

AZ ALGORITMUSRÓL

(bevezetés a programozáshoz)

A bemutató készítéséhez felhasznált tartalmi forrás: (Sz)ámítástechnika 1.4, Budapest, Kvassay Jenő Műszaki Szakközépiskola és TIKETT Nyomdaipari Kft. 1994.
[írták és szerkesztették Hubert Tibor, Mód Géza, Pfeiffer Ferenc a Kvassay Jenő MSZKI tanárai, korábbi és az 1993-94. tanév diákjai]

A 6. és 8. diák piktogram képei Dancsó Tünde-Végh András: Számítástechnika 10-11 éveseknek (Műszaki Könyvkiadó, Budapest, 1997) tankönyvből származnak

Az algoritmus fogalma I.

Mielőtt meghatároznánk az *algoritmus* fogalmát, tisztáznunk kell néhány félreértést ami abból fakadhat, hogy az *algoritmus* kifejezés olyan idegen szó, ami köznapi nyelvünknek nem a sajátja.

Először tehát nézzük meg azt, *mi nem jellemző az algoritmusra?* Remélhetően ez is közelebb visz a fogalom pontos megértéséhez.

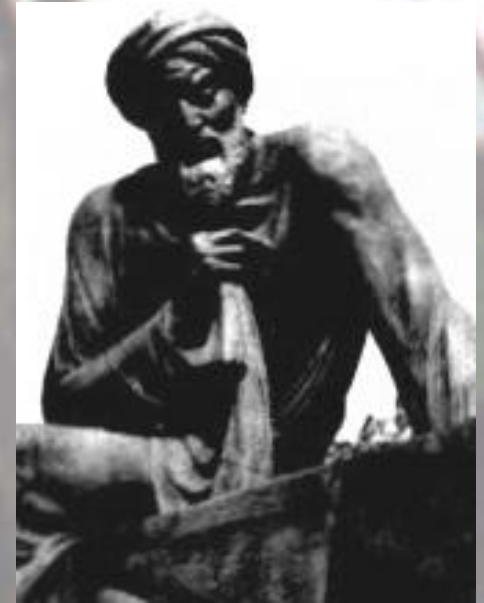
Mi *nem* jellemző az algoritmusra?

- Nincs kapcsolatban az algákkal (moszatokkal)
- Nincs semmi köze a szív ritmusos mozgásához
- Semmi köze a biológiai teljesítőképesség jellemzésére használt bioritmus fogalmához
- Nincs kapcsolatban a nappalok és éjszakák változásának ritmusával
- És nincs köze a zenéből és táncból ismert ritmus kifejezéshez sem



Egy kis szaknyelv történet...

- Hol volt, hol nem volt, élt egyszer - még hozzá Kr.u. a IX. században - egy arab matematikus, akinek a neve Muhammed ibu Muza *al-Khvarizmi* volt
- 825-ben könyvet írt a fontosabb számolási módszerekről és eljárásokról
- Az Ő nevéből származtatják az „*algoritmus*” és az „*algebra*” szavakat (lásd nevének *vastagon és dőlten szedett* részét!)



Az algoritmus fogalma II.

Avagy: Mi az algoritmus?

- Az algoritmus elemi lépésekből álló utasításokat tartalmaz
- Tevékenységek műveletek olyan sorozata, amely biztosan elvezet az adott feladat, probléma megoldásához
- Tartalmazza a megoldás pontos leírását, a megoldáshoz vezető műveletek milyenségét, sorrendjét
- **ÖSSZEGEZVE:** *Az algoritmus egy feladattípus véges számú lépésre bontott megoldása*

Ez utóbbit kell megtanulnod!

Lássunk egy példát!

A telefonálás algoritmusa...



Ez ábrás algoritmus leírás

- Emeld fel a helyéről a telefonkagylót!
- Helyezd be a nyílásba a telefonkártyát!
- Várd meg a tárcsahang bűgását!
- Üsd be a számot!
- Ha a kicsengetés után fölveszik a telefont, akkor kezdj el beszélni!
- Tedd le a telefont!
- Vedd ki a kártyát!

A jobbra olvasható pedig mondatokkal való algoritmus leírás

Az algoritmusok leírása

Az algoritmus leírás eszközei a következők:

- mondatokkal való (hosszadalmas és pontatlan; ezt láthattad az előző dián)
- ábrás, rajzos leírás (piktogram, folyamatábra, struktogram)
- mondatszerű leírás (rövidített formalizált nyelv)
- kódolás (az algoritmus leírása egy adott programozási nyelv szabályai szerint [pl. **Blockly**, Logo, **FreePascal**, Visual Basic, Delphi, **Lazarus**, Java, **AppInventor**, stb.])

1. Algoritmus leírása mondatokkal

Algoritmus eleje

- Emeld fel a helyéről a telefonkagylót!
- Helyezd be a nyílásba a telefonkártyát!
- Várd meg a búgó tárcsahangot!
- Üsd be a számot!
- Ha a kicsengetés után fölveszik a telefont, kezdj el beszélni!
- Tedd le a telefonkagylót!
- Vedd ki a kártyát!



