

# PLANG

## A PLanG környezet futtatása

- A plang.jar fájlt le lehet tölteni és haza lehet vinni, a java -jar plang paranccsal lehet elindítani
- A futtatáshoz szükség van egy Java 5 futtató környezetre. Szükség esetén a [Sun oldaláról](#) lehet letölteni (futtatáshoz elég a JRE).

## A programokban használható típusok és műveleteik

Logikai   Egész   Valós   Karakter   Szöveg   Tömb   Fáj

Műveleti sorrend   Felépítési szabályok

### Logikai típus

Típusértékek:

IGAZ, HAMIS

Műveletek:

Értékadás: logValt := LogKif

Logikai eredményű kifejezések:

Logikai konstansok és logikai típusú változók  
(LogKif): zárójelezés  
NEM LogKif: logikai *nem*  
LogKif1 ÉS LogKif2: logikai és  
LogKif1 VAGY LogKif2: logikai vagy

### Egész szám típus

Típusértékek:

Pozitív és negatív egész számok és a nulla (0, 1, 2, -1, -2, ...)

Műveletek:

Értékadás: egValt := EgKif  
Beolvasás: BE: EgValt  
Kíírás: KI: EgKif

Egész eredményű kifejezések:

Egész konstansok és egész típusú változók  
(EgKif): zárójelezés  
- EgKif: negáció  
|EgKif|: abszolút érték  
EgKif1 + EgKif2: összeadás  
EgKif1 - EgKif2: kivonás  
EgKif1 \* EgKif2: szorzás  
EgKif1 DIV EgKif2: egész osztás (az eredmény törtrésze elveszik)  
EgKif1 MOD EgKif2: osztási maradék  
RND EgKif: véletlenszám előállítás a [0..EgKif) intervallumban

Valós eredményű kifejezés:

VALÓS EgKif: ugyanaz az érték valós típussal

Logikai eredményű kifejezések:

EgKif1 = EgKif2: egyenlőségvizsgálat  
EgKif1 /= EgKif2: nem egyenlő  
EgKif1 < EgKif2: kisebb  
EgKif1 > EgKif2: nagyobb  
EgKif1 <= EgKif2: kisebb vagy egyenlő  
EgKif1 >= EgKif2: nagyobb vagy egyenlő

### Valós szám típus

Típusértékek:

Pozitív és negatív valós számok és a nulla (0, 3.141, -0.06, ...)

Műveletek:

Értékadás: valValt := ValKif  
Beolvasás: BE: ValValt  
Kíírás: KI: ValKif

Valós eredményű kifejezések:

Lebegőpontos konstansok és valós típusú változók  
Egész típusú kifejezések  
(ValKif): zárójelezés  
- ValKif: negáció  
ValKif1 + ValKif2: összeadás  
ValKif1 - ValKif2: kivonás  
ValKif1 \* ValKif2: szorzás  
ValKif1 / ValKif2: osztás  
ValKif1 ^ ValKif2: hatványozás  
|ValKif|: abszolút érték  
SIN ValKif: SZINUSZ; COS, TAN, ARCSIN, ARCCOS, ARCTAN, LOG, EXP: értelemszerűen

Egész eredményű kifejezések:

EGÉSZ ValKif: törtész levágása  
KEREK ValKif: kerekítés

Logikai eredményű kifejezések:

ValKif1 = ValKif2: egyenlőségvizsgálat  
ValKif1 /= ValKif2: nem egyenlő  
ValKif1 < ValKif2: kisebb  
ValKif1 > ValKif2: nagyobb  
ValKif1 <= ValKif2: kisebb vagy egyenlő  
ValKif1 >= ValKif2: nagyobb vagy egyenlő

### Karakter típus

Típusértékek:

Betűk, számjegyek, írásjelek, szóköz, sorvége jel ('A', 'x', ' ', '?', 'ø', sv)

Műveletek:

Értékadás: karValt := KarKif  
Beolvasás: BE: KarValt  
Kíírás: KI: KarKif

Karakter eredményű kifejezések:

Karakter konstansok, karakter típusú változók  
(KarKif): Zárójelezés  
NAGY KarKif: nagybetűvé alakít  
KIS KarKif: kisbetűvé alakít

Logikai eredményű kifejezések:

KarKif1 = KarKif2: egyenlőségvizsgálat  
KarKif1 /= KarKif2: nem egyenlő  
KarKif1 < KarKif2: ábécésorrendben előtte  
KarKif1 > KarKif2: ábécésorrendben utána  
KarKif1 <= KarKif2: előtte vagy egyenlő  
KarKif1 >= KarKif2: utána vagy egyenlő  
BETŰ KarKif: betű-ø?  
SZÁM KarKif: számjegy-ø?

