

Java programozás

1. Bevezetés, alapok

TANTÁRGYI TEMATIKA – SORREND KICSIT ELTÉR

- Az objektumorientált programozás alapjai: osztály, objektum, tulajdonság.
- Java nyelv szintaxis: adattípusok, változók, műveletek, kifejezések.
- Java nyelvi vezérlő elemek. Tömbök, tömbkezelő segédosztályok (rendezés, keresés).
- Egybezártság, adatrejtés.
- Öröklődés.
- Többalakúság.
- Absztrakt osztályok.
- Interfészek.
- Aggregáció és kompozíció.
- Kivételkezelés.
- Összetett adatszerkezetek (lista, verem).
- Összetett adatszerkezetek (halmazok, leképezések).
- Generikus adatszerkezetek.
- Dátum és időkezelés javában.
- Összetett adatszerkezetek kezelése (iterátorok, rendezés).
- Adatfolyamok, fájlkezelés.
- Funkcionális tesztek (unit tesztek) alapjai.

BEVEZETÉS – JAVA NYELV

- Modell alapján → adatstruktúrák, algoritmusok
- **Objektum orientált** (üzenetküldés)
- **Való élet leképezése** → objektum
- Strukturálttal szemben: nyelvtől, géptől, OS-től független

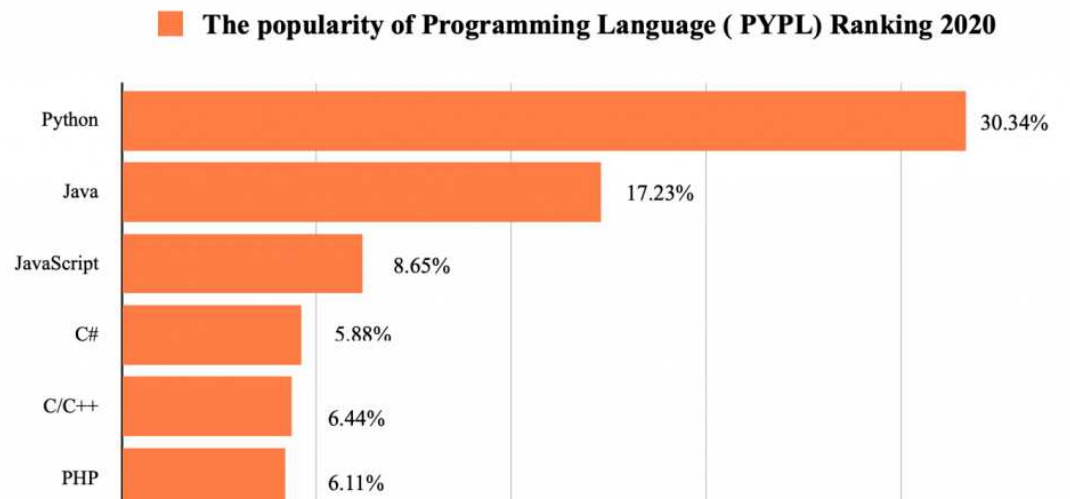
- Patrick Naughton, James Gosling, 1991
- Közbülső VM → bármely gépen futtatható
- **Interpreter jelleg:**
 - Egyszerre egy sort értelmez
 - Utasítás → natív kód, azonnal végrehajt
 - Végrehajtás után eldobja
- **C nyelven alapul**, egyszerűsített, kibővített.
- Első hivatalos verzió: 1995, Sun Microsystems

BEVEZETÉS – JAVA NYELV

2009: **Oracle** felvásárolta



Népszerűség:

