

Probléma-megoldás programozás útján

Célunk: megtanulni informatikus módjára gondolkozni. Ehhez gyakran matematikai, mérnöki és természettudományos elveket kell összkombinálnunk. De mindenekelőtt szükségünk van egy többé-kevésbé formalizált nyelvre, hogy kifejezhessük, leírassuk, megoldhassuk a problémákat melyekkel szembekerülünk. Mint a mérnökök: megtervezzük a részeket, rendszerré építjük össze őket, vagy mint a természettudósok: megfigyeljük a bonyolult rendszereket, hipotéziseket állítunk fel, *teszteljük* azokat.

Az informatikus legfontosabb tevékenysége: a **probléma-megoldás**. Ez komplex feladat: le kell képezni a feladatot olyan nyelvre amit egyrészt ismerünk, másrészt alkalmas a probléma leírására. Módszert, számolási szabályt kell találnunk a megoldás megkeresésére, ami legtöbbször matematikai, természettudományos ismereteket igényel. Ha ezekkel megvagyunk akkor jöhet a megvalósítás, kódolás egy konkrét nyelven.

Mi az a program?

A program egy leképezés, mely bemeneti adatok halmazát egy utasítássorozat segítségével átalakítja kimeneti adathalmazzá: **kimenet = program(bemenet)** A bemenet lehet egy polinom, az utasítássorozat pedig a polinom gyökeinek kiszámítását leíró utasítássorozat, a kimenet pedig a gyökök halmaza. Vagy: lehet a bemenet egy szöveg, amelyen módosításokat akarunk végrehajtani.

Ugyanazon problémákat megoldó programok teljesen másként néznek ki különböző nyelveken, de van néhány alaptevékenység, szerkezet mely minden nyelven kifejezhető:

olvasás

- billentyűzetről, fájlból, internetcímről, egyéb eszközzel.

írás

- képernyőre, fájlba, egyéb eszközre.

matematikai műveletek végrehajtása

- összeadás, szorzás, kivonás, osztás...

feltételes végrehajtás, elágazás

- tevékenységek végrehajtásának egy eseményhez kötése: ha a szám páros akkor oszd el kettővel. (egyébként ne csinálj semmit) ha a szám páros akkor oszd el kettővel, egyébként szorozd meg 3-mal

ismétlés

- tevékenység ismételt végrehajtása: ameddig a szám páros addig oszd el kettővel.

alprogram, függvény

- sokszor használt utasítássorozatok önálló egységbe csomagolásának lehetősége.

A programozás leegyszerűsítve: addig bontjuk szét az előttünk álló feladatot egyszerűbb részfeladatokra, amíg azokat a fenti szerkezetekkel már meg tudjuk oldani, ki tudjuk fejezni.

Az első program

A hagyományokhoz híven:

```
println("helóvilág") # megjegyzés ami a # után van. a kiértékelés: shift+enter
helóvilág
```

A `println` függvényt a `helóvilág` sztring bemenő adattal meghívtuk. Az utasítás (kifejezés) melynek végrehajtása (kiértékelése) során megjelent az adott sztring a képernyőn.

Aritmetikai műveletek

```
1+1
2
2^3
8
1*2*3*4*5
120
(1+2)*(1-2)
-3
20/10 # float lesz ! nincs automatikus int-konverzió
2.0
```

Típusok, értékek

Programozás során számokkal, karakterekkel, karaktersorozatokkal dolgozunk: 1, 42, 'X', "alma", 3.14 Ezek az értékek osztályokba, csoportokba sorolhatók - külön csoportot alkotunk az egész számoknak, a tört számoknak

(ami egy bővebb csoport), vagy karaktereknek és sorozataiknak. A csoportokat típusnak nevezzük. Az egész típus egy példánya az 1-es.

```
typeof(1)
Int64
typeof(42.0)
Float64
typeof('X')
Char
typeof("alma")
String
```

Természetes és formális nyelvek

Természetes nyelv az, amit használunk az emberi kommunikációnk során. Együtt fejlődött, fejlődik, alakul velünk. Hatással vagyunk rá és hatással van ránk. Sajátossága, hogy a jelsorozatok sokszor többféleképpen értelmezhetők, melyet a hallgató (olvasó) a kontextus segítségével általában meg tud szüntetni. Például: **Sorold fel a nyelv tulajdonágait!** A kérdésből még az sem derül ki, hogy a kérdező a nyelv-re mint jelrendszerre, vagy mint szerv-re gondolt.

A **formális** nyelv egy-egy speciális terület által használt, jól definiált szabályok által felépített szimbólum rendszer. Például véve kémiát, a H_2O , H_5O , $Z_{10}L$ jelek közül az első a víz vegyjele, a másodiknak megfelelő molekula a vegyértékek miatt nem létezhet, a harmadiknál már a használt jelekből látjuk hogy hibás van. Vagyis a formális nyelvek egy leegyszerűsített, de sokkal egyértelműbb kommunikációt tesznek lehetővé. A formális nyelvnél a szintaktika és a szakterületi tudás birtokában a jelsorozatoknak egyértelműen dekódolhatónak kell lenni.

Hibakeresés

A géppel, értelmezővel való párbeszéd, a programozás során hibákat követhetünk el, melyek helyét meg kell keresni, okát meg kell szüntetni. Itt egy fontos momentum, az hogy a gépet lehetőleg ne személyesítsük meg, vagy ha mindenáron ezt akarjuk tenni akkor gondoljunk a gépre beosztottként aki a parancsainkat feltétlen, betű szerint végrehajtja. Ha így teszünk sok fejfájástól kíméljük meg magunkat.