

3. Základy programovacího jazyka Java

Programovací jazyk Java nese označení jazyk "3.v generace – 3GL", kde označení 3 označuje imperativní jazyk vysoké úrovně. Tento jazyk se vyznačuje vysokou mírou univerzality, neboť není určen výhradně pro specifickou aplikační oblast. Díky této vlastnosti lze v Javě vytvářet jednoduché či složitější desktopové aplikace, tak i webové aplikace či velké serverově-klientské aplikace. Jak už jsme uvedli dříve, jazyk je čistě objektově orientovaný, což znamená, že výpočet je realizován jako volání metod, neboli zasílání zpráv objektům.

Předchůdcem Javy byl jazyk SmallTalk a C++, přičemž C++ bylo zbaveno zbytečností a nepříjemností. Java je tedy jednodušší než C++. V současné době je jazyk C# od společnosti Microsoft reálným soupeřem, převážně však v operačním systému MS Windows. Odtud může plynout, že to, co se naučíme v Javě, můžeme použít i v C#.

Důležité však je, že Java podporuje vytváření správných návyků v objektovém programování. Naopak: Java brání přenosu některých špatných návyků z jiných jazyků.

Popis prostředí Javy

Program v Javě je meziplatformně přenositelný na úrovni zdrojového kódu i zkompileovaného kódu. Tato schopnost je umožněna tím, že přeložený program je při spuštění uživatelem spuštěn v tzv. Java Virtual Machine (dále jen JVM, více informací na <http://java.sun.com/docs/books/vmspec/>) – v Javovském virtuálním počítači, což je pseudo počítač, spuštěný v operační paměti našeho počítače a emulující virtuální počítač, ve kterém běží Javovské programy.

Vzhledem k tomu, že JVM je vytvořen pro různé operační systémy (UNIX, Windows, MAC OS X, atd.), mohou programy, vytvořené v Javě, fungovat na různých operačních systémech, kde je JVM nainstalován. Při spuštění programu v JVM je možno nastavit úroveň přístupu k hostitelskému operačnímu systému pomocí Security Manageru (více informací na <http://www.securingjava.com/chapter-two/chapter-two-8.html>) nebo je možno ověřovat před spuštěním programu elektronický podpis kódu, těmito způsoby je tedy možno kód programu dobře zabezpečit.

Java pro programátory

Jazyk Java je vhodný pro efektivní – rychlé psaní přehledných programů. Výhodou je vysoká míra kvalitního zdokumentování celého jazyka a jednotlivých prvků. Odtud také plyne v průměru vyšší produktivita programátorské práce v Javě než v C++. Programátorů v Javě je již také dnes srovnatelné množství jako programátorů v C++. Nepopíratelnou výhodou Javy je zdarma dostupné nezměrné množství knihoven pro různorodé aplikační oblasti, které lze nalézt např. na serveru [SourceForge](http://sourceforge.net).

V Javě se velmi dobře píše vícevláknové aplikace, nemusíme se zabývat odklizením vytvořených již nepoužívaných objektů, o uklid v paměti se stará automatický "popelář", neboli "automatic garbage collector".

Pro programátory je k dispozici řada kvalitních vývojových prostředí, jako jsou např. [NetBeans](#), [JBuilder](#), [JCreator](#), [BlueJ](#), [Visual Age for Java](#), [Eclipse](#), [IDEA](#). Některé z nich jsou dostupné také zdarma.

Hlavní využití nalézá Java v škálovatelných výkonných aplikacích běžících an

serverech ([Java Enterprise Edition](#)), v aplikacích pro přenosná a vestavěná zařízení ([Java Micro Edition](#)), pro aplikace, přenositelné mezi operačními systémy (např. klientské, desktopové) a také pro výukové účely, neboť Java v současné době nahrazuje programovací jazyk Pascal jako referenční jazyk. Dalším využitím Javy jsou webové aplikace (servlety, JSP - konkurence proprietárním ASP, SSI, pomalým CGI), dále pak např. zpracování semistrukturovaných dat (XML), přenositelné aplikace s GUI a aplikace distribuované po síti (applety nebo Java Web Start).