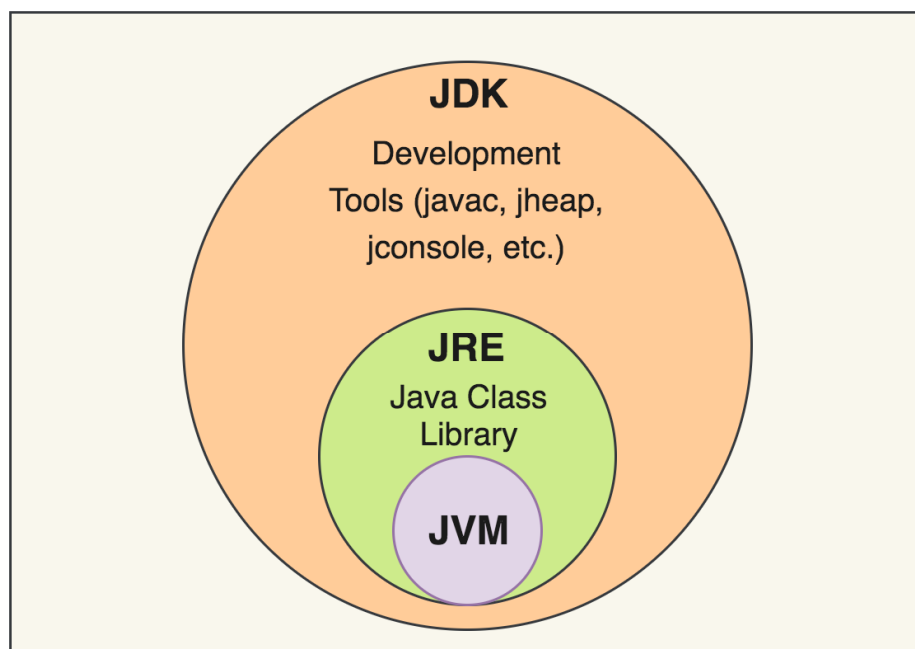


JAVA FŐ SPECIFIKÁCIÓI (SZEMPONTOK):

- Egyszerű (C++ egyszerűsítve, de kibővített lehetőségekkel)
- Tisztán objektumorientált
- Elosztott (hálózatos környezetben – Internet erőforrások)
- Robusztus (hibatűrő – hibák már fordítási időben)
- Biztonságos (hozzáférések, jogosultságok kezelése)
- Hordozható (class fájl bármilyen virtuális gépen)
- Architektúra független (fut bármilyen hw/sw architektúrán)
- Interpretált (kódlétrehozás futásidőben)
- Nagyteljesítményű (részben igaz)
- Többszálú (párhuzamos programozás/algorithmusok)
- Dinamikus (osztálykönyvtárak bővíthetők, fejleszthetők)

BEVEZETÉS – JRE VS. JDK

- Java ME (Micro Edition) – beágyazott eszközök
- Java SE (Standard Edition) – desktop programok
- Java EE (Enterprise Edition) – kliens-szerver, vállalati appok



JRE: java.exe
JDK: javac.exe

Javac.exe:
.\Program Files\java\jdk...
\bin\javac.exe

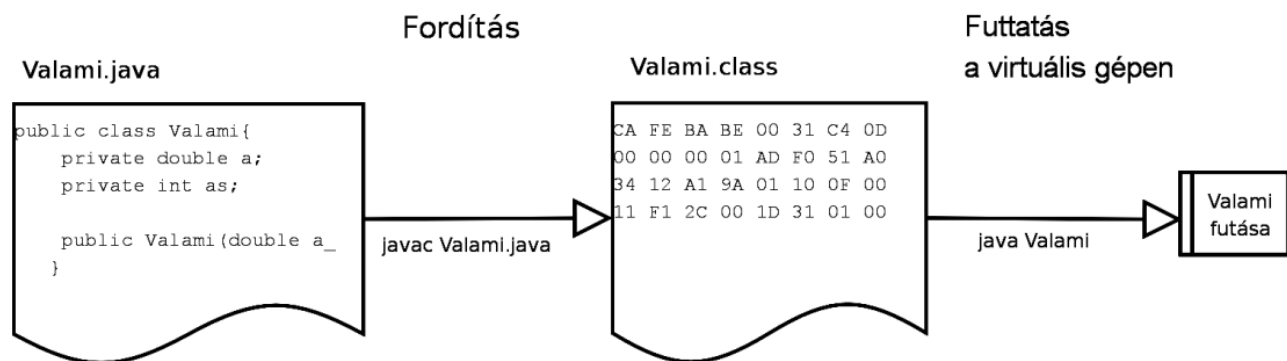
Verzió lekérdezése:
javac.exe - version

JAVA VM ÉS INTEPRETER FOGALMA

- Java virtuális gép: Java alkalmazások futtatására képes. Feladata a Java osztályokat tartalmazó class állományok betöltése, kezelése, gépi kódú utasításokká történő átalakítása. A Java forrásprogram bármilyen fordítóval ugyanazt a class fájlt állítja elő.
- Interpreter (értelmező): Olyan program, amely a forrásprogramnak egyszerre egyetlen utasítását értelmezi. Az utasítást ezután natív kóddá alakítja és azonnal végrehajtja. A lefordított kódot nem jegyzi fel, hanem rögtön el is dobja és a következő utasítás feldolgozásába kezd.

