

Dr. Kiss Attila
kiss@inf.elte.hu

Bevezetés az informatikába

***Programozás és
programozás módszertan***

-A számítástechnika és a programozás története

- A XX. Század negyvenes éveitől Neumann-elvű, elektronikus számítógépeket használunk, amelyeket az őket alkotó fő alkotórész milyensége szerint számítógépgenerációkba sorolunk. Ezek a következők:
 1. generációs számítógépek: elektroncsövekből épülnek fel
 2. generációs számítógépek: fő alkotórészük az egyedi tranzisztor (1958-tól)
 3. generációs számítógépek: integrált áramköröket tartalmaznak (1965-től)
 4. generációs számítógépek: központi egységük a mikroprocesszor (1971, Intel, 4004 típusjelű processzor)
- A különböző megvalósítás azonban nem jelentette a programozás alapelveinek változását

-1. Alapfogalmak

- **Adat:** Egy objektum számunkra fontos tulajdonsága.
- **Ismeret:** Az adatok olyan összessége, amelyet az ember képes észlelni, érzékelni, és összefüggéseiben látni.
- **Információ:** Új ismeretet tartalmazó adathalmaz.

-1. Alapfogalmak

- Program: A számítógép számára érthető **utasítások sorozata**, amely az **adatok** megfelelő számításaival és mozgásaival egy feladat megoldását célozza.
- Következtetés:

PROGRAM=ADAT+ALGORITMUS

-1. Alapfogalmak

- Algoritmus: egy feladat megoldásának leírása.
- Követelmények: milyen legyen egy algoritmus?
 - Általános legyen, lehetőleg a feladattípusra adjon megoldást
 - Véges számú lépésben vezessen eredményre (időben és terjedelemre is véges legyen)
 - Megfelelő bemenő adatokra megfelelő kimenetet adjon

0. Feladatmegoldás számítógéppel

- I. Mintapélda a valós életből: házépítés
1. Igényfelmérés (szempontok: család mérete, elképzelés, pénz)
 2. Tervezés (alaprajz, anyagigény)
 3. Szervezés (ütemterv vagy vállalkozó)
 4. Építkezés (anyagbeszerzés, kivitelezés)
 5. Használatba vétel (szemrevételezés – szépség, kipróbálás – jószág)
 6. Beköltözés, bentlakás (módosítgatás, újabb hibák, karbantartás)

0. Feladatmegoldás számítógéppel

II. A programkészítés folyamata

	Tevékenység	Eredmény
1.	Specifikálás (miből?,mit?)	specifikáció
2.	Tervezés (mivel?,hogyan?)	adat- algoritmusleírás
3.	Kódolás (a géppel hogyan?)	kód (reprezentáció + implementáció)
4.	Tesztelés (hibás?)	hibalista (diagnózis)
5.	Hibakeresés (hol a hiba?)	hibahely, ok (terápia)
6.	Hibajavítás (hogyan jó?)	helyes program
7.	Minőségvizsgálat, hatékonyság (jobbítható?)	jó program
8.	Dokumentálás (hogyan működik, hogyan kell használni?)	használható program
9.	Használat, karbantartás (még mindig jó?)	időtálló program

1. Feladatmegoldás számítógéppel – részletesen az egyes lépésekről - SPECIFIKÁCIÓ

Részei:

- Bemenő adatok (értékhalmoz, mértékegység) + összefüggéseik (**előfeltétel**)
- Eredmények + kiszámítási szabályuk (**utófeltétel**)
- A megoldással szembeni követelmények
- Korlátozó tényezők
- A használt fogalmak definíciói

1. Feladatmegoldás számítógéppel – részletesen az egyes lépésekről - SPECIFIKÁCIÓ

Tulajdonságai:

- Egyértelmű, pontos, teljes
- Rövid, tömör, formalizált
- Szemléletes, érthető

Specifikációs eszközök:

