
Kapitola 1. Úvod do jazyka Java

Tomáš Pitner

Copyright © 2001-2004 Tomáš Pitner

Obsah

Úvod do jazyka Java	1
Java jako programovací jazyk... ..	2
Z toho plyne, že... ..	2
Další charakteristiky	2
Java jako běhové prostředí	2
Java pro programátora (1)	3
Java pro programátora (2)	3
Hlavní domény Javy (1)	3
Hlavní domény Javy (2)	4
Javová platforma	4
Java je tedy dána... ..	4
Vývoj Javy	4
Specifikace a implementace Javy	4
Verze Javy	5
Verze Javy - nově	5
Aktuální verze	5
Získání distribuce Javy	5
Stažení distribuce Sun	6
Obsah vývojové distribuce Javy	6
Obsah vývojové distribuce Javy (2)	6
Nástroje ve vývojové distribuci	6
Pomocné nástroje ve vývojové distribuci	7
Základní životní cyklus javového programu	7
Struktura javového programu	7
Demo "Ahoj!"	8
Překlad a spuštění "Ahoj!"	8
Vytvoření zdrojového textu "Ahoj!" ("for dummies")	8
Překlad "Ahoj!" ("for dummies")	9
Spuštění "Ahoj!" ("for dummies")	9
Co znamená spustit program?	10
Praktické informace (aneb co je nutné udělat)	10
Praktické informace (aneb co je vhodné udělat)	10
Úloha do cvičení 1.	11
Odkazy	11

Úvod do jazyka Java

- Úvod, srovnání s jinými, oblasti použití Javy
- Distribuce, instalace, použití Java SDK
- Životní cyklus programu v Javě
- Základní praktické úkony - *velmi podrobně!!!*

Java jako programovací jazyk...

- jazyk "3. generace - 3GL" (imperativní jazyk vysoké úrovně)
- *univerzální* (není určen výhradně pro specifickou aplikační oblast)
- *objektově-orientovaný* (výpočet je realizován jako volání metod/zasílání zpráv objektů)
- ideovým předchůdcem je C++ (a evt. Delphi) (C++ zbaveno zbytečností a nepříjemností)
- *jednodušší než C++*
- reálným soupeřem je (Microsoft) C# (zatím převážně na platf. Windows)

Z toho plyne, že...

- co se naučíme v Javě, v C# jako když najdeme...
- ale teď vážně: *Java podporuje vytváření správných návyků v objektovém programování*
- a naopak systematicky brání *přenosu některých špatných návyků z jiných jazyků*

Další charakteristiky

- program v Javě je *meziplatformně přenositelný* na úrovni *zdrojového i přeloženého kódu*
- je to umožněno tím, že přeložený javový program běží v tzv. *Java Virtual Machine* [<http://java.sun.com/docs/books/vmspec/>] (JVM)
- zdrojový i přeložený kód je tedy přenositelný mezi všemi obvyklými platformami (UNIX, Windows, MAC OS X, ale také sálové počítače, minipočítače typu IBM AS/400 apod.)
- tedy všude tam, kde *existuje příslušná JVM*

Java jako běhové prostředí

Kód je při běhu dobře zabezpečen:

- je možné nastavit úroveň přístupu k hostitelskému systému pomocí tzv. Security Manageru [<http://www.securingjava.com/chapter-two/chapter-two-8.html>]
- je možné ověřovat před spuštěním elektronický podpis kódu

Java pro programátora (1)

- jazyk vhodný pro *efektivní (rychlé) psaní přehledných programů* (mj. také díky *dokumentačním možnostem*)
- v průměru *vyšší produktivita* programátorské práce v Javě než v C++
- dnes již stejně aktivních programátorů v Javě jako v C++
- zdarma dostupné nezměrné množství knihoven pro různorodé aplikační oblasti, např. na SourceForge [<http://java.foundries.sourceforge.net/>] a tisících dalších místech
- k dispozici je řada kvalitních vývojových prostředí (i zdarma) - NetBeans [<http://www.netbeans.org/>], JBuilder [<http://www.borland.com/>], Visual Age for Java [<http://www.ibm.com/>], Eclipse [<http://eclipse.org/>], IDEA [<http://www.intelij.com/>]

