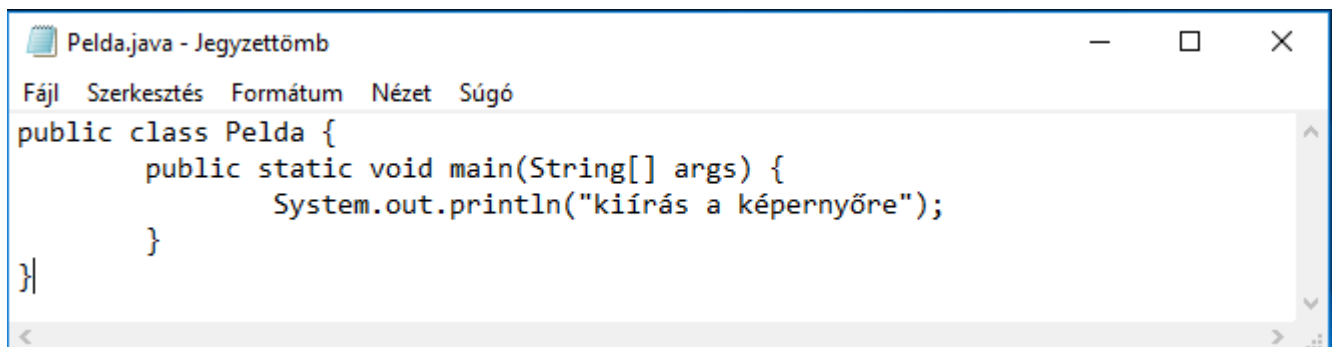


Java-ban készített programok fordítása és futtatása

A Java programkód vagy forráskód egy egyszerű szövegfájl (txt), java kiterjesztéssel. Bármely szövegszerkesztőben elkészíthető, akár **notepad**-el is. Mi **Eclipse**-et fogunk használni, de sok más fejlesztőkörnyezet is van, ami a java programok megírását, bytecode-ra fordítását és futtatását segítik.

Ahhoz, hogy a java forráskódot futtatható állománnyá fordítsuk (ez a java bytecode), java fordítóra van szükségünk, ami a JDK része. A JDK telepítése után (ez a Java Development Kit, pl. a <http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html> oldalról), a fordító alapértelmezetten a `C:\Program Files\Java\jdk-8.x.x\bin\` könyvtárban lesz **javac** néven. Ha egy java programfájlt ezzel lefordítunk, a keletkezett futtatható állomány az ugyanitt található **java** programmal tudjuk indítani.

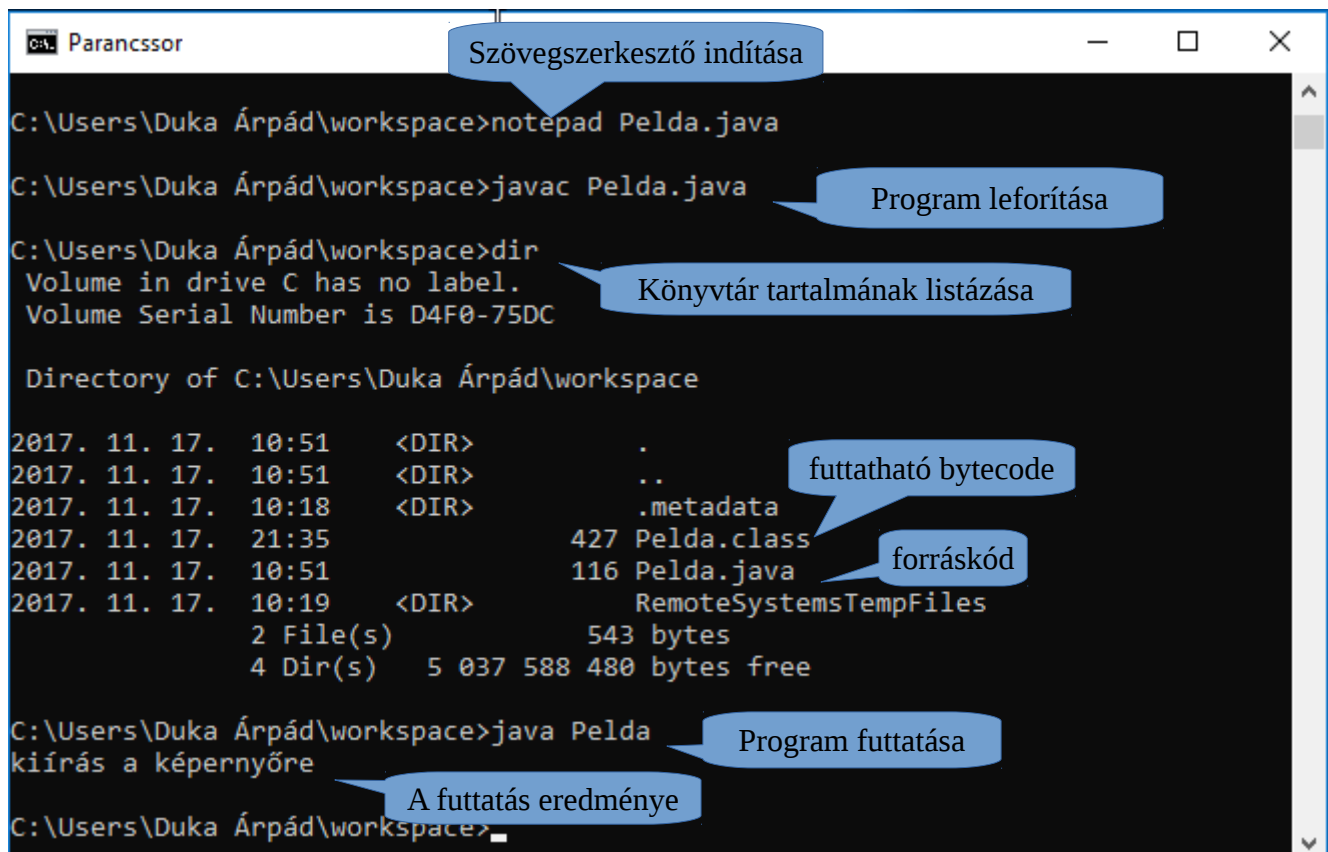
Például ha notepad-el a workspace könyvtárban készítünk egy Pelda.java programfájlt:



```
public class Pelda {
    public static void main(String[] args) {
        System.out.println("kiírás a képernyőre");
    }
}
```

