

**1 GL (első generációs prog. nyelvek):** gépi kód, precíz megfogalmazás, gépfüggő, közvetlen értelmezés. előny: leggyorsabb programok, utasítások egységes/kétcímes formában vannak írva gépi kód után Assembly, egyszerűbb megfogalmazás,

**2GL:**

a programokat minél gyorsabban kellett megírni, a gépi kód és az Assembly nem ilyen

60-as évek: magas szintű prog. nyelvek, itt már a probléma a lényeg, nem a gép

1:N nyelvek, sokkal kisebb a hibalehetőség, áttekinthetőbb forrás.

**3GL:**

a 2GL-ben mindegyik progi 1-1 feladatot tudott jól, ezek már nem. 60-as évek vége. Minden

probléma megoldására nem létezik nyelv. 3 csoportja van: általános célú, magas szintű nyelvek;

objektum-orientált nyelvek; specializált nyelvek.

**4GL:**

Itt már nem a „hogyan” a lényeg, hanem hogy mi a feladat, mert már mindenre van megoldás. A

kérdés nem az algoritmus, hanem a cél, mert a kód generálása már automatikusan készül el.

Vizsgálják a szintaktikai hibáit a forrásnak, és a hatékonyságát is.

**FORTRAN:** Fortran: Az IBM-nél egy kutatócsoport, amelyet John Backus vezetett, az algebrai nyelv fordítási lehetőségeit kutatta. Ennek a kutatásnak az eredményeként jött létre 1954-ben a FORTRAN (FORmula TRANslation -formulafordítás). Egyik legrégebben használatos magas szintű programozási nyelv. Ennek megfelelően elég sok változata létezik, amelyek túlnyomórészt a korszerűsítések eredményei. A magas szintű programozási nyelvek közül talán a FORTRAN áll a legközelebb az assembly szinthez. Elsősorban matematikai számítások, fizikai kutatóintézetekben szükséges számítások elvégzésére készült. Állománykezelése nem igazán fejlett, karakterek kezelésére nem nagyon használják. Azóta több újabb verziója is megjelent. A különféle Fortran verziókat a megjelenési évük feltüntetésével különböztetik meg, a Fortran szó mögötti két számjegy erre utal. A jelenleg is használatban lévő verziók:

- Fortran 66
- Fortran 77
- Fortran 90 (95)

Manapság a leginkább elterjedt a 77-es verzió, de egyre népszerűbb a 90-es is. A Fortran 95 a 90-es verzió módosítása, mely 1996-ban jelent meg. Természetesen ezeken kívül vannak speciális verziók is, jó példa erre a HPF (High Performance Fortran = Magas Szintű Fortran). Fontos, hogy a legtöbb Fortran 77-es szerkesztő speciális kiegészítéseket is támogat, pl nem alapértelmezett kiterjesztéseket is elfogad. E jegyzet azonban az alapértelmezett Fortran 77-et ismerteti. A másik fontos előnye a Fortrannak a mobilitás, vagyis, hogy két követelményrendszernek (ANSI-nak és az ISO-nak) is eleget tesz. Tehát, ha például a programunkat ANSI Fortran 77-ben írtuk és csak a standard alapverziót vettük igénybe, programunkat bármilyen Fortran 77 szerkesztővel rendelkező gépen is le tudjuk majd futtatni, azaz a Fortran programok szabadon mozgathatók a különféle gépek platformok között. 1957, IBM, műszaki és tudományos nyelv, van egy főprogram (program és end között) és egymásba nem ágyazható alprogramok (function..end és subroutine..end). sok verziója volt (66, 77, II. ANSI), a 90-est még ma is használják, 2000-es a legújabb.

**ALGOL:** 1955, matematikai és műszaki nyelv, 60 és 68, szimbólumrendszer, blokk, szabad formátumú nyelv, PROCEDURE...END.

**COBOL:** 1959: CODASYL bizottság, 1963: COBOL - adatfeldolgozós nyelv, 1974: elfogadott szabvány nyelv lett. Minden gépen használható volt, tehát hamar elterjedt. Kötött szerkezetű, 4 szerkezeti blokkja van: Identification Division (azonosító rész), Environment D. (környezethívó), Data D. (adat), Procedure D. (eljáráshívó). Ma is használatos, 89-től objektumorientált. Hátrányok: nagy méretű programok és nem tudja a bonyolult műveleteket.

**BASIC:** 1963, a Fortran egyszerűsített változata, de annál jobb. Gyakran kellett ugró utasításokat alkalmazni (GOTO), interpreter mód (utasításonkénti működés), ezért könnyű tesztelés. Nagyon megváltozott mára, Quick, GFA, MS, Visual a főbb állomások. Visual az egyetlen strukturált.