- Szöveges leírás
- Matematikai leírás

2. Feladatmegoldás számítógéppel – részletesen az egyes lépésekről - TERVEZÉS

- 2 feladat együtt:
 - Algoritmustervezés
 - Adatszerkezet megtervezése

2/a. Feladatmegoldás számítógéppel – részletesen az egyes lépésekről - ALGORITMUSTERVEZÉS

- Eszközei: Algoritmusleíró eszközök
 - **Folyamatábra**
 - Struktogram
 - Jackson-diagramok
 - **Leírás mondatszerű elemekkel
(pseudokód)**
 - Leírás programozási nyelven

2/a. Feladatmegoldás számítógéppel – részletesen az egyes lépésekről - ALGORITMUSTERVEZÉS

- Milyen építőkövekből épül fel egy algoritmus?
 - Állítás: Néhány alapvető elem (vezérlési szerkezet) segítségével minden algoritmus elkészíthető.
 - Az a programozás, amely csak ezeket használja:

STRUKTÚRÁLT PROGRAMOZÁS

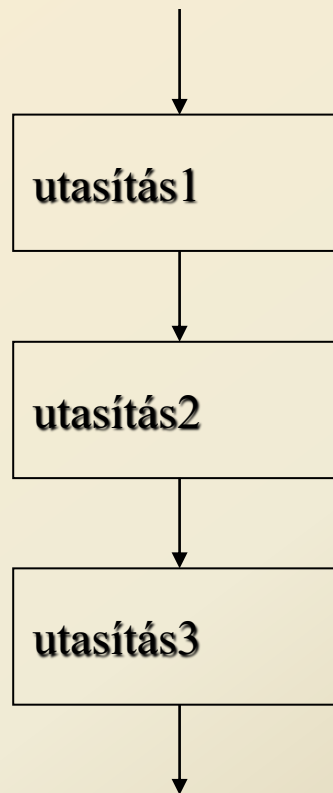
2/a. Feladatmegoldás számítógéppel – részletesen az egyes lépésekről

- ALGORITMUSTERVEZÉS

- Melyek ezek az építőkövek?
 - Tevékenységek egymásutánja (bambán dolgozunk):
SZEKVENCIA
 - Valamilyen döntésre kényszerül a program a végrehajtás során:
ELÁGAZÁS (SZELEKCIÓ)
(van egy-, két-, és többágú)
 - Valamilyen programrészletet többször kell végrehajtani (általában feltételtől függően):
CIKLUS (ITERÁCIÓ)
(többféle létezik)
 - A program részekre bontása alprogramokra (később tárgyaljuk)

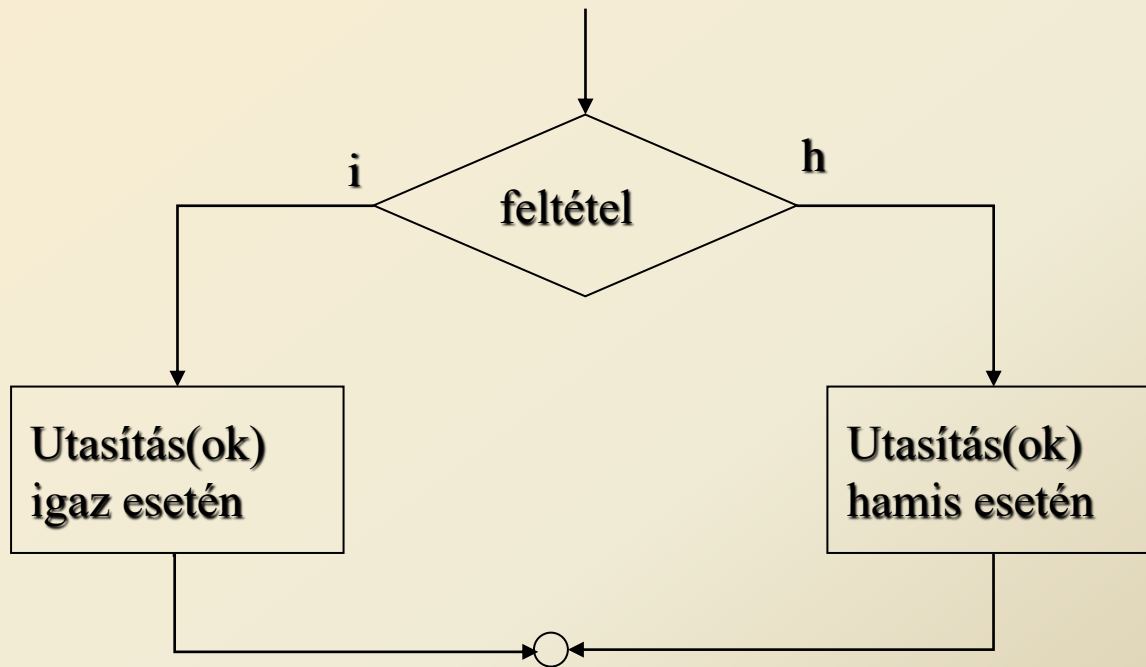
2/a. Feladatmegoldás számítógéppel – részletesen az egyes lépésekről - ALGORITMUSTERVEZÉS