parancssorban ezt lefordíthatjuk és lefuttathatjuk. Ha Pelda.java néven hozunk létre egy fájlt, akkor fontos, hogy benne a Pelda osztályt publikáljuk. A java programok esetén a fájlnev meg kell egyezzen a benne levő osztály nevével, s mivel az osztályok nevét nagy kezdőbetűvel írjuk, a fájl neve is ilyen lesz.

A fenti java programot parancssorban így lehet lefordítani és futtatni:



```
C:\Users\Duka Árpád\workspace>notepad Pelda.java

C:\Users\Duka Árpád\workspace>javac Pelda.java

C:\Users\Duka Árpád\workspace>dir
Volume in drive C has no label.
Volume Serial Number is D4F0-75DC

Directory of C:\Users\Duka Árpád\workspace

2017. 11. 17. 10:51 <DIR>      .
2017. 11. 17. 10:51 <DIR>      ..
2017. 11. 17. 10:18 <DIR>      .metadata
2017. 11. 17. 21:35      427 Pelda.class
2017. 11. 17. 10:51      116 Pelda.java
2017. 11. 17. 10:19 <DIR>      RemoteSystemsTempFiles
                2 File(s)          543 bytes
                4 Dir(s)    5 037 588 480 bytes free

C:\Users\Duka Árpád\workspace>java Pelda
kiírás a képernyőre

C:\Users\Duka Árpád\workspace>
```

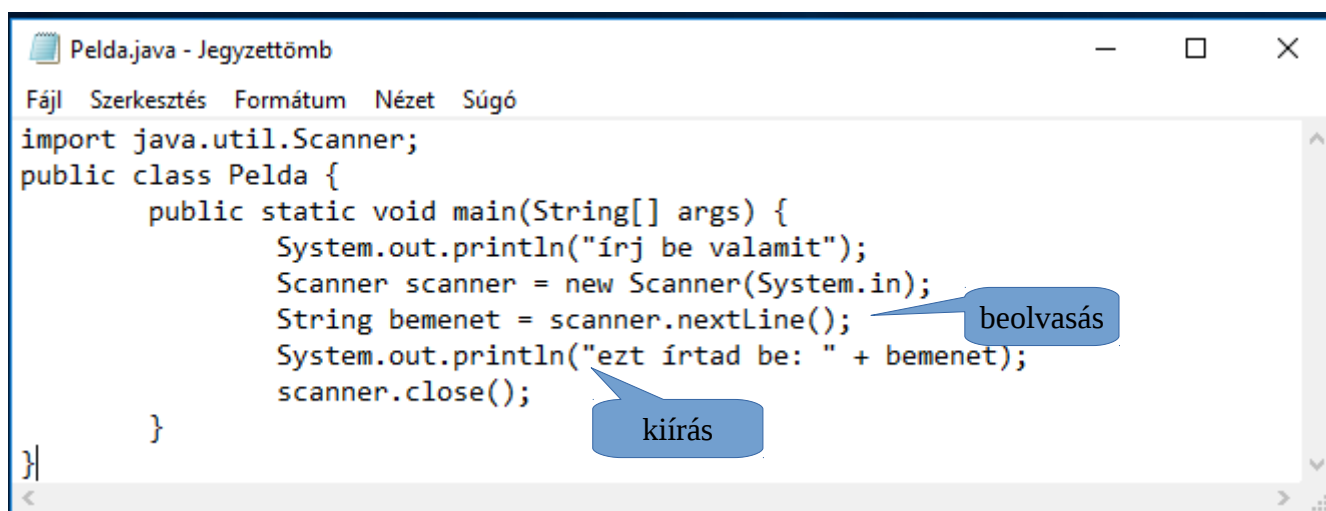
Mint látjuk, a Pelda.java forrás lefordítása után egy Pelda.class fájl jött létre, azonban amikor ezt lefuttatjuk, a fájl kiterjesztését nem kell megadnunk, csak az osztály nevét (**java Pelda.class** helyett csak **java Pelda**).