### Szöveg típus

Típusértékek:

Karakterek (akár üres) sorozatai ("", "alma", "több szó")

Műveletek:

Értékadás: szValt := SzKif  
Karakter megváltoztatása: SzValt[EgKif] := KarKif  
Sor beolvasása: BE: SzValt  
Kíírás: KI: SzKif

Szöveg eredményű kifejezések:

Szövegkonstansok, szöveg típusú változók  
(SzKif): zárójelezés  
SzKif1 + SzKif2: szövegek összefűzése  
KarKif + SzKif: karakter szöveg elé fűzése  
SzKif + KarKif: karakter szöveg után fűzése  
SzKif[EgKif1:EgKif2]: részsöveg lekérdezés, az EgKif1 pozíción kezdődő és EgKif2 pozíció előtt végződő rész az eredmény

Egész eredményű kifejezések:

|SzKif|: hossz lekérdezése  
SzKif @ KarKif: karakter első előfordulásának helye (a karakter indexe; ha nem fordul elő, akkor |SzKif| az eredmény)  
SzKif1 @ SzKif2: részsöveg első előfordulásának helye (a részsöveg első karakterének az indexe; ha nem fordul elő, akkor |SzKif1| az eredmény)

Karakter eredményű kifejezések:

SzKif[EgKif]: karakter lekérdezése

Logikai kifejezések:

SzKif1 = SzKif2: egyenlőségvizsgálat  
SzKif1 /= SzKif2: nem egyenlő  
SzKif1 < SzKif2: ábécésorrendben előtte  
SzKif1 > SzKif2: ábécésorrendben utána  
SzKif1 <= SzKif2: előtte vagy ugyanaz  
SzKif1 >= SzKif2: utána vagy ugyanaz

### Tömb típus

Jelölés:

Alaptípus[Méret]<sub>1</sub>...[Méret]<sub>n</sub>], ahol Méret<sub>i</sub> egész konstans.

Típusértékek:

Ha dim=1: Méret<sub>1</sub> darab *Alaptípus* típusú érték (Méret<sub>1</sub> elemű vektor)  
Ha dim=2: Méret<sub>1</sub> darab *Alaptípus*[Méret<sub>1</sub>] típusú tömb (Méret<sub>1</sub> × Méret<sub>2</sub>-es mátrix)  
Ha dim=n: Méret<sub>1</sub> darab *Alaptípus*[Méret<sub>1</sub>]<sub>1</sub>...[Méret<sub>n</sub>]<sub>1</sub> típusú tömb (n dimenziós tömb)

Műveletek:

Értékadás: TombValt := TombKif  
Értékadás tömbelemnek: TombValt[EgKif] := Kif  
Értékadás tömbelemnek: TombValt[EgKif1]...[EgKifn] := Kif  
Kíírás: KI: TombKif

Egész eredményű kifejezések:

|TombKif|: elemszám lekérdezése (Méret<sub>1</sub> az eredmény)

Tömb eredményű kifejezések:

TombVal[EgKif1]...[EgKifn]: tömbelem lekérdezés; ha n<dim, akkor az eredmény egy dim-n dimenziós tömb

Alaptípus eredményű kifejezések:

TombVal[EgKif1]...[EgKifdim]: tömbelem lekérdezés

### Fáj típusok

Jelölés:

BEFÁJL és KIFÁJL

Típusértékek:

BEFÁJL: karaktorsorozatot tartalmazó fájl (névvel azonosítva), illetve lezárt fájl  
KIFÁJL: karakterekkel feltölthető fájl (névvel azonosítva), illetve lezárt fájl

Műveletek:

Megnyitás: MEONYIT FileValt: "fájlnév.txt"  
Lezárás: LEZÁR FileValt  
Írás: KI KifFileValt: ...  
Olvasás: BE BefileValt: ...

Logikai eredményű kifejezés:

VEGE BefileValt: akkor ad vissza igazat, ha a legutóbbi beolvasó művelet nem sikerült (például mert nincs több adat a fájlban).

### Műveleti sorrend

- Prefix műveletek: NEM, -kif, SIN, ...
- Hatványozás: ^, keresés: @
- Multiplikatív műveletek: \*, /, DIV, MOD
- Additív műveletek: +, -
- Relációs műveletek: =, />, <, ...
- Logikai műveletek: ÉS, VAGY

### A programok felépítésének szabályai

A szabályok leírásakor használt jelölések:

- Az idézőjelek közötti szövegek a programszöveg építőelemei, egy az egyben kerülnek bele a forráskódba
- A szükséges zárójelek elhagyható részeket jelölnek
- A szögletes zárójelek ímeltései jelölnek: a zárójelek közötti teljes rész egymás után tetszőleges sokszor szerepelhet (akár nullaszer is, tehát el is hagyható).
- A betűk a magyar ABC kis- és nagybetűi, a számjegy-ek a számjegyek 0-tól 9-ig, az [Egyéb](#)-ek tetszőleges egyéb karakterek (szóköz, írásjelek, és bármi amit ki lehet csalni a billentyűzetből). A "" a sima felső jobbra dőlő dupla idézőjelet, a "" a sima felső jobbra dőlő aposztrófot jelöl.

