

The background features abstract, overlapping green geometric shapes, primarily triangles and polygons, in various shades of green, creating a modern and dynamic visual effect.

# Programozási nyelvek

kialakulása, fejlődése, generációk

# Program, programozás

## ► Algoritmus

Egy probléma megoldásának véges számú részlépésben történő egyértelmű és teljes leírása.

## ► Program

A számítógép számára érthető utasítássorozat.

## ► Programozás

A feladat kidolgozásától kezdődően a kész program leírásáig tartó folyamat lépései.

# Az első programozók - számítógép-készítők

Charles Babbage

- ▶ Csillagászati táblázatok, vízállás-diagramok, hajózási térképek
- ▶ 1822: Difference Engine
- ▶ 1830-as évek: Analitikal Engine

Herman Hollerith

- ▶ Népszámlálási adatok feldolgozása (USA)
- ▶ 1890: lyukkártyás számítógép

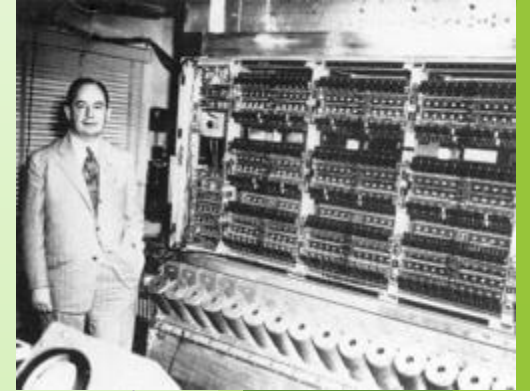
Konrad Zuse

- ▶ 1938 Z-1 számítógép
- ▶ 1946 Plankalkül programozási nyelv (az első)



# A kezdetek:

- ▶ 1952: Neumann János tervei, elkészül az EDVAC, az első Neumann-elvek alapján épített számítógép
- ▶ elsősorban matematikai problémák megoldására specializálták, pl: meteorológia
- ▶ a programok „gép-függőek” voltak
- ▶ a programozás tulajdonképpen nem volt más, mint adott utasítások sorozatának leírása
- ▶ a programok megírása sok időt igényelt



# 60-as, 70-es évek:

- ▶ hosszú, nehéz munkával készített programok gyorsan elavultak
- ▶ új, gyorsabb, egyszerűbb programfejlesztési módszerek
- ▶ első magas szintű programozási nyelvek
- ▶ programozási módszerek, technikák, filozófiák jelentek meg
- ▶ általános célú és a specializált programozási nyelvek

# A 80-as, 90-es évek:

- ▶ objektumorientált programozás
- ▶ Script-nyelvek:
  - ▶ PHP (szerver oldal),
  - ▶ Javascript (kliens oldal), Perl, VBScript (Visual Basic Script), Actionscript

Ezen scriptnyelvek némelyike tekinthető hibridnyelvnek is, ugyanis lehet bennük eljárás és objektumorientáltan is programozni. Pl.: PHP, Javascript, Actionscript

# A programozási nyelvek generációi:

## 1. generáció: alacsony szintű nyelvek

### 1. gépi kód

- ▶ a processzor közvetlenül végre tudja hajtani
- ▶ felépítése:
  - ▶ utasítás, memóriacím vagy
  - ▶ utasítás, memóriacím, memóriacím
- ▶ hátránya a gépfüggőség, nehéz programírás
- ▶ előnye a gyorsaság
- ▶ Példa: 0B0A 0C0E 0D00

# A programozási nyelvek generációi:

## 1. generáció: alacsony szintű nyelvek

### 2. Assembly

- ▶ nem bináris, hanem
  - ▶ Utasításoknak angol nyelvű szöveges kód (mnemonik), pl. ADD
  - ▶ Memóriacímeket el lehet nevezni
- ▶ Gépi kódnak megfelelő felépítés
- ▶ Assembler szükséges (fordítóprogram)
- ▶ Példa: ADD, X, Y
- ▶ *Ma már csak hardver szintű vezérléshez használatos.*

# A programozási nyelvek generációi:

## 2. generáció: magas szintű nyelvek

- ▶ A programokra megnő az igény, széleskörű alkalmazás  
Programfejlesztés idejének csökkentésének igénye
- ▶ Előny: áttekinthetőség, gyors programírás, kevés hibalehetőség,  
gépfüggetlenség
- ▶ Hátrány: fordítóprogram szükséges (lassabb mint az Assembly)
  - ▶ Fortran (matematikai, tudományos területen ma is)
  - ▶ Algol (kutatás, oktatás → Pascal)
  - ▶ Cobol (adatfeldolgozásban ma is)
  - ▶ Basic (oktatás)
- ▶ a nyelvek egy-egy feladatcsoport megoldására specializálódtak

# A programozási nyelvek generációi:

## 3. generáció: magas szintű nyelvek

- ▶ strukturált programozás: utasítások és adatok tekintetében is
- ▶ Általános célú nyelvek
  - ▶ Modula-2
  - ▶ Ada (rendszerprogramozás, gazdasági számítások)
  - ▶ Pascal (általános)
  - ▶ C (UNIX rendszerprogramozás, majd általános)

# A programozási nyelvek generációi:

## 4. generáció: magas szintű nyelvek

- ▶ a 80-as évek közepén jelentek meg
- ▶ nem csak nyelvek, hanem magas szintű nyelvre épülő komplex, objektumorientált programfejlesztő környezetek
- ▶ Előnye: egyszerűbb, hatékonyabb, biztonságosabb programkészítés közelít az emberi gondolkodáshoz
  - ▶ Turbo Pascal
  - ▶ Borland C++
  - ▶ Borland Delphi
  - ▶ Visual Basic
  - ▶ Visual C++
  - ▶ Java

# A programozási nyelvek generációi:

## 5. generáció: magas szintű nyelvek

Párhuzamos programozás

Hálózatok programozása

- ▶ Központilag vezérelt számítógép-hálózatok, döntéshozást támogató rendszerek fejlesztése

Mesterséges intelligencia

- ▶ A „programozónak” csak azt kell a számítógép tudtára adni mi a probléma, milyen feltételek mellett kell megoldani, de a hogyan már a számítógép dolga.
  - ▶ Prolog, Mercury