas	X		
BeallitUnit.pas	X		
BeallitUnit.pas			X
BerendDlgUnit.pas			X
BerendezesekLancUnit.pas	X	X	X
BerendInspUnit.pas			X
BerendTablaUnit.pas		X	X

Beeper.pas			X
ComPort komponens		X	X
DataModulUnit.pas		X	X
DlgUnit.pas			X
EgyebDlgUnit.pas			X
EpuletLancUnit.pas	X	X	X
FileTablaUnit.pas	X	X	X
ForgatUnit.pas	X		
FormFestoUnit.pas	X	X	X
FrmNyomtatUnit.pas			X
FutotestDlgUnit.pas			X
Gomb komponens		X	X
GradientLabel komponens	X		X
HangThreadUnit.pas			X
HomeroBeallUnit.pas			X
HomeroLancUnit.pas		X	X
HozzaadUnit.pas			X
InditUnit.pas		X	X
InstanceHandler.pas	X	X	X
KapcsBerendLancUnit.pas		X	X
KeretUnit.pas	X		
KijelolNegyzetUnit.pas	X		
KodolasUnit.pas		X	X
KuldThreadUnit.pas		X	X
MegnyitasUnit.pas		X	X
MeretFormUnit.pas	X		
NaploDlgUnit.pas			X
NyugtaDlgUnit.pas			X
ObjektumokDlgUnit.pas	X		
ObjektumUnit.pas	X	X	X
ObjListaUnit.pas	X		
ObjNevFormUnit.pas	X		
PenUnit.pas	X	X	X
PontLancUnit.pas		X	X
RajzolasUnit.pas	X	X	X
ResolutionUnit.pas	X	X	X
RiasztoBeallUnit.pas			X
RiasztoLancUnit.pas		X	X
TavolsagUnit.pas	X		
TervezoUnit.pas	X		
TeszteloUnit.pas		X	
TuzjelzoBeallUnit.pas			X

TuzjelzoLancUnit.pas		X	X
UjBerendezesUnit.pas	X		
UjEpEgysegDlgUnit.pas	X		
UjUnit.pas	X		
UtilUnit.pas	X	X	X
VegberendDlgUnit.pas			X
VegBerendLancUnit.pas		X	X
VezerloUnit.pas			X
ZoomUnit.pas	X	X	X

A programcsomagban használt unit-ok

Mindazok a unitok, melyek nem saját fejlesztésük (*Beeper*, *ComPort*, *GradientLabel*, *Gomb* - részben) nyílt forrásúak és szabadon felhasználhatók. Ezen unit-ok forrása a CD mellékleten elérhetők és megtekinthetők.

Adatszerkezetek

Ebben a szakaszban azok a változók, eljárások, függvények, típusok és osztályok kerülnek bemutatásra, melyek minden a három programban előfordulnak. Ezeket le tudjuk olvasni az előbbi listából, ahol az egyes unit-ok fel vannak sorolva, az őket tartalmazó programokkal együtt. Részletesebben csak az *EpuletLancUnit.pas* unit kerül bemutatásra, egyszerűt mert ez a unit tartalmazza egy épület szerkezetének felépítését, mely fontos pillére a dolgozatnak, másrészt a többi láncot leíró unit is hasonló felépítésű.

A részletes leírások az egyes unit-okról megtalálhatók a forrásban.

Globális konstansok, változók

Itt csak a fontosabb konstansok, változók és típusok szerepelnek.

Ezek a konstansok és változók a programok egész működése során elérhetők.

(*BerendezesekLancUnit.pas*):

- *TBerendezesTip* = (*btHomero*, *btRiaszto*, *btTuzjelzo*, *btFutotest*, *btEgyeb*)
A berendezések felsorolástípusa. A berendezéseket ezekkel az értékekkel tudjuk azonosítani.

(*EpuletLancUnit.pas*):

- *Ajto* : *PAjtoAlakzat* : ajtó alakzatra mutató pointer
- *Ablak* : *PAbnakAlakzat* : ablak alakzatra mutató pointer
- *Vonal* : *PVonalAlakzat* : vonal alakzatra mutató pointer

- *Negyzet* : *PNegyzetAlakzat* : téglalap alakzatra mutató pointer
 - *Ellipszis* : *PEllipszisAlakzat* : ellipszis alakzatra mutató pointer
 - *AlakzatLanc* : *PAlekzatLanc* : alakzatok láncára mutató pointer
 - *AlakzatElem* : *PAlekzatElem* : egy alakzatláncbeli elemre mutató pointer
 - *EpuletEgyseg* : *PEpuletEgyseg*: egy épületegységre mutató pointer
(*FormFestoUnit.pas*):
- *AblakSzin* : *T AblakSzin* : Színátmenet tulajdonsága

(*RajzolasUnit.pas*):

- *RAJZMERET* = 1200;
Rajzasztal mérete pixelben.
- *ELLPONTSZAM* = 200;
Ellipszis pontjainak a száma.

(*ResolutionUnit.pas*):

- *Bitek* : *Array[0..2] Of Integer* = (8,16,32);
Monitor színmélysége.
- *ResXek* : *Array[0..2] Of Integer* = (640,800,1024);
Monitor horiz. felbontása.
- *ResYok* : *Array[0..2] Of Integer* = (480,600,768);
Monitor vert. Felbontása.

(*ZoomUnit.pas*):

- *MAXZOOM* = 4;
Maximális nagyítás-kicsinyítés (négyszeres).

Globális eljárások, függvények

Ezek a konstansok és változók a programok egész működése során meghívhatók:

(*FileTablaUnit.pas*):

- *Function Keres(Szoveg: String) : Boolean;*
Felteszi a Szoveg kérdést. Ha 'Igen' lett lenyomva -> True, különben -> False.
- *Function AktDirNev(Const Dir: String) : String;*
Visszaadja a paraméterben átadott útvonalból az aktuális könyvtárat.

- *Function KonyvtarMegnyit(Const SubDir: String) : String;*
Az épületet tartalmazó könyvtár megnyitása. Ha siker -> visszaadja az elérési útvonalát, különben -> ''.
- *Procedure AliasModosit(Const Dir: String);*
Az alias nevet átállítjuk a paraméterben átadott könyvtárra.
- *Procedure RajzIniBetolt(Const Path,EpuletNev: String; Var AlapMeret,EgysMeret: Integer);*
A tervező program beállításainak betöltése.
- *Procedure TervezoIniMent(Const FNev: String; Const SzinAtmenet: Boolean; Const AblakMeret: Integer);*
A tervező program beállításainak mentése.
- *Procedure VezerloIniBetolt(FNev: String; Form: TForm; SzinBox,InditBox: TCheckBox; InputEdit: TEdit; CPort: TComPort; Var Hang: Integer; Var Epulet: String);*
A vezérlő program beállításainak betöltése.
- *Procedure VezerloIniMent(FNev,Epulet: String; SzinAtmenet,AutIndit: Boolean; Com: String; Stop,Baud,Data,Flow,Hang: Integer);*
A vezérlő program beállításainak mentése.
- *Procedure TeszteloIniBetolt(FNev: String; Form: TForm; SzinBox,FormaBox, InditBox: TCheckBox; InputEdit: TEdit; CPort: TComPort; Var Epulet: String; Var Beallitas: TTeszteloBeallit);*
A tesztelő program beállításainak betöltése.
- *Procedure TeszteloIniMent(FNev,Epulet: String; SzinAtmenet,Forma, AutIndit: Boolean; Com: String; Stop,Baud,Data,Flow: Integer; Beallitas: TTeszteloBeallit);*
A tesztelő program beállításainak mentése.
- *Procedure EpuletTablaMentes(Const EpLista: TEpuletLista; EpuletNev: String; EpuletTabla: TTable; Const AlapMeret,EgysMeret: Integer);*
EpuletTabla mentése.
- *Procedure EpuletLancMentes(Const EpLista: TEpuletLista; Const EpuletTabla: TTable);*
Épület lánc mentése a táblába.
- *Procedure EpuletTablaKeszit(Tabla: TTable; Nev: String; Tipus: TTableType);*
EpuletTabla létrehozása.
- *Procedure ObjektumTablaMentes(Const EpLista: TEpuletLista; EpuletNev: String; Const ObjLista: TObjektumLista; ObjektumTabla: TTable);*
ObjektumTabla mentése.
- *Procedure ObjektumLancMentes(Const EpLista: TEpuletLista; Const ObjektumTabla: TTable; Const ObjLista: TObjektumLista);*
Objektum lánc mentése a táblába.

- *Procedure ObjektumTablaKeszit(Tabla: TTable; Nev: String; Tipus: TTableType);*
ObjektumTabla létrehozása.
- *Procedure BerendezesTablaMentes(Const EpLista: TEpuletLista; Const ObjLista: TObjektumLista; EpuletNev: String; Const BerendLista: TBerendezesLista; BerendezesTabla: TTable);*
BerendezesTabla mentése.
- *Procedure BerendezesLancMentes(Const EpLista: TEpuletLista; Const ObjLista: TObjektumLista; Const BerendezesTabla: TTable; Const BerendLista: TBerendezesLista);*
Berendezés lánc mentése a táblába.
- *Procedure BerendezesTablaKeszit(Tabla: TTable; Nev: String; Tipus: TTableType);*
BerendezesTabla létrehozása.

(*FormFestoUnit.pas*):

- *Procedure FormFesto(Form: TForm);*
Ablakfestő rutin.
- *Procedure Label_Bevel_Allit(Form: TForm; TombLabel: Array Of TLabel; TombBevel: Array Of TBevel; Ertek: Boolean; Szin: Integer; Stilus: TBevelStyle);*
Beállítja a betűk és a keretek színét.
- *Procedure Label_Allit(Form: TForm; TombLabel: Array Of TLabel; Ertek: Boolean; Szin: Integer);*
Beállítja a betűk színét.
- *Procedure Bevel_Allit(Form: TForm; TombBevel: Array Of TBevel; Ertek: Boolean; Stilus: TBevelStyle);*
Beállítja a keretek színét.

(*PenUnit.pas*):

- *Procedure GetPenStyle(Const Form : TForm; Const Image : TImage);*
Toll tulajdonságainak lekérdezése.
- *Procedure SetPenStyle(Const Form : TForm; Const Image : TImage; Const Stilus : TPenStyle; Const Szin : TColor; Const Meret : Integer);*
Toll tulajdonságainak visszaállítása a paraméterek szerint.
- *Procedure AlapPenBeallit(Const Form : TForm; Const Image : TImage);*
Toll alaptulajdonságainak beállítása.

(RajzolasUnit.pas):

- *Function SzogToRad(Const Szog: Integer) : Extended;*
Szög átalakítása radiánba.
- *Procedure DrawShape(Const Form: TForm; Const Image: TImage; Const DT: TDrawingTool; Const TopLeft, BottomRight: TPoint; Const AMode: TPenMode);*
Rajzolási rutin. Ez rajzol a BitMap-re.
- *Procedure NegyzetRajzol(Const Form: TForm; Const Image: TImage; Const BF,JF,JA,BA: TPoint);*
Négyszet rajzolása.
- *Procedure EllipszisRajzol(Const Form: TForm; Const Image: TImage; Const A,B: Double; Const Kozep: TPoint; Const Szog: Integer; Const Rajzol: Boolean; Var Min,Max: TPoint);*
Ellipszis rajzolása.

(UtilUnit.pas):

- *Function BenneVan(Const BF,JA,Mi: TPoint) : Boolean;*
Egy pont benne van-e a két pont által határolt területben. Ha benne van -> True, különben -> False.
- *Function PontForgat(Const Pont: TPoint; Const Szog: Integer) : TPoint;*
Egy pont elforgatása egy adott szöggel és visszaadja az új pontot.
- *Function KeresComboBox(Const Form: TForm; Const CB: TComboBox; Const S: String) : Boolean;*
ComboBoxban keres egy karaktersorozatot. Ha megvan -> True, különben -> False.
- *Function ObjektumLancBetolt(Const EpLista: TEpuletLista; Tabla: TTable) : TObjektumLista;*
Objektumok betöltése. Visszaadja az objektumok láncát.
- *Function BerendezesLancBetolt(Tabla: TTable) : TBerendezesLista;*
Berendezések betöltése. Visszaadja az berendezések láncát.
- *Procedure AblakRajzol(Const Form: TForm; Const Image: TImage; Const AblakAlakzat: TAbnakAlakzat; Const Szin: TColor; Const Zoom: Extended);*
Egy ablak rajzolása.
- *Procedure AjtoIvSzamol(Const Kezdo, Vegzo: TPoint; Const JB: Byte; Var Iv1,Iv2: TPoint);*
Az ajtó ívének kiszámolja két pontját és vissza is adja.
- *Procedure AjtoRajzol(Const Form: TForm; Const Image: TImage; Const AjtoAlakzat: TAjtoAlakzat; Const Zoom: Extended);*
Egy ajtó kirajzolása.

- *Procedure RajzasztalKeszit(Const Form: TForm; Const Image: TImage; Const Meret: Integer);*
Rajzasztal elkészítése.
- *Procedure AlakzatLancAtRajzol(Const Form: TForm; Const Image: TImage; Const Zoom: Extended);*
Épületegység újra kirajzolása.
- *Procedure AlakzatLancAtRajzol(Const Form: TForm; Const Image: TImage; Const Szin: TColor; Const Zoom: Extended);*
Épületegység újra kirajzolása.
- *Procedure AlakzatLancAtRajzol(Const Form: TForm; Const Image: TImage; Const Szin: TColor; Const Mit: Pointer; Const Zoom: Extended);*
Egy adott alakzat újraraajzolása a megadott színnel.
- *Procedure AlakzatLetrehoz(Const Form: TForm; Const Image: TImage; Const EpLista: TEpuletLista; Const Szog: Integer; Const Tip: TDrawingTool; Const PM: Integer; Const PS: TPenStyle; Const PC: Integer; Const K1,K2,K3,K4: TPoint; Const A,B: Double; Const Zoom: Extended);*
Egy alakzat létrehozása.
- *Procedure Asztalfrissit(Const Form: TForm; Const Image: TImage; Const Zoom: Extended);*
Rajzasztal újraraajzolása .

(ZoomUnit.pas):

- *Function ErtekZoom(Const Ertek: Integer;Const Zoom: Extended) : Integer;*
A kapott érték konvertálása a Zoom mértéke szerint.
- *Function KoordZoom(Const Pont: TPoint;Const Zoom: Extended) : TPoint;*
A kapott koordináta konvertálása a Zoom mértéke szerint.

Típus- és osztályleírások

(BerendezesekLancUnit.pas):

- *TBerendArray = Array Of Array[1..5] Of String*
A berendezéseket tartalmazó dinamikus tömb.
- *TMutatoArray = Array Of PLancElem*
A LancElem-ek pointereit tartalmazó dinamikus tömb
- *TLancElem = Object*
BerendezLista láncéléme.
- *TBerendezLanc = Object*
BerendezLista lánc fejeleme.

- $TBerendezesLista = Class(TPersistent)$
A láncot tartalmazó osztály.

(*EpuletLancUnit.pas*):

- $TDrawingTool = (dtNyil, dtVonal, dtNegyzet, dtEllipszis, dtIv, dtAjto, dtAblak, dtFal, dtKijelol, dtMeretez, dtMozgatas, dtPoligon, dtObjektum)$
Rajzolás típusát, aktuális művelet típusát adja meg.
- Egy alakzat alaptulajdonságait leíró objektum:

TAlakzatTip = Object ()

Szog : Integer;
Tipus : TDrawingTool;
PenStyle : TPenStyle;
PenColor : TColor;
PenMeret : Integer;

End;

Leírás:

- *Szog* : elforgatás szöge
- *Tipus* : alakzat típusa
- *PenStyle* : toll stílusa
- *PenColor* : toll színe
- *PenMeret* : toll vastagsága

- Vonal alakzat tulajdonságai:

TVonalAlakzat = Object(AlakzatTip)

Koord : Array[1..2] Of TPoint;

End;

Leírás:

- *Koord*: vonal alakzat két végpontjának koordinátái.