BEVEZETÉS – JAVA VM MŰKÖDÉSE

Fordítás, futtatás virtuális gépen



- Előzetesen lefordított **tárgykódú állományok** (class fájlok)
Minden környezetben bájról-bájtra ua. (hordozható!)
- Paraméterek
- Végrehajtás, **futtatás**

BEVEZETÉS – JAVA FEJLESZTŐI CSOMAG

- JDK: **Java Development Kit** (ajánlott: **Oracle**)



- Jelenleg: 17-es verzió, elegendő: **Java 8**

-Letöltés:

<https://www.oracle.com/java/technologies/downloads/#java8-windows>

- X64 illetve x86 verzió

- Van egy kiegészítő csomag is, példaprogramokkal

Product/file description	File size	Download
x86 Installer	156.45 MB	 jdk-8u301-windows-i586.exe
x64 Installer	169.46 MB	 jdk-8u301-windows-x64.exe

FEJLESZTŐKÖRNYEZETEK

- IntelliJ Idea (Community Edition)
(főleg Androidra, fejlett kódkiegészítés, magas gépigény)
- Oracle JDeveloper
(Oracle saját környezete, főleg ipari felhasználás)
- Eclipse
(gépigény, kiváló memóriakezelés)
- **NetBeans**
(Sun saját környezete volt
→ Apache égisze alatt)
- stb.



NETBEANS LETÖLTÉSE

- Letöltés:

<https://netbeans.apache.org/download/index.html>

- Régebbi verzió (Apache előtti):

<https://www.oracle.com/technetwork/java/javase/downloads/jdk-netbeans-jsp-3413139-esa.html>

- Installers:

- Apache-NetBeans-12.5-bin-windows-x64.exe (SHA-512, PGP ASC)
- Apache-NetBeans-12.5-bin-linux-x64.sh (SHA-512, PGP ASC)
- Apache-NetBeans-12.5-bin-macosx.dmg (SHA-512, PGP ASC)

Lépésről-lépésre telepítési útmutató (JDK, IDE, Path):

