

# Programozás 1

Alapfogalmak

# A kurzus céljai

- Absztrakt algoritmikus gondolkodásmód kialakítása
  - A valós életből vett dolgok matematikai leírása (a lehető legegyszerűbb, de a célnak tökéletesen megfelelő módon).
  - Mivel egy gép „csak” számolni tud, a problémákra végső soron matematikai megoldást adunk.
  - Ne jelentsen gondot egy egyszerűbb probléma matematikai leírása, illetve magasabb szintű matematikai struktúrák használata egy probléma leírására.

# A kurzus céljai

- Programozási struktúrák megismerése
  - Az imperatív programozás adatokból és rajtuk végzett műveletekből építkezik.
  - Ahhoz, hogy egy problémát részekre bontsunk, tudnunk kell, hogy ezt milyen módon tehetjük meg.
  - Ahhoz, hogy valamit adatként modellezni tudjunk szintén ismernünk kell, hogy milyen alapelemekből hogyan építkezhetünk.
  - Ne jelentsen gondot eldönteni, hogy mikor és hogyan kell egy (tágabb értelemben vett) problémát részekre bontani, vagy mikor lehet triviálisan megoldhatónak tekinteni.

# A kurzus céljai

- A C programozási nyelv megismerése
  - Egy probléma megoldása akkor teljes, ha azt a számítógép számára érthető formában le tudjuk írni, ennek eszköze a programozási nyelv.
  - A C egy igen elterjedt, alacsony és magasabb szintű programozásra is alkalmas nyelv.
  - Ne jelentsen gondot egy megtervezett program implementálása, illetve annak eldöntése, hogy egy probléma megoldása közvetlenül leírható-e C nyelven vagy sem.

# Fogalmak

- Programozás
  - Valós problémák számítógépes megoldása
- Számítógép tulajdonságai
  - Általános célú (univerzális)
    - A határok egyre inkább elmosódnak (pl. okostelefon)
  - Automatikus vezérlésű
    - A programját külső beavatkozás nélkül képes végrehajtani
  - Elektronikus
    - Az áramkör kapcsolás nem mechanikus
  - Digitális
    - Két értékből (0,1) felépített jól elkülöníthető állapotok

# A számítógép modellje

- Hogyan számol az ember?
  - Olvassa az utasításokat
  - Értelmezi
  - Végrehajtja
  - Rátér a következő utasításra, vagy arra, amit előírnak
- Milyen utasítások vannak?
  - Bemeneti/kimeneti (input/output)
  - Adatkezelési
  - Aritmetikai
  - Vezérlési

# A számítógép modellje

- A számolási tevékenységet végző ember számára magyar nyelven lehet olyan utasításokat adni, amit képes végrehajtani.
- A számítógép számára a vezérlő egység által értelmezhető és végrehajtható utasítások (parancsok) adhatók.
- Kezdetben az ember megtanulta ezeket az utasításokat (a számítógép nyelvét). A kommunikáció ezen a nyelven lassú, nehézkes, sok hibával jár.

# A számítógép modellje

- Miért nem tanul meg inkább a számítógép magyarul?
  - Valakinek meg kell rá tanítani
  - A nyelv kétértelműsége
  - Szövegkörnyezettől való függőség
  - Szemantika (el kellene magyarázni a szavak jelentését)
- Az ember is és a számítógép is tanuljon meg egy nyelvet!



# A számítógép modellje

- A nyelv egy kifejező eszköz valaminek a leírására. De mit akarunk a számítógép számára leírni?
- Azt, hogy mit csináljon, vagyis milyen tevékenységet hajtson végre. Egy számítógép csak azt tudja megcsinálni, amit megmondunk neki.
- Amiről meg tudjuk mondani, hogy hogyan kell csinálni, azt a számítógép meg tudja csinálni.

# Algoritmus

- Alapvetően azt kell megmondani a számítógépnek, hogy milyen adatokon milyen műveleteket kell elvégezni. Ezt pontosan meg kell fogalmazni, vagyis el kell készíteni a problémát megoldó algoritmust.
- Az algoritmus
  - Adott típusú összes feladat megoldására vonatkozó pontos előírás, amely megmondja, hogy a kezdeti adatokon milyen műveleteket milyen sorrendben kell elvégezni

# Algoritmus

- Tulajdonságai
  - meghatározott
    - véges módon leírható, mindig ugyanazt eredményezi
  - széleskörű
    - egész feladatosztályra vonatkozik
  - véges
    - véges lépésben véget ér
  - potenciálisan megvalósítható
    - a technika fejlődésével
- Nincs olyan algoritmus, amelyik el tudja dönteni, hogy egy tetszőleges program a kezdőadatokkal végrehajtva véges lépés után megáll-e?