- Téglalap alakzat tulajdonságai:

TNegyzetAlakzat = Object(AlakzatTip)

Koord : Array[1..4] Of TPoint;

End;

Leírás:

- *Koord*: téglalap alakzat négy sarkának koordinátái.
- Ellipszis alakzat tulajdonságai:

TEllipszisAlakzat = Object(TVonalAlakzat)

A : Double;
 B : Double;
 Kozep : TPoint;
End;

Leírás:

- *A* : magasság sugár
- *B* : szélesség sugár
- *Kozep* : középpont
- *TAjtoAlakzat = TNegyzetAlakzat;*
Ajtó alakzat.
- *TAblakAlakzat = TNegyzetAlakzat;*
Ablak alakzat.
- AlakzatLanc egy eleme:

TAlakzatElem = Object

Tipus : TDrawingTool;
 AlakMut : Pointer;
 Kovetkezo : PAlakzatElem;
 Constructor Create;
End;

Leírás:

- *Tipus* : alakzat típusa
- *AlakMut* : alakzatra mutató pointer
- *Kovetkezo* : következő alakzatra mutató pointer
- *Constructor Create* : alakzat létrehozása
- AlakzatLanc fejeleme:

TAlakzatLanc = Object

Elszo : PAlakzatElem;

Aktualis : PAlakzatElem;
Constructor Create;
Destructor Destroy;
Procedure LancVegere;
Procedure LancraFuz(EpLista: TEpuletLista; Mutato: Pointer; DT:
 TDrawingTool);
End;

Leírás:

- *Első* : első alakzatra mutató pointer
 - *Aktualis* : aktuális alakzatra mutató pointer
 - *Constructor Create* : lánc létrehozása
 - *Destructor Destroy* : lánc megszüntetése
 - *Procedure LancVegere* : elmegy a lánc utolsó elemére
 - *Procedure LancraFuz(EpLista: TEpuletLista; Mutato: Pointer; DT:*
 TDrawingTool) : egy elemet felfűz a láncre
- EpuletLanc egy eleme:

TEpuletEgyseg = Object
 EgysegNev : String;
 AlakzatLancMutato : PAlakzatLanc;
 Elozo : PEpuletEgyseg;
 Kovetkezo : PEpuletEgyseg;
 Constructor Create;
End;

Leírás:

- *EgysegNev* : melyik épületegységhez tartozik
 - *AlakzatLancMutato* : AlakzatLanc-ra mutató pointer
 - *Elozo* : előző épületegység-re mutató pointer
 - *Kovetkezo* : következő épületegység-re mutató pointer
 - *Constructor Create* : új épületegység objektum létrehozása
- Egy épület szerkezeti tulajdonságai (fejelem)

TEpuletLanc = Object
 Nev : String;
 EpuletMeret : Integer;
 EgysegMeret : Extended;
 Első : PEpuletEgyseg;
 Aktualis : PEpuletEgyseg;

*Procedure Create(EL: TEpuletLista);
End;*

Leírás:

- *Nev* : épület neve
- *EpuletMeret* : mérete pixelben. Lásd *RAJZMERET* konstans.
- *EgysegMeret* : méretarány. Egy pixel hány milliméternek felel meg.
- *Elso* : első épületegység-re mutató pointer
- *Aktualis* : aktuális épületegység-re mutató pointer
- *Procedure Create(EL: TEpuletLista)* : egy EpuletLanc létrehozása.
- Épület szerkezeti tulajdonságait tartalmazó osztály:

TEpuletLista = Class(TPersistent)

Private

*FValtozas : Boolean;
EpuletLanc : TEpuletLanc;
Procedure SetNev(Ertek: String);
Procedure SetValt(Ertek: Boolean);
Procedure SetEpuletMeret(Ertek: Integer);
Procedure SetEgysegMeret(Ertek: Extended);*

Public

*Property Elso : PEpuletEgyseg Read EpuletLanc.Elso Write
EpuletLanc.Elso;
Property Aktualis : PEpuletEgyseg Read EpuletLanc.Aktualis Write
EpuletLanc.Aktualis;
Property Valtozas : Boolean Read FValtozas Write SetValt;
Property Nev : String Read EpuletLanc.Nev Write SetNev;
Property EpuletMeret : Integer Read EpuletLanc.EpuletMeret Write
SetEpuletMeret;
Property EgysegMeret : Extended Read EpuletLanc.EgysegMeret Write
SetEgysegMeret;
Function EpuletUres : Boolean;
Function EgysegUres(Const Nev: String) : Boolean;
Function EgysegKeresNev(Const Nev: String) : Boolean; Overload;
Function EgysegKeresNev(Const Nev: String; Var AlakLanc:
PALakzatLanc; Var EpEgyseg: PEpuletEgyseg) : Boolean; Overload;
Constructor Create;
Destructor Destroy; Override;
End;*

Leírás:

- *FValtozas* : volt-e változás
- *EpuletLanc* : EpuletLanc-ra mutató pointer
- *Procedure SetNev(Ertek: String)* : épületnév beállítása
- *Procedure SetValt(Ertek: Boolean)* : változott-e az EpuletLista
- *Procedure SetEpuletMeret(Ertek: Integer)* : épületméret beállítása
- *Procedure SetEgysegMeret(Ertek: Extended)* : méretarány beállítása
- *Property Elso* : hívatkozás az *EpuletLanc.Elso* elemre
- *Property Aktualis* : hívatkozás az *EpuletLanc. Aktualis* elemre
- *Property Valtozas* : hívatkozás az *FValtozas* mezőre
- *Property Nev* : hívatkozás az *EpuletLanc. Nev* elemre
- *Property EpuletMeret* : hívatkozás az *EpuletLanc. EpuletMeret* elemre
- *Property EgysegMeret* : hívatkozás az *EpuletLanc. EgysegMeret* elemre
- *Function EpuletUres* : üres-e az EpuletLista
ha üres -> True, különben -> False.
- *Function EgysegUres(Const Nev: String)* : üres-e a paraméterben átadott épületegység. Ha nem üres -> True, különen False.
- *Function EgysegKeresNev(Const Nev: String)* : a láncban megkeressük a paraméterben szereplő épületegység nevét. Ha megvan -> True és rá is állunk, különben -> False.
- *Function EgysegKeresNev(Const Nev: String; Var AlakLanc: PAalkatLanc; Var EpEgyseg: PEpuletEgyseg)* : a láncban megkeressük a paraméterben szereplő épületegység nevét. Ha megvan -> True és visszaadjuk az AlakzatLanc,EpuletEgyseg címét, különben -> False.
- *Constructor Create* : TEpuletLista osztály létrehozása
- *Destructor Destroy* : TEpuletLista osztály megszüntetése

(*FormFestoUnit.pas*):

- *TAblakSzín = Object*
Színátmény tulajdonságait leíró objektum

(*ObjektumUnit.pas*):

- *TPontKoordTomb = Array Of TPoint;*
Pontokat tartalmazó dinamikus tömb.
- *TObjektumArray = Array Of Array[1..3] Of String;*
Objektumok tulajdonságait tartalmazó dinamikus tömb.
- *TObjektum = Class(TPersistent)*
Objektum osztály.

- *TObjektumTip* = *Object*
Egy objektum tulajdonsága.
- *TObjektumLanc* = *Object*
Egy objelktumláncbeli elem.
- *TObjektumLista* = *Class(TPersistent)*
Objektum láncot tartalmazó osztály (fejelem is)

(*PenUnit.pas*):

- *TRegiPenStyle* = *Object*
A toll tulajdonságainak tárolása.

(*RajzolasUnit.pas*):

- *TEpuletTul* = *Record*
Egy épületrajz tulajdonságai.

(*ResolutionUnit.pas*):

- *TDevTul* = *Record*
Felbontás tulajdonságai.
- *TResolution* = *Class(TPersistent)*
Monitor felbontását tartalmazó osztály.

(*UtilUnit.pas*):

- *TEgerPos* = *Record*
Egér pozíciója.

(*ZoomUnit.pas*):

- *TZoom* = *Object*
A nagyítást-kicsinyítést nyilvántartó objektum.

ADATÁLLOMÁNYOK FIZIKAI RENDSZERTERVE

1. ALAPRAJZ.DB

- *Adatbázis fizikai neve:* ALAPRAJZ.DB
- *Adatbázis alias neve:* EPULET
- *Indexfájl neve:* ALAPRAJZ.VAL
- *Feladata:* az épület alaprajzának a tárolása.

Mező neve	Típusa	Hossza
SORINDEX	Short (16-bit integer)	
ALAKSORREND	Short (16-bit integer)	
EGYSEGNEV	String	25
TIPUS	Short (16-bit integer)	
SZOG	Short (16-bit integer)	
PENSTYLE	Short (16-bit integer)	
PENCOLOR	Integer (32-bit integer)	
PENMERET	Short (16-bit integer)	
K1X	Short (16-bit integer)	
K1Y	Short (16-bit integer)	
K2X	Short (16-bit integer)	
K2Y	Short (16-bit integer)	
K3X	Short (16-bit integer)	
K3Y	Short (16-bit integer)	
K4X	Short (16-bit integer)	
K4Y	Short (16-bit integer)	
A	Float	
B	Float	

Az egyes mezők tartalmi jelentése:

- **SORINDEX:** A tábla elsődleges indexének meghatározására szükséges. Csak paradox típusú tábla esetén töltődik fel. Sorszámozza a mezőket.
- **ALAKSORREND:** Értéke 0 vagy 1 lehet és függ az alakzat típusától. A vonal, téglalap, ellipszis 0-ás értékét vehetnek fel, az ajtó és ablak alakzatok pedig 1-es értéket vehetnek fel. Ez a sorrendiségek meghatározása a kirajzolás sorrendjében segít.

- **EGYSEGNEV:** Annak az épületegységnek a nevét tartalmazza, amelyben az adott alakzat található. Formai megkötés nincs, értéke programból nem módosítható.
- **TIPUS:** Az alakzat típusát tartalmazza, a típusok felsorolása között elfoglalt helye szerint.
- **SZOG:** A koordináta-tengellyel bezárt szöget tartalmazza. Csak az ellipszisnél van jelentőssége.
- **PENSTYLE:** A rajztoll stílusát tartalmazza. Ez lehet folytonos vonal, amelynek értéke 0, vagy lehet szaggatott vonal, melynek értéke 1.
- **PENCOLOR:** A rajztoll színét tartalmazza.
- **PENMERET:** A rajztoll vonalvastagságát tartalmazza.
- **K1X:** Az 1. koordináta X összetevője
- **K1Y:** Az 1. koordináta Y összetevője
- **K2X:** Az 2. koordináta X összetevője
- **K2Y:** Az 2. koordináta Y összetevője
- **K3X:** Az 3. koordináta X összetevője
- **K3Y:** Az 3. koordináta Y összetevője
- **K4X:** Az 4. koordináta X összetevője
- **K4Y:** Az 4. koordináta Y összetevője
- **A:** Ellipszis vertikális sugara
- **B:** Ellipszis horizontális sugara

2. OBJEKTUM.DB

- *Adatbázis fizikai neve:* OBJEKTUM.DB
- *Adatbázis alias neve:* EPULET
- *Indexfájl neve:* OBJEKTUM.VAL
- *Feladata:* az épület objektumainak a tárolása.

Mező neve	Típusa	Hossza
EGYSEGNEV	String	25
OBJNEV	String	25
ID	Short (16-bit integer)	
MERET	Short (16-bit integer)	
X	Short (16-bit integer)	
Y	Short (16-bit integer)	

Az egyes mezők tartalmi jelentése:

- **EGYSEGNEV:** Annak az épületegységnek a nevét tartalmazza, amelyben az adott objektum található. Formai megkötés nincs, értéke programból nem módosítható.
- **OBJNEV:** Az objektum nevét tartalmazza.
- **ID:** Megadja, hogy az adott alkotóelem, az objektumon belül hányadik helyet foglal el. Ennek az objektum kirajzolásakor van szerepe, amikor fontos a rajzolás sorrendje.
- **MERET:** Azt adja meg, hogy az objektum mennyi alkotóelemből épül fel.
- **X:** A koordináta X összetevője
- **Y:** A koordináta Y összetevője

3. BERENDEZES.DB

- *Adatbázis fizikai neve:* BERENDEZES.DB
- *Adatbázis alias neve:* EPULET
- *Indexfájl neve:* BERENDEZES.VAL
- *Feladata:* az épület berendezéseinek a tárolása.

Mező neve	Típusa	Hossza
TIPUS	Short (16-bit integer)	
EGYSEGNEV	String	25
OBJNEV	String	25
BERENDNEV	String	25
ID	Short (16-bit integer)	

Az egyes mezők tartalmi jelentése:

- **TIPUS:** A berendezés típusát adja meg. Ez lehet:
 - 0 – Hőmérő
 - 1 – Riasztó
 - 2 – Tűzjelző
 - 3 – Fűtőtest
 - 4 – Más berendezés
- **EGYSEGNEV:** Annak az épületegységnek a nevét tartalmazza, amelyben az adott berendezés található. Formai megkötés nincs, értéke programból nem módosítható.
- **OBJNEV:** Az objektum nevét tartalmazza, amelyben a berendezés található.
- **BERENDNEV:** A berendezés neve.
- **ID:** Az adott berendezés azonosítószáma. Ez az azonosító az egész épületre vonatkozóan egyedi.

4. HOMERO.DB

- *Adatbázis fizikai neve:* HOMERO.DB
- *Adatbázis alias neve:* EPULET
- *Indexfájl neve:* HOMERO.VAL
- *Feladata:* az épület hőmérő-berendezéseinek a tárolása.

Mező neve	Típusa	Hossza
ID	Short (16-bit integer)	
MIN	Short (16-bit integer)	
MAX	Short (16-bit integer)	
BERTEK	Short (16-bit integer)	
NAPLOZ	Boolean	

Az egyes mezők tartalmi jelentése:

- **ID:** Az adott hőmérő azonosítószáma. Ez az azonosító az egész épületre vonatkozóan egyedi.
- **MIN:** A hőmérő mérési tartományának minimuma.
- **MAX:** A hőmérő mérési tartományának maximuma.
- **BERTEK:** A hőmérőn beállított érték.
- **NAPLOZ:** Kell-e naplózni, ha változik a berendezés valamelyik tulajdonsága vagy a hőmérőn mért értéke.
True → kell naplózni
False → nem kell naplózni

5. FUTOTEST.DB

- *Adatbázis fizikai neve:* FUTOTEST.DB
- *Adatbázis alias neve:* EPULET
- *Indexfájl neve:* FUTOTEST.VAL
- *Feladata:* az épület fűtőtest-berendezéseinek a tárolása.

Mező neve	Típusa	Hossza
ID	Short (16-bit integer)	
NAPLOZ	Boolean	
AUTO	Boolean	
DIREKT	Boolean	

Az egyes mezők tartalmi jelentése:

- **ID:** Az adott fűtőtest azonosítószáma. Ez az azonosító az egész épületre vonatkozóan egyedi.
- **NAPLOZ:** Kell-e naplózni, ha változik a berendezés valamelyik tulajdonsága.
 $True \longrightarrow$ kell naplózni
 $False \longrightarrow$ nem kell naplózni
- **AUTO:** A berendezés automata üzemmódban van-e.
 $True \longrightarrow$ a berendezés automata üzemmódban van
 $False \longrightarrow$ a berendezés kézi üzemmódban van
- **DIREKT:** A berendezés kézi üzemmódjának állapota.
 $True \longrightarrow$ a berendezés be van kapcsolva.
 $False \longrightarrow$ a berendezés ki van kapcsolva.

6. RIASZTO.DB

- *Adatbázis fizikai neve:* RIASZTO.DB
- *Adatbázis alias neve:* EPULET
- *Indexfájl neve:* RIASZTO.VAL
- *Feladata:* az épület riasztó-berendezéseinek a tárolása.

Mező neve	Típusa	Hossza
ID	Short (16-bit integer)	
NAPLOZ	Boolean	

Az egyes mezők tartalmi jelentése:

- **ID:** Az adott riasztó azonosítószáma. Ez az azonosító az egész épületre vonatkozóan egyedi.
- **NAPLOZ:** Kell-e naplózni, ha változik a berendezés valamelyik tulajdonsága.
 $True \longrightarrow$ kell naplózni
 $False \longrightarrow$ nem kell naplózni

7. TUZJELZO.DB

- *Adatbázis fizikai neve:* TUZJELZO.DB
- *Adatbázis alias neve:* EPULET
- *Indexfájl neve:* TUZJELZO.VAL
- *Feladata:* az épület tűzjelző-berendezéseinek a tárolása.

Mező neve	Típusa	Hossza
ID	Short (16-bit integer)	
NAPLOZ	Boolean	

Az egyes mezők tartalmi jelentése:

- **ID:** Az adott tűzjelző azonosítószáma. Ez az azonosító az egész épületre vonatkozóan egyedi.
- **NAPLOZ:** Kell-e naplózni, ha változik a berendezés valamelyik tulajdonsága.
True → kell naplózni
False → nem kell naplózni

8. EGYEB.DB

- *Adatbázis fizikai neve:* EGYEB.DB
- *Adatbázis alias neve:* EPULET
- *Indexfájl neve:* EGYEB.VAL
- *Feladata:* az épület más berendezéseinek a tárolása.

Mező neve	Típusa	Hossza
ID	Short (16-bit integer)	
NAPLOZ	Boolean	
AUTO	Boolean	
DIREKT	Boolean	

Az egyes mezők tartalmi jelentése:

- **ID:** Az adott berendezés azonosítószáma. Ez az azonosító az egész épületre vonatkozóan egyedi.
- **NAPLOZ:** Kell-e naplózni, ha változik a berendezés valamelyik tulajdonsága.
True → kell naplózni
False → nem kell naplózni
- **AUTO:** A berendezés automata üzemmódban van-e.
True → a berendezés automata üzemmódban van
False → a berendezés kézi üzemmódban van
- **DIREKT:** A berendezés kézi üzemmódjának állapota.
True → a berendezés be van kapcsolva.
False → a berendezés ki van kapcsolva.