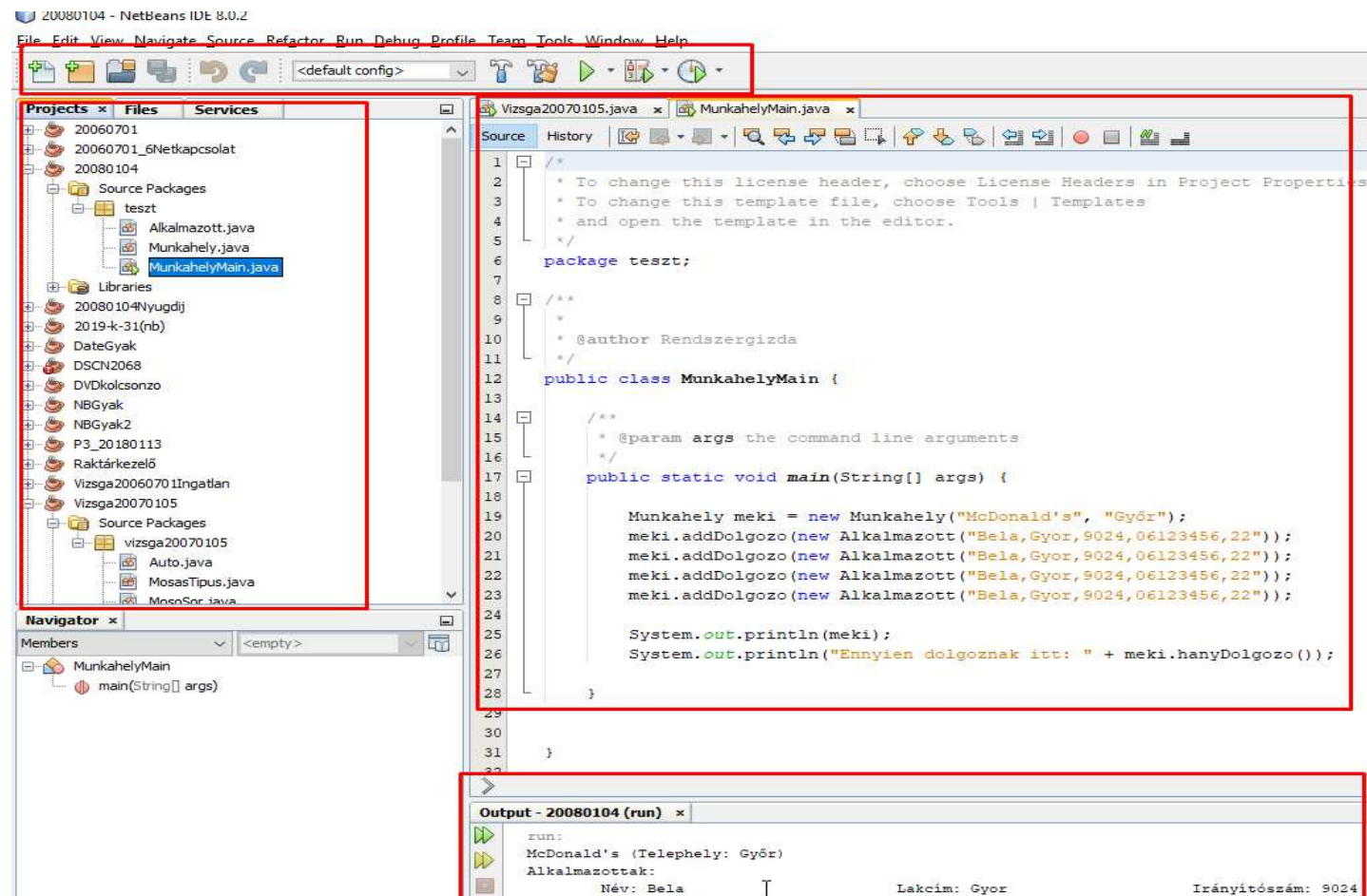
http://szt1.sze.hu/java/java_segedlet.html

NETBEANS KEZELŐFELÜLETE

IDE:

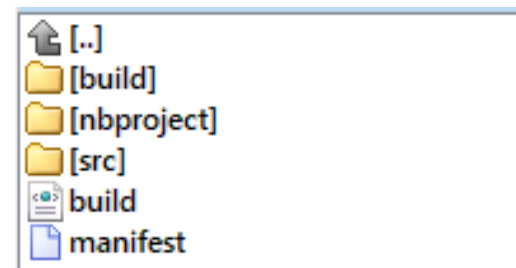
- Eszköztár
- Projekt
- Editor
- Konzol

Window ►
RESET




NETBEANS JELLEMZŐI

- IDE – Integrated Development Environment, „**ablakozós**”
- Grafikus felület (nagyobb jelentősége: Swing, Spring Boot)
- Kódszerkesztőben: sima, szövegfájlok (kiterjesztés: .java)
~~Jegyzet~~ → IDE **kódszerkesztő**
- **Case sensitive** (magyar ékezeteket NE használjunk)
- Buildelni lehet javac.exe-vel akár
~~Parancssorból~~ → IDE build funkciója
- **Könyvtárakba szervezett** projektek
 - Class fájlok (\build\classes)
 - XML projektleíró (\nbproject)
 - Java forrásfájlok (**\SRC**)

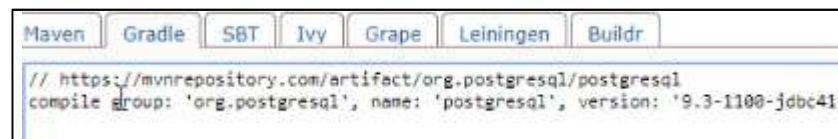


BUILD TOOL-OK

- Projektleíró fájl szerkezetét adják meg
 - **Ant (alapértelmezett)** – csak ezzel dolgozunk
 - Szerveroldali, 3rd party, adatbázis-appok:
Automatizálhatunk lépéseket (pl. .jar, szerverre másolás)
 - Maven (XML alapú függőségek használata)
 - Gradle (szöveges alapú konfigurációs fájl)



