

Compilálás és interpretálás

Vastag Atila

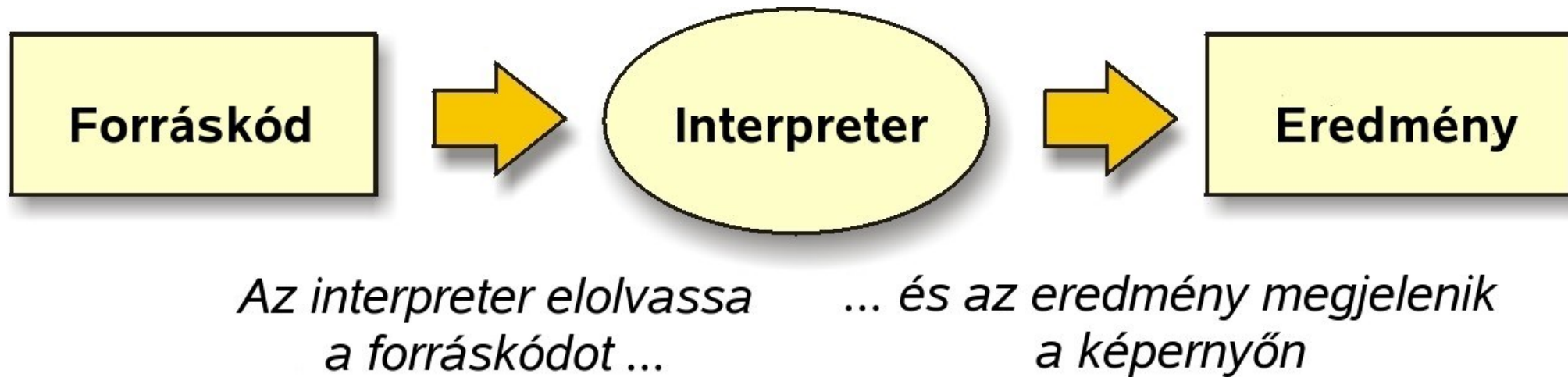
2020

A programot, ahogyan azt egy szerkesztő programmal (egyfajta speciálizált szövegszerkesztővel) megírjuk mostantól fogva **forrásprogramnak** (vagy forráskódnak) nevezzük.

Két fő technika létezik arra, hogy egy ilyen forráskódot a gép által végrehajtható bináris kódra fordítsunk :

- *interpretáció*
- *compilatio*

- Az interpretáció esetén minden egyes alkalommal, amikor végre akarjuk hajtatni a programot, az interpreter programot kell használnunk. Ennél a technikánál a fordító a forrásprogram minden egyes elemzett sorát néhány gépi nyelvű utasításra lefordítja, amiket azonnal végre is hajt. Semmilyen tárgyprogram sem generálódik (nincs futtatható állomány -> exe file).



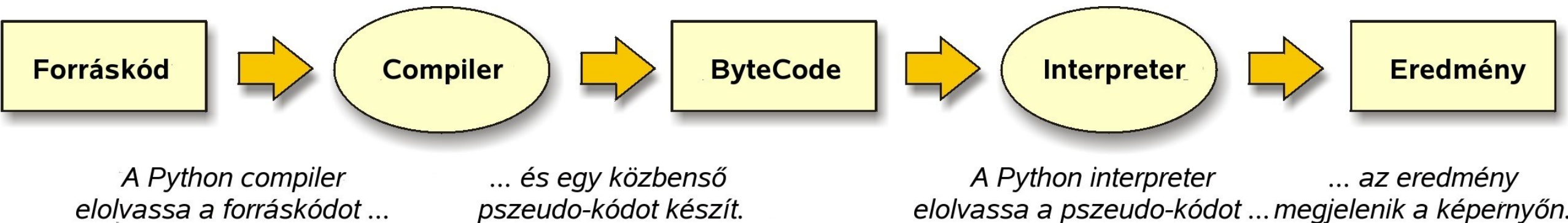
- A compilálás a teljes forrásszöveg egyszeri lefordításából áll. A fordító program elolvassa a forrásprogram összes sorát és egy új kódot állít elő, amit tárgykódnak (*object* kód) hívunk. Ez utóbbit mostmár a compilertől függetlenül végrehajthatjuk és tárolhatjuk egy fileban (*exe* file)



Mindegyik technikának vannak előnyei és hátrányai :

- Az interpretáció ideális amikor egy nyelvet tanulunk, vagy egy projekttel kísérletezünk. Ezzel a technikával közvetlenül tesztelhető a forráskód minden megváltoztatása anélkül, hogy átmennénk egy compilálási fázison, ami mindig időigényes.
- Ezzel szemben, amikor egy projektnek összetett funkciói vannak, amiket gyorsan kell végrehajtani, a compilatiót részesítjük előnyben. Nyilvánvaló, hogy egy compilált program mindig lényegesen gyorsabban fog működni, mint az interpretált változata, mivel a számítógépnek a végrehajtás előtt nem kell minden egyes utasítást újra bináris kódra lefordítani.

Egyes modern nyelvek megpróbálják a két módszert kombinálni, hogy mindegyikből a legjobbat hozzák ki. Ez a helyzet a **Python** és a **Java** esetében is. Amikor egy forráskódot adunk át a Pythonnak, az először egy gépi kódhoz hasonló közbenső kódot ú.n. bytecodeot hoz létre, amit aztán egy interpreternek ad át végrehajtásra. A számítógép szempontjából a bytecode ot nagyon egyszerű gépi nyelven interpretálni. Ez az interpretáció tehát sokkal gyorsabb lesz, mint egy forráskód interpretációja.



Ennek a módszernek az előnyei érzékelhetőek :

- Az, hogy állandóan rendelkezünk egy interpreterrel, lehetővé teszi bármilyen kis programrészlet közvetlen tesztelését. Egy alkalmazás bármelyik alkotójának helyes működését igazolhatjuk annak létrehozását követően.
- A lefordított bytecode interpretációja nem olyan gyors, mint egy igazi bináris kódé, de számos program számára, beleértve a grafikus programokat is, nagyon megfelel.
- A bytecode portábilis. Ahhoz, hogy egy **Python** vagy egy **Java** program különböző gépeken végrehajtsódjon elég ha mindegyik gépen rendelkezésre áll egy adaptált interpreter.