Java pro programátora (2)

Konkrétní možnosti:

- v Javě se dobře píše *vícevláknové aplikace* (multithreaded applications)
- Java má automatické *odklizení nepoužitelných objektů* (automatic garbage collection)
- Java je jednodušší než C++ (méně syntaktických konstrukcí, méně nejednoznačností v návrhu)

Hlavní domény Javy (1)

- Škálovatelné výkonné *aplikace běžící na serverech* (Java Enterprise Edition [<http://java.sun.com/j2ee/>])
- Aplikace na *přenosných a vestavěných zařízeních* (Java Micro Edition [<http://java.sun.com/j2me/>])
- *Výukové účely* (nahrazuje Pascal jako referenční jazyk)
- Další přenositelné aplikace - např. klientské/desktopové

Hlavní domény Javy (2)

- webové aplikace (servlety, JSP - konkurence proprietárním ASP, SSI, ... a pomalým CGI)
- zpracování semistrukturovaných dat (XML)
- přenositelné aplikace s GUI
- aplikace distribuované po síti (applety nebo Java Web Start)

Javová platforma

Javovou platformu tvoří:

- **Java Virtual Machine**
- **překladač** (přístupný např. příkazem **javac**) a další vývojové nástroje
- **Java Core API** (základní knihovna tříd)

Java je tedy dána...

- **definicí jazyka** (Java Language Definition) - syntaxe a sémantika jazyka
- **popisem chování JVM**
- **popisem Java Core API**

Vývoj Javy

- *nejrychleji* se vyvíjí Java Core API
- chování JVM se mění např. pokud se objeví bezpečnostní "díra" nebo nelze-li dosáhnout požadované změny chování pomocí modifikace Java Core API
- daleko konzervativnější je samotný *jazyk* - *mění se zřídka*, ale přece: např. Java2, v1.4 přidáván nové klíčové slovo "assert", v1.5 přidá enum a další.

Specifikace a implementace Javy

- **Specifikace** Javy (tzv. "Editions") - např.: *Java 2 Standard Edition, v1.4*
- **Implementace** Javy ("Development Kits" nebo "Runtime Environments") - např.: *Java 2 Software*

Development Kit, v1.4.2 - obsahuje vývojové nástroje

- *Java 2 Runtime Enviroment, v1.4* - obsahuje jen **běhové prostředí** pro spouštění hotových přeložených pg.

Verze Javy

- hrubé členění - na verze "*Java* (před Java 2)" a "*Java 2*"
- číslování verzí:
 - tzv. major číslo, např. Java 2, **v1.4**
 - tzv. minor číslo, např. Java 2, v1.4.**2**
- změnu minor (třetího) čísla doprovází jen odstraňování chyb
- při změně major (druhého) čísla se může měnit Core API a někdy i jazyk
- ke změně prvního čísla dojde až teď se změnou pojmenovovacího schématu

Verze Javy - nově

- hrubé členění - na verze *Java 1.x.y* (před Java 2 v1.5) a *Java 5.z*

Aktuální verze

Stav k září 2004:

- Java 2 Standard Edition v1.4.2 pro všechny platformy je stabilní verzí
- Java 2 Standard Edition v1.5 neboli *Java 5* je k dispozici jako RC (Release Candidate), tj. je tzv. feature-complete (=nebude se nic přidávat, jen se vychytají chyby).
- Ostrá verze Java 5 bude dostupná v Q4 2004.
- aktuálně vždy na webu java.sun.com [<http://java.sun.com>]

Získání distribuce Javy

- používání Javy pro běžný vývoj (i komerční) je zdarma
- redistribuce javového **vývojového prostředí** je povolena pouze s **licencí** od Sunu

- redistribuce javového **běžového prostředí** je možná **zdarma**
- distribuce vyvíjí Sun Microsystems Inc. (Javasoftware) i další výrobci (např. IBM) a tvůrci Open Source