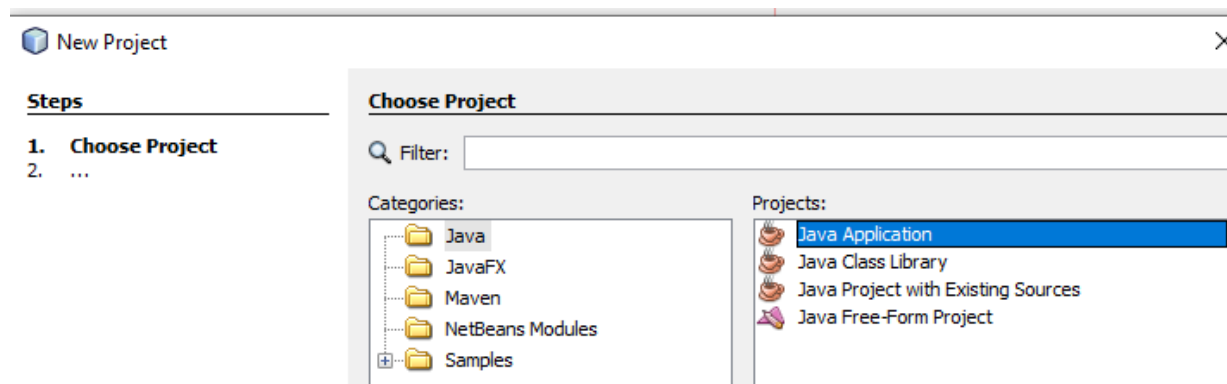
```
Maven  Gradle  SBT  Ivy  Grape  Leiningen  Buildr
<!-- https://mvnrepository.com/artifact/org.postgresql/postgresql -->
<dependency>
  <groupId>org.postgresql</groupId>
  <artifactId>postgresql</artifactId>
  <version>9.3-1100-jdbc41</version>
</dependency>
```



```
Maven  Gradle  SBT  Ivy  Grape  Leiningen  Buildr
// https://mvnrepository.com/artifact/org.postgresql/postgresql
compile group: 'org.postgresql', name: 'postgresql', version: '9.3-1100-jdbc41'
```

ÚJ PROJEKT LÉTREHOZÁSA

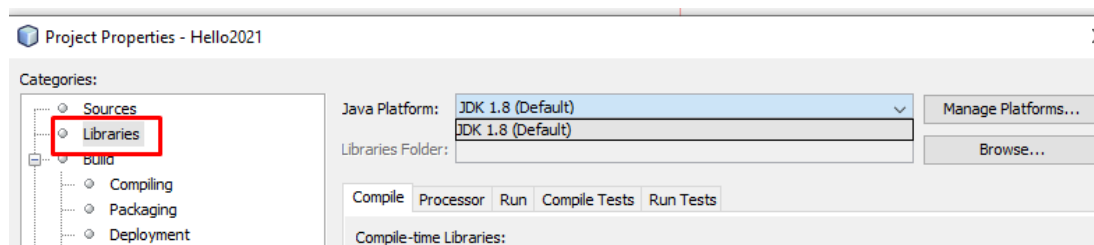
- File ► New Project ► Java Application



- Elnevezzük (angol ABC!), megadjuk az elérési utat
- Legyen írásjogunk a mappához
- Ne túl mélyen a fájlstruktúrában
- Main vagy sem?
- Finish: legyártja (☕ ikon)