Ezek piktogramok, nem mondatok

A mondatokkal való algoritmus leírás
hosszadalmas, pontatlan

Algoritmus vége

2. A folyamatábra

A teakészítés algoritmusának folyamatábrája

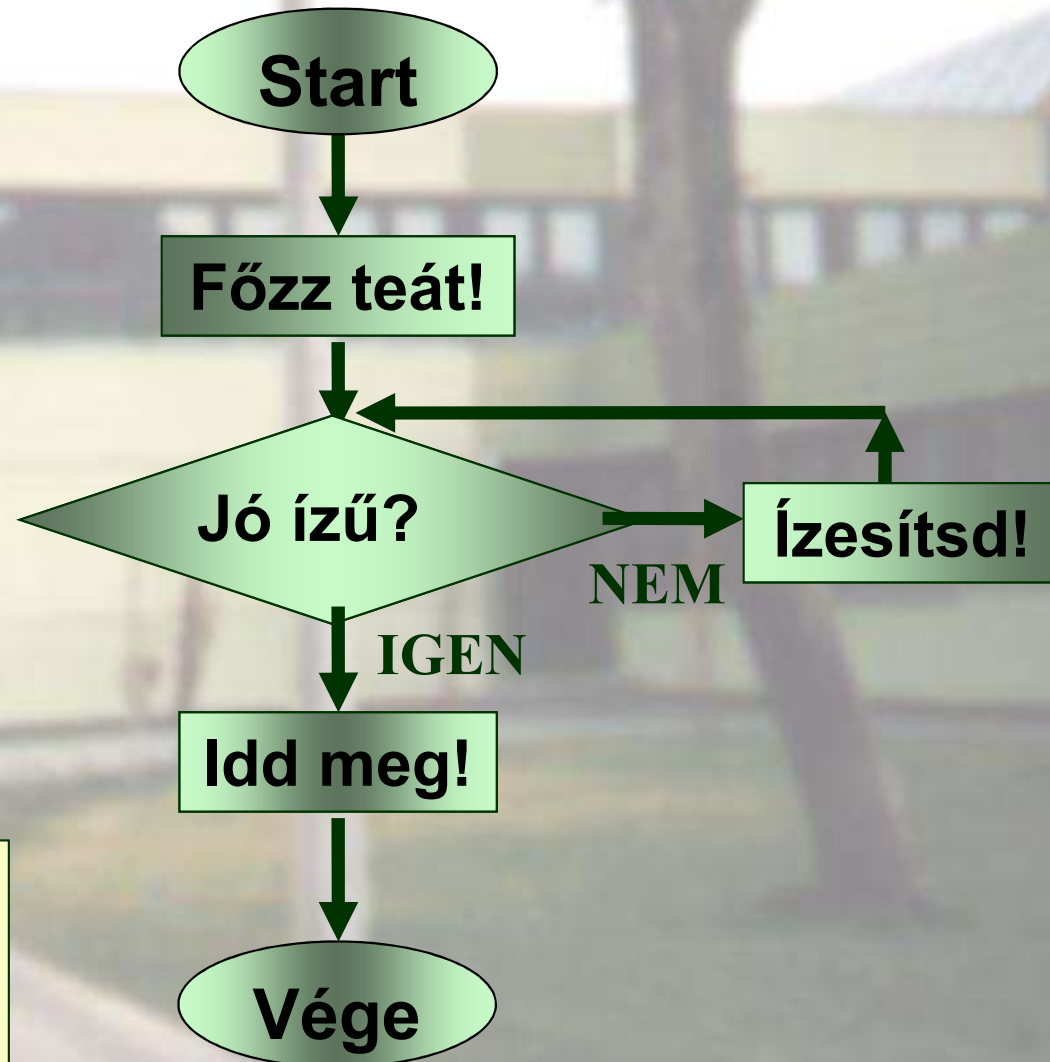
Jelmagyarázat:

ellipszis: algoritmus eleje, vége

téglalap: utasítás

rombusz: döntés, elágazás

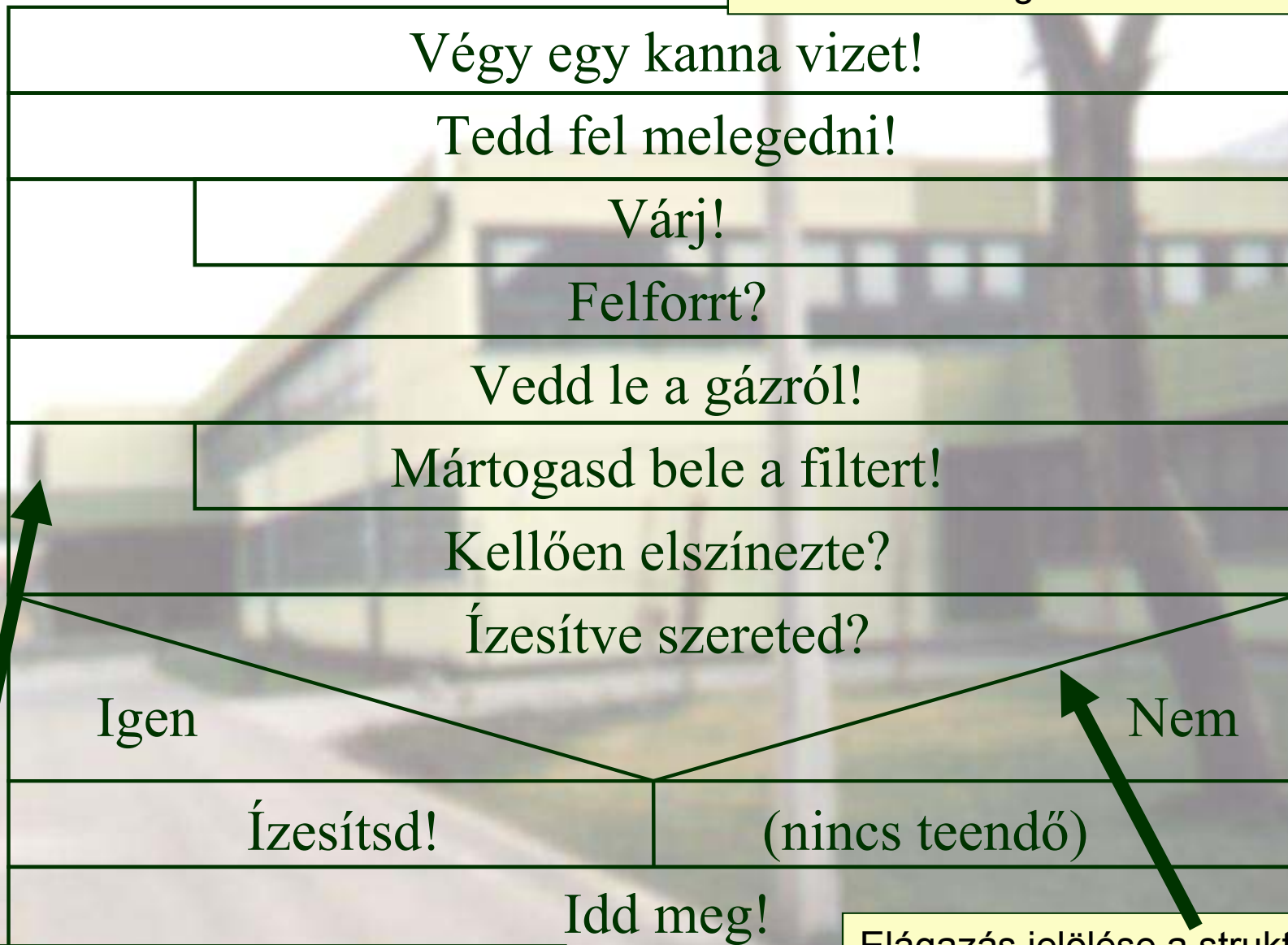
paralelogramma: input, output adatok (a jelzett folyamatábra ilyet nem tartalmaz)