- Szekvencia megvalósítása folyamatábrával



2/a. Feladatmegoldás számítógéppel – részletesen az egyes lépésekről - ALGORITMUSTERVEZÉS

- Elágazás megvalósítása folyamatábrával



- Az ágak valamelyike el is maradhat: egyágú elágazás
- Többágú elágazásra nincs folyamatábra-jelölés, több kétágú elágazással írható le

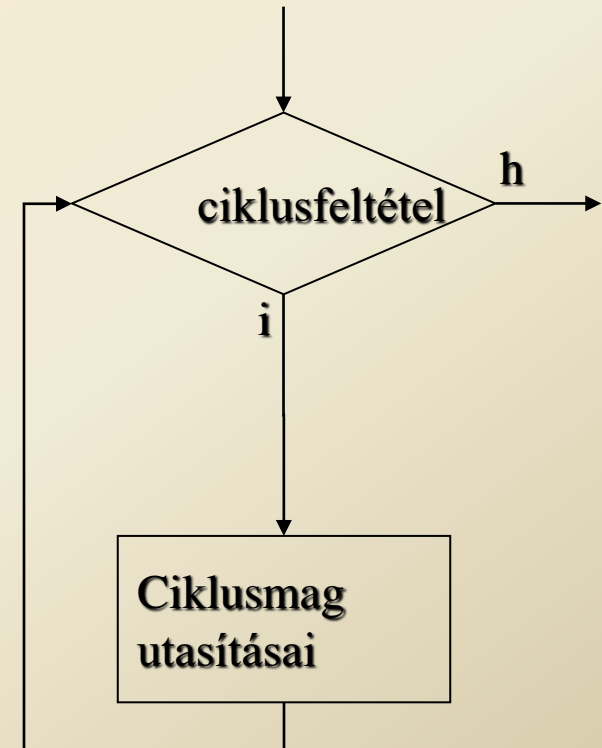
2/a. Feladatmegoldás számítógéppel – részletesen az egyes lépésekről - ALGORITMUSTERVEZÉS

- Ciklusok megvalósítása folyamatábrával
 - A megismétlésre kerülő utasítások összefoglaló neve:
CIKLUSMAG
 - Fajtái:
 - Elöltesztelő feltételes ciklus
 - Hátultesztelő feltételes ciklus
 - Számlálós (növekményes) ciklus (speciális elöltesztelő ciklus)

2/a. Feladatmegoldás számítógéppel – részletesen az egyes lépésekről - **ALGORITMUSTERVEZÉS**