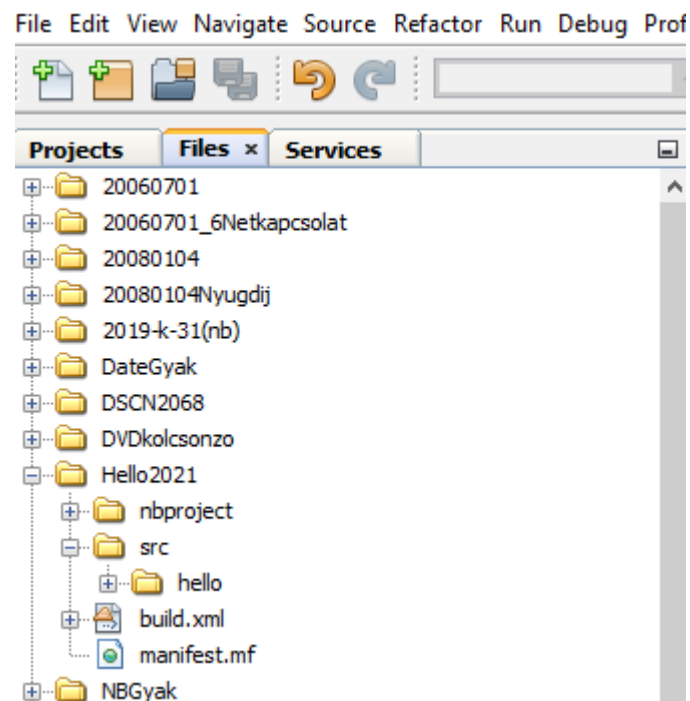
FELÉPÍTÉSE A PROJEKT TALLÓZÓBAN

- **Projekt neve** (Hello2021 – beszédes név legyen!)
- **Alapértelmezett csomag** (később)
- A JDK már be van hivatkozva a projektbe (Libraries\JDK...)
(változtatható: Properties ► Libraries)
- XML (build.xml → app neve, build tool, beállítások)



PROJEKT TALLÓZÓ - NÉZETEK

- Projects – teljes projekt felépítése
- Files – projekt fájlrendszer-szinten
- Services (web hozzáférés, Maven repó, adatbázisok)



JAVA FÁJL FELÉPÍTÉSE

- `/* blabla */` - kommentezés, törölhető
- **Kommentek** fajtái:
 - Blokk komment: `/* többsoros szöveg */`
 - Sor komment: `// megjegyzés, adott sorban`
 - Javadoc leíró: `/** dokumentációs kommentek */` → html
- **package** csomagnev; - csomag megadása (később)
- **public class app_neve** { ... }
 - Hozzáférés (public, protected... később)
 - Class előtag – osztály jelzése, név == java fájl neve (**elírva nem ok**)
- **Belépési pont** (csak main-nál): void – C-ből ismerős lehet
 - `public static void main(String[] args {.....})` (args: paraméterek)
- **Utasítások**
 - Itt: szöveg kiírása (`System.out.println("Hello 2021!");`);)

A MAIN FÜGGVÉNY

- A Java-ban írt programok **belépési pontja**.
- Minden alkalmazásnak tartalmaznia kell a következőképpen:

```
public static void main(String[] args) {...} ahol:
```
- **public:** jelzi, hogy a metódust más osztálybeli objektumokból is meg lehet hívni
- **static:** jelzi, hogy a Main osztálymetódus
- **void:** a metódusnak nincs visszatérési értéke
- A `String[] args` részben paramétereket adhatunk meg parancssori futtatáshoz: `<Osztály><p1><p2>...`
- Az alkalmazás futtatható parancssorból a `javac` fordítóval:
`javac Programneve.java param1, param2...`

ALAPTÍPUSOK, REFERENCIATÍPUSOK

byte	8 bites előjeles egész	byte largestByte = 127; * 2^8
short	16 bites előjeles egész	short largestShort = 32767; 2^15
int	32 bites előjeles egész	int largestInteger = 2147483647; 2^31
long	64 bites előjeles egész	long largestLong = 9223372036854775807L;
float	32 bites egyszeres lebegőpontosságú	float largestFloat = 3.4028235E38f; 10^38
double	64 bites kétszeres lebegőpontosságú	float largestFloat = 1.7976931348623157E308; 10 ^308
char	16 bites Unicode-karakter	char aChar = 'S';
boolean	logikai érték (igaz / hamis)	boolean aBoolean = true;

- A referenciatípusok objektumok.
- A legtöbb primitívnek van referencia párja, pl.:
int – Integer, char – String, boolean – Boolean

Referencia típus: osztály, tömb, enum, interface

Köszönöm a figyelmet!