Stažení distribuce Sun

- `java.sun.com` [<http://java.sun.com>] (pro Windows, Solaris, Linux)
- dokumentace se stahuje z téhož místa, ale *samostatně* (nebo lze číst z WWW)
- celkově *vývojové prostředí J2SDK 1.4.2 vč. dokumentace* zabere cca **220 MB** na disku
- potřebná *velikost operační paměti* - min 64 MB, doporučeno **128 MB** (i více :-))

Obsah vývojové distribuce Javy

- **Vývojové nástroje (Development Tools)** v `bin` -- určené k vývoji, spouštění, ladění a dokumentování programů v Javě.
- **Běžové prostředí Javy (Java Runtime Environment)** se nalézá v `jre`. Obsahuje Java Virtual Machine (JVM), knihovnu tříd Java Core API a další soubory potřebné pro běh programů v Javě.
- **Přídavné knihovny (Additional libraries)** v podadresáři `lib` jsou další knihovny nutné pro běh vývojových nástrojů.
- **Ukázkové applety a aplikace (Demo Applets and Applications)** v `demo`. Příklady zahrnují i zdrojový kód.

Obsah vývojové distribuce Javy (2)

- **Hlavičkové soubory pro C (C header Files)** - v `include` - představují podporu pro psaní tzv. nativních metod přímo v jazyce C.
- **Staré hlavičkové soubory (Old Native Interface Headers)** - totéž, ale pro starší verzi rozhraní.
- **Zdrojový kód (Source Code)** knihoven z Java Core API se nalézá v archivu `src.jar`.
- **Dokumentace (Documentation)** - v podadresáři `docs` - obsahuje dokumentaci k dané verzi JDK, k API, nejruznější průvodce vývojem, dokumentaci k nástrojům, ukázkové programy a odkazy na související dokumentaci.

Nástroje ve vývojové distribuci

Pod Windows jsou to `.exe` soubory umístěné v podadresáři `bin`

- **java** - spouštěč (přeloženého bajtkódu)
- **javac** - překladač (.java -> .class)
- **javadoc** - generátor dokumentace API
- **jar** - správce archivů JAR (sbalení, rozbalení, výpis)
- **jdb** - debugger
- **appletviewer** - referenční prostředí pro spouštění appletů

Pomocné nástroje ve vývojové distribuci

- **javah** - generátor hlavičkových souborů pro C
- **javap** - disassembler bajtkódu (např. pro ruční optimalizace, hledání chyb)

Základní životní cyklus javového programu

- Program sestává z jedné (ale obvykle více) **tříd** (class)
- Zdrojový kód každé veřejně přístupné třídy je umístěn v jednom souboru (`NazevTridy.java`)
- Postup:
 - vytvoření zdrojového textu (libovolným editorem čistého textu) -> **Pokus.java**
 - překlad (nástrojem `javac`) **Pokus.java** -> **Pokus.class**
 - spuštění, např. **java Pokus**
- překládá se `javac` název souboru se třídou (včetně přípony `.java!!!`)
- spouští se vždy udáním `java` a názvu třídy (bez přípony `.class!!!`)

Struktura javového programu

- Každý netriviální javový program sestává z *více tříd* (class)
- Třídy jsou členěny do *balíků* (package)
- **Zařazení do balíků znamená mj. umístění zdrojového souboru do příslušného adresáře!!!**
- U běžné "desktopové" aplikace představuje vždy jedna (evt. více) tříd *vstupní bod* do programu - je to třída/y obsahující metodu `main`.

Demo "Ahoj!"

Zdrojový kód v souboru `tomp\ucebnice\Pozdrav.java`

```
package tomp.ucebnice;
public class Pozdrav {
    // Program spouštíme aktivací funkce "main"
    public static void main(String[] args) {
        System.out.println("Ahoj!");
    }
}
```

Třída `Pozdrav` je umístěna do balíku `tomp.ucebnice` ->

její zdrojový soubor musí být uložen v podadresáři `tomp\ucebnice`.

Překlad a spuštění "Ahoj!"