- Elöltesztelő ciklus megvalósítása
folyamatábrával

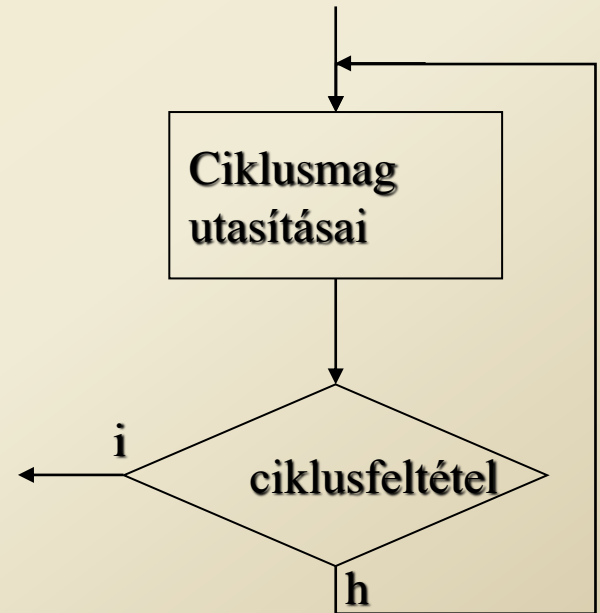
– A ciklusmag nem
biztos, hogy
végrehajtódik!



2/a. Feladatmegoldás számítógéppel – részletesen az egyes lépésekről - ALGORITMUSTERVEZÉS

- Hátultesztelő ciklus megvalósítása
folyamatábrával

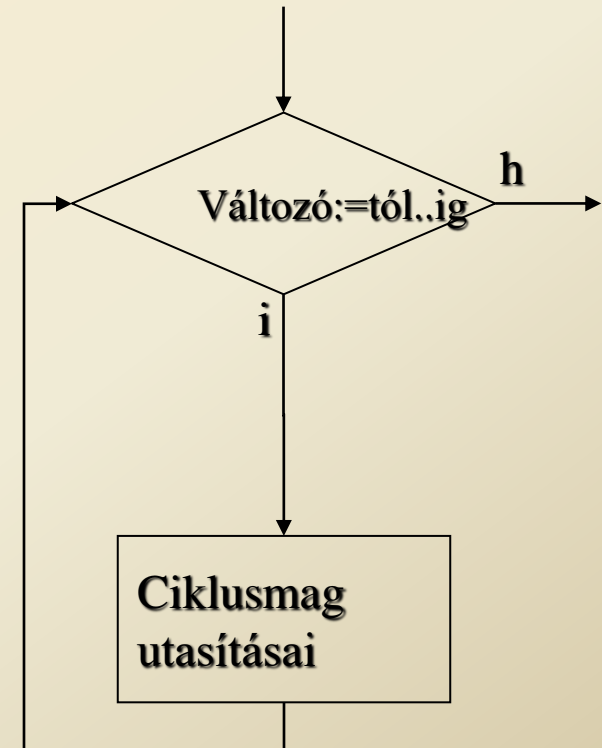
– A ciklusmag egyszer
biztosan végrehajtódik!



2/a. Feladatmegoldás számítógéppel – részletesen az egyes lépésekről - **ALGORITMUSTERVEZÉS**

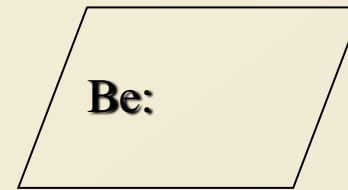
- Számlálós ciklus megvalósítása
folyamatábrával

– A ciklusmag nem
biztos, hogy
végrehajtódik!



