

Logikai programozás

BEVEZETÉS

ADMINISZTRATÍV KÉRDÉSEK

Bármilyen kérdéssel (akár tananyag, akár nem),
örömmel, bánattal:

achs.agnes@gmail.com

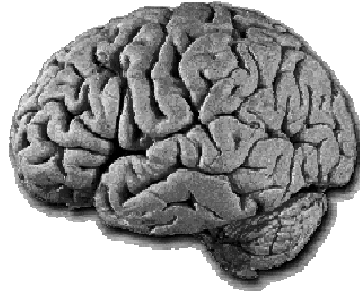
(Ha két napon belül nem válaszolok, akkor kérek egy
figyelmeztető levelet. ☺)

Ha a kérdés a követelményrendszerre vonatkozik, akkor
előbb ld. witch ☺, illetve neptun.

(De ha mégsem érthető, akkor persze lehet írni.)

KÖVETELMÉNYRENDSZER – FŐBB PONTOK

Jót, s jól. Ebben áll a nagy titok.
Ezt ha nem érted...



Néhány számszerű tudnivaló:

- Két zh
- Megajánlott jegy a két zh átlaga alapján:
 - 50% elégtelen
 - 51 - 65% elégséges
 - 66 - 82% közepes
 - 83 - jó

**A jelesért
vizsgázni kell.**

NÉHÁNY BIZTATÓ SZÓ

A tárgy „csalóka”,
tanuljon! ☺

„Csak akkor értesz
valamit, ha be tudod
programozni.
Te magad és nem
valaki más!

Ha nem tudod
beprogramozni, akkor
csak úgy gondold,
hogy érted.”



— Gregory Chaitin

A TANULÁS MOTTÓJA:

**„Akár azt hiszed, hogy képes vagy rá,
akár azt, hogy nem, igazad lesz”**

Henry Ford

NYELV

Lee Whorf szerint (a hopi indiánok nyelvét próbálta tanulmányozni) a nyelv több, mint a gondolatközlés eszköze. A nyelv valójában fontos szerepet játszik a gondolat megformálásában.

Sőt, azt is mondhatjuk, hogy az ember oly módon érzékeli az őt körülvevő világot, ahogyan azt az anyanyelve már eleve beprogramozta számára.

Hasonló a helyzet a programozási nyelvekkel is.

NYELVEK

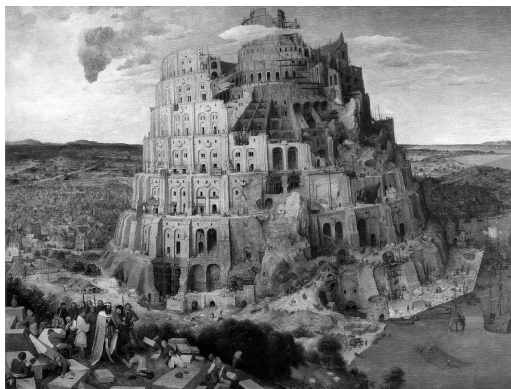
A középkorban hátrányos helyzetben volt az a diák, aki nem tudott latinul **ÉS** görögül, mert nem tudta két különböző nézőpontból szemlélni a világot.

(írja Patrick Winston cambridge-i professzor
Ivan Bratko: Prolog című könyvének előszavában)

Hasonló a helyzet a programozási nyelvekkel is.

NYELVEK

Hány nyelvet beszélnek a világon?



Brueghel: Bábel tornya

kb.:

3 – 10 ezer között

ebből írott nyelv kb.:

2-300

PROGRAMOZÁSI NYELVEK

Imperatív nyelvek:

„Hogyan” oldjuk meg a feladatot – a Turing gép modellen alapszik.

Változó: egy memóriarekesz kap nevet és értéket.

A program állapota: a változó nevek/értékek és az éppen végrehajtott utasítás. Program: állapotátmenetek sorozata.

Imperatív program: állapotok és az őket módosító utasítások.

Eljárás: állapotátmenetek bizonyos értelemben jellemző sorozata.

Procedurális programozás: imperatív programozási eljárások együttese.

PROGRAMOZÁSI NYELVEK

Deklaratív nyelvek:

Nem az a kérdés, hogy „hogyan”, hanem, hogy „mi” a megoldandó feladat. ⇒ A feladat pontos megfogalmazásán van a hangsúly.

Változó: egy érték kap nevet (mint a matematikai változó fogalma).

A megoldás megkeresése a futtató környezet dolga.