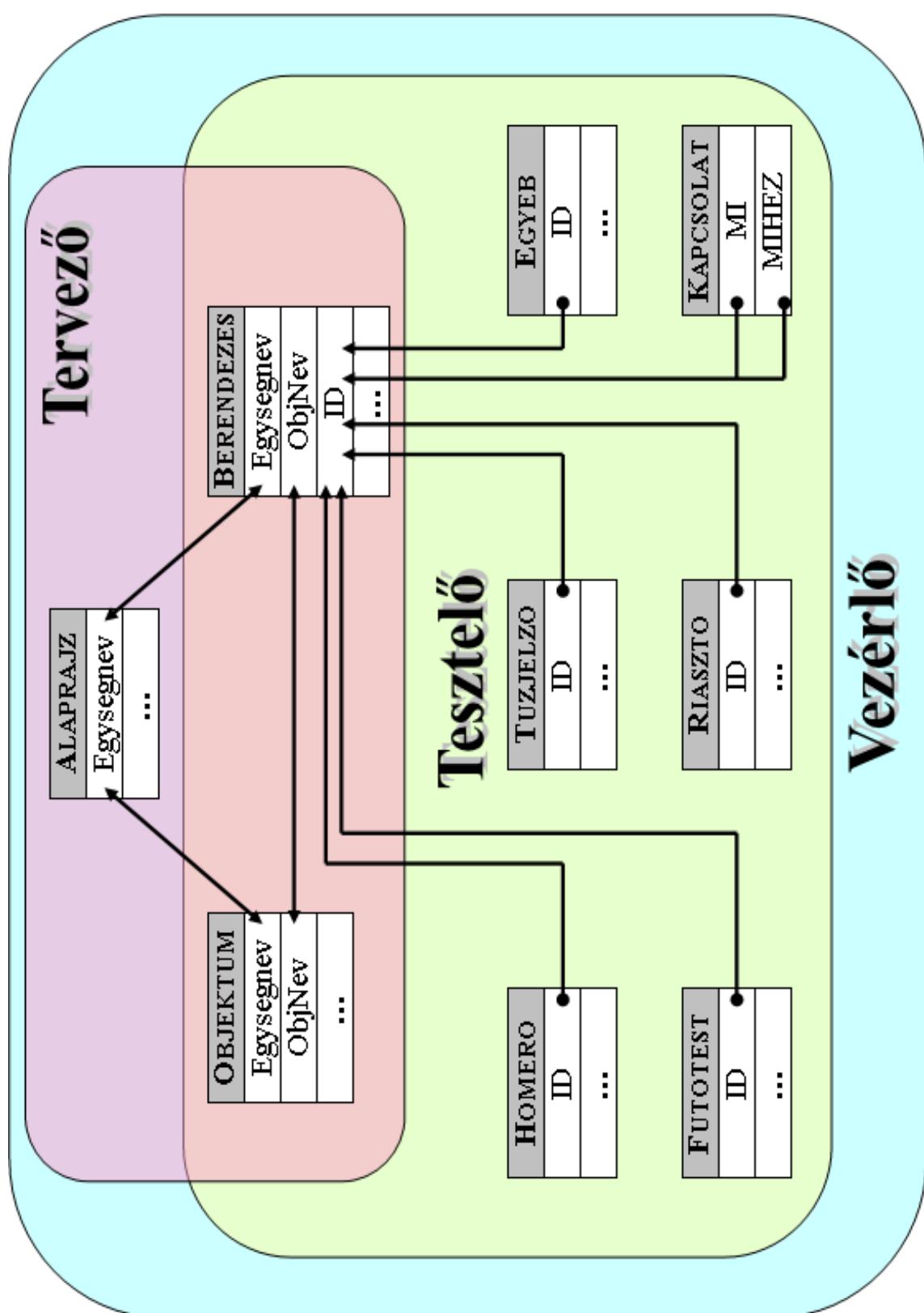
9. KAPCSOLAT.DB

- *Adatbázis fizikai neve:* KAPCSOLAT.DB
- *Adatbázis alias neve:* EPULET
- *Indexfájl neve:* KAPCSOLAT.VAL
- *Feladata:* az egyes berendezéseknek az egymáshoz való kapcsolatát írja le.

Mező neve	Típusa	Hossza
MI	Short (16-bit integer)	
MIHEZ	Short (16-bit integer)	

Az egyes mezők tartalmi jelentése:

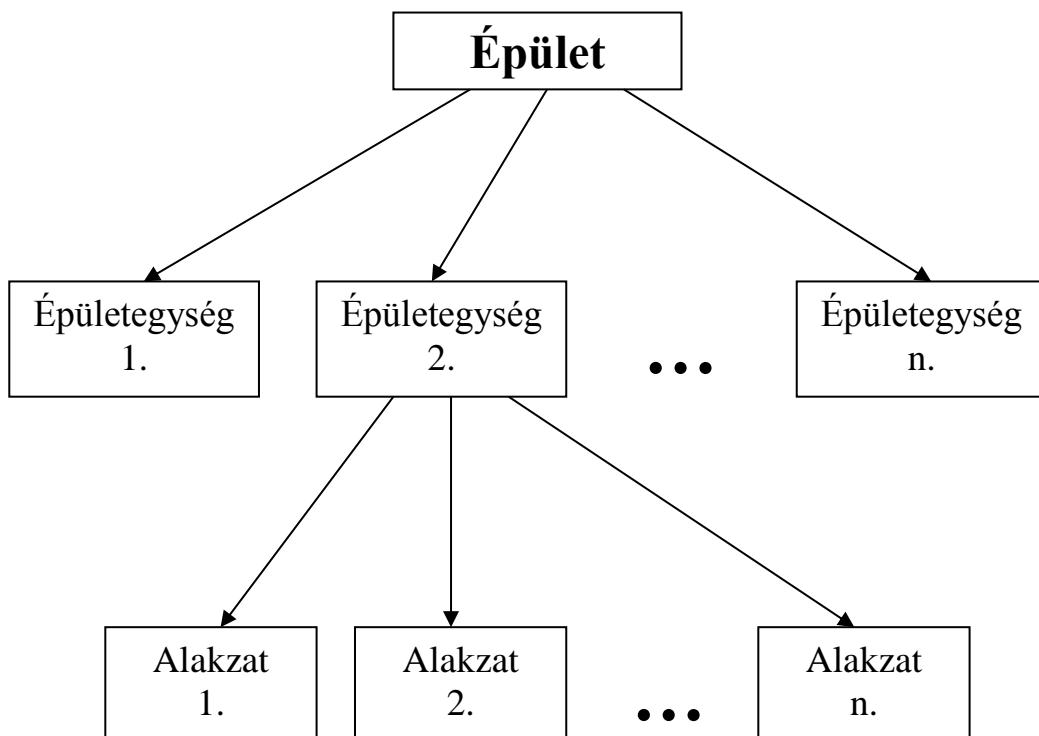
- **MI:** Annak a berendezésnek az azonosítószáma, amely a MIHEZ berendezéshez kapcsolódik.
- **MIHEZ:** Annak a berendezésnek az azonosítószáma, amelyhez a MI berendezés kapcsolódik.



EGY ÉPÜLET FELÉPÍTÉSÉNEK ADATSZERKEZETI MEGVALÓSÍTÁSA

Épület szerkezeti felépítése

Az egyes épületegységek, azon belül az egyes alakzatok kezelése szempontjából fontos elvárás, hogy külön egységekkel kezelhessük őket. Ezért olyan adatszerkezeti megoldást kellett keresni, ahol az alakzatok egyértelműen azonosíthatóknak kell lenniük, és a rajzolás alkalmával a műveletek (mozgatás, méretezés, forgatás, törlés) alkalmazhatóak legyenek a kiválasztott alakzat(ok)-on. A következő ábrán látható egy épület felépítésének vázlatra:

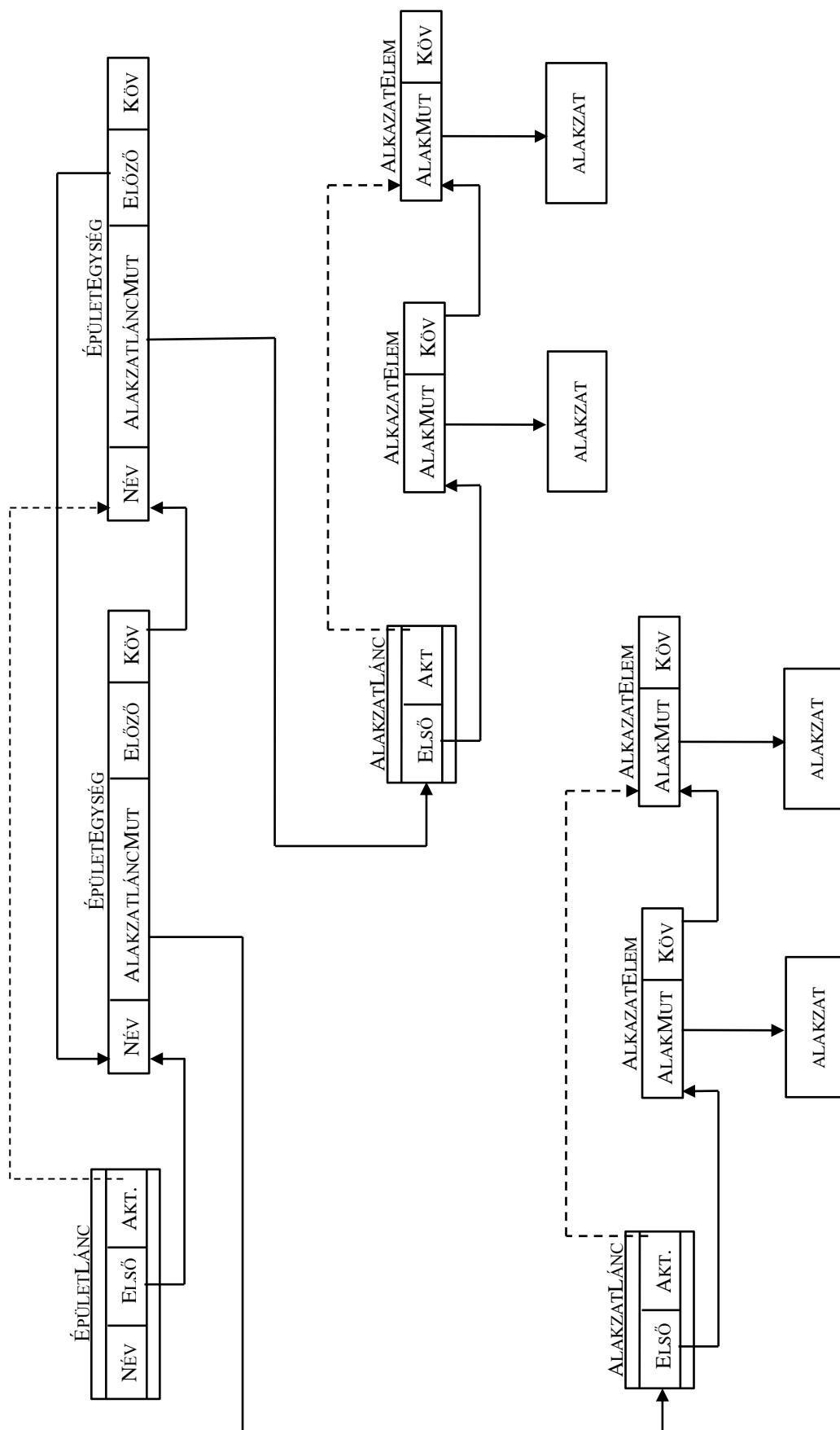


Az ábrán látható szerkezet, láncok segítségével tárolódik a memóriában, amely gyors elérést biztosít az egyes alakzatokhoz. Így megvalósítható az a cél is, hogy az egyes épületegységeket és alakzatokat külön-külön kezelhessük.

Az egyes szerkezeti szintek külön-külön, a memóriában lévő láncokban tárolódnak, melyek bejárásával érjük el az egyes alakzatokat.

Hivatkozás egy alakzatra: *Épület.Épületegység.Alakzat*

A következő ábra tartalmazza egy épület memóriabeli felépítését és az egyes láncok egymáshoz való kapcsolatát.



Műveletek az épületláncon:

Itt csak a fontosabb műveletek és algoritmusok kerülnek bemutatásra terjedelmi és hasonlósági okok miatt.

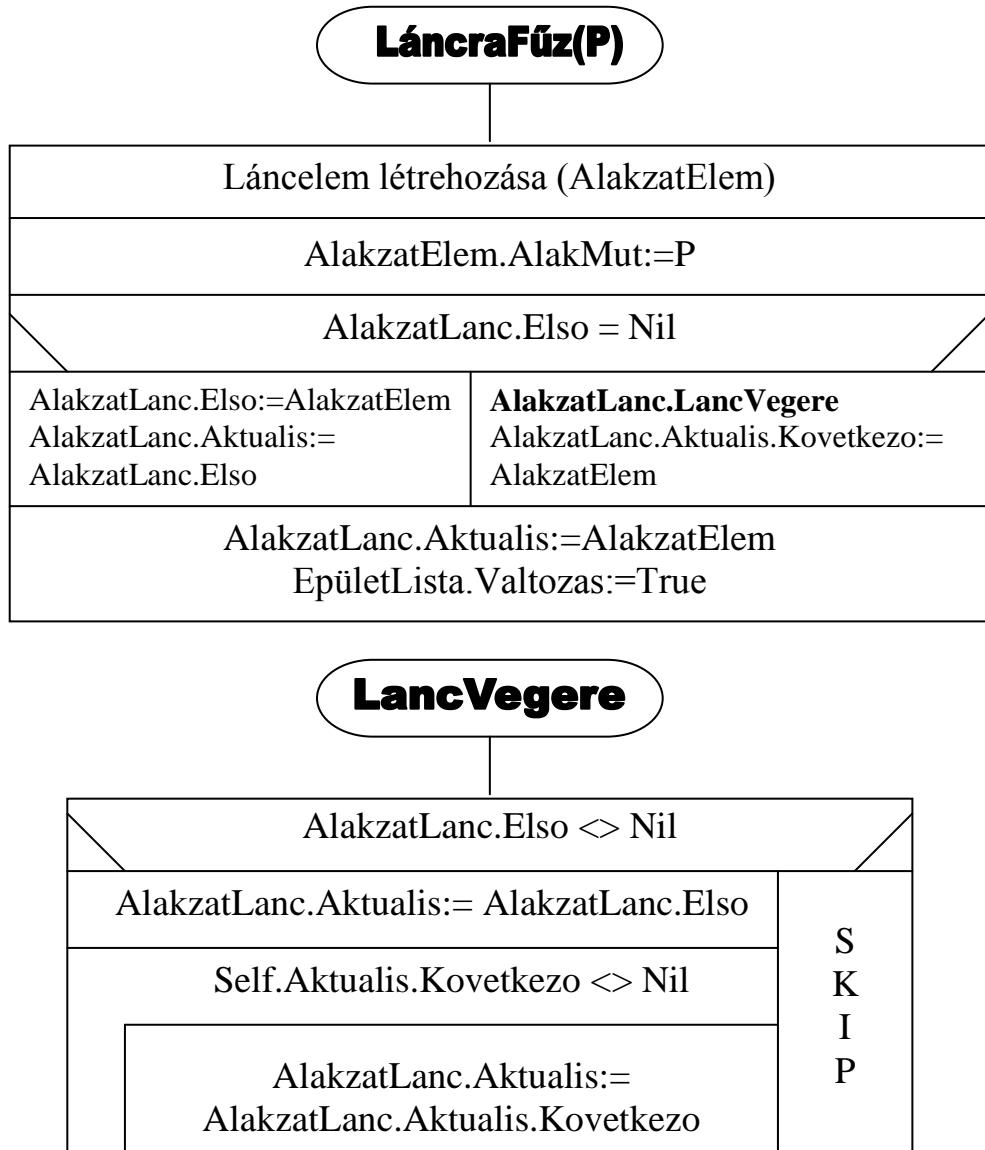
1. Új épület létrehozása

Új épület létrehozásakor automatikusan létrejön egy új épületegység is.

2. Új alakzat hozzáfűzése az épülethez

Ez valójában az éppen aktuális épületegységhez tartozó alakzat lesz, amely az épületegység alakzatokat tartalmazó láncára lesz felfűzve.

P: Alakzatra mutató pointer, amely már korábban létre lett hozva a memóriában.



3. Alakzatlánc megszüntetése

Egy épületegységet tartalmazó lánc felszabadítása, a hozzá tartozó alakzatokkal együtt.



4. Épület megszüntetése

Egy épület megszüntetése, a hozzá tartozó épületegységekkel együtt.