2/a. Feladatmegoldás számítógéppel – részletesen az egyes lépésekről - ALGORITMUSTERVEZÉS

- Egyéb folyamatábra-elemek
 - Bevitel, kiírás:



- Algoritmus eleje, vége:



2/a. Feladatmegoldás számítógéppel – részletesen az egyes lépésekről - ALGORITMUSTERVEZÉS

- Algoritmusleírás mondatszerű elemekkel
 - A beszélt nyelvhez közeli leírás
 - Logikája hasonlít a Pascal nyelvhez
 - Az összetartozó elemeket egy oszlopban kezdjük (tabulált írásmód)

2/a. Feladatmegoldás számítógéppel – részletesen az egyes lépésekről - ALGORITMUSTERVEZÉS

- A mondatszerű leírás szabályainak definiálásához ún. metanyelvi szimbólumokat használunk:
 - $\langle \dots \rangle$ Be kell helyettesíteni valamit
 - $[\dots]$ Nem kötelező elem

2/a. Feladatmegoldás számítógéppel – részletesen az egyes lépésekről - ALGORITMUSTERVEZÉS

- Szekvencia mondatszerű elemekkel

<utasítás(ok) 1>

<utasítás(ok) 2>

•

•

2/a. Feladatmegoldás számítógéppel – részletesen az egyes lépésekről - ALGORITMUSTERVEZÉS

- Elágazás mondatszerű elemekkel

Ha <feltétel> akkor

<utasítás(ok) 1>

[Egyébként

<utasítás(ok) 2>]

Elágazás vége

2/a. Feladatmegoldás számítógéppel – részletesen az egyes lépésekről - ALGORITMUSTERVEZÉS

- Elágazás mondatszerű elemekkel

– **Többágú:**

Elágazás

<feltétel 1> esetén <utasítás(ok) 1>

<feltétel 2> esetén <utasítás(ok) 2>

.

.

<feltétel n> esetén <utasítás(ok) n>

[Egyébként

<utasítás(ok) n+1>]

Elágazás vége

2/a. Feladatmegoldás számítógéppel – részletesen az egyes lépésekről - ALGORITMUSTERVEZÉS

- Elöltesztelő ciklus mondatszerű elemekkel

Ciklus amíg <feltétel

<ciklusmag utasítása(i)>

Ciklus vége

2/a. Feladatmegoldás számítógéppel – részletesen az egyes lépésekről - ALGORITMUSTERVEZÉS

- Hátultesztelő ciklus mondatszerű elemekkel

Ciklus

<ciklusmag utasítása(i)>

Amíg nem <feltétel>

Ciklus vége

2/a. Feladatmegoldás számítógéppel – részletesen az egyes lépésekről - ALGORITMUSTERVEZÉS

- Számlálós ciklus mondatszerű elemekkel

Ciklus <változónév>:=<kezdőérték>-től <végérték>-ig
<ciklusmag utasítása(i)>

Ciklus vége

2/a. Feladatmegoldás számítógéppel – részletesen az egyes lépésekről - ALGORITMUSTERVEZÉS

- Egyéb algoritmuselemek mondatszerű elemekkel
 - Beolvasás:
Be:
 - Kiírás:
Ki: ...

2/b. Feladatmegoldás számítógéppel – részletesen az egyes lépésekről - ADATTERVEZÉS

- Hol helyezkedhetnek el a program futásához szükséges adatok?
 - Háttértáron
 - > fájlkezelés, külön foglalkozunk vele
 - Operatív tárban (memória, RAM)

2/b. Feladatmegoldás számítógéppel – részletesen az egyes lépésekről - ADATTERVEZÉS

- Memóriában elhelyezkedő adatok kezelése
 - Két alapvetően különböző fajtája van:
 - Ami nem változik a program futása során: **állandó, konstans** (ha többször használjuk, érdemes nevet adni neki)
 - Ami változik a futás során: **változó**

2/b. Feladatmegoldás számítógéppel – részletesen az egyes lépésekről - ADATTERVEZÉS

- Változók tulajdonságai
 - Neve van, ezzel hivatkozhatunk rá (**azonosító**)
 - A program bizonyos szakaszában használható csak (**érvényességi kör, hatáskör**)
 - A program futási ideje alatt sem mindig létezik (**élettartam**)
 - Meghatározott értékeket vehet fel, meghatározott műveletek végezhetők vele, meghatározott méretű helyet foglal a memóriában, és meghatározott szerkezete van (**típus**)

