

## HOGYAN KEZDŐDIK A PROGRAMOZÁS? MI AZ A PROGRAMOZÁS?

**1. Specifikáció:** A feladat készítője kitalálja, mit kell tudnia a programnak, milyen végkimenetel legyen, milyen legyen a kinézete, a funkciója, milyen szintű legyen a felhasználóval való kapcsolattartása. A tisztázott igények a *specifikációk*.

**2. Tervezés:** A megoldandó feladat *adatszerkezeteket* és *algoritmusokat* kell használni. Ezek megtervezése, összegyűjtése, esetleges programbővítés kérdése mind a tervezés feladata.

**3. Kódolás:** A tervezés és a specifikáció alapján egy programnyelvre alakítjuk át a feladatot. Az elkészült *forráskódból* egy fordítóprogram hozza létre a futtatható *gépi kódú* programot.

**4. Tesztelés:** A kész program futását ellenőrizni kell, először tartalmilag. Minden esetben helyesen működik-e? Mennyi tárhelyet használ, mennyire gyors, mennyire érthető a felhasználók számára.

**5. Hibajavítás:** A hibákat javítani kell, új kódot (remélhetőleg) nem kell írni, a meglévőket kell optimalizálni.

**6. Dokumentáció:** A program futásának leírása. Egyrészt a felhasználók számára, másrészt a fejlesztők számára.

### PROGRAMOZÁSI KÖRNYEZETEK:

- **Operációs rendszer:** *platform*.
- **Fejlesztő környezet:**

**szövegszerkesztő** (source editor): *forráskód elkészítése*

**fordítóprogram** (compiler): a forráskódot lefordítja, jelzi a szintaktikai hibákat, és elkészíti a *tárgykódot*

**szerkesztő** (linker): a tárgykódból elkészíti a *futtatható programot*

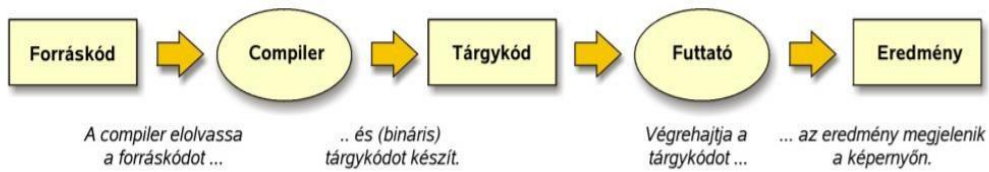
**hibakereső** (debugger): a program lépésenkénti futtatása, és a változók értékének alakulása

### PROGRAMOZÁS FOLYAMATA:

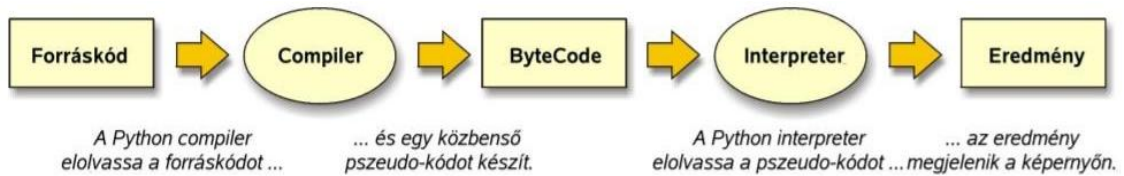
Interpreter: nem generálódik tárgyprogram, csak gépi program szintjén



## Compiler: Tárgyprogramot készít



## Python:



## HIBAKERESÉS: (DEBUG)

Szintaktika: formai szabályok megszegése

Szemantika: tartalmi hibák megjelenése

Run time error: futtatás közbeni hibák megjelenése