AZ ADATKAPCSOLATI RÉTEG

FELADATA ÉS MŰKÖDÉSE

(KuldThreadUnit.pas)

Az adatkapcsolati réteg három feladatot hajt végre:

- A kommunikációs porton kapott információkat keretekbe rendezi
- Hibavédelem megvalósítása
- Az üzenetek (keretek) továbbítása és nyugtázása

A feladatkörből látszik, hogy az egyik legfontosabb feladat a keretek összeállítása, de mik is azok a keretek? A keret az adat valamilyen határoló jelek közé foglalt része, amely a tényleges adaton kívül egyéb információkat is tartalmaz. Mivel nagyon fontos, a biztonságos és hibamentes átvitel biztosítása, ezért nagyon fontos, hogy olyan ellenőrzést is kell a keretbe foglalni, amely alapján felfedezhetők a hibák. Ilyen esetben a rossz keretet újra le lehet kérni a küldőtől. Hibavédelmet egy ellenőrző kód valósítja meg, amely a keret létrehozásakor jön létre és épül be a keretbe. A keret vételekor a vevő a kód alapján kinyeri az adatot az üzenetből és ha a dekódolás során bármilyen hiba előfordul, akkor a keret megsérült, a vevő eldobja és kéri újra a küldőtől.

Átvitel során bináris információt, vagyis biteket kell továbbítani az adó és a vevő között. A hatékonyabb működés érdekében egyszerre a lehető legtöbb bitet kell továbbítani. A számítógépek kialakulásakor nagyon sokszor kellett szöveges információt, vagyis karakterSORozatokat küldeni. A karakterek mindenazonos módon, az ASCII kódnak megfelelően tárolódtak. Az ilyen információtovábbítást **karakterorientált átvitelnek** nevezzük. A dolgozat is ezt a karakterorientált átvitelt valósítja meg.

A fejlődéssel együtt járt, hogy már egyre többször kellett nem karakteralapú adatokat továbbítani. Ilyen esetekben a biteket sorasan küldjük egymás után át a vonalon, de célszerű több (például védelmi) szempontból ezeket csoportokba foglalni. Ezt a módszert nevezzük **bitorientált átvitelnek**.

Keretek képzése

Egy keret létrehozása során az üzenet fejlécében megadjuk, az átvitelre kerülő adatok számát és ezeket az értékeket egy elválasztó karakterrel fűzzük össze. Vezérlőkarakterek beszúrásával jelezhetők az egyes keretek végét. Az ASCII kódkészletben meglehetősen sok vezérlőkarakter található. Ezek nem jeleníthetők meg csupán jelzési és vezérlési feladatra használhatók.

Alkalmazhatjuk ezeket is, de alkalmazhatunk olyan karaktert is, amelyről biztosan tudjuk, hogy adatként nem fordulhat elő.

Nézzünk egy konkrét példát!

Tegyük fel, hogy egy hőmérő hőmérsékletváltozást érzékel és ezt az új értéket továbbítani szeretné a vezérlőprogram felé. Legyen a berendezés azonosító száma 10, és a hőmérőn mért érték pedig 21°C fok.

A kódolás előtt létrejön egy dinamikus tömb, amelynek a mérete attól függ, hogy mennyi adatot szeretnénk elküldeni. Jelen esetben a tömb mérete kettő lesz. A tömb első eleme a berendezés azonosítóját tartalmazza, a második pedig az új hőmérséklet értékét. A kódoló rutin (*func KodolasUnit.Kodolo*) ezt a tömböt fogja megkapni és ez alapján fogja elkészíteni az elküldendő karaktersort. Az értékeket a '#' karakter választja el egymástól. Egy általános üzenet felépítése:

adatok száma # 1. adat # 2. adat # ... # n. adat

Tehát a példa alapján a lekódolt sorozat: #2#10#21#

Az egyes sorozatokhoz szöveges üzenetet is kapcsolhatunk, amelyek ugyan nem kerülnek elküldésre, de a felhasználót informálhatják a küldés eseményéről.

Adatkapcsolati protokoll

Az adatkapcsolati réteg, mint azt már tudjuk, a keretek elkészítését valósítja meg, majd a kereteket átadja a fizikai rétegnak, amely azt továbbítja a vevő irányába. Felmerülhet a kérdés, hogy hogyan értesül az adó, ha a vevő hibásan kapta meg a csomagot, vagy még rosszabb esetben (bár a végeredmény szempontjából majdnem mindegy) meg sem kapta a csomagot? A hibalehetőségek száma csak növekszik, ha az adó és a vevő egyszerre adhat. Ezeknek a problémáknak a kezelésére több módszert fejlesztettek ki:

- Korlátozás nélküli szimplex protokoll
- Szimplex megáll- és vár protokoll
- Egyirányú összetett protokoll
- Kétirányú protokollok
- Egybites csúszóablakos protokoll
- N visszalépést alkalmazó protokoll
- Szelektív ismétlő protokoll

Részletesebben a második pontban említett *szimplex megáll- és vár protokollra* térnék ki, mert a dolgozatban elkészített protokoll ezt használja fel alapul.

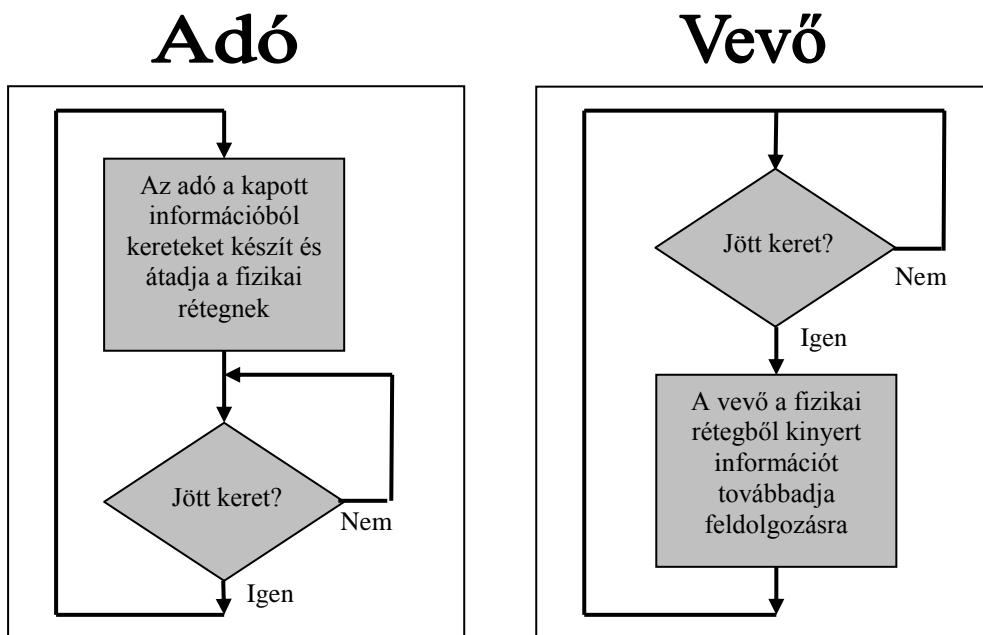
Szimplex megáll- és vár protokoll

Ez a protokoll az első pontban említett *korlátozás nélküli szimplex protokoll* egy továbbfejlesztett változata.

Az első protokoll típus feltételezi, hogy az egyes résztvevők minden készen állnak az adatátvitelre, a feldolgozó idő elhanyagolhatóan rövid és a keretek tárolására végtelen kapacitás áll a rendelkezésre. Ez amilyen egyszerű, a gyakorlatban annyira használhatatlan is. A *szimplex megáll- és vár protokoll* az előzőekben említett hibák bizonyos mértékű lecsökkentése.

Az előző megoldás legnagyobb problémája az, hogy szinte soha sem képes a vevő olyan sebességgel feldolgozni a kereteket, ahogy az adó küldi, vagy a hibás keretek adását meg kell ismételni. A hibalehetőség és az elárasztás lehetősége fokozottan jelentkezik akkor, ha a vevőnek több vonalra is figyelnie kell.

A megoldás meglehetősen kézenfekvő, a vevőnek közölnie kell az adóval, hogy mikor adhatja a következő keretet, amelyet már képes fogadni. Ezt legegyeszerűbben nyugtázással lehet megvalósítani. A vevőnek egy nyugtázást kell küldenie az adónak akkor, ha végzett az előző keret vételével, feldolgozásával. A nyugta megérkezéséig az adó egyszerűen várakozik.



Szimplex megáll- és vár protokoll

A programban használt protokoll

Mind a *Vezérlő*, mind a *Tesztelő* program ugyanazt a protokoll-t használja, amely az adó része a *KuldThreadUnit.pas*-ban található. A protokoll az előbb említett *szimplex megáll- és vár protokoll*-t építi magába mégpedig úgy, hogy minden adó, minden vevő feladatait is ellátja. Tehát a protokoll képes adóként és

vevőként is funkcionálni. Természetesen minden adó minden vevő jól elkülöníthető egységet képez.

A kommunikációban használt konstans változók

Neve	Értéke	Leírás
rKapcsVege	#1	kapcsolat vége
rNyugta	#2	üzenetet megkaptam
rHiba	#3	hiba az átvitelben
rKapcsKer	#13	kapcsolat létrehozása
rKapcsKesz	'OK'	kapcsolat létrejött
sHatJel	' '	üzeneteket elválasztó jel

A kommunikációban használt típusok

1. Szöveg jellemzői

Leírás: A kommunikációs folyamat során kiírandó üzenetek tulajdonságainak tárolása.

TTextAttrib = Object

```
Color : TColor;      //szín
Height : Integer;    //méret
End;
```

2. Láncelem

Leírás: Az üzenetláncban szereplő láncelem, amely egy üzenetet (keretet) tartalmaz.

TUzenetElem = Object

```
Prior      : Boolean;        //üzenet prioritása
Uzenet     : String;         //üzenet
Szoveg     : String;         //üzenethez kapcsolódó kiírandó szöveg
Kovetkezo : PUzenetElem;    //következő elemre mutat
Attrib     : TTextAttrib;    //szöveg jellemzője
Procedure Init(UL: TUzenetLista; P: Boolean; Uz: String); Overload;
Procedure Init(UL: TUzenetLista; P: Boolean; Uz,Szov: String; Atb:
TTextAttrib); Overload;
End;
```

Eljárások, függvények:

1. **Procedure Init(UL: TUzenetLista; P: Boolean; Uz: String)**

Egy láncelem létrehozása a három paraméter alapján.

- UL : az üzeneteket tartalmazó lánc
- P : üzenet prioritása (True - magas, False - alacsony)
- Uz : üzenet tartalma

2. **Procedure Init(UL: TUzenetLista; P: Boolean; Uz, Szov: String; Atb: TTextAttrib)**

Egy láncelem létrehozása az öt paraméter alapján.

- UL : az üzeneteket tartalmazó lánc
- P : üzenet prioritása (True - magas, False - alacsony)
- Uz : üzenet tartalma
- Szov : kiírandó szöveg
- Atb : a kiírandó szöveg tulajdonságai

3. Fejelem

Leírás: Az üzenetláncban szereplő fejelem, amely tartalmazza a lánc első- és aktuális elemére (keretre) mutató pointert.

TUzenetLanc = Object

```
  Elso    : PUzenetElem; //első elemre mutat
  Aktualis : PUzenetElem; //aktuális elemre mutat
  Procedure Init(UL: TUzenetLista);
End;
```

Eljárások, függvények:

1. **Procedure Init(UL: TUzenetLista)**

Egy láncelem létrehozása a paraméter alapján.

- UL : az üzeneteket tartalmazó lánc

4. Üzenetláncot tartalmazó típus

Leírás: Az üzenetláncot megvalósító típus.

TUzenetLista = Class(TPersistent)

Private

```
  Form      : TForm;           //főform
  ListBox   : TListBox;        //főformon található listbox
```

```

FMasolat      : PUzenetElem;    //egy üzenet másolata
UzenetElem    : PUzenetElem;    //egy láncelemre mutat
UzenetLanc    : TUzenetLanc;   //üzenetláncra mutat
Procedure Fuz(UzElem: PUzenetElem);
Public
  Property Masolat : PUzenetElem Read FMasolat;
  Property Elso : PUzenetElem Read UzenetLanc.Elso Write UzenetLanc.Elso;
  Property Aktualis : PUzenetElem Read UzenetLanc.Aktualis Write
    UzenetLanc.Aktualis;
  Function Torles : Boolean;
  Function MasolatTorles : Boolean;
  Function MasolAndTorles : Boolean;
  Function Ures : Boolean;
  Procedure Vegere;
  Procedure Felfuz(P: Boolean; Str: String); Overload;
  Procedure Felfuz(P: Boolean; Str,Szov: String; Atb: TTextAttrib); Overload;
  Constructor Create; Overload;
  Constructor Create(F: TForm; LB: TListBox); Overload;
  Destructor Destroy; Override;
End;

```

Eljárások, függvények:

1. Procedure Fuz(UzElem: PUzenetElem)

Egy elem (keret) felfűzése a láncra.

- UzElem : felfűzendő elem mutatója

2. Function Torles: Boolean

Az első láncelem törlése. Ha sikerült -> True, különben -> False.

3. Function MasolatTorles: Boolean

Régi másolat törlése. Ha sikerült -> True, különben -> False.

4. Function MasolAndTorles: Boolean

Üzenet másolatának létrehozása és az üzenet törlése. Ha sikerült -> True, különben -> False.

5. Function Ures: Boolean

Üres-e a lánc. Ha üres -> True, különben -> False.

6. Procedure Vegere

Elmegy a lánc végére.

7. Procedure Felfuz(*P*: Boolean; *Str*: String)

Egy elem (üzenet) felfűzése a láncra, a két paraméter alapján.

- *P* : üzenet prioritása (True - magas, False - alacsony)
- *Str* : üzenet

8. Procedure Felfuz(*P*: Boolean; *Str*, *Szov*: String; *Atb*: TTextAttrib)

Egy elem (üzenet) felfűzése a láncra, a négy paraméter alapján.

- *P* : üzenet prioritása (True - magas, False - alacsony)
- *Str* : üzenet
- *Szov* : kiírandó szöveg
- *Atb* : a kiírandó szöveg tulajdonságai

9. Constructor Create

Üzenetláncot tartalmazó típus létrehozása.

10. Constructor Create(*F*: TForm; *LB*: TListBox)

Üzenetláncot tartalmazó típus létrehozása a két paraméter alapján.

- *F* : a meghívó form
- *LB* : a meghívó formon található listbox

11. Destructor Destroy

Üzenetláncot tartalmazó típus megszüntetése.

5. Kommunikációs csatorna állapota

Leírás: Mindig megadja, hogy a másik gép készen áll-e az adatok fogadására.

TPortAllapot = Object

 Szabad : Boolean; //lehet-e küldeni adatot
End;

6. Üzenetek küldéséért felelős szál

Leírás: Mikor egy üzenet (keret) feldolgozásra került, akkor küld egy nyugta üzenetet (#2) az adónak, hogy jöhet a következő keret. Mikor ez a nyugta megérkezett, akkor a szál megpróbálja elküldeni a következő üzenetet, ami az üzenetláncban található. Egy üzenet elküldésére 3-szori próbálkozási lehetőség van, ha ezután is hibásan menne el az üzenet az adó oldalról, akkor az üzenet törlődik a láncból.

TKuldThread = Class(TThread)

Private

```

FMemo      : TRichEdit;      //főformon található richedit
FUzenet    : String;        //üzenet
FSzoveg    : String;        //kiírandó szöveg
FAttrib    : TTextAttrib;   //szöveg jellemzője
FKuldhet  : Boolean;       //lehet-e küldeni
FComPort   : TComPort;      //comport komponens
FUzenetHossz : Integer;    //mennyi értékből áll az üzenet
FUzenetLista : PUzenetLista; //üzenetláncra mutat
FFPortAllapot : PPortAllapot; //kommunikációs csatorna állapota
Procedure Kiir;
Procedure UzenetKuld;
Protected
  Procedure Execute; Override;
Public
  Property Kuldhet : Boolean Read FKuldhet Write FKuldhet;
  Constructor Create(CP: TComPort; PA: PPortAllapot; UL: PUzenetLista;
    Mem: TRichEdit);
  Destructor Destroy; Override;
End;

```

Eljárások, függvények:

1. Procedure Kiir

A szálból származó üzenetek kiírása.

2. Procedure UzenetKuld

Az üzenet tényleges elküldése.

3. Procedure Execute

A szál főrutinja.

4. Constructor Create(CP: TComPort; PA: PPortAllapot; UL: PUzenetLista; Mem: TRichEdit)

Szál létrehozása a négy paraméter alapján.

- CP : a TComPort típus átadása
- PA : a port állapotát leíró mutató
- UL : az üzeneteket tartalmazó lánc
- Mem : a főformon található richedit

5. Destructor Destroy

Szál megszüntetése.

Globálisan elérhető függvény

1. Időpont lekérdezése

Leírás: Visszaadja a lekérdezés (meghívás) időpontját stringben.