Překlad

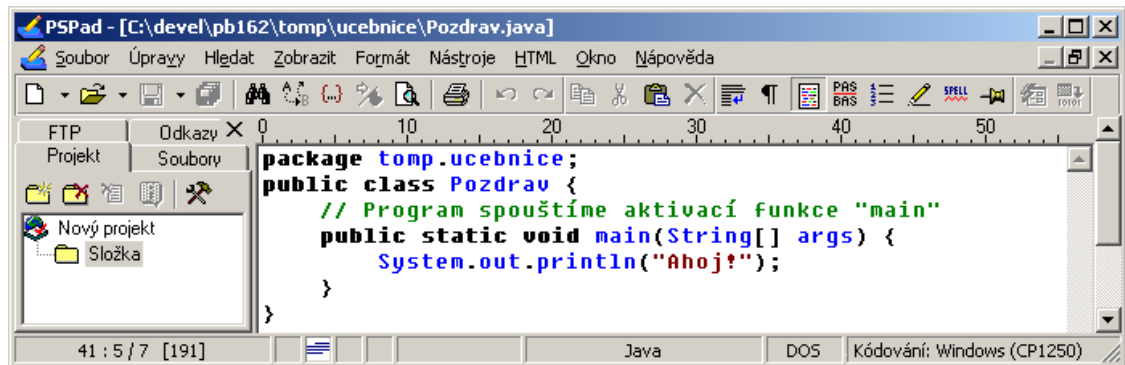
1. Máme nainstalován J2SDK 1.4.2
2. Jsme v adresáři `c:\devel\pb162`, v něm je podadresář `tomp\ucebnice`, v něm je soubor `Pozdrav.java`
3. Spustíme *překlad* `javac tomp\ucebnice\Pozdrav.java`
4. Je-li program správně napsán, přeloží se "mlčky"
5. (výsledný `.class` soubor bude v téže adresáři jako zdroj)

Spuštění

1. Poté spustíme *program* `Pozdrav: java -classpath . tomp.ucebnice.Pozdrav`
2. Volba překladače `-classpath` adresář zajistí, že (dříve přeložené) třídy používané při spuštění této třídy budou přístupné pod adresářem *adresář*.
3. `-classpath .` tedy značí, že třídy (soubory `.class`) se budou hledat v odpovídajících podadresářích aktuálního adresáře (adresáře `.`)
4. Je-li program správně napsán a přeložen, vypíše se `Ahoj !`

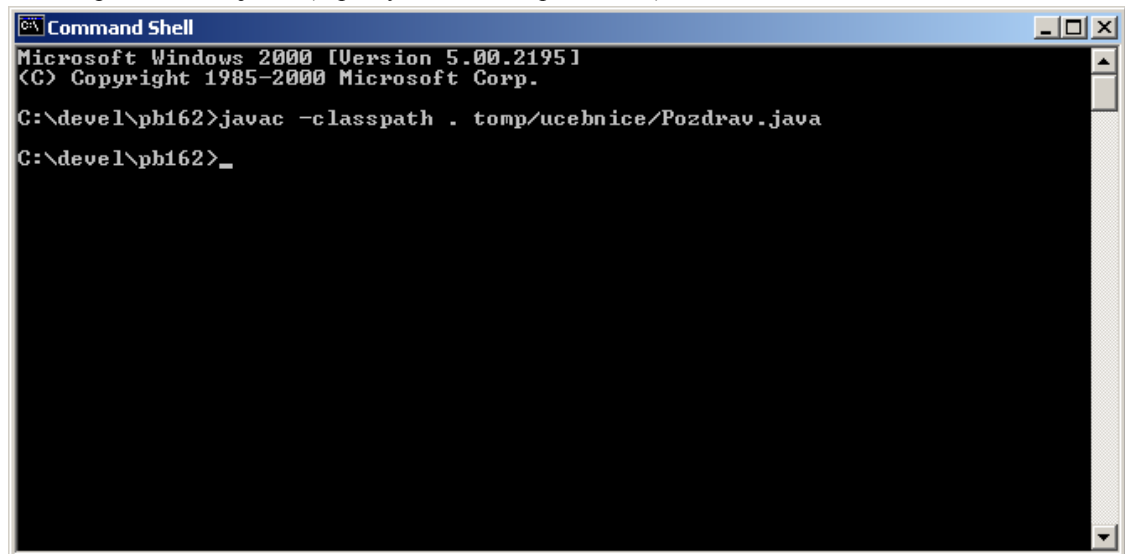
Vytvoření zdrojového textu "Ahoj!" ("for dummies")

