
Informace o charakteru, průběhu a hodnocení předmětu. Úvod do jazyka Java a prostředí BlueJ

Obsah

Informace o průběhu, hodnocení, apod.	2
O předmětu... ..	2
Návaznosti - co dál studovat	3
Složky hodnocení předmětu	3
Kritéria hodnocení předmětu	4
Obsah (syllabus) předmětu	4
Cvičení	4
O přednášejících (2)	5
O přednášejících (2)	5
Konzultační hodiny	5
Informační zdroje	6
Úvod do prostředí Java	6
Java jako programovací jazyk... ..	7
Proč Java	7
Z toho plyne, že... ..	7
Další charakteristiky	7
Java jako běhové prostředí	8
Java pro programátora (1)	8
Java pro programátora (2)	8
Hlavní domény Javy (1)	9
Hlavní domény Javy (2)	9
Javová platforma	9
Java je tedy dána... ..	9
Vývoj Javy	9
Specifikace a implementace Javy	10
Verze Javy - starší konvence	10
Verze Javy - konvence od verze 5.0 (1.5 postaru)	10
Aktuální verze	11
Co bylo nového v Javě 5.0	11
Licence k použití (a redistribuci) Javy	11
Stažení distribuce Sun	11
Typy distribucí Javy (Sun)	12
Obsah vývojové distribuce Javy	12
Obsah vývojové distribuce Javy (2)	12
Nástroje ve vývojové distribuci	13
Pomocné nástroje ve vývojové distribuci	14

První kroky javového programování v prostředí BlueJ	14
Struktura javového programu	14
Spuštění a běh javového programu	14
Prostředí BlueJ	15
Stažení BlueJ	15
Instalace BlueJ	15
Spuštění BlueJ	16
Nastavení českého prostředí BlueJ	16
Otevření hotového programu (projektu) v BlueJ	17
Spuštění hotového programu	17
Interaktivní vytvoření objektu (instance)	17
Práce s objektem (instancí)	18
Úprava hotového programu v BlueJ	19
Sestavení a spuštění programu bez BlueJ	19
Základní životní cyklus javového programu - bez BlueJ	19
Demo "Ahoj!"	20
Překlad a spuštění "Ahoj!"	21
Vytvoření zdrojového textu "Ahoj!" ("for dummies")	21
Překlad "Ahoj!" ("for dummies")	22
Spuštění "Ahoj!" ("for dummies")	22
Co znamená spustit program?	22
Praktické informace (aneb co je nutné udělat)	22
Praktické informace (aneb co je vhodné udělat)	22
Sestavení a spuštění programu s třídami v balících	23
Demo "Ahoj!"	23
Překlad a spuštění "Ahoj!"	24
Vytvoření zdrojového textu "Ahoj!" ("for dummies")	25
Překlad "Ahoj!" ("for dummies")	25
Spuštění "Ahoj!" ("for dummies")	25
Co znamená spustit program?	25
Odkazy	26

Informace o průběhu, hodnocení, apod.

- Souvislosti, prekvizity, návaznosti
- Hodnocení a jeho složky
- Kontakty a konzultační hodiny
- Odkazy na informační zdroje

O předmětu...

Prakticky zaměřený bakalářský předmět

Cílem je naučit základním principům objektového programování a algoritmizace

Souvisí s

- IB001 - **Úvod do programování** (předpokládají se znalosti na úrovni IB001)
- IB002 - **Návrh algoritmů I** (v PB162 se prakticky implementují vybrané algoritmy probírané v IB002)

Předpokládají se základní znalosti strukturované algoritmizace a programování (v rozsahu Úvodu do programování), tj. např.:

- základní příkazy, sestavování jednoduchých výrazů
- základní datové typy (celá a reálná čísla, logické proměnné, řetězce)
- základní řídicí struktury - větvení, cykly, procedury/funkce

Návaznosti - co dál studovat

Na tento základní kurz navazují na FI

- PV168 - **Seminář z jazyka Java** (jarní semestr), náplní je zvládnutí Javy umožňující vývoj jednodušších praktických aplikací s GUI, databázemi, s použitím JavaME... Učí kolektiv zkušených cvičících pod vedením Petra Adámka (Jan Pavlovič, Pavel Drášil, Luděk Bártek a další).
- PA165 - **Vývoj aplikací v jazyce Java** (podzimní semestr) - pokročilejší předmět spíše magisterského určení, předpokládá znalosti/zkušenosti z oblasti databází, částečně sítí a distribuovaných systémů - a také Javy zhruba v rozsahu PB162 a PV168. Náplní je zvládnutí netriviálních, převážně klient/server aplikací na platformě JavaEE.