A folyamatábrát főleg a BASIC alapú programozás előkészítésénél lehet igen jól használni

3.1. A struktogram* (*szerkezeti diagram)

A teakészítés algoritmusának struktogramja



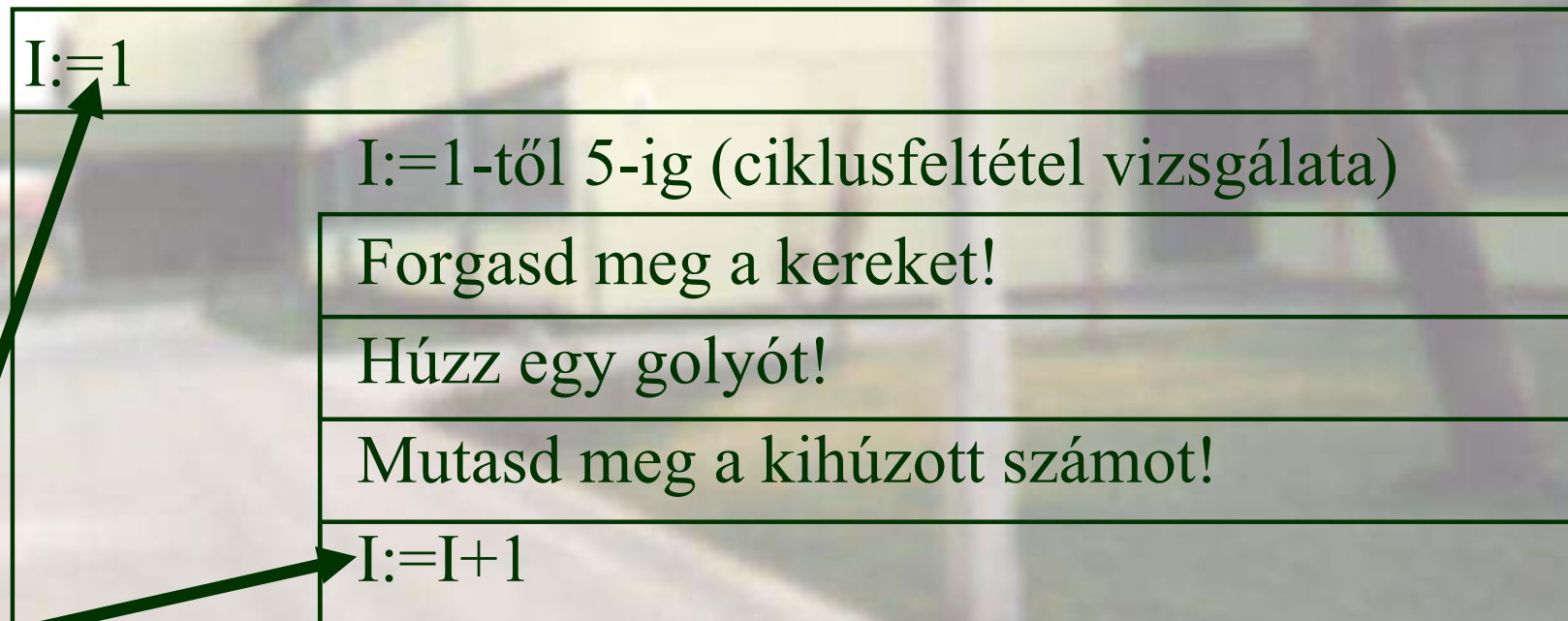
Ismétlődés - **ciklus** - jelölése az „L” alak.
A ciklusnak ez a típusa hátultesztelő

Elágazás jelölése a struktogramban

3.2. Számláló ciklus ábrázolása a struktrogram segítségével

Az algoritmusos ismétlődéseket ciklusoknak hívjuk. Típusai:

- *számláló ciklus*
- **hátról tesztelő ciklus**
- **előlről tesztelő ciklus**



Ez nem matematikai egyenlőség,
hanem értékadás

A lottóhúzás algoritmusában számláló ciklust tartalmaz

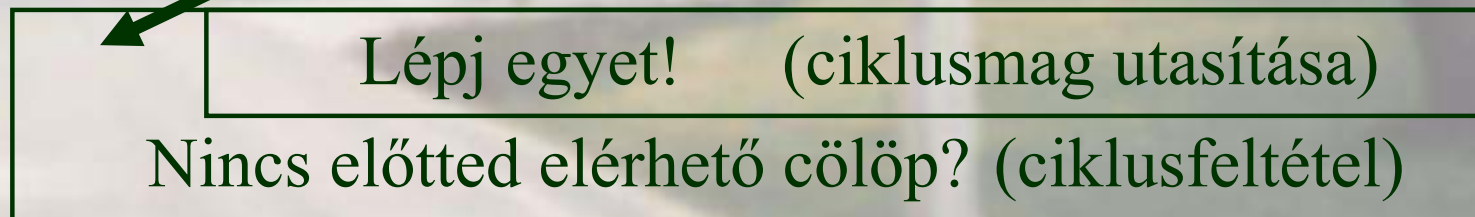
3.3. Hátraltesztelő ciklus ábrázolása a struktogrammal

Az algoritmusos ismétlődéseket ciklusoknak hívjuk. Típusai:

- számláló ciklus
- *hátraltesztelő ciklus*
- előltesztelő ciklus

Piócáktól hemzsegő csatornán kell átkelned, de se híd, se csónak nincs a közelben. Viszont cölöpök állnak ki a vízből egymástól lépésnyi távolságra, ezeken átsétálhatsz!

Hátraltesztelő ciklus struktogramja.
Figyeld meg jellegzetes „L” alakját!



Ezzel az algoritmussal szerencsés esetben a túlpartra juthatsz. Viszont, ha a legelső cölöp hiányzik... Akkor bizony a piócák járnak jól...

3.4. Elől tesztelő ciklus ábrázolása a struktogrammal

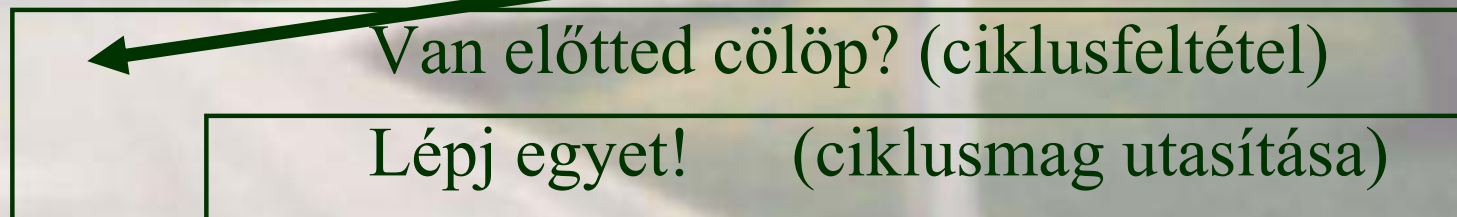
Az algoritmusos ismétlődéseket ciklusoknak hívjuk. Típusai:

- számláló ciklus
- hátul tesztelő ciklus
- *elől tesztelő ciklus*

Mi lenne tehát a megoldás?

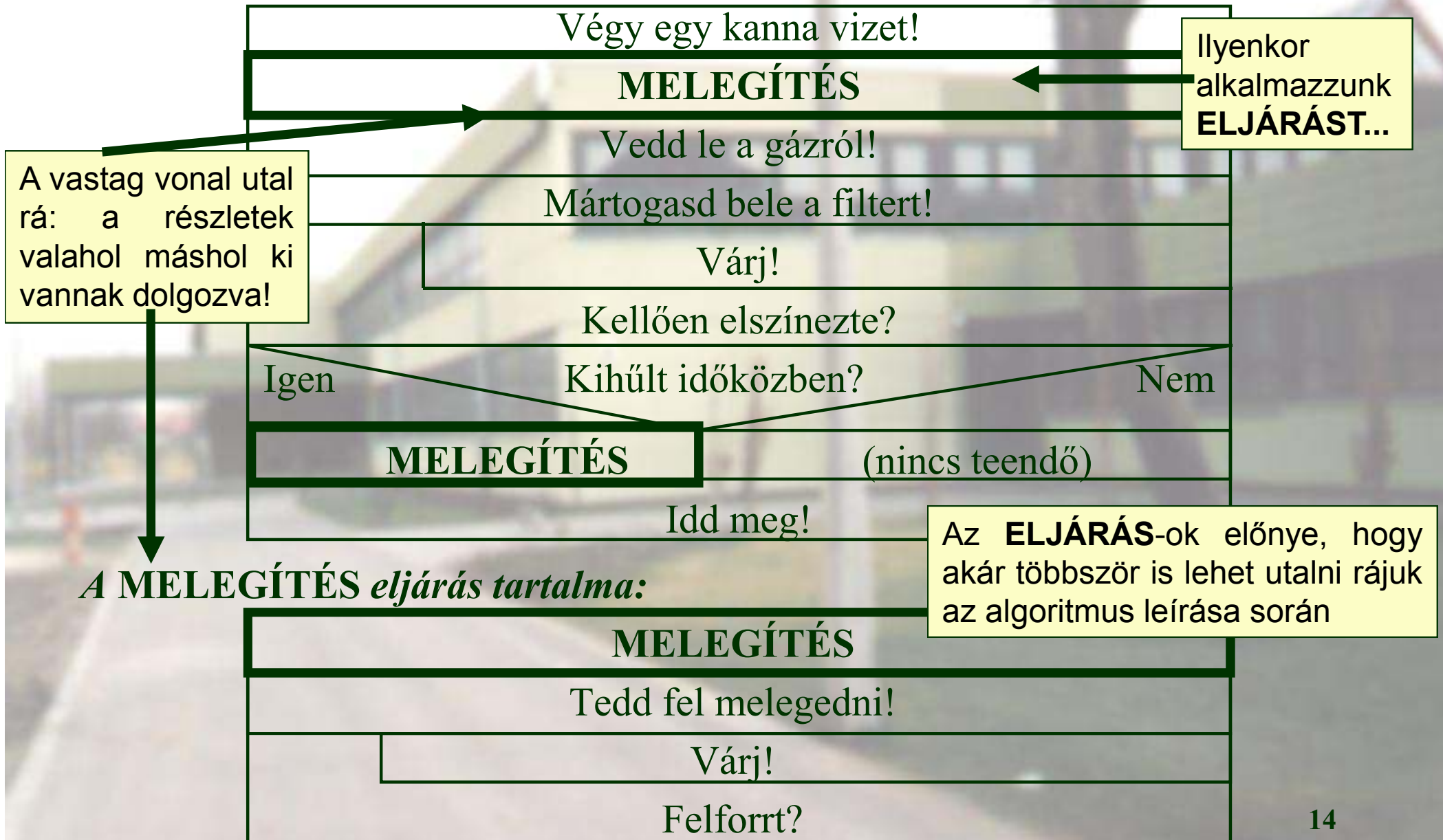
Az ilyen és hasonló esetekben (piócák, kezdő cölöp hiányzik, stb...) javaslom, hogy előbb vizsgálódj és csak azután cselekedj! Vagyis struktogram ábrázolással leírva:

Elől tesztelő ciklus struktogramja.
Figyeld meg fejre állított „L” alakját!



Elől tesztelő ciklusnál a feltételt mindig úgy fogalmazd meg, hogy nemleges válasz esetén kelljen a cikusból kilépni, tehát a ciklusba lépés feltételét add meg!

3.5. Mi a teendő, ha a struktogram áttekinthetetlenül hosszú lenne?



4.1. Mondatszerű algoritmus-leírás

- A mondatokkal való algoritmus-leírás pontatlan és hosszadalmas, felhasználhatatlan a tényleges programkészítés előtt
- A bizonyos megkötéseket tartalmazó - *formalizált nyelvet használó* - leírást *mondatszerű algoritmus-leírásnak* nevezzük
- Az így elkészített algoritmus-leírás könnyedén felhasználható a megadott programnyelvre való átíráshoz, az un. kódoláshoz

A formalizált nyelv főbb elemei:

- *Algoritmus (program eleje és vége)*
- *Értékadás*
- *Ciklusok (elől tesztelő, hátul tesztelő, számláló)*
- *Elágazás és többirányú elágazás*
- *Eljárás definiálása*
- *Adat be- vagy kivitel (INPUT, OUTPUT)*

4.2. A formalizált nyelv elemei: a program eleje és vége

**A formalizált nyelvben a program elejét és
végét így add meg:**

Algoritmus

utasítások sorozata

Algoritmus vége

4.2. A formalizált nyelv elemei: az értékadás

A formalizált nyelvben az adatok különböző típusainak - a változóknak - értéket kell adni

Értékadás:

változó:=értéke vagy **változó:=kifejezés értéke**

Például:

$x:=5$ (az x legyen egyenlő öttel)

$x:=x+1$ (x új értéke legyen x régi értéke $+1$)

$t:=a*m/2$

4.3. A formalizált nyelv elemei: ciklusok

Ha egy feladatot sokszor kell egymás után elvégezni a mondatszerű algoritmus-leírásnál (formalizált nyelvben) is **ciklust** használunk.

Fajtái:

- ***számláló ciklus***

Ciklus $x:=x1$ -től $x2$ -ig (lépésköz δ)

ciklusmag utasításai (ezek az ismétlődő utasítások)

Ciklus vége

- ***hátralétesztelő ciklus***

Ciklus

ciklusmag utasításai

amíg *feltétel*

Ciklus vége

- ***előlétesztelő ciklus***

Ciklus amíg *feltétel*

ciklusmag utasításai

Ciklus vége

A lépésköz nem kell feltétlenül megadni, ilyenkor alapértelmezetten értéke 1.

4.4. A formalizált nyelv elemei: az elágazások

- **Elágazás**

Ha *feltétel* akkor
utasítások(1)
különben
utasítások(2)
elágazás vége

- **Többirányú elágazás**

változó=érték1 esetén *utasítások(1)*
változó=érték2 esetén *utasítások(2)*
változó=érték.. esetén *utasítások(..)*
elágazás vége

A mondatszerű leírás formáinak fontos elemei az elágazások

4.5. A formalizált nyelv elemei: *eljárások definiálása*

A struktogramoknál már talákoztál az **ELJÁRÁS** fogalmával. Szerepe, hogy egy - az algoritmus során különböző esetekben - többször végrehajtandó utasítássorozatot nem kell minden alkalommal megírni. Elég egyszer, - nevet adva neki - majd a névvel hivatkozni rá a szükséges helyeken.