Vytvoření a editace zdrojového kódu v editoru PSPad 4.2.2 [<http://pspad.zde.cz>] (dostupný zdarma, instalovaný na všech Win strojích v učebnách na FI)



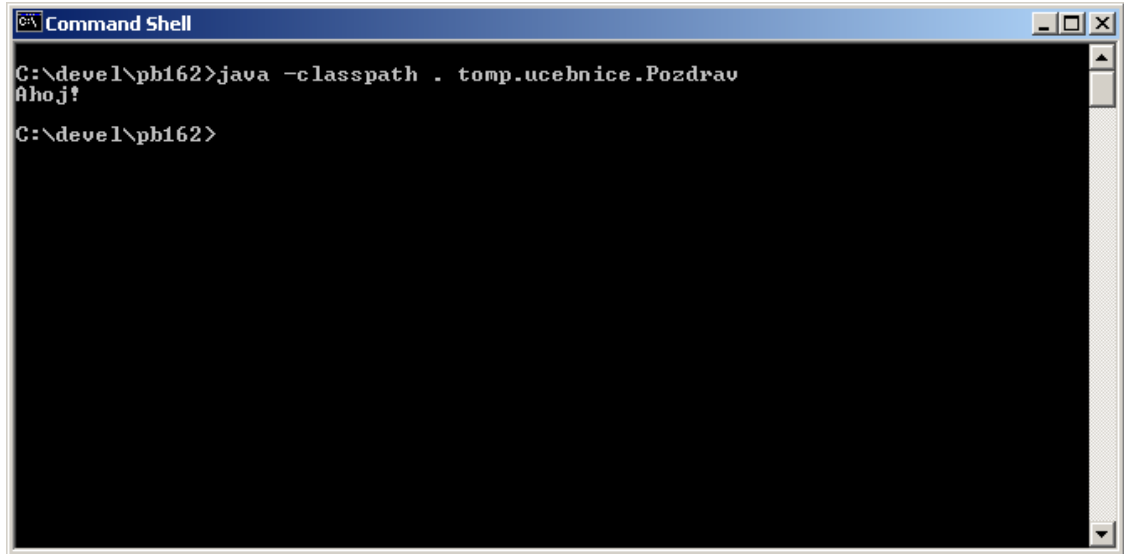
Překlad "Ahoj!" ("for dummies")

Překlad překladačem **javac** (úspěšný, bez hlášení překladače)



Spuštění "Ahoj!" ("for dummies")

Spuštění voláním **java**



```
Command Shell
C:\devel\pb162>java -classpath . temp.ucebnice.Pozdrav
Ahoj!
C:\devel\pb162>
```

Co znamená spustit program?

Spuštění javového programu

= spuštění metody `main` jedné ze tříd tvořících program

Tato funkce může mít parametry:

- podobně jako např. v Pascalu nebo v C
- jsou typu `String` (řetězec)
- předávají se při spuštění z příkazového řádku do pole `String[] args`

Metoda `main` nevrací žádnou hodnotu - návratový typ je vždy(!) `void`

Její hlavička musí *vypadat vždy přesně tak*, jako ve výše uvedeném příkladu, jinak nebude spuštěna!

Praktické informace (aneb co je nutné udělat)

Cesty ke spustitelným programům (`PATH`) musejí obsahovat i adresář `JAVA_HOME\bin`

Praktické informace (aneb co je vhodné udělat)

Systémové proměnné by měly obsahovat:

- `JAVA_HOME`=kořenový adresář instalace Javy, např. `JAVA_HOME=c:\j2sdk1.4.2`
- `CLASSPATH`=cesty ke třídám (podobně jako v `PATH` jsou cesty ke spustitelným souborům), např. `CLASSPATH=c:\devel\pb162`

Úloha do cvičení 1.

...Zde bude zadání úlohy č. 1...

Odkazy

- Odkazy na zdroje (učebnice)<http://www.fi.muni.cz/~tomp/java/ucebnice/resources.html>
- Další tutoriály:<http://www.mike-levin.com/learning-java/toc.html>