Složky hodnocení předmětu

Hodnocení má tři složky:

- **32 bodů - hodnocení úloh** řešených samostatně v průběhu semestru (na cvičeních, ve volném čase...)
- **8 bodů - první písemka** řešení jednoduché praktické úlohy přímo u počítače (říjen)
- **14 bodů - druhá písemka** řešení jednoduché praktické úlohy přímo u počítače (listopad)
- **18 bodů - třetí písemka** řešení praktické úlohy přímo u počítače, v době cvičení, na konci výuky v semestru (prosinec).
- **28 bodů - závěrečná/zkoušková písemka** řešení rozsáhlejší praktické úlohy přímo u počítače, ve zkouškovém období.

- **Celkem 100 bodů**



Poznámka

docházka do cvičení je povinná, ale netvoří součást bodového hodnocení

Kritéria hodnocení předmětu

Doporučené ukončení je zkouškou. K tomu potřebujete:

- A 94 - 100 bodů
- B 88 - 93 bodů
- C 82 - 87 bodů
- D 76 - 81 bodů
- E 70 - 75 bodů
- F 0 - 69 bodů

K ukončení zápočtem postačí:

- 60 bodů

Obsah (syllabus) předmětu

- viz aktuální informace o předmětu PB162
[<https://is.muni.cz/auth/predmety/predmet.pl?fakulta=1433;id=427864>] na IS MU

Cvičení

Cvičení jsou dvouhodinová, konají se pod vedením příslušných kvalifikovaných cvičících v počítačových učebnách, zpravidla B116, B117, B130 - viz váš rozvrh.

Náplň

- Hlavním obsahem je *konzultovaná samostatná práce* na bodovaných úlohách + aktivní opakování témat z přednášek s důrazem na praktické zvládnutí;
- cvičení jsou *dvouhodinová*, účast na přednáškách je však přesto žádoucí;

- předpokládá se i jistý netriviální podíl *práce mimo cvičení*;
- odměnou vám, kromě získaných znalostí a dovedností, budou *celkem 4 kr. + za ukončení*

O přednášejících (2)

Radek Ošlejšek

- kanc. B306 (3. patro budovy B)
- tel. 54949 6121 (z tlf. mimo budovu), kl. 6121 (volání v rámci fakulty i celé MU)
- e-mail: oslejsek@fi.muni.cz [<mailto:oslejsek@fi.muni.cz>]
- www RO: <http://www.fi.muni.cz/~oslejsek>

materiály PB162-Java (IS): <https://is.muni.cz/auth/el/1433/podzim2007/PB162/>
[<https://is.muni.cz/auth/el/1433/podzim2007/PB162/>]

O přednášejících (2)

Tomáš Pitner

- kanc. B307 (3. patro budovy B)
- tel. 54949 5940 (z tlf. mimo budovu), kl. 5940 (volání v rámci fakulty i celé MU)
- e-mail: tomp@fi.muni.cz [<mailto:tomp@fi.muni.cz>]
- www TP: <http://www.fi.muni.cz/~tomp>

materiály PB162-Java (IS): <https://is.muni.cz/auth/el/1433/podzim2007/PB162/>
[<https://is.muni.cz/auth/el/1433/podzim2007/PB162/>]

Konzultační hodiny

Primárním konzultačním bodem jsou vaši cvičící. Cvičení jsou vedena mj. právě z důvodu možnosti konzultací.

Konzultace přímo s přednášejícími

Tomáš Pitner: vždy v kanc. B307, prosíme o dodržování časů

- **Út 10.00 - 11.30**
- nebo jindy, dle dohody

Radek Ošlejšek - viz web [<http://www.fi.muni.cz/~oslejsek>]