A windows parancssorában mindig látható az aktuális könyvtár útvonala. Könyvtárt váltani a **cd** paranccsal lehet (**cd alkonyvtar1** → belépés az alkonyvtar1-be, **dc ..** → visszalépés a szülő könyvtárba).

Kimenet, bemenet

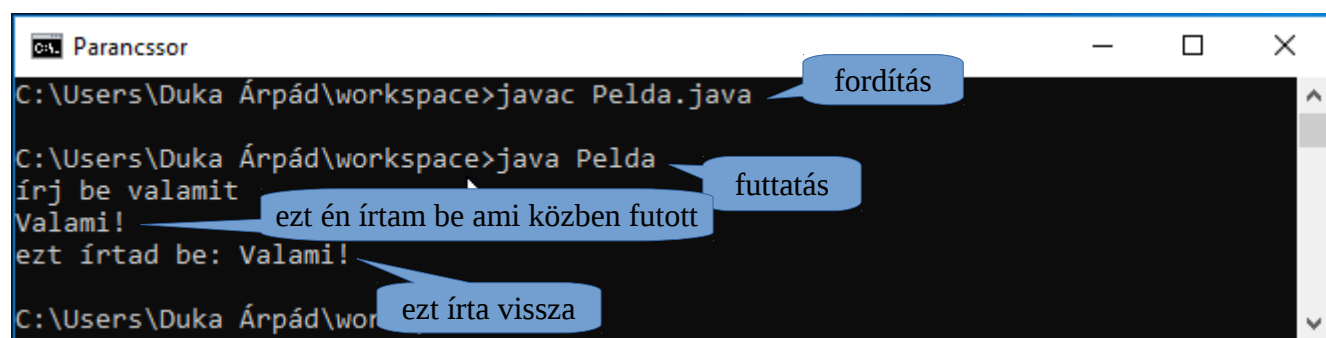
Amikor egy programot futtatunk egy számítógépen többféle bemenete és többféle kimenete lehet. Ezek azok a csatornák, amelyeken a program adatokat fogad és adatokat közöl vissza. Bemenetek: billentyűzet, eger, fájl, hálózati kapcsolat, adatbázis, mikrofon stb., kimenetek: képernyő, hangszóró, hálózat, fájl, nyomtató stb.

Az egyszerű java programok parancssorban futtathatók (mint fent láttuk), ide írják ki a kimenetüket és itt is fogadják a bemenetet. Amikor a program a bemenetről szeretne olvasni (pl. a **scanner.nextLine()** utasítással), a parancssor kurzora ott fog villogni, várja hogy valamit beírunk, beírás és Enter után pedig a program feldolgozza az általunk beírtakat. Példa:



```
Pelda.java - Jegyzettömb
Fájl Szerkesztés Formátum Nézet Súgó
import java.util.Scanner;
public class Pelda {
    public static void main(String[] args) {
        System.out.println("írj be valamit");
        Scanner scanner = new Scanner(System.in);
        String bemenet = scanner.nextLine();
        System.out.println("ezt írtad be: " + bemenet);
        scanner.close();
    }
}
```

Futtatása:



```
Parancssor
C:\Users\Duka Árpád\workspace>javac Pelda.java
C:\Users\Duka Árpád\workspace>java Pelda
írj be valamit
Valami!
ezt írtad be: Valami!
C:\Users\Duka Árpád\workspace>
```