Alakja: '[óó:pp:mm] - '

Function MemoIdo : String;

ÜZENET KÓDOLÁSA, DEKÓDOLÁSA

(KodolasUnit.pas)

A kódolás előtt létrejön egy dinamikus tömb, amelynek a mérete attól függ, hogy mennyi adatot szeretnénk elküldeni. Típusa:

TAdatUzenetTomb = Array Of Integer;

A tömb első eleme mindig a berendezés azonosítóját tartalmazza, a többi elem pedig a berendezésre vonatkozó adatokat, értékeket. A kódoló rutin (*func KodolasUnit.Kodolo*) ezt a tömböt fogja megkapni és ez alapján fogja elkészíteni az elküldendő karaktersort. Az értékeket a '#' karakter választja el egymástól. Egy általános üzenet felépítése:

adatok száma # 1. adat # 2. adat # ... # n. adat

Kódolás

Feladat specifikáció:

Visszatérési értéke a kódolt üzenet.

T : TAdatUzenetTomb típus

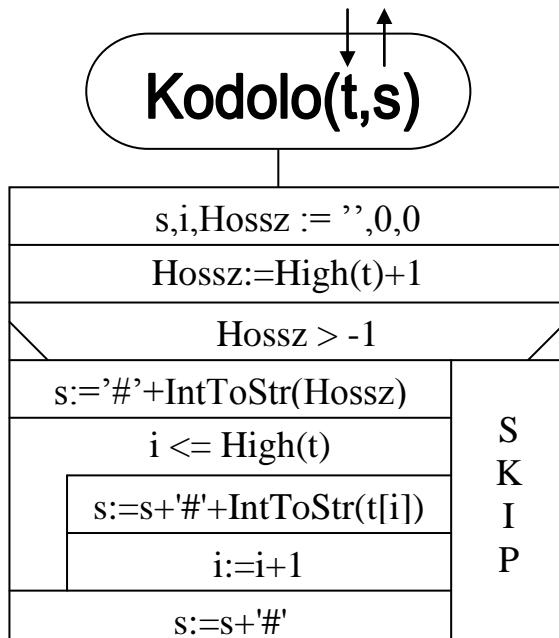
S : String típus

A: $\frac{\mathbf{T}}{t} \times \frac{\mathbf{S}}{s}$

B: $\frac{\mathbf{T}}{t'}$

Q: $(t = t' \wedge |t| > 0)$

R: $(Q \wedge s = \text{Kodolo}(t'))$



Beépített függvények:

- **High(X: Array) : Integer**
Visszaadja a paraméterben megadott tömb hosszát.
- **IntToStr(X: Integer) : String**
A paraméterben megadott értéket átalakítja String-é.

Dekódolás

Feladat specifikáció:

Visszatérési értéke a dekódolt üzenet egy dinamikus tömbben.

T : TAdatUzenetTomb típus

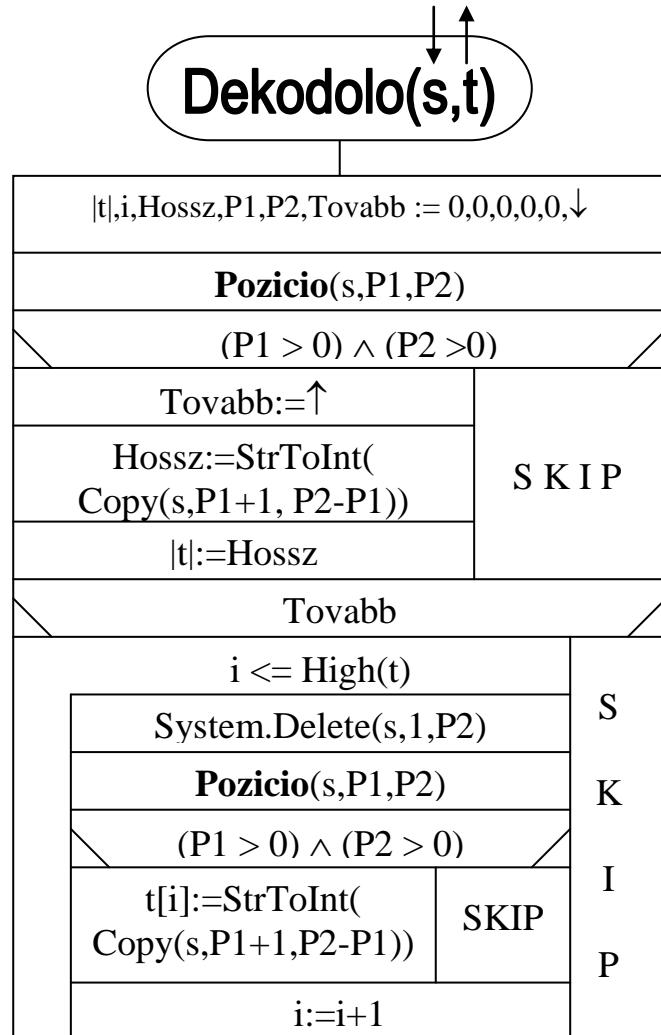
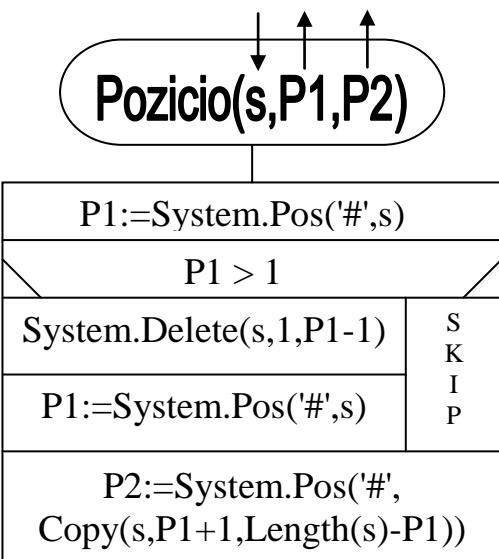
S : String típus

A: $\underset{s}{S} \times \underset{t}{T}$

B: $\underset{s'}{S}$

Q: $(s = s' \wedge |s| > 0)$

R: $(Q \wedge t = \text{Dekodolo}(s'))$



Beépített függvények:

- **High(X: Array) : Integer**
Visszaadja a paraméterben megadott tömb hosszát.
- **StrToInt(Str: String) : Integer**
A paraméterben megadott értéket átalakítja String-é.
- **Pos(Str, Source: String): Integer**
Megmondja, hogy a keresett sztring (Str) hol található a Source sztringben.
- **Copy(Str: String; Pos, Length: Integer) : String**
Az Str sztringből a Pos helytől Length hosszan visszaadja a sztringet.

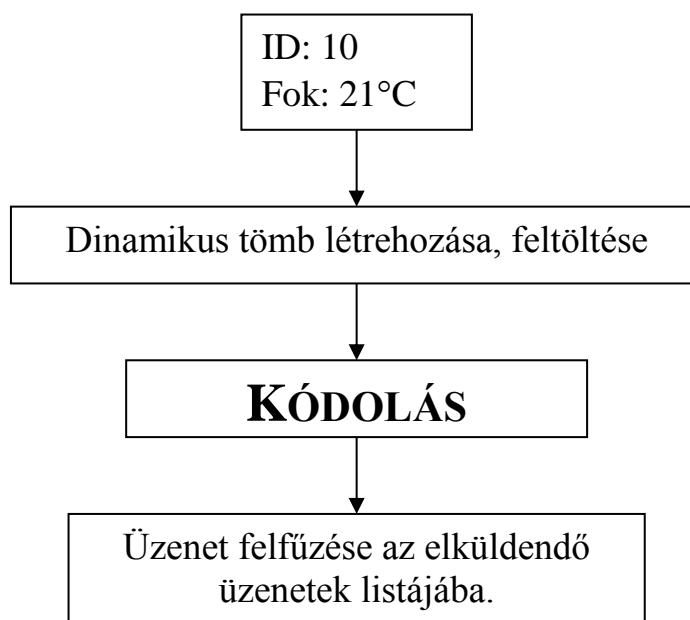
- Delete(Str: String; Pos, Length: Integer)
Az Str sztringből a Pos helytől Length hosszan töröl.
- Length(Str: String) : Integer
Visszaadja az Str sztring hosszát.

Kódolás-dekódolás folyamata:

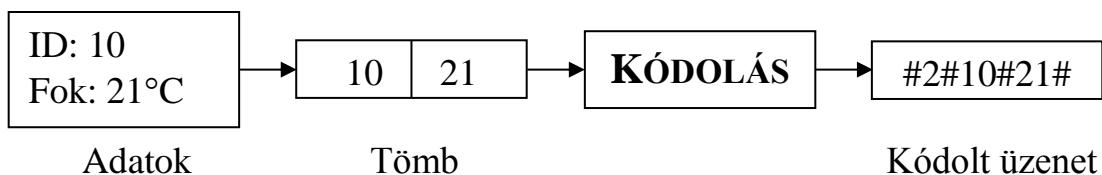
Nézzünk egy konkrét példát!

Tegyük fel, hogy egy hőmérő hőmérsékletváltozást érzékel és ezt az új értéket továbbítani szeretné a vezérlőprogram felé. Legyen a berendezés azonosító száma 10, és a hőmérőn mért érték pedig 21°C fok.

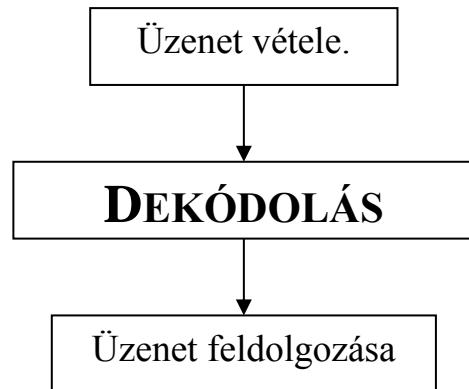
- Kódolás folyamat áttekintése:



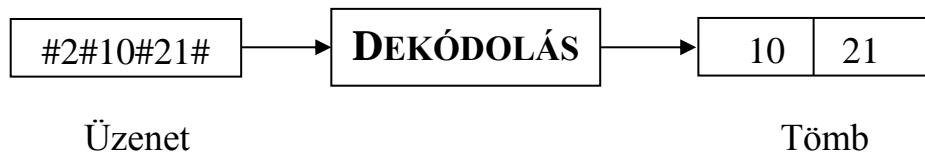
- Kódolás adatfolyam áttekintése:



- Dekódolás folyamat áttekintése:



- Dekódolás adatfolyam áttekintése:



**TERVEZŐ
FEJLESZTŐI DOKUMENTÁCIÓ**

Specifikáció

A Tervező program egy épület tulajdonságainak a bevitelére szolgál, ahol egyszerű rajzolási eszközökkel meg lehet rajzolni egy épület vagy épületegységek alaprajzát, másrészt az épülethez tartozó objektumokat, berendezéseket tudjuk megadni.

A program főbb funkciói

- Alaprajz bevitel
- Objektumok kezelése
- Berendezések kezelése

- *Alaprajz bevitel*

Egyszerű rajzolási eszközökkel (vonal, téglalap, ellipszis) egy épületnek az alaprajzát lehet megrajzolni, a rajzot elmenteni, visszatölteni és módosítani. Egy épület állhat egy vagy több épületegységből, attól függően, hogy mennyire szeretnénk az egyes részeket megkülönböztetni egymástól, vagy hogy maga az épület is több fizikailag különálló egységből áll.

- *Objektumok kezelése*

Az épület alaprajzán kitüntetett területeket lehet bejelölni (objektum), melyek lehetnek például szobák, melléképület, kert, stb...

A programban lehetőség van új objektumok felvitelére és törlésére is.

- *Berendezések kezelése*

Ha már az épület rendelkezik objektummal, akkor azt feltölthetjük különféle berendezésekkel. Berendezést hozzákapcsolhatunk az épülethez anélkül is, hogy objektumba zárnánk.

Kétféle berendezéstípus áll a rendelkezésre:

- Érzékelő berendezés: hőmérő, riasztó, tűzjelző
- Véghajtó berendezés: fűtőtest, egyéb berendezés.

A programban használt UNIT-ok:

- *AboutUnit.pas*
A készítőről és a programról ad információt.
- *AjtoUnit.pas*
Az ajtó form eljárásait, függvényeit tartalmazza. Form-hoz tartozó unit.
- *BeallitUnit.pas*
Beállítás form eljárásait, függvényeit tartalmazza. Form-hoz tartozó unit.
- *BerendezesekLancUnit.pas*
Berendezéseket tartalmazó lánc típusának eljárásai, függvényei.

- *EpuletLancUnit.pas*
Épület szerkezetét tartalmazó láncok.
- *FileTablaUnit.pas*
Adatbázisokkal kapcsolatok eljárások, függvények.
- *ForgatUnit.pas*
Forgatás form eljárásait, függvényeit tartalmazza. Form-hoz tartozó unit.
- *FormFestoUnit.pas*
Egy form átfestésének eljárásai.
- *GradientLabel komponens*
Színátmenetes label. Más által írt komponens. Használat előtt telepíteni kell.
- *KeretUnit.pas*
Kijelölő keret eljárásai, függvényei.
- *KijelolNegyzetUnit.pas*
A kijelölő négyzet eljárásai, függvényei.
- *MeretFormUnit.pas*
Rajzasztal beállítása form eljárásait, függvényeit tartalmazza. Form-hoz tartozó unit.
- *ObjektumokDlgUnit.pas*
Objektumok megjelenítése form. Form-hoz tartozó unit.
- *ObjektumUnit.pas*
Objektum lánc eljárásai, függvényei.
- *ObjListaUnit.pas*
Objektumok listája form. Form-hoz tartozó unit.
- *ObjNevFormUnit.pas*
Új objektum nevének felvitele form. Form-hoz tartozó unit.
- *PenUnit.pas*
A tollal kapcsolatos eljárások.
- *RajzolasUnit.pas*
Rajzolással kapcsolatos eljárások, függvények.
- *ResolutionUnit.pas*
Monitor felbontásának beállítása.
- *TavolsagUnit.pas*
Két pont távolságát kiszámító eljárások, függvények.
- *TervezoUnit.pas*
A Tervező program főformjának unit-ja. Form-hoz tartozó unit.
- *UjBerendezesUnit.pas*
Új berendezés hozzáadása form. Form-hoz tartozó unit.
- *UjEpEgysegDlgUnit.pas*
Új épületegység felvétele form. Form-hoz tartozó unit.
- *UjUnit.pas*
Indulási form. Form-hoz tartozó unit.

- *UtilUnit.pas*
Általános eljárások, függvények.
- *ZoomUnit.pas*
Nagyítás-kicsinyítés függvényei.

Globális eljárások, függvények a programban

Ezek a konstansok és változók a programok egész működése során meghívhatók:

(*KeretUnit.pas*):

- *Function RelHely(Pont1,Pont2 : Extended) : Extended;*
Két pont relatív helyét, távolságát adja vissza.
- *Function UjKoord(Const Sarok: TSarokKoord; Const Irany,Szog: Integer; Const P1,P2: Extended) : TPoint;*
Egy pont elforgatása egy sarok körül. Leírás ld.: *Forgatás egy pont körül*

Típus- és osztályleírások:

(*KeretUnit.pas*):

- *TSarokKoord = Object*
A kijelölő keret egy sarkának koordinátái.
- *TKeretClass = Class*
A keretet tartalmazó ősosztály.
- *TKeret = Class(TKeretClass)*
A keret osztály.

(*KijelolNegyzetUnit.pas*):

- *TKijelolNegyzet = Object*
A kijelölő négyzet típusa.

(*TavolsagUnit.pas*):

- *TTavolsag = Object*
Két pont távolságát nyilvántartó és számító típus.

(*TervezoUnit.pas*):

- *TAranyok = Record*
A kijelölő keret méretei és arányai.
- *TVezerloAllapot = Class(TPersistent)*
A rajzolással kapcsolatos eszközök adatai.

Műveletek az alakzatokon

Ha módosítani szeretnénk egy vagy több alakzaton, akkor előbb azokat ki kell jelölni. Ezt a kijelölés funkcióval tudjuk elvégezni. A kijelölés során egy új lánc jön létre, amely csak a kijelölt elemeket tartalmazza. Ez felfogható úgy is, mint egy biztonsági másolat. Ha valami miatt nem lehetséges a kiválasztott művelet, akkor a másolat alapján vissza lehet állítani az eredeti láncot. Ilyen hiba lehet például, ha forgatás, vagy mozgatás esetén az alakzat „lelögna” a rajzlapról.

Ha a művelet rendben lezajlott, akkor a másolatot tartalmazó lánc megszűnik.