### A szabályok

```
Program ::= { Megjegyzés ... }  
           "PROGRAM" Azonosító  
           Deklarációk  
           Utasítások  
           "PROGRAM_VEGE"  
  
Deklaráció ::= [ "VÁLTOZÓK" ":"  
               VÁLTOZÓDeklaráció  
               { ", " VÁLTOZÓDeklaráció ... } ]  
  
VÁLTOZÓDeklaráció ::= Azonosító { ", " Azonosító ... } ":" Típus  
  
Utasítások ::= Utasítás { [ ", " ] Utasítás ... }  
  
Utasítás ::=      Értékadás      |  
                  Előtagadás    |  
                  Ciklus       |  
                  InputOutput  |  
                  Megjegyzés   |  
  
Értékadás ::=     BalÉrték ":"=" Kifejezés  
  
Előtagadás ::=    "HA" Kifejezés "AKKOR"  
                  Utasítások  
                  [ "KÜLÖNBEN"  
                    Utasítások ]  
                  "HA_VEGE"  
  
Ciklus ::=        "CIKLUS"  
                  Utasítások  
                  "AMEG" Kifejezés |  
                  "CIKLUS AMIG" Kifejezés  
                  Utasítások  
                  "CIKLUS_VEGE"  
  
InputOutput ::=  "BE" [ Azonosító ] ":"  
                  BalÉrték { ", " BalÉrték ... } |  
                  "KI" [ Azonosító ] ":"  
                  Kifejezés { ", " Kifejezés ... } |  
                  "MEGNYIT" Azonosító ":" Szöveg |  
                  "LEZÁR" Azonosító  
  
Megjegyzés ::=  """" Jel { Jel ... }  
  
BalÉrték ::=     Azonosító { "[ " Kifejezés "]" } ... }  
  
Kifejezés ::=    Kifejezés "ÉS" RelációKifejezés |  
                  Kifejezés "VAGY" RelációKifejezés  
  
RelációKifejezés ::= RelációKifejezés "=" ÖsszegKifejezés |  
                     RelációKifejezés ">" ÖsszegKifejezés |  
                     RelációKifejezés "<" ÖsszegKifejezés |  
                     RelációKifejezés "<=" ÖsszegKifejezés |  
                     RelációKifejezés ">=" ÖsszegKifejezés  
  
ÖsszegKifejezés ::= ÖsszegKifejezés "+" SzorzatKifejezés |  
                     ÖsszegKifejezés "-" SzorzatKifejezés  
  
SzorzatKifejezés ::= SzorzatKifejezés "*" IndexKifejezés |  
                     SzorzatKifejezés "/" IndexKifejezés |  
                     SzorzatKifejezés "DIV" IndexKifejezés |  
                     SzorzatKifejezés "MOD" IndexKifejezés  
  
IndexKifejezés ::= IndexKifejezés "" ElőtagKifejezés |  
                     IndexKifejezés "8" ElőtagKifejezés  
  
ElőtagKifejezés ::= "-" ElőtagKifejezés |  
                     "NEM" ElőtagKifejezés |  
                     "SIN" ElőtagKifejezés |  
                     "COS" ElőtagKifejezés |  
                     "TAN" ElőtagKifejezés |  
                     "ARCSIN" ElőtagKifejezés |  
                     "ARCCOS" ElőtagKifejezés |  
                     "ARCTAN" ElőtagKifejezés |  
                     "LOG" ElőtagKifejezés |  
                     "EXP" ElőtagKifejezés |  
                     "RND" ElőtagKifejezés |  
                     "KIS" ElőtagKifejezés |  
                     "NAGY" ElőtagKifejezés |  
                     "BETŰ" ElőtagKifejezés |  
                     "SZÁM" ElőtagKifejezés |  
                     "KEREK" ElőtagKifejezés |  
                     "EGÉSZ" ElőtagKifejezés |  
                     "VALÓS" ElőtagKifejezés |  
                     "VEGE" Azonosító  
                     AlapKifejezés [ KifejezésUtótag ] |  
                     Konstans  
  
AlapKifejezés ::= Azonosító |  
                  "[" Kifejezés "]" |  
                  "(" Kifejezés ")"  
  
KifejezésUtótag ::= "[" Kifejezés "]" [ KifejezésUtótag ] |  
                    "[" Kifejezés "]"  
  
Típus ::=         "EGÉSZ" |  
                  "VALÓS" |  
                  "LOGIKAI" |  
                  "KARAKTER" |  
                  "SZÖVEG" |  
                  "BEFÁJL" |  
                  "KIFÁJL" |  
                  Típus "!" "EgészSzám "]"  
  
Konstans ::=      EgészSzám |  
                  Tizedestört |  
                  Szöveg |  
                  Karakter |  
                  "SV" |  
                  "IGAZ" |  
                  "HAMIS"  
  
EgészSzám ::=    Számjegy { Számjegy ... }  
  
Tizedestört ::=  Számjegy { Számjegy ... }  
                  "." Számjegy { Számjegy ... }  
  
Szöveg ::=       """" Jel ... """  
  
Karakter ::=     "" Jel ""  
  
Azonosító ::=    Betű { AzonosítóJel ... }  
  
AzonosítóJel ::= Betű |  
                  Számjegy  
  
Jel ::=          Betű |  
                  Számjegy |  
                  EgyébJel
```