**PL/1 (Programming Language 1):** 1964, IBM, előnyei: mindenféle alkalmazásra jó, könnyű, egyszerű, de bonyolult feladatokat is ellát, szabad formátumú, blokkszerkezet (strukturálthoz jól illeszkedik).

1976: PL/C, ezzel jó tanulni.

#### PASCAL: Pascal:

Ez volt az első igazi és legteljesebb körű megvalósítása a struktúrált szemléletű programozás elveinek. Niklaus Wirth professzor, a zürichi Műszaki Főiskola oktatója dolgozta ki a nyelvet a 70-es években, és Blaise Pascal francia matematikus-filozófusról nevezte el. Wirth professzor az ALGOL programozási nyelv alapján definiálta a PASCAL szabályait, a cél egy az ALGOL-hoz hasonló, de általános célú programozási nyelv elkészítése volt. Általános célú programozási nyelv, amelyet elsősorban az oktatásban, a műszaki-tudományos számítások területén és a komplex programrendszerek fejlesztése során alkalmaznak elterjedten. Szabad formátumú nyelv, mely viszonylag kevés utasítást tartalmaz. A programok szerkezete kötött, szigorú előírások vonatkoznak rá. A PASCAL programok blokkokból épülnek fel, a blokkok a begin foglalt szóval kezdődnek és az end foglalt szóval érnek véget. Minden utasítás végére „;” -t kell tenni. Támogatja a moduláris programozást, függvények és eljárások definiálhatók a function és a procedure foglalt szavak segítségével. A függvények, eljárások egymásba ágyazhatók, sőt a nyelv támogatja a rekurziót is. Erősen típusos nyelv. A programban használni kívánt változókat a főprogram előtt elhelyezkedő deklarációs részben fel kell tüntetni. A PASCAL fordítók compiler típusú fordítók. A programok szabályos, szigorú, egyértelmű felépítése miatt a nyelv az oktatás kedvelt programozási nyelve. Jó alapot biztosít más programozási nyelvek tanulásához, a struktúrált programozás alapelveinek megértéséhez.

#### C, C++:

1970-ben dolgozta ki Ken Thompson a B nyelvet és ezen készült el az első UNIX operációs rendszer. Mivel a B nyelv nem volt elég hatékony, ezért 1971-ben Dennis Ritchie kifejlesztette a C nyelvet. 1983-ban létrejött az első szabványos C nyelv. Végül is az ANSI C szabvány végleges változata 1989-ben jött létre. C nyelven írják a UNIX operációs rendszert, és ehhez hozzátartozik általában a C fordító is. Elterjedését segíti, hogy nagyon hatékony programozási nyelv és a C forrásprogramok elég jól hordozhatóak különböző platformok között. Ha egy C programban csak szabványos dolgokat használunk, akkor valószínűleg mindenhol futni fog, minimális átalakítással. Manapság a C nyelv szerepét a C++ veszi át. Végül is a C++ a C nyelv objektum-orientált kiterjesztése. Általában a C programok gond nélkül használhatóak C++ rendszerben is és a fordítók többsége is ismeri mindkettőt. Szokás a C nyelvet alacsony szintű programozási nyelvként is említeni. Ez azonban csak azt jelenti, hogy általában a számítógép által támogatott adattípusokkal dolgozik. Támogat olyan alacsony szintű műveleteket is, amelyet más magas szintű programozási nyelvek nem támogatnak. A C sajátossága, hogy támogatja bitstruktúrák kezelését is. A C nyelv nem tartalmaz I/O utasításokat. Ezeknek a műveleteknek az elvégzésére függvénykönyvtárak vannak létrehozva. A C nyelv nem tartalmaz műveleteket stringek, halmazok, listák és tömbök kezelésére sem. Azonban a függvénykönyvtárak segítségével több hasznos eszközhöz juthatunk, mint más programozási nyelvekben.

**Objektum orientált programozási nyelvek:** egyszerűbb, hatékonyabb és biztonságosabb, mint a többi. Az elemek előre definiáltak, csak a tulajdonságokat kell átírni. Sok új nyelv, amik csak OO-nyelvek (TISZTA OO nyelvek, pl Eiffel, SmallTalk); a régieket pedig kiegészítették OO-ra (VEGYES nyelvek, pl TurboPascal, Borland C++).

Lekérdező nyelvek: adatbázisok adataihoz szolgáltatják a hozzáférést

Programgenerátorok: előre definiált minta alapján egy hibátlan 3GL program létrehozása

Egyéb: döntéstámogató rendszerek nyelvei, prototípus nyelvek, formális specifikáció nyelvek.