A kijelölés után az alábbi műveleteket tudjuk elvégezni:

- mozgatás,
- forgatás,
- méretezés,
- törlés

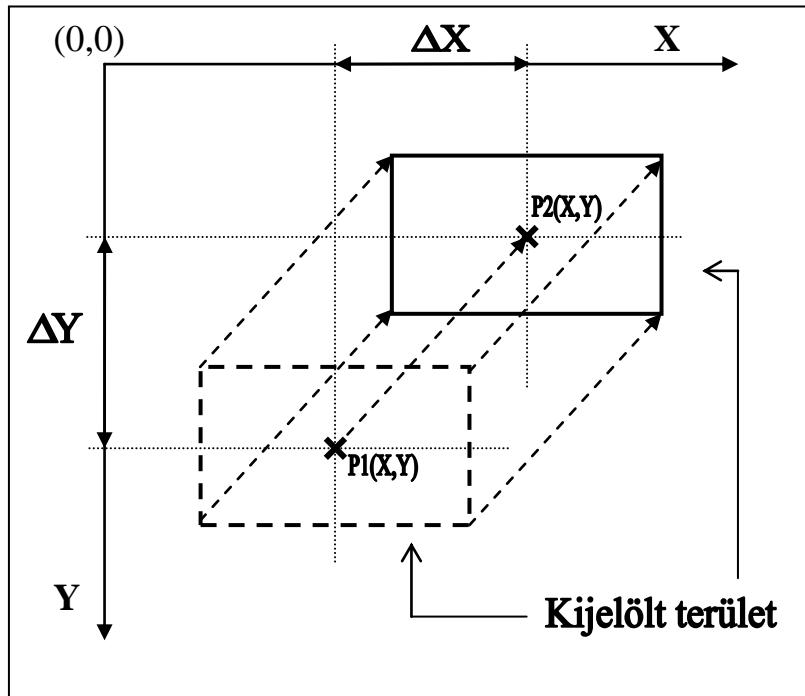
Mozgatás

Egy alakzat mozgatásakor, a keret elmozdulásának alapján számolódnak ki az új koordináta értékek. A mozgatás művelet aktiválásakor a keret koordinátái elmentésre kerülnek és a mozgatás végén - amit az egér bal gombjával jelzünk – a keret újra kiértékelődik. A kezdeti- és véghelyzet közötti eltéréssel fognak az alakzatok koordinátái megváltozni. Ezek az értékek automatikusan számolódnak ki (*Procedure TKeret.KeretSzamit*).

$$\Delta X = P2.X - P1.X$$

$$\Delta Y = P2.Y - P1.Y$$

Ezzel a ΔX , ΔY értékkal fognak megváltozni a kijelölt alakzatok koordinátái. Ha valamelyik koordináta kívül esne a rajzolási területen, akkor a mozgatás során nem rögzülnek az új koordináták.



Forgatás egy pont körül

(Function *UjKoord*)

Feladat: egy pont elforgatása egy másik pont körül.

A függvény input értékei:

- *Sarok*: annak a pontnak a koordinátája, amelyik körül a forgatást szeretnénk elvégezni.
- *Irany*: azt adja meg, hogy az óramutató járásával megegyező vagy ellentétes irányba akarunk forgatni. Értéke lehet:
 - (1), ha az óramutató járásával megegyező,
 - (-1), ha az óramutató járásával ellentétes irányba forgatunk.
- *Szog*: a forgatás szögét adja meg
- *P1,P2*: a forgatandó pont X,Y koordinátája.

A függvény visszatérési értéke az elforgatott pont koordinátája, amely TPoint típusú.

Feladat specifikáció:

S : TSarokKoord típus

P : TPoint típus

N : Integer típus

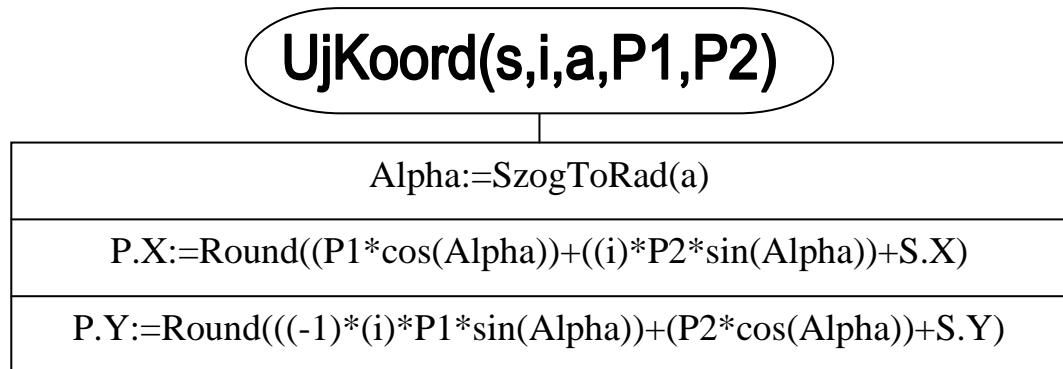
R : Extended típus

$$A: \underset{s}{\mathbf{S}} \times \underset{i}{\mathbf{N}} \times \underset{a}{\mathbf{N}} \times \underset{P1}{\mathbf{R}} \times \underset{P2}{\mathbf{R}} \times \underset{P}{\mathbf{P}}$$

$$B: \underset{s'}{\mathbf{S}} \times \underset{i'}{\mathbf{N}} \times \underset{a'}{\mathbf{N}} \times \underset{P1'}{\mathbf{R}} \times \underset{P2'}{\mathbf{R}}$$

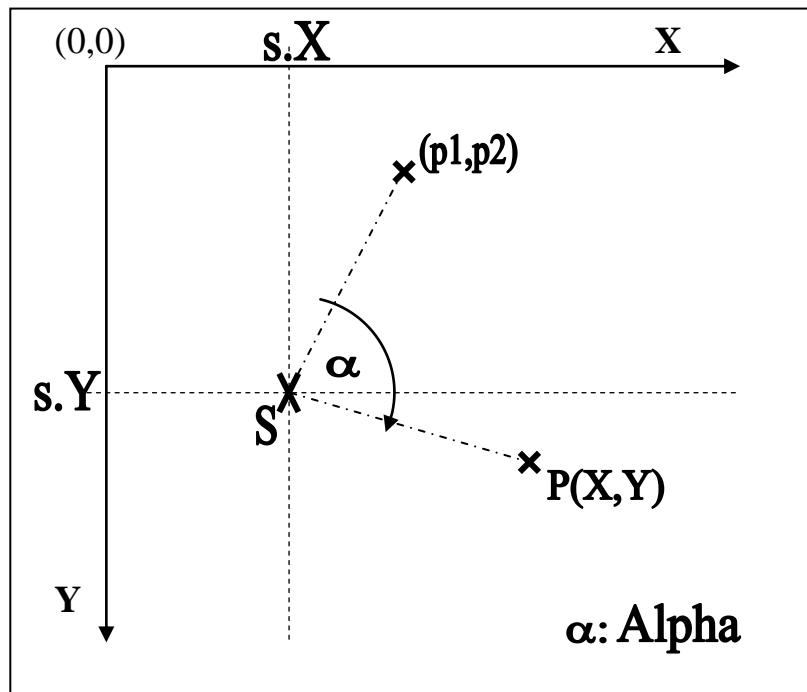
$$Q: (s = s' \wedge i = i' \wedge a = a' \wedge P1 = P1' \wedge P2 = P2')$$

$$R: (Q \wedge P = UjKoord(s', i', a', P1', P2'))$$



Felhasznált függvények:

- SzogToRad(Szog: Integer) : Extended
Szög átalakítása radiánba. (*RajzolasUnit.pas*)
- Round(X: Extended) : Integer
Az X-et visszaadja kerekítve (beépített függvény).



Méretezés

Egy alakzat méretezésekor, a keret méretváltozásának (arányának) alapján számolódnak ki az új koordináta értékek.

$$\Delta X = X_2 - X_1$$

$$\Delta Y = Y_2 - Y_1$$

Minden lépésben kiértékelődik egy új arányszám, amivel majd az új koordináta értékek lesznek beszorozva.

$$\text{AranyX} = X_2 / X_1$$

$$\text{AranyY} = Y_2 / Y_1$$

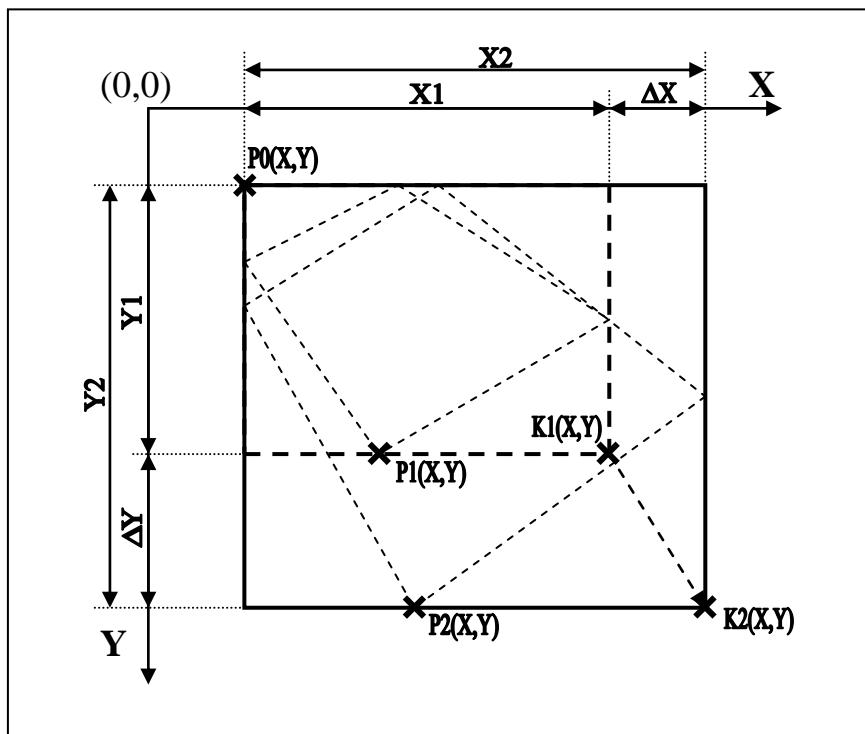
Ezekkel az arányszámokkal fognak beszorzódni a koordináták.

Ha az arányszám $\left\{ \begin{array}{l} > 1, \text{ akkor nagyítást végzünk} \\ = 1, \text{ helyben marad} \\ < 1, \text{ akkor kicsinyítést végzünk} \end{array} \right\}$

$$P2.X := (\text{AranyX} * (P1.X - P0.X)) + P0.X$$

$$P2.Y := (\text{AranyY} * (P1.Y - P0.Y)) + P0.Y$$

Ha valamelyik koordináta kívül esne a rajzolási területen, akkor a méretezés során nem rögzülnek az új koordináták.



Törlés

A kijelölt terület törlésekor az alakzatokat egyszerűen kifűzzük az alakzatokat tartalmazó láncból. Ha sikerült, akkor megszüntetjük a másolatot is.
(*TervezoUnit.pas - Procedure TFormTervezo.TorlesButtonClick*)



**TESZTELŐ
FEJLESZTŐI DOKUMENTÁCIÓ**

Specifikáció

A Tesztelő program egy épület berendezéseinek a tesztelését végzi, valóságos események szimulálásával. Ezért fontos, hogy az egyes berendezések külön-külön is kezelhetőek, irányíthatóak legyenek. Figyelemmel kell kísérni a küldési folyamatot és a küldés során esetlegesen előforduló hibákat.

A program főbb funkciói

- Kapcsolat a vezérlő programmal.
- Események generálása az egyes berendezésekre.
 - *Kapcsolat a vezérlő programmal*
Adatok küldése és fogadása a kommunikációs porton keresztül. A küldési folyamat figyelemmel kísérése és az esetleges hibák kezelése.
 - *Események generálása az egyes berendezésekre*
Egy épület berendezéseinek működését szimuláló események generálása.
Riasztásokat lehet létrehozni a tűzjelző és riasztó berendezésekre, hőmérséklet változást a hőmérő berendezésekre és ki-be kapcsolást a végberendezésekre (fűtőtest, egyéb berendezések).

A programban használt UNIT-ok:

- *BerendezesekLancUnit.pas*
Berendezéseket tartalmazó lánc típusának eljárásai, függvényei.
- *BerendTablaUnit.pas*
A berendezéseket tartalmazó adatbázisok eljárásai, függvényei.
- *ComPort* komponens
Soros port vezérlése. Más által írt komponens. Használat előtt telepíteni kell.
- *DataModulUnit.pas*
Adattáblákat tartalmazó unit.
- *EpuletLancUnit.pas*
Épület szerkezetét tartalmazó láncok.
- *FileTablaUnit.pas*
Adatbázisokkal kapcsolatok eljárások, függvények.
- *FormFestoUnit.pas*
Egy form átfestésének eljárásai.
- *Gomb.pas*
Kapcsoló komponens. Részben más által írt komponens. Használat előtt telepíteni kell.
- *HomeroLancUnit.pas*

Hőmérőket tartalmazó lánc eljárásai, függvényei.

- *InditUnit.pas*

Az InditasDlg form eljárásai. Épület betöltésekor hívódik meg. Form-hoz tartozó unit.

- *KapcsBerendLancUnit.pas*

Az egymáshoz tartozó berendezéseket tartalmazó lánc eljárásai, függvényei.

- *KodolasUnit.pas*

Üzenetek kódolása és dekódolása.

- *KuldThreadUnit.pas*

Kommunikációhoz szükséges eljárások, függvények. Az üzenetek küldéséért felelős szál eljárásai, függvényei.

- *MegnyitasUnit.pas*

Megnyitás form eljárásait, függvényeit tartalmazza. Form-hoz tartozó unit.

- *ObjektumUnit.pas*

Objektum lánc eljárásai, függvényei.

- *PenUnit.pas*

A tollal kapcsolatos eljárások.

- *PontLancUnit.pas*

Pontokat tartalmazó lánc eljárásai, függvényei.

- *RajzolasUnit.pas*

Rajzolással kapcsolatos eljárások, függvények.

- *ResolutionUnit.pas*

Monitor felbontásának beállítása.

- *RiasztoLancUnit.pas*

Riasztókat tartalmazó lánc eljárásai, függvényei.

- *TeszteloUnit.pas*

A Tesztelő program főformjának unit-ja. Form-hoz tartozó unit.

- *TuzjelzoLancUnit.pas*

Tűzjelzőket tartalmazó lánc eljárásai, függvényei.

- *UtilUnit.pas*

Általános eljárások, függvények.

- *VegBerendLancUnit.pas*

Végberendezéseket tartalmazó lánc eljárásai, függvényei.

- *ZoomUnit.pas*

Nagyítás-kicsinyítés függvényei.

Globális konstansok, változók

Ezek a konstansok és változók a programok egész működése során elérhetők.

(*BerendTablaUnit.pas*):

- Adatbázisok nevei (Const):

sEgyebDB = 'Egyeb'
sHomeroDB = 'Homero'
sRiasztoDB = 'Riaszto'
sTuzjelzoDB = 'Tuzjelzo'
sFutotestDB = 'Futotest'
sKapcsDB = 'Kapcsolat'

- Üzenetek típusa (Const):

rKapcsVege = #1 - kapcsolat vége
rNyugta = #2 - üzenetet megkaptam
rHiba = #3 - hiba az átvilelben
rKapcsKer = #13 - kapcsolat létrehozása
rKapcsKesz = 'OK' - kapcsolat létrejött
sHatJel = '/' - üzeneteket elválasztó jel

Globális eljárások, függvények a programban

Ezek a konstansok és változók a programok egész működése során meghívhatók:

(*BerendTablaUnit.pas*):

- *Function HomeroTablaBeolvas(Const Tabla: TTable; Const EpuletNev: String; HL: THomeroLista; BL: TBerendezesLista) : Boolean;*
Hőmérők beolvasása a láncba az adatbázisból.
Ha sikerült -> True, különben -> False.
- *Function RiasztoTablaBeolvas(Const Tabla: TTable; Const EpuletNev: String; RL: TRiasztoLista; BL: TBerendezesLista) : Boolean;*
Hőmérők beolvasása a láncba az adatbázisból.
Ha sikerült -> True, különben -> False.
- *Function TuzjelzoTablaBeolvas(Const Tabla: TTable; Const EpuletNev: String; TL: TTuzjelzoLista; BL: TBerendezesLista) : Boolean;*
Tűzjelzők beolvasása a láncba az adatbázisból.
Ha sikerült -> True, különben -> False.
- *Function FutotestTablaBeolvas(Const Tabla: TTable; Const EpuletNev: String; FL: TVegBerendLista; BL: TBerendezesLista) : Boolean;*
Fűtőtestek beolvasása a láncba az adatbázisból.
Ha sikerült -> True, különben -> False.
- *Function EgyebTablaBeolvas(Const Tabla: TTable; Const EpuletNev: String; EL: TVegBerendLista; BL: TBerendezesLista) : Boolean;*

Más berendezések beolvasása a láncba az adatbázisból.

Ha sikerült -> True, különben -> False.