Informační zdroje

- Studijní materiály předmětu: <https://is.muni.cz/auth/el/1433/podzim2007/PB162/...> budou vystavovány postupně, pro celkový přehled o obsahu lze použít materiály loňské -> <https://is.muni.cz/auth/el/1433/podzim2006/PB162/>
- Aktuální materiály na wiki <http://kore.fi.muni.cz/wiki/index.php/PB162> přístup ze sítě mimo FI: <http://kore.fi.muni.cz:5080/wiki/index.php/PB162>
- Další zdroje, materiály z (hodně) minulých let: <http://www.fi.muni.cz/~tomp/java>
- Knihy
 - TP: **Java - začínáme programovat**, Grada Publishing [<http://www.gradapublishing.cz>], 2002, <http://www.fi.muni.cz/~tomp/java/ucebnice>
 - Pavel Herout: **Učebnice jazyka Java**, Kopp [<http://www.kopp.cz>], 2000
 - (Pavel Herout: **Java - grafické uživatelské rozhraní a čeština**, Kopp [<http://www.kopp.cz>], 2001) - pro pokročilé
 - Rudolf Pecinovský: **Myslíme v jazyce Java 5.0**, Grada Publishing [<http://www.gradapublishing.cz>], 2005
 - Bruce Eckel: **Myslíme v jazyce Java - příručka programátora**, Grada Publishing [<http://www.gradapublishing.cz>], 2000
 - (Bruce Eckel: **Myslíme v jazyce Java - příručka zkušeného programátora**, Grada Publishing [<http://www.gradapublishing.cz>], 2000) - pro pokročilé
 - a další, viz např. <http://www.vltava.cz>
 - Joshua Bloch: **Java efektivně 57 zásad softwarového experta**, Grada Publishing [<http://www.gradapublishing.cz>]
 - Bogdan Kiszka: **1001 tipů a triků pro programování v jazyce Java**, Computer Press, 2003, informace na Vltavě [<http://www.vltava.cz/Store/GoodsDetail.asp?c=BogdanKiszka&sCGoodsID=SE00679120>]
 - Bruce Eckel: **Thinking in Java** Stáhnout zdarma (PDF) [<http://www.mindview.net/Books/TIJ/javabook.html>]

Úvod do prostředí Java

- Úvod, srovnání s jinými, oblasti použití Javy

- Distribuce, instalace, použití Java SDK

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++*
- perspektivním směrem vývoje je zachování Java platformy (JVM, stávající knihovny a aplikace) + skriptovací jazyky na této platformě (Groovy, JRuby, Jython...)

Proč Java

- Java je jazyk pro vývoj a běh jednoduchých i rozsáhlých aplikací
- Vývoj je efektivnější než na jejich předchůdcích (C++) a výsledné aplikace "běží všude"
- Nezavádí zbytečnosti a vede ke správným a dále uplatnitelným návykům
- Je velmi perspektivní platformou pro vývoj open-source i komerčního SW, mj. pro extrémně velké množství volně dostupných knihoven

Z toho plyne, že...

- co se naučíme v Javě, v C# jako když najdeme...
- rovněž moderní skriptovací jazyky (Groovy) koncepčně i syntakticky vychází z Javy
- ale teď vážně: *Java podporuje vytváření správných návyků v objektovém programování*
- a naopak systematicky brání *přenosu některých špatných návyků z jiných jazyků*
- začít v imperativním paradigmatu ihned se *slabě typovaným* (např. skriptovacím) jazykem není vhodné; je lépe nejdříve dobře zvládnout klasický (*silně typovaný*) jazyk a pak snadno přejít

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 více 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.intellij.com/>], BlueJ - výukové vývojové prostředí - používáme v tomto předmětu! [<http://bluej.org>]

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/>])
- Přenositelné *desktopové* i *grafické/okénkové (GUI)* aplikace
- *Výukové účely* (nahrazuje Pascal jako referenční jazyk)
- Další přenositelné aplikace

Hlavní domény Javy (2)

- Webové aplikace (servlety, JSP, webové rámce)
- Zpracování semistrukturovaných dat (XML)
- Aplikace distribuované po síti (applety nebo Java Web Start)

Javová platforma

Javovou platformu tvoří:

- **Java Virtual Machine**
- **Java Core API** (základní knihovna tříd)
- **překladač** (přístupný např. příkazem **javac**  [<http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=javac>]) a další vývojové nástroje

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 přidává nové klíčové slovo *assert*, Java 5.0 (postaru 1.5) obohacuje jazyk o *enum*, generické typy a další rysy.

Specifikace a implementace Javy

- **Specifikace** Javy (tzv. "Editions") - např.: *Java 2 Standard Edition, 1.4* nebo *Java 2 Standard Edition 6.0*
- **Implementace** Javy ("Development Kits" nebo "Runtime Environments") - např.: *Java 2 Software Development Kit, 6.0* - obsahuje vývojové nástroje
- *Java 2 Runtime Environment, 6.0* - obsahuje jen **běžové prostředí** pro spouštění hotových přeložených pg.

