

Spuštění programu z příkazové řádky

Table of Contents

Základní životní cyklus javového programu	1
Rychlé spuštění	1
Běžný postup	2
Nástroje ve vývojové distribuci	2
Překlad "Ahoj!"	2
Zdrojový kód "Ahoj!"	2
Spuštění "Ahoj!"	3
Překlad & Spuštění	3
Co když je třída v adresáři (balíku)	3
Od Java 11: překlad a spuštění v jednom	4
Praktické informace (aneb co je nutné udělat)	4
Spuštění sbaleného programu	4

Základní životní cyklus javového programu

- V Javě sice existuje interaktivní (tzv. REPL *read-eval-print-loop*) prostředí,
- většinou však kód *napišeme do souboru, uložíme, přeložíme a spustíme*.
- Zdrojový kód každé veřejné (**public**) třídy je umístěn v jednom souboru
- např. třída **Hello** je v **Hello.java**

Rychlé spuštění

- V nových verzích Javy lze psát i hodně zhuštěně, podobně jiným jazykům.
- Prakticky nepřilíš použitelné, hodí se skutečně na pár-řádkové věci.

Shebang (!) spouštění

```
#!/usr/bin/env java --source 23 --enable-preview
void main() {
    println("In new Javas, writing extremely simple apps is finally possible");
    var year = readln("Enter the current year (YYYY): ");
    println("Current year is " + year);
}
```



Soubor se nesmí jmenovat s příponou `.java`.

Běžný postup

- vytvoření zdrojového textu (libovolným editorem)
- překlad (nástrojem `javac`)
- spuštění (nástrojem `java`)
- anebo totéž jednodušeji zelenou šipkou v některém z IDE, třeba v IntelliJ IDEA

Nástroje ve vývojové distribuci

`javac`

překladač, tj. `Hello.java` → `Hello.class`

`java`

(nebo `jexec`) spouštěč přeloženého bytecode

`javadoc`

generátor dokumentace

`jar`

správce archivů JAR (sbalení, rozbalení, výpis)

`jshell`

interaktivní (REPL) prostředí

Pod Windows to jsou `.exe` soubory `java`, `javac`... umístěné v podadresáři `bin` instalace Javy.

Překlad "Ahoj!"

- Máme nainstalován Java **SDK** 23 (jen příklad, funguje i s dalšími verzemi)
- Jsme v adresáři `~/devel/pb112`, v něm je soubor `Hello.java`
- Spustíme *překlad* `javac Hello.java` název souboru je včetně přípony `.java`
- Je-li program správně napsán, přeloží se "mlčky"
- Vytvoří se soubor `Hello.class`

Zdrojový kód "Ahoj!"

Hello.java

```
public class Hello {  
    public static void main(String[] args) {
```

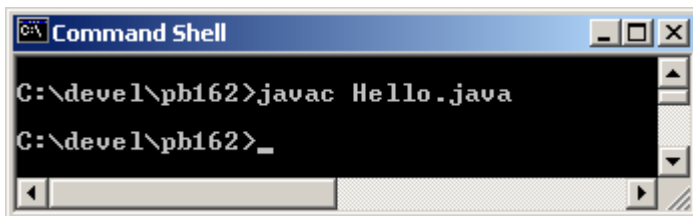
```
System.out.println("Ahoj!");  
}  
}
```

Spuštění "Ahoj!"

- Spustíme program Hello příkazem `java Hello`, název třídy je bez přípony `.class`
- V nejnovějších verzích Javy lze spustit rovnou včetně překladače: `java Hello.java`, pak se přeloží s hned spustí, viz dále.
- Je-li program správně napsán a přeložen, vypíše se `Ahoj!`

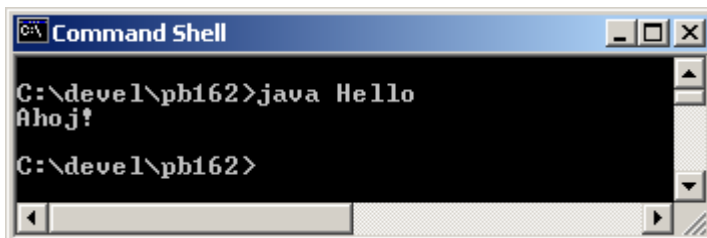
Překlad & Spuštění

Překlad překladačem `javac` (úspěšný, bez hlášení překladače):



```
C:\devel\pb162>javac Hello.java  
C:\devel\pb162>_
```

Spuštění voláním `java`:



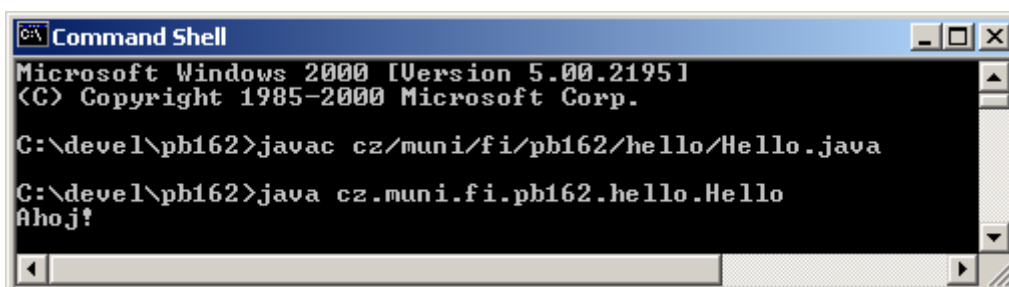
```
C:\devel\pb162>java Hello  
Ahoj!  
C:\devel\pb162>
```

Co když je třída v adresáři (balíku)

Když je třída v balíku, tj. na začátku souboru je:

```
package cz.muni.fi.pb112.hello;
```

Kompilace a spuštění pak vypadá následovně:



```
Microsoft Windows 2000 [Version 5.00.2195]  
<C> Copyright 1985-2000 Microsoft Corp.  
C:\devel\pb162>javac cz/muni/fi/pb162/hello/Hello.java  
C:\devel\pb162>java cz.muni.fi.pb162.hello.Hello  
Ahoj!
```

Pro **maven** projekty (všechny projekty na cvičení) je nutno být ve adresáři `src/main/java`.

Od Java 11: překlad a spuštění v jednom

- Počínaje Java 11 lze na jednoduché(!) programy použít přímý postup:
- `java HelloWorld.java` → program se přeloží a následně spustí
- funguje dokonce i tehdy, nebude-li se třída s metodou `main` jmenovat `HelloWorld`
- [330: Launch Single-File Source-Code Programs](#)

Praktické informace (aneb co je nutné udělat)

- Cesty ke spustitelným programům `PATH` musejí obsahovat i adresář `<JAVA_HOME>/bin`
- jako `C:\Program Files\Java\jdk9.0\bin`
- Systémové proměnné by měly obsahovat `JAVA_HOME=<adresář Javy>`
- Např. `JAVA_HOME=C:\Program Files\Java\jdk9.0`
- Možné je nastavit i proměnnou `CLASSPATH=<cesty ke třídám>`
- Např. `CLASSPATH=c:\devel\pb162`

Spuštění sbaleného programu

- Java má formát JAR (Java ARchive) pro distribuci programů složených z mnoha tříd.
- Je podobný ZIP s pomocnými soubory navíc.
- V archívu se deklaruje, která třída (=její metoda `main`) se spouští.
- Lze pak spouštět jakoby celý archív: `java -jar myapp.jar`