- **Eljárás definiálása formalizált nyelven**

Eljárás eljárásnév

utasítás(1), utasítás(2), utasítás(n)

Eljárás vége

- **Eljárás hívása**

Eljárásnév

A teakészítésnél megismert **MELEGÍTÉS** eljárás meghívása a **MELEGÍTÉS** eljárásnévnek az algoritmus megfelelő helyén való elhelyezésével lehetséges

4.6. A formalizált nyelv elemei: adat be- vagy kivitel (INPUT, OUTPUT)

Az adat be- vagy kivitel ilyen egyszerű a mondatszerű algoritmus-leírásnál:

- **Be: változók értékei**

Ki: változók vagy kifejezés értéke

- Például:

Be: a, m

$t := a * m / 2$

← Ez a sor nem INPUT vagy OUTPUT, hanem ÉRTÉKADÁS

Ki: t

5.1. Kódolás

- **Mondatszerűen megfogalmazott, vagy struktogrammal leírt algoritmusainkat valamely programnyelv segítségével tehetjük a számítógép számára is érthetővé.**
- Ezt a folyamatot nevezzük **kódolásnak**.
- A legtöbb magas szintű programozási nyelvnek sok **utasítása** van, az utasítások a legtöbbször **angol szavak vagy rövidítések**.
- Minden magas szintű programnyelvhez **szükség van fordítóprogramra**, amely le tudja fordítani gépi (bináris) kódra az utasításokat. A processzor csak ezeket az utasításokat képes „értelmezni”, végrehajtani.
- **Szükség van szövegszerkesztőre (editorra)**, amelyben megírhatjuk a programot.
- **Szükség van** - beépített - hibakeresést megkönnyítő szoftvereszközökre (**debugger**)

5.2. A kódolás eszközei a programozási nyelvek

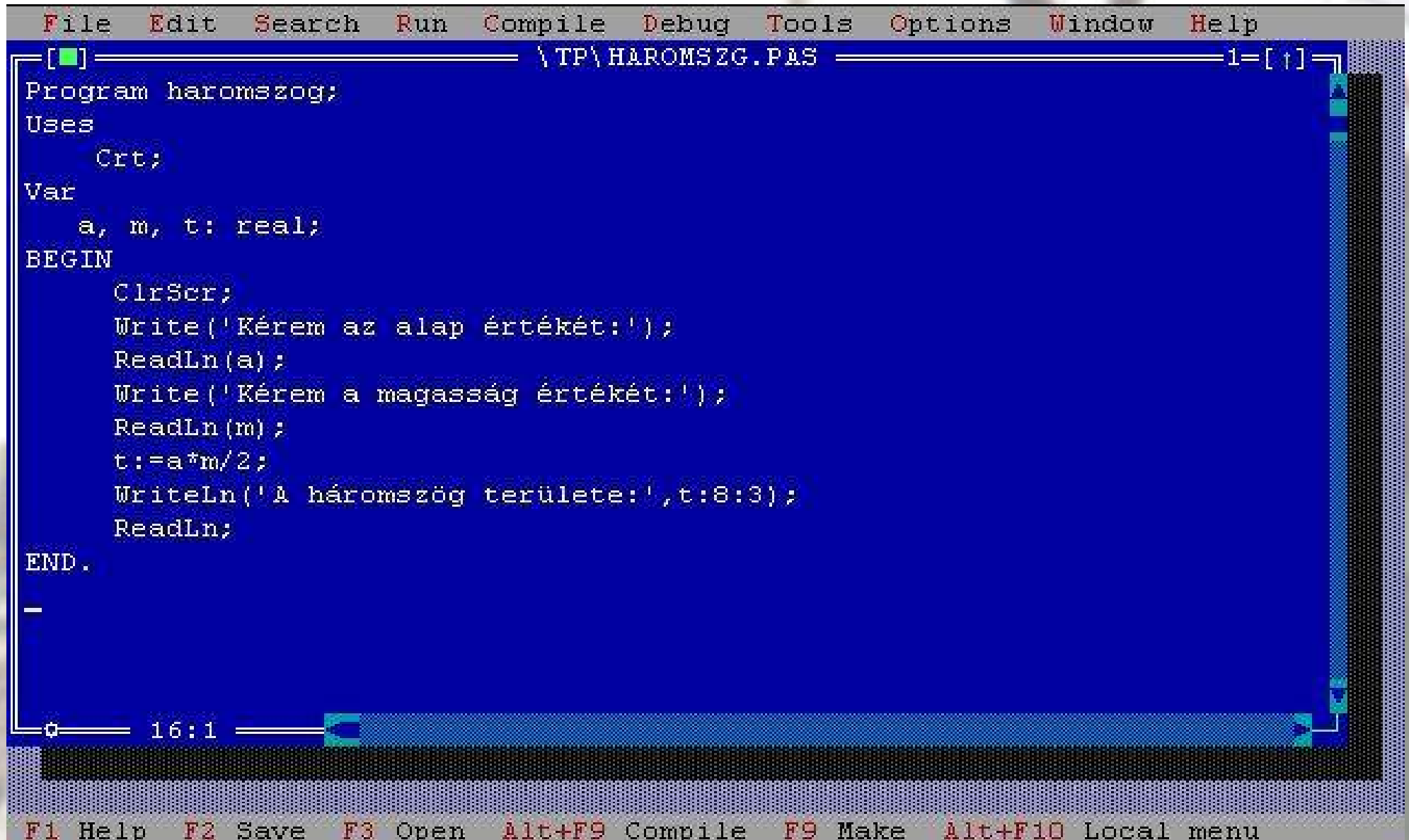
- A kódolást különböző - karakteres vagy grafikus felületű - fejlesztő rendszerek (programozási nyelvek) segítségével végezhetjük el
- Például:
 - TURBO PASCAL, FREE PASCAL
 - SCRATCH, BLOCKLY
 - LOGO, COMENIUS LOGO
 - VISUAL BASIC, DELPHI, LAZARUS
 - App Inventor, stb...