- *Function KapcsTablaBeolvas(Const Tabla: TTable; Const EpuletNev: String; KBL: TKapcsBerendLista; BL: TBerendezesLista) : Boolean;*
Kapcsolatok beolvasása a láncba az adatbázisból.
Ha sikerült -> True, különben -> False.
- *Procedure HomeroTablaMentes(HomeroTabla: TTable; Const EpuletNev: String; HL: THomeroLista);*
Hőmérőket tartalmazó adatbázis mentése.
- *Procedure HomeroLancMentes(HL: THomeroLista; Const HomeroTabla: TTable);*
A hőmérőket tartalmazó lánc mentése.
- *Procedure HomeroTablaKeszit(Tabla: TTable; Nev: String; Tipus: TTableType);*
Adatbázis létrehozása.
- *Procedure RiasztoTablaMentes(RiasztoTabla: TTable; Const EpuletNev: String; RL: TRiasztoLista);*
Riasztókat tartalmazó adatbázis mentése.
- *Procedure RiasztoLancMentes(RL: TRiasztoLista; Const RiasztoTabla: TTable);*
A riasztókat tartalmazó lánc mentése.
- *Procedure RiasztoTablaKeszit(Tabla: TTable; Nev: String; Tipus: TTableType);*
Adatbázis létrehozása.
- *Procedure TuzjelzoTablaMentes(TuzjelzoTabla: TTable; Const EpuletNev: String; TL: TTuzjelzoLista);*
Tűzjelzőket tartalmazó adatbázis mentése.
- *Procedure TuzjelzoLancMentes(TL: TTuzjelzoLista; Const TuzjelzoTabla: TTable);*
A tűzjelzőket tartalmazó lánc mentése.
- *Procedure TuzjelzoTablaKeszit(Tabla: TTable; Nev: String; Tipus: TTableType);*
Adatbázis létrehozása.
- *Procedure FutotestTablaMentes(FutotestTabla: TTable; Const EpuletNev: String; VBL: TVegBerendLista);*
Fűtőtesteket tartalmazó adatbázis mentése.
- *Procedure FutotestLancMentes(VBL: TVegBerendLista; Const FutotestTabla: TTable);*
A fűtőtesteket tartalmazó lánc mentése.
- *Procedure FutotestTablaKeszit(Tabla: TTable; Nev: String; Tipus: TTableType);*

Adatbázis létrehozása.

- *Procedure EgyebTablaMentes(EgyebTabla: TTable; Const EpuletNev: String; VBL: TVegBerendLista);*
Más berendezéseket tartalmazó adatbázis mentése.
- *Procedure EgyebLancMentes(VBL: TVegBerendLista; Const EgyebTabla: TTable);*
A más berendezéseket tartalmazó lánc mentése.
- *Procedure EgyebTablaKeszit(Tabla: TTable; Nev: String; Tipus: TTableType);*
Adatbázis létrehozása.
- *Procedure KapcsTablaMentes(KapcsTabla: TTable; Const EpuletNev: String; KBL: TKapcsBerendLista);*
Kapcsolatokat tartalmazó adatbázis mentése.
- *Procedure KapcsLancMentes(KBL: TKapcsBerendLista; Const KapcsTabla: TTable);*
A kapcsolatokat tartalmazó lánc mentése.
- *Procedure KapcsTablaKeszit(Tabla: TTable; Nev: String; Tipus: TTableType);*
Adatbázis létrehozása.

(*KodolasUnit.pas*):

- *Function Dekodolo(Str: String) : TAdatUzenetTomb;*
Üzenet dekódolása.
Visszatérési értéke a dekódolt üzenet egy dinamikus tömbben.
- *Function Kodolo(Tomb: TAdatUzenetTomb) : String;*
Üzenet kódolása. Visszatérési értéke a kódolt üzenet.
- *Function BoolToStr(Ertek: Boolean) : String;*
Logikai értékből stringet állít elő.
- *Function StrToInt(Ertek: String) : Boolean;*
Stringből logikai értéket állít elő.
- *Function BoolToInt(Ertek: Boolean) : Integer;*
Logikai értékből integert állít elő.
- *Function IntToBool(Ertek: Integer) : Boolean;*
Integerből logikai értéket állít elő.

(*KuldThreadUnit.pas*):

- *Function MemoIdo : String;*
Visszaadja a lekérdezés időpontját stringben.

Típus- és osztályleírások

(*HomeroLancUnit.pas*):

- *THomeroElem* = *Object*
Egy hőmérőt tartalmazó láncelem.
- *THomeroLanc* = *Object*
Egy hőmérőlánc fejeleme.
- *THomeroLista* = *Class(TPersistent)*
Hőmérőláncot tartalmazó osztály.

(*KapcsBerendLancUnit.pas*):

- *TKapcsBerendElem* = *Object*
Egy kapcsolatot tartalmazó láncelem.
- *TKapcsBerendLanc* = *Object*
Egy kapcsolatlánc fejeleme.
- *TKapcsBerendLista* = *Class(TPersistent)*
Kapcsolatláncot tartalmazó osztály.

(*KodolasUnit.pas*):

- *TAdatUzenetTomb* = *Array Of Integer*
Egy üzenetet tartalmazó dinamikus tömb.

(*KuldThreadUnit.pas*):

- *TKuldThread* = *Class(TThread)*
Az üzenetek küldéséért felelős szál.
- *TPortAllapot* = *Object*
Kommunikációs csatorna állapota.
- *TTextAttrib* = *Object*
Szöveg jellemzői.
- *TUzenetElem* = *Object*
Egy üzenetet tartalmazó láncelem.
- *TUzenetLanc* = *Object*
Egy üzenetlánc fejeleme.
- *TUzenetLista* = *Class(TPersistent)*
Üzenetláncot tartalmazó típus.

(*PontLancUnit.pas*):

- *TPontElemTip* = *Object*
Egy pontot tartalmazó láncelem.
- *TPontLancTip* = *Object*
Egy pontlánc fejeleme.
- *TPontLista* = *Class(TPersistent)*
Pontláncot tartalmazó típus.

(*RiasztoLancUnit.pas*):

- *TRiasztoElem* = *Object*
Egy riasztót tartalmazó láncelem.
- *TRiasztoLanc* = *Object*
Egy riasztólánc fejeleme.
- *TRiasztoLista* = *Class(TPersistent)*
Egy riasztóláncot tartalmazó osztály.

(*TuzjelzoLancUnit.pas*):

- *TTuzjelzoElem* = *Object*
Egy tűzjelzöt tartalmazó láncelem.
- *TTuzjelzoLanc* = *Object*
Egy tűzjelzőlánc fejeleme.
- *TTuzjelzoLista* = *Class(TPersistent)*
Egy tűzjelzőláncot tartalmazó osztály.

(*VegBerendLancUnit.pas*):

- *TVegBerendElem* = *Object*
Egy végberendezést tartalmazó láncelem.
- *TVegBerendLanc* = *Object*
Egy végberendezéslánc fejeleme.
- *TVegBerendLista* = *Class(TPersistent)*
Egy végberendezésláncot tartalmazó osztály.

**VEZÉRLŐ
FEJLESZTŐI DOKUMENTÁCIÓ**

Specifikáció

A Vezérlő program egy épület berendezéseinek az irányítását végzi, valóságos események szimulálásával. Figyelemmel kell kísérni a vezérlési folyamatokat és a vezérlés során esetlegesen előforduló hibákat. Fontos követelmény, hogy az eseménykről naplófájl készüljön, hogy a későbbiekben a történések ellenőrizhetőek legyenek.

A program főbb funkciói

- Kapcsolat a tesztelő programmal.
- Folyamatok figyelemmel kísérése
- Berendezések adatainak kezelése
- Berendezések működésének vezérlése

- *Kapcsolat a tesztelő programmal*

Adatok küldése és fogadása a kommunikációs porton keresztül. A küldési folyamat figyelemmel kísérése és az esetleges hibák kezelése, naplázása.

- *Folyamatok figyelemmel kísérése*

Az irányítási folyamatok nyomon követése és az eredmények kijelzése. Ha szükséges, akkor a beavatkozás lehetőségének biztosítása.

- *Berendezések adatainak kezelése*

Az egyes berendezések tulajdonságainak beállítása, amelyek a működést befolyásolják.

- *Berendezések működésének vezérlése*

Az egyes irányítási folyamatokba való beavatkozások elvégzése. A berendezések üzembe helyezése esetleg kikapcsolása.

A programban használt UNIT-ok:

- *AboutUnit.pas*
A készítőről és a programról ad információt.
- *BeallitUnit.pas*
Beállítás form eljárásait, függvényeit tartalmazza. Form-hoz tartozó unit.
- *BerendDlgUnit.pas*
A berendezéset megjelenítő form eljárásait, függvényeit tartalmazza. Form-hoz tartozó unit.
- *BerendezesekLancUnit.pas*
Berendezéseket tartalmazó lánc típusának eljárásai, függvényei.
- *BerendInspUnit.pas*
A berendezések irányítópaneljének eljárásai, függvényei.

- *BerendTablaUnit.pas*
A berendezéseket tartalmazó adatbázisok eljárásai, függvényei.
- *Beeper.pas*
A rendszerhangszóró vezérlése. Más által írt unit.
- *ComPort komponens*
Soros port vezérlése. Más által írt komponens. Használat előtt telepíteni kell.
- *DataModulUnit.pas*
Adattáblákat tartalmazó unit.
- *DlgUnit.pas*
Az berendezések láncait tartalmazó típus.
- *EgyebDlgUnit.pas*
Az egyéb berendezések form eljárásait, függvényeit tartalmazza. Form-hoz tartozó unit.
- *EpuletLancUnit.pas*
Épület szerkezetét tartalmazó láncok.
- *FileTablaUnit.pas*
Adatbázisokkal kapcsolatok eljárások, függvények.
- *FormFestoUnit.pas*
Egy form átfestésének eljárásai.
- *FrmNyomtatUnit.pas*
Eseménynapló nyomtatási riportja.
- *FutotestDlgUnit.pas*
A fütőtest berendezések form eljárásait, függvényeit tartalmazza. Form-hoz tartozó unit.
- *Gomb.pas*
Kapcsoló komponens. Részben más által írt komponens. Használat előtt telepíteni kell.
- *GradientLabel komponens*
Színátmenetes label. Más által írt komponens. Használat előtt telepíteni kell.
- *HangThreadUnit.pas*
Figyelmeztető hangjelzést vezérlő szál eljárásai, függvényei.
- *HomeroBeallUnit.pas*
Egy hőmérő berendezés beállításának formja.
- *HomeroLancUnit.pas*
Hőmérőket tartalmazó lánc eljárásai, függvényei.
- *HozzaadUnit.pas*
Berendezések egymáshoz való rendelése form eljárásait, függvényeit tartalmazza. Form-hoz tartozó unit.
- *InditUnit.pas*
Az InditasDlg form eljárásai. Épület betöltésekor hívódik meg. Form-hoz tartozó unit.

- *KapcsBerendLancUnit.pas*
Az egymáshoz tartozó berendezéseket tartalmazó lánc eljárásai, függvényei.
- *KodolasUnit.pas*
Üzenetek kódolása és dekódolása.
- *KuldThreadUnit.pas*
Kommunikációhoz szükséges eljárások, függvények. Az üzenetek küldéséért felelős szál eljárásai, függvényei.
- *MegnyitasUnit.pas*
Megnyitás form eljárásait, függvényeit tartalmazza. Form-hoz tartozó unit.
- *NaploDlgUnit.pas*
Naplózó form. Form-hoz tartozó unit.
- *NyugtaDlgUnit.pas*
Riasztást nyugtázó form. Form-hoz tartozó unit.
- *ObjektumUnit.pas*
Objektum lánc eljárásai, függvényei.
- *PenUnit.pas*
A tollal kapcsolatos eljárások.
- *PontLancUnit.pas*
Pontokat tartalmazó lánc eljárásai, függvényei.
- *RajzolasUnit.pas*
Rajzolással kapcsolatos eljárások, függvények.
- *ResolutionUnit.pas*
Monitor felbontásának beállítása.
- *RiasztoBeallUnit.pas*
Egy riasztó beállításának formja. Form-hoz tartozó unit.
- *RiasztoLancUnit.pas*
Riasztókat tartalmazó lánc eljárásai, függvényei.
- *TuzjelzoBeallUnit.pas*
Egy tűzjelző beállításának formja. Form-hoz tartozó unit.
- *TuzjelzoLancUnit.pas*
Tűzjelzőket tartalmazó lánc eljárásai, függvényei.
- *UtilUnit.pas*
Általános eljárások, függvények.
- *VegBerendLancUnit.pas*
Végberendezéseket tartalmazó lánc eljárásai, függvényei.
- *VezerloUnit.pas*
A Vezérlő program főformjának unit-ja. Form-hoz tartozó unit.
- *ZoomUnit.pas*
Nagyítás-kicsinyítés függvényei.

Globális konstansok, változók

Ezek a konstansok és változók a program egész működése során elérhetők.

(*HangThreadUnit.pas*):

- Hang tulajdonságai:

Freq = 200 - frekvencia

Msecs = 500 - milliszekundum

(*VezerloUnit.pas*):

- Napló tulajdonságai:

NAPLOKDIR = 'Naplo' - naplófájlok könyvtára

NAPLOKIT = '.npl' - naplófájl kiterjesztése

Típus- és osztályleírások

(*BeallitUnit.pas*):

- *TRegiErtekek* = *Object*

Régi beállítási értékek tárolása.

(*DlgUnit.pas*):

- *TOsszesLista_Dlg* = *Class(TPersistent)*

Az összes láncot tartalmazó típus.

(*HangThreadUnit.pas*):

- *THangThread* = *Class(TThread)*

Hang kezeléséért felelős szál.

(*HozzaadUnit.pas*):

- *TCelElem* = *Object*

Egy hozzáadandó berendezést tartalmaz.

- *TCelLanc* = *Object*

Hozzáadandó berendezések lánca

- *TCelLista* = *Class(TPersistent)*

Hozzáadandó berendezések lánc típusa

(*VezerloUnit.pas*):

- *TPictIndexTomb = Array Of Integer*
A táblázatban szereplő ikonokat tartja nyilván.

Összefoglaló

Tesztelés

A dolgozathoz elkészítettem egy kitalált épület alaprajzát *Lakás* néven, amit a Tervező programmal rajzoltam és feltöltöttem berendezésekkel. A rajzolás során, a program nyújtotta rajzolási lehetőségeket, műveleteket kipróbáltam és felhasználtam. A Tesztelő és a Vezérlő programot ezzel a *Lakás* nevű épülettel teszteltem. Mind a két programban, az elérhető funkciókat leteszteltem.

A *Lakás* épület alaprajzát és berendezéseit tartalmazó adatbázisok megtalálhatók a dolgozathoz mellékelt CD-n az '\Alaprajzok\Lakas' alkönyvtárban.

Fő feladatként a programok közötti kommunikáció tesztelését tűztem ki célul, mivel a biztonságos adatforgalom elengedhetetlen feltétele a berendezések vezérlésének.

Fejlesztési lehetőségek

Köztudott, hogy egy programot soha sem lehet befejezni, csak a fejlesztését abbahagyni. Így van ez ezzel a dolgozattal is. Rengeteg funkciót lehet a későbbiekben hozzáadni. Például:

1. Nyomtatás bővítése

- alaprajz nyomtatása
- berendezések listájának és tulajdonságainak nyomtatása

2. Időzítő bevezetése

Berendezések működésének időponthoz kötése. Például a kerti öntöző minden nap 17-től 18 óráig lépjen működésbe.

A fejlesztés során próbáltam a programot modulokból felépíteni, hogy a későbbiekben a fejlesztéseket, a változtatásokat egyszerűbben és biztonságosabban lehessen végrehajtani.