Verze Javy - starší konvence

- Verze Javy byly až do verze 1.5 (ponovu 5.0) děleny na "*Java* (před Java 2)" a "*Java 2*"
- číslování verzí:
 - tzv. major číslo, např. Java 2, v**1.4**
 - 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 dochází až teď, s verzí 5.0 (postaru 1.5) - tj. s celkovou změnou pojmenovacího schématu

Verze Javy - konvence od verze 5.0 (1.5 postaru)

Nové schéma číslování verzí počínaje 5.0 (postaru 1.5):

- tzv. major číslo, např. Java 2, **5.0**
- tzv. minor číslo, např. Java 2, **5.0**

Aktuální verze

Stav k září 2007:

- Java 2 Standard Edition 6.0 a Enterprise Edition 5.0 je stabilní verzí pro všechny platformy.
- Aktuální informace najdete vždy na webu <http://java.sun.com>.

Co bylo nového v Javě 5.0

Generics	generika (generické datové typy) technika umožňující silnou typovou kontrolu v době překladu - např. v metodách manipulujících s prvky dynamických datových struktur
Enhanced for Loop	umožňuje snadnou a typově bezpečnou iteraci po prvcích kolekcí a polí
Autoboxing/Unboxing	zajišťuje automatickou konverzi mezi primitivními typy a příslušnými objektovými "obalujícími" typy
Typesafe Enums	typově bezpečné výčtové typy
Varargs	umožňují specifikovat proměnný počet parametrů metod
Static Import	dovoluje odkazovat se na statické prvky bez uvádění jména třídy (rozhraní)
Metadata (Annotations)	dovolují specifikovat u tříd, rozhraní, metod metadata (anotace) dostupná i za běhu

Licence k použití (a redistribuci) Javy

- používání Javy pro běžný vývoj (i komerční) je zdarma
- licence vývojového prostředí viz <http://java.sun.com/javase/6/jdk-6u2-license.txt>
licence běhového prostředí viz <http://www.java.com/en/download/license.jsp>
- distribuce vyvíjí Sun Microsystems Inc. (Javasoft) i další výrobci (např. IBM) a tvůrci Open Source

Stažení distribuce Sun

- java.sun.com [<http://java.sun.com>] (pro Windows, Solaris, Linux)
- lze stáhnout jak samotné vývojové prostředí (JDK), jen běhové prostředí (JRE) nebo JDK v balíčku s

IDE (Integrated Development Environment) NetBeans [<http://netbeans.org>].





- dokumentace se stahuje z téhož místa, ale *samostatně* (nebo lze číst z WWW)
- celkově *vývojové prostředí Java SE 6* zabere bez dokumentace cca **200 MB** na disku
- potřebná *velikost operační paměti* - min 64 MB, ale doporučeno alespoň **128 MB** (při použití Netbeans apod. rozhodně více)

Typy distribucí Javy (Sun)


Lze stáhnout:

- samotné vývojové prostředí (JDK),
- jen běhové prostředí (JRE),
- JDK v balíčku s grafickým (okénkovým) integrovaným vývojovým prostředím (IDE, Integrated Development Environment) NetBeans [<http://netbeans.org>].



Obsah vývojové distribuce Javy

- **Vývojové nástroje (Development Tools)** v `bin` 
[<http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=bin>] -- určené k vývoji, spuštění, ladění a dokumentování programů v Javě.
- **Běhové prostředí Javy (Java Runtime Environment)** se nalézá v `jre` 
[<http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=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` 
[<http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=lib>] jsou další knihovny nutné pro běh vývojových nástrojů.
- **Ukázkové applety a aplikace (Demo Applets and Applications)** v `demo` 
[<http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=demo>]. Příklady zahrnují i zdrojový kód.