5.3. A Pascal programnyelv története...

Nincs köze a zuglói Paskál területhez, nevét Blaise Pascal híres matematikus (XVII. sz.) iránti tiszteletből kapta

- Niklaus Wirth professzor 1968-ban rakta le a Pascal nyelv alapjait azzal a céllal, hogy a nyelvvel a programozást okítsák, de a nyelv túl jól sikerült, így óriási karriert futott be...
- 1970-ben készült el első fordító programja (compiler)
- A Turbo Pascal (TP) egyike az első PC-re készült programnyelveknek, a TP 1.0 1984-ben jelent meg.
- Az **ingyenes FreePascal** alkalmas matematikai számításokra, adatfeldolgozásra, grafikus megjelenítésre stb...
- A FP-hoz hasonlóan **ingyenes** az objektumorientált programozáshoz könnyedén használható Pascal alapú nyelv a LAZARUS
- Gyors fordítóprogramjuk, jól használható hibakeresőjük van 24

5.4. Pascal karakteres editora



The image shows a screenshot of a Pascal character editor window. The window has a menu bar at the top with the following items: File, Edit, Search, Run, Compile, Debug, Tools, Options, Window, and Help. The title bar of the window displays the file path \TP\HAROMSZG.PAS and the line number 1=[↑]. The main area of the window contains the following Pascal code:

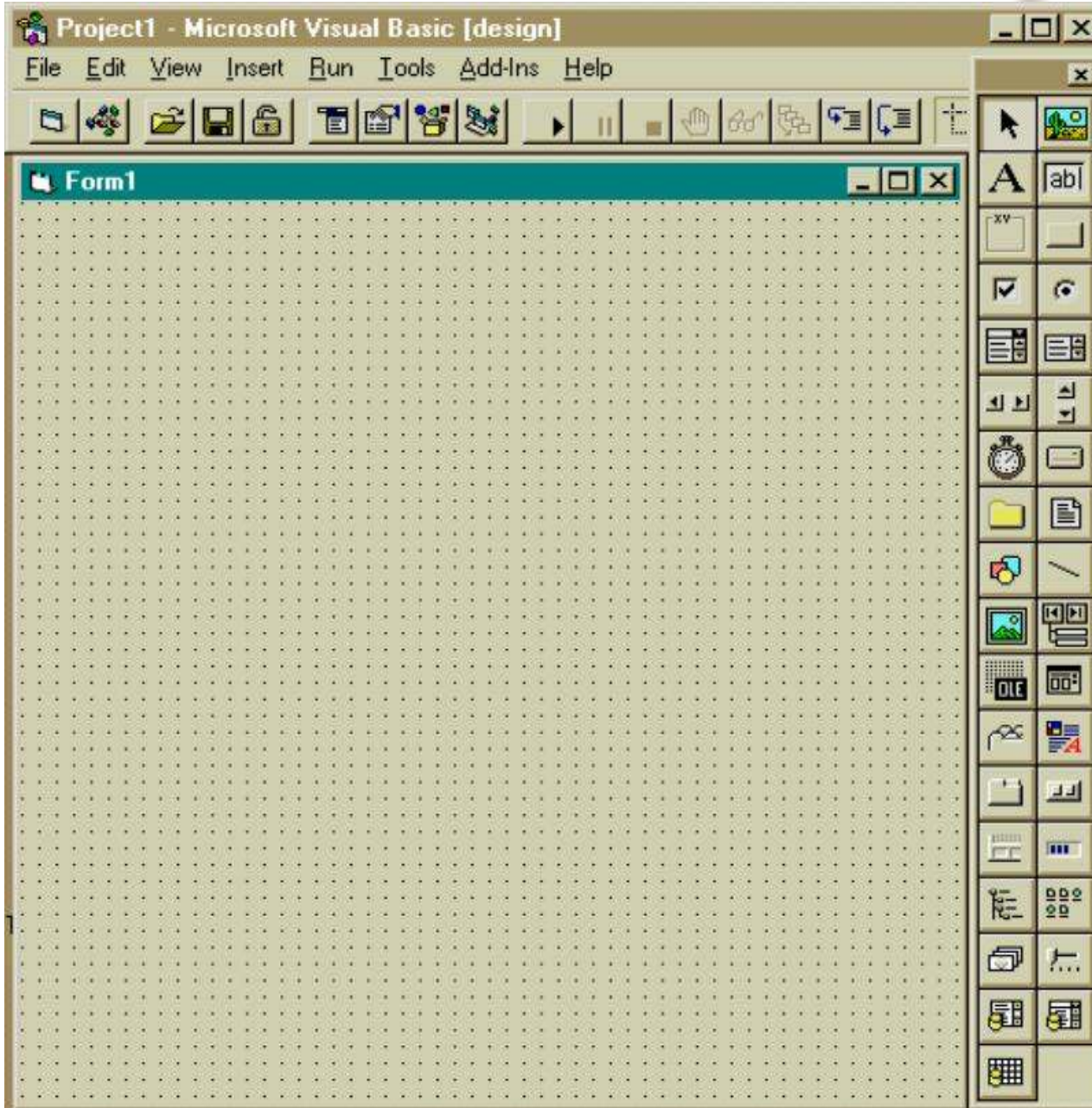
```
Program haromszog;  
Uses  
    Crt;  
Var  
    a, m, t: real;  
BEGIN  
    ClrScr;  
    Write('Kérem az alap értékét:');  
    ReadLn(a);  
    Write('Kérem a magasság értékét:');  
    ReadLn(m);  
    t:=a*m/2;  
    WriteLn('A háromszög területe:',t:8:3);  
    ReadLn;  
END.  
-
```

The status bar at the bottom of the window shows the cursor position 16:1. Below the window, there is a row of function key shortcuts: F1 Help, F2 Save, F3 Open, Alt+F9 Compile, F9 Make, and Alt+F10 Local menu.