Javovská platforma

Javovskou platformu tvoří **Java Virtual Machine**, neboli virtuální stroj, na kterém běží přeložené aplikace, dále **překladač** (přístupný např. příkazem **javac**) a vývojové nástroje a nakonec **Java Core API**, což je základní knihovna tříd. Java je tedy dána definicí jazyka, což je syntaxe a sémantika jazyka, dále popisem chování virtuálního stroje JVM a popisem Java Core API.

Co se týče vývoje Javy, nejrychleji se vyvíjí Java Core API, základní knihovna tříd se neustále rozšiřuje a jednotlivé třídy jsou stále zdokonalovány, upravovány a je urychlován jejich kód. Naopak chování JVM se mění tehdy pokud se např. objeví bezpečnostní díra v Javě nebo pokud nelze dosáhnout určité požadované změny chování určité třídy pomocí modifikace Java Core API. Nejméně se mění jazyk samotný. Ten se mění opravdu velice zřídka. V případech rozšíření možností jazyka se např. přidává klíčové slovo či datový typ.

Specifikace, implementace a verze Javy

Specifikací rozumíme tzv. edice Javy, kdy rozlišujeme edice pro desktopové aplikace (**Java 2 Standard Edition, v1.4**), edice pro přenosná zařízení (**Java 2 Micro Edition v1.4**) a další.

Implementací rozumíme rozdělení celého balíku Javy na vývojovou část ("**Development Kits**" - **Java 2 Software Development Kit, v1.4.2**) a běhovou část ("**Runtime Environments**" - **Java 2 Runtime Environment, v1.4.2**). Běhová část obsahuje pouze běhové prostředí, to znamená virtuální stroj, pro spouštění přeložených programů.

Programovací jazyk Java se dělí na verzi "**Java**" a verzi "**Java 2**". Názvem "Java" se označovala verze 1.0 a 1.1. Od verze 1.2 se Java označuje "Java 2"

Distribuce Javy

Licenční politika společnosti Sun Microsystems Inc. je velmi příznivá pro jakéhokoliv programátora, ať už nadšence nebo profesionálního vývojového týmu jakéhokoliv společnosti. Používání Javy pro běžný **nekomerční i komerční vývoj** je **zdarma**. Redistribuce javového **vývojového prostředí** je povolena pouze s **licencí** od společnosti Sun Microsystems Inc. Redistribuce javového **běhového prostředí** je **zdarma**.

Pro vývoj vlastních programů budete potřebovat instalaci celého balíku běhového a vývojového prostředí. Tento balík pod označením "**Java 2 Software Development Kit, v1.4.2**" můžete stáhnout z internetových stránek <http://java.sun.com>. Na těchto stránkách naleznete instalační program jak pro operační systém MS Windows, tak i pro

OS Solaris či pro OS Linux. Dokumentaci můžete stáhnout z těchto stránek také jako samostatný balík, nebo můžete využívat online dokumentaci přímo na stránkách <http://java.sun.com>.

Je nutno na tomto místě upozornit, že nároky na počítač, na kterém budete provozovat Javu, nejsou malé. Potřebná velikost operační paměti je minimálně 64 MB, doporučeno je 128 MB a více. Také velikost volného místa na pevném disku je nezanedbatelná. Vývojové prostředí J2SDK 1.4.2_07 včetně dokumentace zabere na disku cca. 260 MB.

Ve vývojové distribuci Javy se nachází několik následujících komponent:

- **Vývojové nástroje – Development Tools**, jsou umístěny v adresáři **/bin**, jsou určeny hlavně k vývoji, spouštění, ladění a dokumentace programů.
- **Běhové prostředí – Java Runtime Environment**, je umístěno v adresáři **/jre**, používá se pro spouštění programů, napsaných v Javě, obsahuje Java Virtual Machine, knihovnu tříd Java Core API a další potřebné soubory, nutné pro správné fungování programů, napsaných v Javě.
- **Přídavné knihovny – Additional libraries**, jsou umístěny v adresáři **/lib**, tyto knihovny jsou nutné pro správné fungování a běh vývojových nástrojů.
- **Ukázkové applety a aplikace – Demo Applets and Applications**, jsou umístěny v adresáři **/demo**, jsou to ukázkové příklady, zahrnující i zdrojový kód.
- **Hlavičkové soubory pro jazyk C – C Header Files**, jsou umístěny v adresáři **/include**, jsou zde umístěny pro podporu psaní tzv. nativních metod přímo v jazyce C.
- **Staré hlavičkové soubory – Old Native Interface Headers**, jsou uloženy ve stejném adresáři, avšak jsou užívány pro starší verzi rozhraní.
- **Zdrojový kód – Source Code**, je umístěn v kořenovém adresáři instalace Javy a nachází se v souboru **src.jar** nebo **src.zip**. Jedná se o zdrojové kódy knihoven z Java Core API.
- **Dokumentace – Documentation**, je umístěna v adresáři **/docs**, obsahuje dokumentaci k dané verzi JDK, k Java Core API, dále nejruznější průvodce pro vývoj aplikací, dokumentaci k nástrojům, vzorové a ukázkové programy a v poslední řadě také odkazy na související dokumentaci na Internetu.

Nástroje ve vývojové distribuci

Nástroje se ve vývojových distribucích pro různé operační systémy příliš neliší, neboť v každé distribuci se nachází překladač programů a spouštěč programů plus několik podpůrných programů. My si zde popíšeme distribuci pro operační systém MS Windows.

Pro tento OS se nástroje nacházejí v adresáři **/bin** a kromě jiných jsou to následující (asi nejdůležitější) programy:

- **java** – spouštěč programu, přeloženého do tzv. bajtkódu – bytecode.
- **javac** – překladač programů ze zdrojového tvaru (soubor s příponou .java) do spustitelného tvaru (soubor s příponou .class)
- **javadoc** – generátor dokumentace API
- **jar** – správce archivů JAR, který se stará o jejich sbalení, rozbalení a výpis
- **jdb** – debugger javovských programů
- **appletviewer** – referenční prostředí pro spouštění appletů
- **javah** – generátor hlavičkových souborů pro C
- **javap** – disassembler bajtkódu, využívá se např. pro ruční optimalizace, pro hledání chyb v programu, atd.

Javovský program – životní cyklus, struktura, vytvoření, přeložení a spuštění

Každý program, vytvořený v Javě, se skládá z jedné či více **tříd – class**. Každá třída, která je veřejně přístupná, je umístěna v jednom souboru, který je pojmenován podle názvu třídy a přípona je java (tedy např. MojePrvniTrida.java). Jméno názvu souboru musí být shodné s názvem třídy i co se velkých a malých písmen týče.

Třídy jsou členěny do **balíků – package**. Zařazení třídy do balíku mimo jiné znamená umístění zdrojového souboru třídy do **příslušného adresáře**. U běžné desktopové aplikace představuje **vždy jedna třída** (eventuálně více tříd) vstupní bod do programu – konkrétní třída obsahuje metodu **main**.

Životní cyklus programu je následující:

- vytvoření zdrojového textu v libovolném editoru či ve vývojovém prostředí. Zdrojový soubor se pojmenuje podle názvu třídy (např. **Pokus.java**).
- poté se zdrojový soubor pomocí překladače **javac** přeloží. Ze souboru **Pokus.java** se vytvoří soubor **Pokus.class**. Překládá se tedy takto, že se spustí z příkazové řádky příkaz: **javac Pokus.java**.
- nakonec se program **Pokus.java** spustí pomocí spouštěče **java**. Spouští se tedy tak, že se z příkazové řádky zavolá příkaz: **java Pokus** (bez uvedení přípony class).

Ukázkový program

Následující příklad ukazuje zdrojový kód jednoduchého programu:

```
jdaslfj asjf lsadjf lsadjf lúasjfd jsaůf dslajf ladsjf lsakj fldjaf lúdajsf ůljdsaf jldsafj
dsajf ilsajd fldsajfůl jdsalfj
```