2/b. Feladatmegoldás számítógéppel – részletesen az egyes lépésekről - ADATTERVEZÉS

- Néhány egyszerűbb típus
 - Egész típus
 - Valós típus
 - Karakter típus
 - Karakterlánc típus
 - Logikai típus
 -

3. Feladatmegoldás számítógéppel – részletesen az egyes lépésekről - KÓDOLÁS

- A program kódolása (megírása) valamilyen programozási nyelven történik.
- Ahhoz, hogy ki tudjuk választani az általunk használni kívánt programozási nyelvet, tudnunk kellene, milyenek vannak egyáltalán
- És azt is, hogy hogyan lehet ezeket jellemezni, csoportosítani

3. Feladatmegoldás számítógéppel – részletesen az egyes lépésekről - KÓDOLÁS

- Programozási nyelvek csoportosítási szempontjai

I. Emberközelség szerint

1. Alacsony szintű

a) Gépi kód

- Az utasítások és az adatok is számok
- Az ember számára szinte kezelhetetlen
- A programozó közvetlenül a processzor utasításait írja

b) Assembly

- A gépi kódú utasításoknak pár karakterből álló szimbólumok felelnek meg (mnemonik)

3. Feladatmegoldás számítógéppel – részletesen az egyes lépésekről - KÓDOLÁS

- Programozási nyelvek csoportosítási szempontjai

I. Emberközeliég szerint

2. Magas szintű

- A beszélt emberi nyelvhez (általában angolhoz) közeli programozási nyelvek
- Az ipari méretű programozás hívta életre őket
- Néhány példa: BASIC, Pascal, Java, PL-1, C, Ada...

3. Feladatmegoldás számítógéppel – részletesen az egyes lépésekről - KÓDOLÁS

- Programozási nyelvek csoportosítási szempontjai

II. Strukturáltság szerint

1. Strukturált

- Csak a strukturált programozás eszközeit megvalósító utasításai vannak
- Támogatja a program részekre, alprogramokra bontását (eljárások, függvények)
- Pl.: C, Pascal, Java

2. Nem strukturált

- Pl.: Assembly, eredeti BASIC nyelv

3. Feladatmegoldás számítógéppel – részletesen az egyes lépésekről - KÓDOLÁS

- Programozási nyelvek csoportosítási szempontjai

III. Alkalmazási kör szerint

1. Általános célú

- Elméletileg bármilyen programozási feladat megoldására alkalmas
- Emiatt azonban nincs optimalizálva egy speciális feladatra
- Pl.: C, Pascal, BASIC

2. Speciális célú nyelvek

- Általában egy feladatra optimalizáltak, pl. szövegfeldolgozás, szimuláció, adatbáziskezelés
- Pl.: PROLOG, SIMULA, SQL, dBase

3. Feladatmegoldás számítógéppel – részletesen az egyes lépésekről - KÓDOLÁS

- Programozási nyelvek csoportosítási szempontjai

IV. Fejlettség szerint

- 1. Generációs nyelvek
 - Gépi kód
- 2. Generációs nyelvek
 - Már vannak benne vezérlési szerkezetek, az utasítások elvannak nevezve
 - Pl. Assembly, ALGOL60, FORTRAN
- 3. Generációs nyelvek
 - A magas szintű programnyelvek többsége, struktúrált utasításaival, alprogramok lehetőségével
- 4. Generációs nyelvek
 - Programgenerátorok, beépített adatbáziskezelőt tartalmaznak, a szoftver a háttérben írja a kódot helyettünk
 - Pl.: Visual Basic, Visual C++, Delphi, Python

3. Feladatmegoldás számítógéppel – részletesen az egyes lépésekről - KÓDOLÁS

- Programozási nyelvek csoportosítási szempontjai

V. Fordítás típusa szerint

- A forráskódot, amit a programozó létrehoz, a futtatáshoz gépi kódba kell alakítani, ez a fordítás

a) Értelmező (interpreter) típusú nyelvek

- A fordítás a program futása közben, az adott sor lefuttatása előtt történik
- A futtatás emiatt lassúbb
- A szintaktikai (nyelvtani) programhibák is csak futás közben derülnek ki
- Pl. : korai BASIC-ek (Commodore 64, ZX Spectrum, Quick Basic)