EGY KIS TÖRTÉNELEM

- Prolog: Robert Kowalski ötlete
- Első Prolog megvalósítás: Alain Colmerauer 1972, Marseille-i egyetem
- Első magyar megvalósítás: Szeredi Péter 1975.
- Első fordítóprogram D.H.D Warren 1977, (DEC10, WAM, Warren Abstract Machine)
- 1981: japán 5. generációs elképzelések
- Magyar Prolog rendszerek: MPROLOG (Szeredi P. és társai), CS-Prolog (Futó Iván)
- Prolog ISO szabvány: 1995.

LOGIKAI PROGRAMOZÁS – PROLOG

(PROgramming in LOGic / PROgrammation en LOGique)

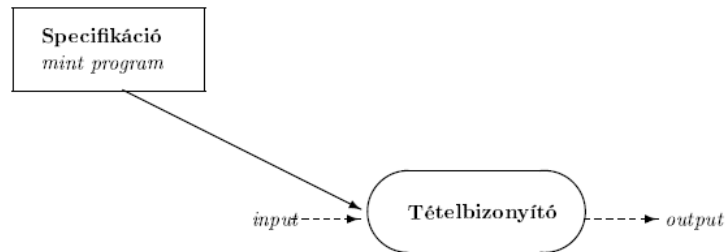
Egy logikai program axiómák (tények) és következtetési szabályok halmaza.

A tények bizonyos ismereteket reprezentálnak, melyekből a szabályok segítségével következtethetünk újabb ismeretekre.

A program kiértékelése ezen axiómák és szabályok következményeinek meghatározása.

Az összes lehetséges következmény halmazát tekintjük a program jelentésének.

A LOGIKAI PROGRAMOZÁS SÉMÁJA



Egy Prolog program futási eredménye:
yes vagy no (true vagy false).

LOGIKAI PROGRAMOZÁS – PROLOG

Axiómák (tények): igaznak tekintett állítások.

Amiről állítunk valamit: TERM

Amit állítunk: PREDIKÁTUM (prédikálni)

Pl.: kék(ég)

Másképp: színe(ég, kék)

Később még pontosítunk kicsit.

LOGIKAI PROGRAMOZÁS – PROLOG

Következtetési szabályok:

HA *feltétel*₁ ÉS *feltétel*₂ ÉS ... *feltétel*_n AKKOR *következmény*

A következményen van a hangsúly, ezért a szabály:

következmény HA *feltétel*₁ ÉS *feltétel*₂ ÉS ... *feltétel*_n.

Prolog szintaktikával:

következmény :- feltétel₁, feltétel₂, ...feltétel_n.

LOGIKAI PROGRAMOZÁS – PROLOG

Logikai háttér:

Implikáció: $A \rightarrow B$.

Horn klóz: $A_1 \wedge A_2 \wedge \dots \wedge A_n \rightarrow B$.

Fordított sorrendben:

$B \leftarrow A_1 \wedge A_2 \wedge \dots \wedge A_n$.

LOGIKAI PROGRAMOZÁS – PROLOG

Feladatok:

Hogyan lehet megfogalmazni azt, hogy

1. olcsó a sör?

a/ tényállításként

b/ szabállyal (HF)

2. minden diák okos?

LOGIKAI PROGRAMOZÁS – PROLOG

A tárgyalandó Prolog: SWI-Prolog

<http://www.swi-prolog.org>



Szintaktikai szabályok:

- Minden mondat végén pont. (Adatbevitelkor is.)
- Nagy kezdőbetű: **változó**, kis kezdőbetű: **konstans**.

FONTOS:

- A paraméter nélküli predikátumnál NINCS üres zárójel!
- Predikátumnév és zárójel között NEM LEHET szóköz!

LOGIKAI PROGRAMOZÁS – PROLOG

MÉG EGY FONTOS MEGJEGYZÉS:

A predikátumok korrekt jelölése:

predikátumnév/argumentumszám



Minden predikátum kis betűvel kezdődik!!

PÉLDÁK

1. Feladat:

Ünnepeljük meg az első Prolog órát, menjünk kocsmázni! Vigyázat, csak virtuálisan 😊!

Vagyis: tudjuk, hogy melyik diák milyen italt szeret.
Azt is tudjuk, melyik kocsmában milyen italt árulnak.
Kérdés: ki hova megy, ha mindenki olyan helyre jár,
ahol mérnek olyan italt, amit ő szeret.

Hogyan oldaná meg ugyanezt a feladatot
C-ben (vagy C#-ban vagy Java-ban)?



PÉLDÁK

2. Feladat:

Valaki sertést, kecskét és juhot vásárolt, összesen 100 db-t, pontosan 100 aranyért.

A sertés darabja 3 és fél arany, a kecskéé 1 és egyharmad, a juhoké fél arany. Hány darabot vett az egyes állatokból?