drojový kód v souboru temp\ucebnice\Pozdrav.java

```
package temp.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 temp.ucebnice -> její zdrojový soubor musí být uložen v podadresáři temp\ucebnice.

Překlad a spuštění "Ahoj!"

Překlad

1. Máme nainstalován J2SDK 1.4.2

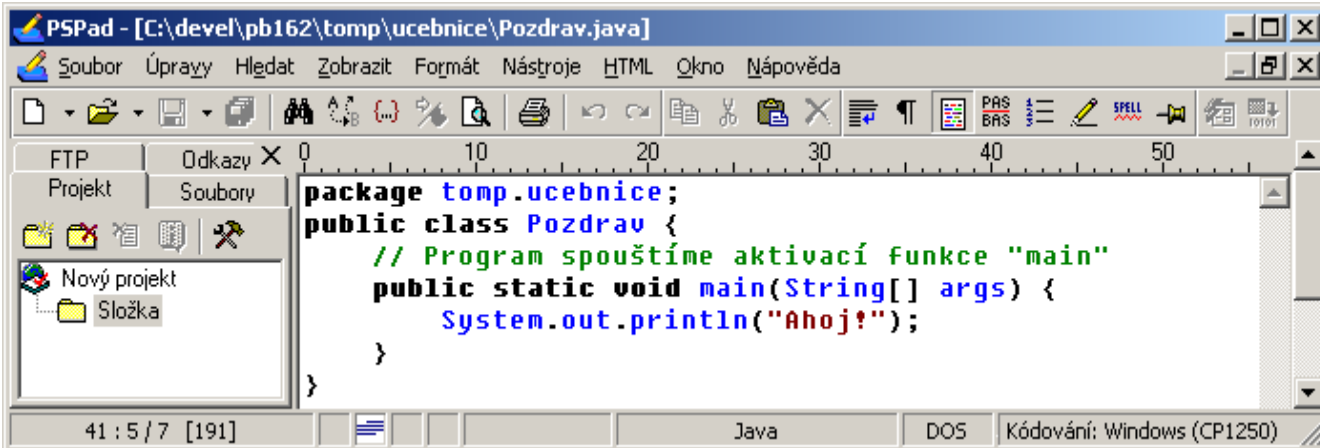
2. Jsme v adresáři `c:\devel\pb162`, v něm je podadresář `tomp\ucebnice`, v něm je soubor `Pozdrav.java`
3. Spustíme **překladač** `javac tomp\ucebnice\Pozdrav.java`
4. Je-li program správně napsán, přeloží se "mlčky"
5. (výsledný `.class` soubor bude v témže adresáři jako zdroj)