- A háromszög területét számoló program kódja

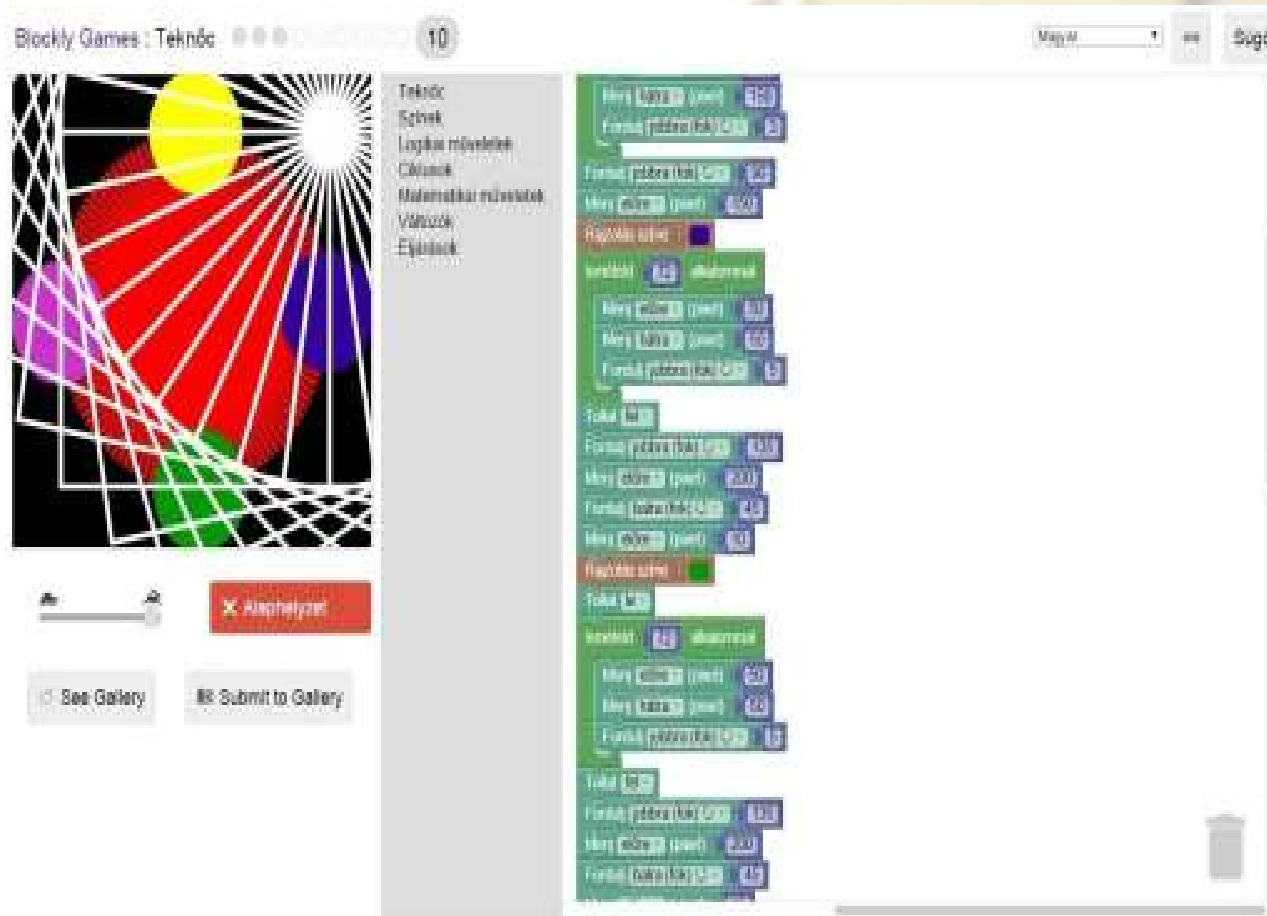
5.5. Példa a kódolásra grafikus felületen:

VISUAL BASIC



- Windows-os grafikus felület a kódoláshoz
- Windows-os jelleget viselő új szoftverek hozhatók létre
- **Objektumorientált és eseményorientált** programozás jellemzi
- Egérrel létrehozható grafikus elemek

5.6. Példa a kódolásra online vizuális felületen: BLOCKLY



- Online - böngészőben megnyíló - grafikus felület a kódoláshoz
- Puzzle szerű blokkokkal dolgozhatunk
- Grafikus rajzolója a **Teknőc**
- Interaktív felülete a **Kódszerkesztő**
- Megtekinthető, és esetenként szerkeszthető **forráskód**

Ellenőrző kérdések

- Mi az algoritmus?
- Mi az összefüggés az algoritmus és a program között?
- Milyen általános algoritmus-leíró formákat, eszközöket ismersz?
- Mi a folyamatábra és a struktogram?
- Mi a ciklus és milyen fajtái vannak?
- Milyen elágazási formákat ismersz?
- Mikor írunk ELJÁRÁS-t?
- Mit nevezünk kódolásnak?
- Milyen fejlesztőrendszerekről tudsz?