# Fogalmak

- Programozási nyelv
  - A problémát megoldó algoritmus leírására szolgál
  - Lehet az emberi nyelvhez közel álló (magas szintű) vagy a számítógép nyelvéhez közel álló (alacsony szintű)
- Program
  - Egy algoritmusnak egy adott programozási nyelven történő leírása

# Fogalmak

- Hardver
  - A számítást végző fizikai-technikai rendszer
- Szoftver
  - A hardvert működtető programok és parancsok összessége
- Adatok
  - A hardver és a szoftver által feldolgozott információ

# Hardver környezet

- Nagyon sokféle hardver létezik, de az alapfelépítése mindnek nagyon hasonló
  - Központi feldolgozó egység (CPU, Processzor)
    - felépítés (CISC, RISC), sebesség (MHz, GHz)
  - Főttár (Memória)
    - típus (ROM, RAM), méret (bit, bájt(byte), KB, MB, GB)
  - Háttértár (HDD, CD, DVD, Flash)
    - méret (MB, GB, TB)
  - Felhasználói terminál és perifériák (I/O eszközök)
    - billentyűzet, egér, képernyő, hanggenerátor
    - nyomtató, modem, hálózati csatoló, speciális eszközök

# Szoftver környezet

- Egy hierarchikus felépítésű programrendszer
- Főbb szintjei felhasználó szemszögből
  - felhasználói programok
  - programfejlesztői rendszerek
  - operációs rendszer
  - gépi alapszoftver

# Gépi alapszoftver

- BIOS
  - Basic Input Output System
- A ROM-ba (EPROM, EEPROM) égetett programok.
- Ezek funkciója
  - A hardver tesztelése
  - Az operációs rendszer betöltése és indítása
  - Gépi szintű be- és kimenet megvalósítása
    - Ezt a feladatot a modern operációs rendszerek az elindításuk után részben vagy teljesen átveszik



# Operációs rendszer

- Olyan programrendszer, amely közvetítő szerepet tölt be a számítógép hardver erőforrásai és a felhasználó között
- Főbb funkciói:
  - Programok végrehajtása
  - erőforrások elosztása
  - input/output műveletek végzése
  - háttértárakon tárolt adatállományok kezelése (fájlrendszer)
  - a működés közben fellépett hibák lekezelése
  - a felhasználó által kiadott parancsok értelmezése és végrehajtása

# Programfejlesztő rendszer

- Feladata, hogy támogassa a programozás során a programok (programrendszerek) létrehozását, módosítását és végrehajtását; fő funkciói:
  - könyvtárkezelés
  - szövegszerkesztés
  - fordítás
  - végrehajtás
  - hibakeresés
  - rendszerparaméterezés
  - Dokumentáció elérése, help