MELLÉKLET

[22:20:37] - A rendszer elindult
[22:20:37] - A(z) Lakás nevű épület betöltve.
[22:20:58] - Kapcsolatfelvétel a COM1 porton...
[22:20:58] - Kapcsolat létrejött a COM1 porton
(1) Fürdő-hőmérő - beállítások elküldve.
[22:20:59] - (14) Vendég-tűzjelző állapota: Normál
(3) Konyha-hőmérő - beállítások elküldve.
(6) Nappali-hőmérő - beállítások elküldve.
(11) Dolgozó-hőmérő - beállítások elküldve.
[22:20:59] - (13) Dolgozó-riasztó állapota: Normál
[22:20:59] - (12) Dolgozó-radiátor állapota: Bekapcsolva
[22:20:59] - (11) Dolgozó-hőmérő értéke: 17
[22:20:59] - (10) Térvilágítás állapota: Ki
[22:21:00] - (9) Nappali-radiátor állapota: Kikapcsolva
[22:21:00] - (8) Nappali-tűzjelző állapota: Normál
[22:21:00] - (7) Nappali-riasztó állapota: Normál
[22:21:00] - (6) Nappali-hőmérő értéke: 25
[22:21:00] - (5) Konyha-radiátor állapota: Kikapcsolva
[22:21:00] - (4) Konyha-tűzjelző állapota: Normál
[22:21:00] - (3) Konyha-hőmérő értéke: 27
[22:21:00] - (2) Fürdő-radiátor állapota: Bekapcsolva
[22:21:00] - (1) Fürdő-hőmérő értéke: 20
[22:21:51] - (11) Dolgozó-hőmérő értéke: 18
[22:21:51] - (12) Dolgozó-radiátor bekapcsolása...
HIBA: Adatvételei hiba. Ismételt adatkérés...
[22:21:52] - (3) Konyha-hőmérő értéke: 26
[22:21:52] - (6) Nappali-hőmérő értéke: 24
[22:21:52] - (1) Fürdő-hőmérő értéke: 21
[22:21:52] - (12) Dolgozó-radiátor állapota: Kikapcsolva
[22:21:52] - (2) Fürdő-radiátor bekapcsolása...
[22:21:52] - (2) Fürdő-radiátor állapota: Kikapcsolva
[22:21:54] - (11) Dolgozó-hőmérő értéke: 17
[22:21:54] - (6) Nappali-hőmérő értéke: 23
[22:21:54] - (12) Dolgozó-radiátor kikapcsolása...
[22:21:54] - (3) Konyha-hőmérő értéke: 25
[22:21:54] - (1) Fürdő-hőmérő értéke: 20
[22:21:54] - (2) Fürdő-radiátor kikapcsolása...
[22:21:54] - (12) Dolgozó-radiátor állapota: Bekapcsolva
[22:21:54] - (2) Fürdő-radiátor állapota: Bekapcsolva
[22:21:56] - (11) Dolgozó-hőmérő értéke: 18
[22:21:56] - (6) Nappali-hőmérő értéke: 22
[22:21:56] - (12) Dolgozó-radiátor bekapcsolása...
[22:21:56] - (3) Konyha-hőmérő értéke: 24
[22:21:56] - (9) Nappali-radiátor kikapcsolása...
[22:21:56] - (1) Fürdő-hőmérő értéke: 21
[22:21:56] - (12) Dolgozó-radiátor állapota: Kikapcsolva
[22:21:56] - (2) Fürdő-radiátor bekapcsolása...
[22:21:56] - (9) Nappali-radiátor állapota: Bekapcsolva
[22:21:56] - (2) Fürdő-radiátor állapota: Kikapcsolva
[22:21:58] - (11) Dolgozó-hőmérő értéke: 17
[22:21:58] - (12) Dolgozó-radiátor kikapcsolása...
[22:21:58] - (6) Nappali-hőmérő értéke: 23
[22:21:58] - (3) Konyha-hőmérő értéke: 23
[22:21:58] - (9) Nappali-radiátor bekapcsolása...
[22:21:58] - (1) Fürdő-hőmérő értéke: 20
[22:21:58] - (12) Dolgozó-radiátor állapota: Bekapcsolva
[22:21:58] - (2) Fürdő-radiátor kikapcsolása...
[22:21:58] - (9) Nappali-radiátor állapota: Kikapcsolva
[22:21:58] - (2) Fürdő-radiátor állapota: Bekapcsolva
[22:22:00] - (11) Dolgozó-hőmérő értéke: 18
[22:22:00] - (12) Dolgozó-radiátor bekapcsolása...

[22:22:00] - (6) Nappali-hőmérő értéke: 22
[22:22:00] - (9) Nappali-radiátor kikapcsolása...
[22:22:00] - (3) Konyha-hőmérő értéke: 22
[22:22:00] - (1) Fürdő-hőmérő értéke: 21
[22:22:00] - (12) Dolgozó-radiátor állapota: Kikapcsolva
[22:22:00] - (2) Fürdő-radiátor bekapcsolása...
[22:22:00] - (9) Nappali-radiátor állapota: Bekapcsolva
[22:22:00] - (2) Fürdő-radiátor állapota: Kikapcsolva
[22:22:02] - (11) Dolgozó-hőmérő értéke: 17
[22:22:02] - (6) Nappali-hőmérő értéke: 23
[22:22:02] - (12) Dolgozó-radiátor kikapcsolása...
[22:22:02] - (3) Konyha-hőmérő értéke: 21
[22:22:02] - (9) Nappali-radiátor bekapcsolása...
[22:22:02] - (1) Fürdő-hőmérő értéke: 20
[22:22:02] - (2) Fürdő-radiátor kikapcsolása...
[22:22:02] - (12) Dolgozó-radiátor állapota: Bekapcsolva
[22:22:02] - (9) Nappali-radiátor állapota: Kikapcsolva
[22:22:02] - (2) Fürdő-radiátor állapota: Bekapcsolva
[22:22:04] - (11) Dolgozó-hőmérő értéke: 18
[22:22:04] - (12) Dolgozó-radiátor bekapcsolása...
[22:22:04] - (6) Nappali-hőmérő értéke: 22
[22:22:04] - (3) Konyha-hőmérő értéke: 20
[22:22:04] - (9) Nappali-radiátor kikapcsolása...
[22:22:04] - (1) Fürdő-hőmérő értéke: 21
[22:22:04] - (12) Dolgozó-radiátor állapota: Kikapcsolva
[22:22:04] - (2) Fürdő-radiátor bekapcsolása...
[22:22:04] - (9) Nappali-radiátor állapota: Bekapcsolva
[22:22:04] - (2) Fürdő-radiátor állapota: Kikapcsolva
[22:22:06] - (11) Dolgozó-hőmérő értéke: 17
[22:22:06] - (6) Nappali-hőmérő értéke: 23
[22:22:06] - (12) Dolgozó-radiátor kikapcsolása...
[22:22:06] - (3) Konyha-hőmérő értéke: 19
[22:22:06] - (9) Nappali-radiátor bekapcsolása...
[22:22:06] - (1) Fürdő-hőmérő értéke: 20
[22:22:06] - (12) Dolgozó-radiátor állapota: Bekapcsolva
[22:22:06] - (2) Fürdő-radiátor kikapcsolása...
[22:22:06] - (9) Nappali-radiátor állapota: Kikapcsolva
[22:22:06] - (2) Fürdő-radiátor állapota: Bekapcsolva
[22:22:08] - (11) Dolgozó-hőmérő értéke: 18
[22:22:08] - (6) Nappali-hőmérő értéke: 22
[22:22:08] - (12) Dolgozó-radiátor bekapcsolása...
[22:22:08] - (9) Nappali-radiátor kikapcsolása...
[22:22:08] - (3) Konyha-hőmérő értéke: 18
[22:22:08] - (1) Fürdő-hőmérő értéke: 21
[22:22:08] - (12) Dolgozó-radiátor állapota: Kikapcsolva
[22:22:08] - (2) Fürdő-radiátor bekapcsolása...
[22:22:08] - (9) Nappali-radiátor állapota: Bekapcsolva
[22:22:08] - (2) Fürdő-radiátor állapota: Kikapcsolva
[22:22:10] - (11) Dolgozó-hőmérő értéke: 17
[22:22:10] - (6) Nappali-hőmérő értéke: 23
[22:22:10] - (12) Dolgozó-radiátor kikapcsolása...
[22:22:10] - (9) Nappali-radiátor bekapcsolása...
[22:22:10] - (3) Konyha-hőmérő értéke: 17
[22:22:10] - (1) Fürdő-hőmérő értéke: 20
[22:22:10] - (5) Konyha-radiátor kikapcsolása...
[22:22:10] - (2) Fürdő-radiátor kikapcsolása...
[22:22:10] - (12) Dolgozó-radiátor állapota: Bekapcsolva
[22:22:10] - (9) Nappali-radiátor állapota: Kikapcsolva
[22:22:10] - (5) Konyha-radiátor állapota: Bekapcsolva
[22:22:11] - (2) Fürdő-radiátor állapota: Bekapcsolva
[22:22:12] - (7) Nappali-riasztó állapota: Riasztás

- [22:22:12] - (10) Térvilágítás állapota: Be
[22:22:12] - (11) Dolgozó-hőmérő értéke: 18
[22:22:12] - (12) Dolgozó-radiátor bekapcsolása...
[22:22:12] - (6) Nappali-hőmérő értéke: 22
[22:22:12] - (9) Nappali-radiátor kikapcsolása...
[22:22:12] - (3) Konyha-hőmérő értéke: 18
[22:22:12] - (5) Konyha-radiátor bekapcsolása...
[22:22:12] - (1) Fürdő-hőmérő értéke: 21
[22:22:12] - (2) Fürdő-radiátor bekapcsolása...
[22:22:12] - (12) Dolgozó-radiátor állapota: Kikapcsolva
[22:22:12] - (9) Nappali-radiátor állapota: Bekapcsolva
[22:22:13] - (5) Konyha-radiátor állapota: Kikapcsolva
[22:22:13] - (2) Fürdő-radiátor állapota: Kikapcsolva
[22:22:14] - (11) Dolgozó-hőmérő értéke: 17
[22:22:14] - (12) Dolgozó-radiátor kikapcsolása...
[22:22:14] - (6) Nappali-hőmérő értéke: 23
[22:22:14] - (9) Nappali-radiátor bekapcsolása...
[22:22:14] - (3) Konyha-hőmérő értéke: 17
[22:22:14] - (5) Konyha-radiátor kikapcsolása...
[22:22:14] - (1) Fürdő-hőmérő értéke: 20
[22:22:14] - (2) Fürdő-radiátor kikapcsolása...
[22:22:14] - (12) Dolgozó-radiátor állapota: Bekapcsolva
[22:22:14] - (9) Nappali-radiátor állapota: Kikapcsolva
[22:22:14] - (5) Konyha-radiátor állapota: Bekapcsolva
[22:22:14] - (2) Fürdő-radiátor állapota: Bekapcsolva
[22:22:16] - (11) Dolgozó-hőmérő értéke: 18
[22:22:16] - (12) Dolgozó-radiátor bekapcsolása...
[22:22:16] - (6) Nappali-hőmérő értéke: 22
[22:22:16] - (9) Nappali-radiátor kikapcsolása...
[22:22:16] - (3) Konyha-hőmérő értéke: 18
[22:22:16] - (5) Konyha-radiátor bekapcsolása...
[22:22:16] - (1) Fürdő-hőmérő értéke: 21
[22:22:16] - (2) Fürdő-radiátor bekapcsolása...
[22:22:16] - (12) Dolgozó-radiátor állapota: Kikapcsolva
[22:22:16] - (9) Nappali-radiátor állapota: Bekapcsolva
[22:22:17] - (5) Konyha-radiátor állapota: Kikapcsolva
[22:22:17] - (2) Fürdő-radiátor állapota: Kikapcsolva
[22:22:18] - (11) Dolgozó-hőmérő értéke: 17
[22:22:18] - (12) Dolgozó-radiátor kikapcsolása...
HIBA: Adatvételi hiba. Ismételt adatkérés...
[22:22:18] - (3) Konyha-hőmérő értéke: 17
[22:22:18] - (5) Konyha-radiátor kikapcsolása...
[22:22:18] - (6) Nappali-hőmérő értéke: 23
(7) Nappali-riasztó: Kikapcsolása...
[22:22:18] - (1) Fürdő-hőmérő értéke: 20
[22:22:18] - (9) Nappali-radiátor bekapcsolása...
[22:22:18] - (2) Fürdő-radiátor kikapcsolása...
[22:22:19] - (12) Dolgozó-radiátor állapota: Bekapcsolva
[22:22:19] - (5) Konyha-radiátor állapota: Bekapcsolva
[22:22:19] - (7) Nappali-riasztó állapota: Normál
[22:22:19] - (9) Nappali-radiátor állapota: Kikapcsolva
[22:22:19] - (2) Fürdő-radiátor állapota: Bekapcsolva
[22:22:19] - (10) Térvilágítás állapota: Ki
[22:22:20] - (11) Dolgozó-hőmérő értéke: 18
[22:22:20] - (6) Nappali-hőmérő értéke: 22
[22:22:20] - (12) Dolgozó-radiátor bekapcsolása...
[22:22:20] - (3) Konyha-hőmérő értéke: 18
[22:22:20] - (9) Nappali-radiátor kikapcsolása...
[22:22:20] - (1) Fürdő-hőmérő értéke: 21
[22:22:20] - (5) Konyha-radiátor bekapcsolása...
[22:22:20] - (2) Fürdő-radiátor bekapcsolása...

[22:22:20] - (12) Dolgozó-radiátor állapota: Kikapcsolva
[22:22:20] - (9) Nappali-radiátor állapota: Bekapcsolva
[22:22:20] - (5) Konyha-radiátor állapota: Kikapcsolva
[22:22:21] - (2) Fürdő-radiátor állapota: Kikapcsolva
[22:22:21] - (11) Dolgozó-hőmérő értéke: 17
[22:22:21] - (12) Dolgozó-radiátor kikapcsolása...
[22:22:22] - (6) Nappali-hőmérő értéke: 23
[22:22:22] - (9) Nappali-radiátor bekapcsolása...
[22:22:22] - (3) Konyha-hőmérő értéke: 17
[22:22:22] - (5) Konyha-radiátor kikapcsolása...
[22:22:22] - (1) Fürdő-hőmérő értéke: 20
[22:22:22] - (2) Fürdő-radiátor kikapcsolása...
[22:22:22] - (12) Dolgozó-radiátor állapota: Bekapcsolva
[22:22:22] - (9) Nappali-radiátor állapota: Kikapcsolva
[22:22:23] - (5) Konyha-radiátor állapota: Bekapcsolva
[22:22:23] - (2) Fürdő-radiátor állapota: Bekapcsolva
[22:22:24] - (11) Dolgozó-hőmérő értéke: 18
[22:22:24] - (12) Dolgozó-radiátor bekapcsolása...
[22:22:24] - (6) Nappali-hőmérő értéke: 22
[22:22:24] - (9) Nappali-radiátor kikapcsolása...
[22:22:24] - (3) Konyha-hőmérő értéke: 18
[22:22:24] - (5) Konyha-radiátor bekapcsolása...
[22:22:24] - (1) Fürdő-hőmérő értéke: 21
[22:22:24] - (2) Fürdő-radiátor bekapcsolása...
[22:22:24] - (12) Dolgozó-radiátor állapota: Kikapcsolva
[22:22:25] - (9) Nappali-radiátor állapota: Bekapcsolva
[22:22:25] - (5) Konyha-radiátor állapota: Kikapcsolva
[22:22:25] - (2) Fürdő-radiátor állapota: Kikapcsolva
[22:22:46] - Kapcsolat lezárása...
[22:22:47] - Kapcsolat lezárvva a COM1 porton
[22:22:59] - Kilépés a programból.

FÜGGELÉK

<i>TAlakzatTip</i>	79	<i>TRegiErtekek</i>	131
<i>TAbnakAlakzat</i>	80	<i>TRegiPenStyle</i>	84
<i>TAbnakSzin</i>	83	<i>TResolution</i>	84
<i>TAdatUzenetTomb</i>	125	<i>TRiasztoElem</i>	126
<i>TAjtoAlakzat</i>	80	<i>TRiasztoLanc</i>	126
<i>TAlakzatElem</i>	80	<i>TRiasztoLista</i>	126
<i>TAlakzatLanc</i>	80	<i>TSarokKoord</i>	120
<i>TAranyok</i>	121	<i>TTavolsag</i>	120
<i>TBerendArray</i>	78	<i>TTTextAttrib</i>	125
<i>TBerendezesLanc</i>	78	<i>TTuzjelzoElem</i>	126
<i>TBerendezesLista</i>	79	<i>TTuzjelzoLanc</i>	126
<i>TBerendezesTip</i>	73	<i>TTuzjelzoLista</i>	126
<i>TCelElem</i>	131	<i>TUzenetElem</i>	125
<i>TCelLanc</i>	131	<i>TUzenetLanc</i>	125
<i>TCelLista</i>	131	<i>TUzenetLista</i>	125
<i>TDevTul</i>	84	<i>TVegBerendElem</i>	126
<i>TDrawingTool</i>	79	<i>TVegBerendLanc</i>	126
<i>TEgerPos</i>	84	<i>TVegBerendLista</i>	126
<i>TEllipszisAlakzat</i>	80	<i>TVezerloAllapot</i>	121
<i>TEpuletEgyseg</i>	81	<i>TVonalAlakzat</i>	80
<i>TEpuletLanc</i>	81	<i>TZoom</i>	84
<i>TEpuletLista</i>	82		
<i>TEpuletTul</i>	84		
<i>THangThread</i>	131		
<i>THomeroElem</i>	125		
<i>THomeroLanc</i>	125		
<i>THomeroLista</i>	125		
<i>TKapcsBerendElem</i>	125		
<i>TKapcsBerendLanc</i>	125		
<i>TKapcsBerendLista</i>	125		
<i>TKeret</i>	120		
<i>TKeretClass</i>	120		
<i>TKijelolNegyzet</i>	120		
<i>TKuldThread</i>	125		
<i>TLancElem</i>	78		
<i>TMutatoArray</i>	78		
<i>TNegyzetAlakzat</i>	79		
<i>TObjektum</i>	83		
<i>TObjektumArray</i>	83		
<i>TObjektumLanc</i>	84		
<i>TObjektumLista</i>	84		
<i>TObjektumTip</i>	84		
<i>TOsszesLista_Dlg</i>	131		
<i>TPictIndexTomb</i>	131		
<i>TPontElemTip</i>	126		
<i>TPontKoordTomb</i>	83		
<i>TPontLancTip</i>	126		
<i>TPontLista</i>	126		
<i>TPortAllapot</i>	125		