3. Feladatmegoldás számítógéppel – részletesen az egyes lépésekről - KÓDOLÁS

- Programozási nyelvek csoportosítási szempontjai

V. Fordítás típusa szerint

- b) Fordító (compiler) típusú nyelvek
 - A fordítás a program futása előtt történik
 - A futtatás emiatt gyorsabb (a gépi kódú program fut)
 - A szintaktikai (nyelvtani) programhibák már futás előtt kiderülnek
 - Pl. : Pascal, C, C++, Delphi...

3. Feladatmegoldás számítógéppel – részletesen az egyes lépésekről - KÓDOLÁS

- Programozási nyelvek csoportosítási szempontjai

VI. Számítási modell szerint

a) Neumann elvű nyelvek

- A program felépítése erőteljesen kihasználja, hogy a számítógép, amin a program futni fog, Neumann-elvű (címezhető memória, változók, stb)
- A ma létező legtöbb nyelv ilyen

b) Automata elvű nyelvek (ipari robot, festőautomata programozásához)

- Az adatok állapotok, illetve bemeneteken mérhető értékek
- A végrehajtás állapotok sorozata
- Tevékenységorientált
- Pl: LOGO grafikus része, ipari robotok pr. nyelvei

3. Feladatmegoldás számítógéppel – részletesen az egyes lépésekről - **KÓDOLÁS**

- Programozási nyelvek csoportosítási szempontjai

VI. Számítási modell szerint

c) Logikai nyelvek

- A program egy logikai kifejezés
- A végrehajtás ennek kiértékelése
- Erős matematikai kidolgozottság jellemzi ezeket a nyelveket
- A program és az adatok nem különülnek el (nincsenek változók)
- Vezérlési szerkezetek helyett logikai kifejezések és rekurzió
- Pl.: PROLOG

d) Funkcionális nyelvek

- Az adatok állapotok, illetve bemeneteken mérhető értékek
- A program egy függvény
- A végrehajtás függvénykiértékelés
- Erős matematikai kidolgozottság
- Nincsenek változók, helyette függvényparaméterek
- Pl.: LOGO szövegkezelő része, FORTH, LISP

3. Feladatmegoldás számítógéppel – részletesen az egyes lépésekről - KÓDOLÁS

- Programozási nyelvek csoportosítási szempontjai

VI. Számítási modell szerint

e) Objektumelvű nyelvek

- Objektum=tulajdonságai+műveletei
- Az adat és a kód nem választható szét
- Objektumtípus=osztály
- Az objektumpéldányok eljárásokkal kommunikálnak
- Eseményvezérelt programozás jellemzi
- Öröklés: a származtatott objektumtípus rendelkezik egy másik objektum összes tulajdonságával és műveletével, de újak is lehetnek neki
- Pl.: Turbo Pascal 6.0-tól, Visual Basic, Delphi, Smalltalk

3. Feladatmegoldás számítógéppel – részletesen az egyes lépésekről - KÓDOLÁS

- Programozási nyelvek csoportosítási szempontjai

VI. Típusosság szerint

- a) Erősen típusos: minden változó, objektum típusát meg kell adni a használat előtt, ezt deklarációnak hívják. (Pl.: Pascal, C)
- b) Gyengén típusos: Vannak előre definiált típusok (szöveges, numerikus), de a pontos típus futás közben derül ki (Pl. hogy egy változó egész vagy valós típusú). Bizonyos típusokat deklarálni kell (tömb). Ilyen nyelv a BASIC.
- c) Nem típusos nyelvek: A változók típusát nem kell megadni, sőt, bizonyos esetekben nincsenek is változók.