# Felhasználói program(rendszer)

- A megoldandó probléma számítógépes megoldását szolgáltatja

# A programozás fázisai

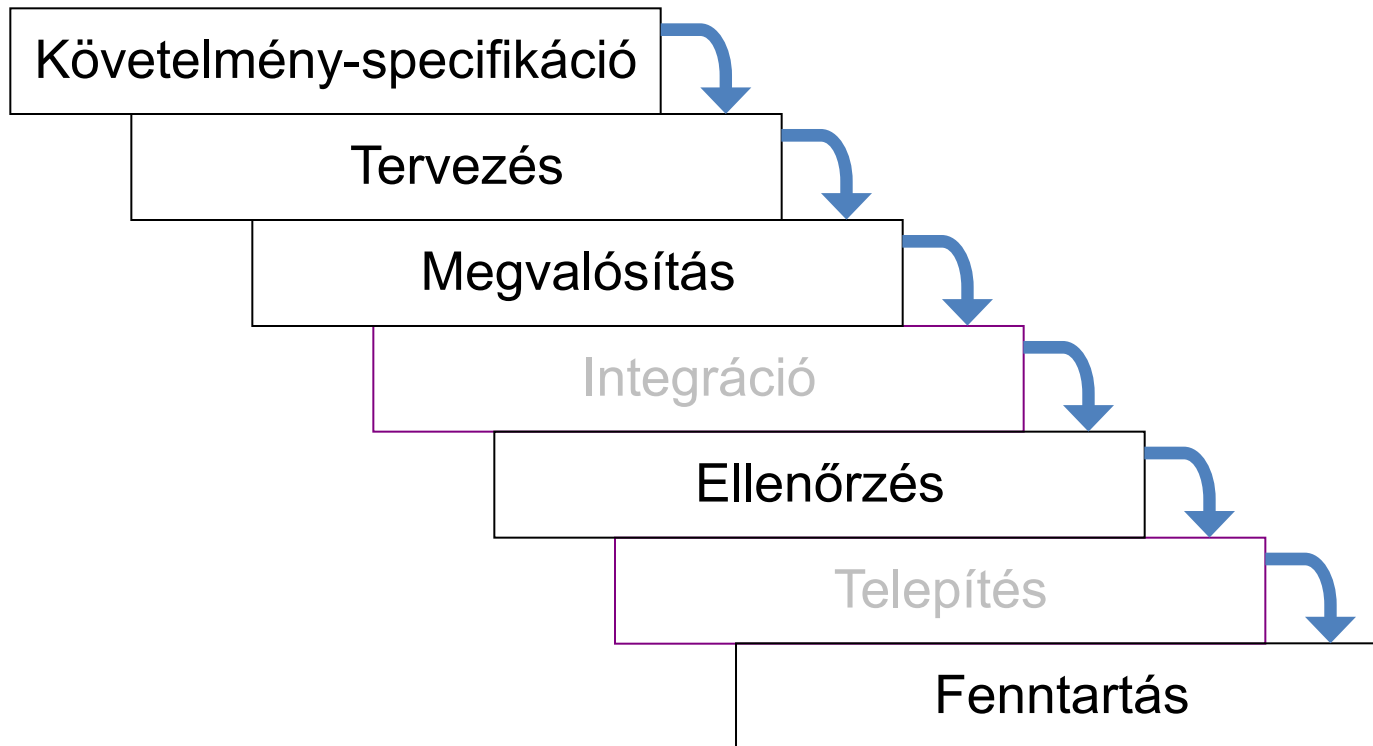
- A számítógépes problémamegoldás (a programozás) egymástól jól elkülöníthető fázisokból épül fel, amelyek sajátos kölcsönhatásban vannak egymással; ezt a kapcsolatot fejezi ki a szoftverfejlesztési modell
- A világon többféle modell létezik, ezek közül nincs jó vagy rossz, csak adott feladathoz jobban vagy kevésbé alkalmas
  - vízesés-modell, V-modell, spirál-modell, O-modell, extrém programozás, stb...

# A programozás fázisai

- Az egyik első és legegyszerűbb a vízesés modell
  - Követelmény-specifikáció (Problémafelvetés, Specifikáció)
  - Tervezés (Algoritmustervezés)
  - Megvalósítás
  - Integráció
  - Ellenőrzés (Helyességigazolás, Költségelemzés, Tesztelés)
  - Telepítés
  - Fenntartás (Végrehajtás, Fenntartás)

# A vízesés modell

- Az modellben az előbbi fázisokat sorban egymás után hajtjuk végre



# Követelmény-specifikáció

- A (későbbi) felhasználó a saját szakterületének megfelelő nyelven meg tudja fogalmazni a megoldandó problémát (problémafelvetés), ez viszont az algoritmus elkészítéséhez nem elég; ahhoz pontosan meg kell határozni, hogy milyen feltételt elégítenek ki a probléma bemenő adatai, és hogy adott bemenő adatok esetén milyen feltételt kell kielégíteni a kiszámított adatoknak (specifikáció)

# Követelmény-specifikáció

- A specifikáció tehát egy  $(B,K)$  bemeneti-kimeneti feltételpár: akkor fogadjuk el az algoritmust a probléma megoldásának, ha valahányszor a  $B$  feltétel teljesül a bemenő adatokra, mindannyiszor a  $K$  feltétel teljesül a kiszámított adatokra



# Tervezés

- Itt történik a specifikációt kielégítő algoritmus létrehozása (algoritmustervezés)
- A felülről lefelé haladó módszer lényege:
  - A kiindulási  $P$  problémát  $P_1, \dots, P_n$  részproblémákra bontjuk
  - Minden  $P_i$ -nek valamely  $M_i$  megoldását adjuk
  - Az  $M_i$  műveleteket alkalmas módon összetéve  $P$ -t megoldó algoritmushoz jutunk
  - Ezt addig folytatjuk, amíg a választott programozási nyelv elemi műveletével közvetlenül meg nem adható problémákhoz jutunk