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")

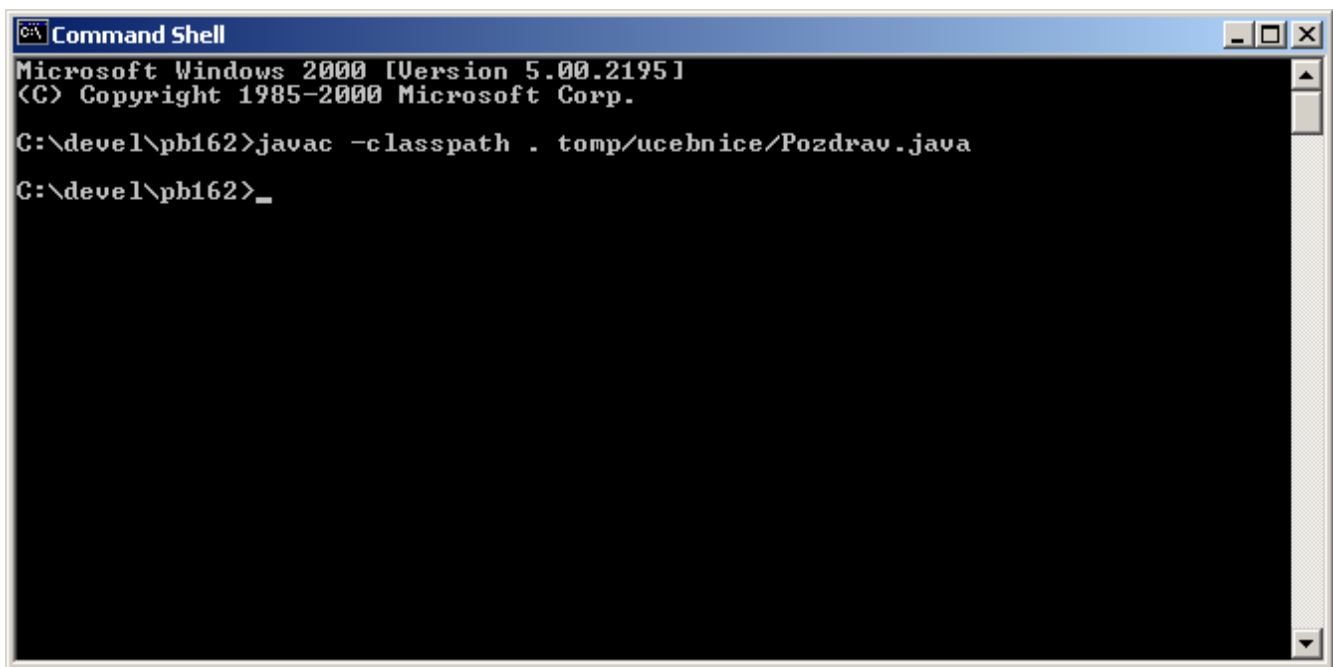
Vytvoření a editace zdrojového kódu v editoru [PSPad 4.2.2](#) (dostupný zdarma, instalovaný na všech Win strojích v učebnách na FI)



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

Překlad "Ahoj!" ("for dummies")

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

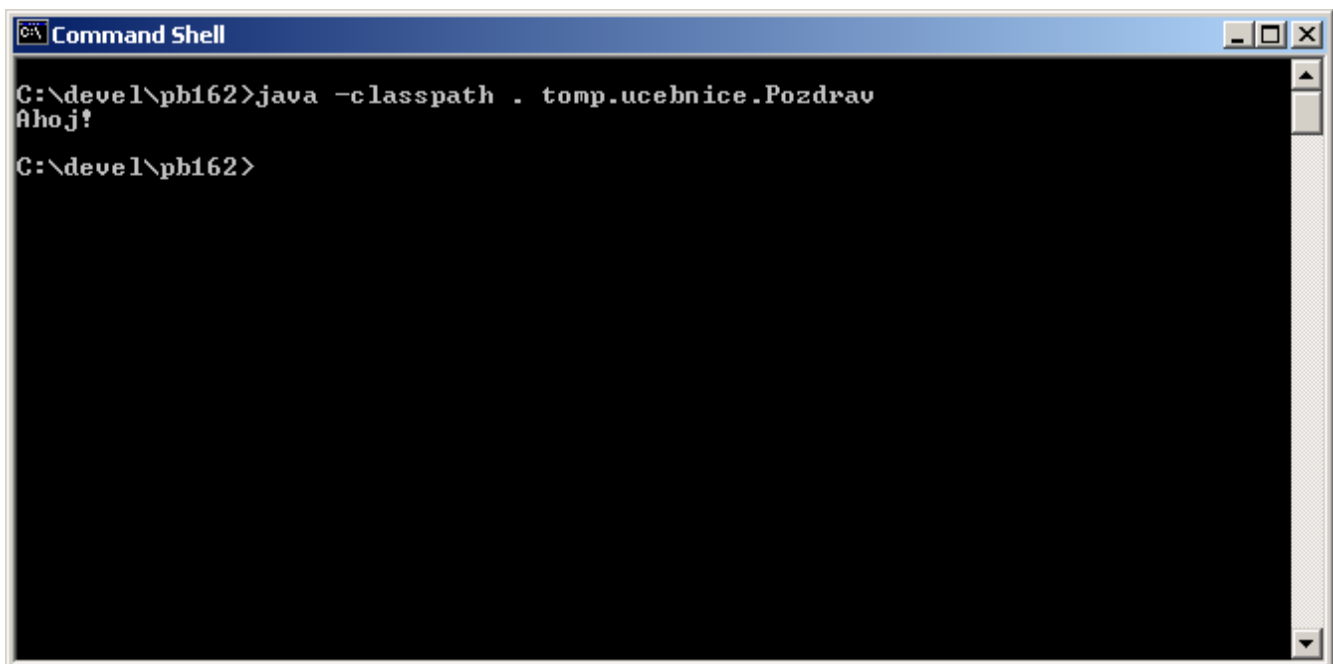


```
Command Shell
Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.

C:\devel\pb162>javac -classpath . temp/ucebnice/Pozdrav.java
C:\devel\pb162>_
```

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