# Megvalósítás

- Megvalósításon az algoritmustervezés során kifejlesztett algoritmusnak egy adott programozási nyelven történő leírását értjük (kódolás)
- A lépés eredménye a program

# Integráció

- Amennyiben az általunk megoldott probléma egy nagyobb, összetettebb probléma része, úgy a programunknak is szükségképpen illeszkednie kell egy a nagyobb probléma egyéb komponenseinek megvalósításához
- Ilyen esetekben ez a lépés szolgálja a saját kódunk nagyobb programrendszerbe történő beillesztését, és a beillesztés során esetleg felvetődő problémák megoldását

# Ellenőrzés

- Az ellenőrzés része a
  - helyességigazolás, minek során megmutatjuk, hogy a kifejlesztett algoritmus valóban a kiindulási probléma megoldását adja (nincs tervezési hiba)
  - költségelemzés, ami az algoritmus idő- és tárigényének (költségének) a bemenő adatok függvényében történő meghatározása
  - tesztelés, amivel a program megvalósítása során elkövetett hibák egy részét tudjuk felderíteni
    - a helyességigazolás és költségelemzés kiváltására nem alkalmas, de segíthet ezek elvégzésében is

# Telepítés

- Amennyiben szükséges a programot telepíteni kell, azaz el kell helyezni és működőképesé kell tenni a felhasználó által kívánt környezetben

# Fenntartás

- A programkészítés végső célja az, hogy a megoldandó probléma konkrét bemenő adataira végrehajtsuk a kifejlesztett algoritmust
- A program fenntartásán a végrehajtás során felmerült problémák megoldását értjük, ide tartozik például:
  - a használat során felmerülő hibák kijavítása
  - apróbb módosítások elvégzése

# Dokumentáció

- Nagyon fontos, hogy a teljes folyamat dokumentálva legyen, és ne csak ott, ahol ez nyilvánvalóan szükséges
  - A specifikáció önmagában dokumentum
  - A tervek valamilyen formájú leírása
  - Megjegyzések a kódban
  - Integráció során történt módosítások leírása
  - Számítások eredménye, teszt-jegyzőkönyv
  - Telepítési kézikönyv
  - Felhasználói kézikönyv, hibajavítások, módosítások rögzítése

# A programok fő komponensei

- Minden programnak két fő komponense van:
  - Az adatok
  - Az adatokon végzett műveletek
- Az adatok is és a műveletek is lehetnek összetettek.
- Adatokból (amelyek végső soron elemiek) adat-összetételi, műveletekből (amelyek végső soron elemiek) műveletképzési szabályokkal összetett adatok illetve műveletek gyárthatók.



# Az algoritmus vezérlése

- Az algoritmusnak, mint műveletnek a legfontosabb komponense az az előírás, amely az algoritmus minden lépésére (részműveletére) kijelöli, hogy a lépés végrehajtása után melyik lépés végrehajtásával folytatódják (esetleg fejeződnek be) az algoritmus végrehajtása.
- Ezt az előírást nevezzük az algoritmus vezérlésének.

# Vezérlési módok

- A vezérlési mód azt fejezi ki, hogy egyszerűbb műveletekből hogyan építünk fel összetett műveletet és ennek milyen lesz a vezérlése.
- Négy fő vezérlési módot különböztetünk meg:
  - Szekvenciális vezérlési mód
  - Szelekciós vezérlési mód
  - Ismétléses vezérlési mód
  - Eljárás vezérlési mód

# Vezérlési módok röviden

- Szekvenciális vezérlés
  - Véges sok adott művelet rögzített sorrendben egymás után történő végrehajtása
- Szelekciós vezérlés
  - Véges sok rögzített művelet közül adott feltétel alapján valamelyik végrehajtása
- Ismétléses vezérlés
  - Adott művelet adott feltétel szerinti ismételt végrehajtása
- Eljárásvezérlés
  - Adott művelet alkalmazása adott argumentumokra, ami az argumentumok értékének pontosan meghatározott változását eredményezi

# Felhasznált anyagok

- Dévényi Károly (SZTE): Programozás alapjai
- Simon Gyula (PE): A programozás alapjai
- Pohl László (BME): A programozás alapjai
- B. W. Kernighan - D. M. Ritchie: A C programozási nyelv