Obsah vývojové distribuce Javy (2)









- **Hlavičkové soubory pro C (C header Files)** - v `include` 
[<http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=include>] - představují pod-

poru 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.zip`  [<http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=src.zip>].
- **Dokumentace (Documentation)** - v podadresáři `docs`  [<http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=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`  [<http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=.exe>] soubory umístěné v podadresáři `bin`  [<http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=bin>]

- **java**  [<http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=java>] - spouštěč (přeloženého bajtkódu)
- **javac**  [<http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=javac>] - překladač (`.java`  [<http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=.java>] \rightarrow `.class`  [<http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=.class>])
- **javadoc**  [<http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=javadoc>] - generátor dokumentace API
- **jar**  [<http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=jar>] - správce archivů JAR (sbalení, rozbalení, výpis)
- **jdb**  [<http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=jdb>] - debugger
- **appletviewer** 

[<http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=appletviewer>] - referenční prostředí pro spouštění appletů

- **javaws**  The Free Encyclopedia

[<http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=javaws>] - referenční prostředí pro spouštění aplikací typu "Java Web Start" prostřednictvím Java Network Launching Protocol (JNLP)

Pomocné nástroje ve vývojové distribuci

- **javah**  The Free Encyclopedia

[<http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=javah>] - generátor hlavičkových souborů pro C - používá se při programování tzv. nativních (platformově závislých) metod dostupných přes *Java Native Interface* (JNI)

- **javap**  The Free Encyclopedia

[<http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=javap>] - disassembler bajtkódu (např. pro ruční optimalizace, hledání chyb)

První kroky javového programování v prostředí BlueJ

1. Seznámíme se se strukturou programu v Javě,
2. naučíme se spouštět hotové jednoduché programy ve výukovém prostředí BlueJ
3. a tyto programy upravovat.

Struktura javového programu

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

Spuštění a běh javového programu

1. Spuštění programu se zahajuje vstupem do metody main.
2. Běh programu spočívá ve vytváření objektů a interakci s nimi, tj. volání jejich metod.
3. Běh končí, jakmile jsou provedeny všechny příkazy aktivované metodou main.

Prostředí BlueJ

- Pro výuku v kurzu PB162 budeme převážně používat jednoduché, pro výuku určené, vývojové prostředí BlueJ.
- BlueJ je pro všechny platformy (Win, Linux...) zdarma dostupné na <http://bluej.org>.
- BlueJ je lokalizované pro češtinu vč. základního manuálu.
- Pro BlueJ existuje i jednoduchý animátor běhu programů - vizualizuje spouštění a chod programů.
- Toto prostředí je již předinstalované v síti FI.

Stažení BlueJ

- BlueJ je již v síti FI instalováno, pro vlastní potřebu jej lze zdarma stáhnout z <http://bluej.org>.
- Doporučujeme aktuální verzi - v současnosti BlueJ 2.2.0
- Předtím je třeba mít korektně nainstalováno samotné vývojové prostředí Java.

Obrázek 1. Stažení BlueJ

Instalace BlueJ


Po stažení příslušného instalačního balíčku postupujte v závislosti na platformě:

Windows poklepat (double-click) na staženém programu `bluejsetup-verze.exe`

 WIKIPEDIA
The Free Encyclopedia

[<http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=bluejsetup->]




(např.

`bluejsetup-220.exe`  WIKIPEDIA
The Free Encyclopedia

[<http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=bluejsetup-220.exe>])

následovat pokyny instalačního programu

instalační program přitom nechá vybrat z předem instalovaných běhových prostředí Javy


Unix na konzole (v prostředí shellu) se přepněte (pomocí ) do adresáře [\[http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=cd\]](http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=cd) do adresáře se staženým balíčkem, např. `bluej-220.jar` 
[\[http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=bluej-220.jar\]](http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=bluej-220.jar)
spustíte instalaci příkazem `java -jar bluej-220.jar` 
[\[http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=java -jar bluej-220.jar\]](http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=java%20-jar%20bluej-220.jar)

při instalaci vyberte vhodný cílový adresář a zvolte z předem instalovaných běhových prostředí Javy

Spuštění BlueJ

Po úspěšné instalaci na:

Windows najdete v menu Start systému Windows složku *BlueJ* a zde spouštěcí odkaz *BlueJ* s ikonou ptáka

Unix v adresáři, kam se BlueJ nainstalovalo, najdete spustitelný soubor `bluej` 
[\[http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=bluej\]](http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=bluej) - tím lze prostředí spustit.

Obrázek 2. Spuštění BlueJ

Obrázek 3. BlueJ spuštěno



Nastavení českého prostředí BlueJ

Chcete-li, aby nabídky (menu) a hlášky BlueJ byly v češtině, postupuje podle pokynů v Konfigurační doplněk pro prostředí BlueJ [\[http://vyuka.pecinovsky.cz/mojj50/index.html#BlueJConfig\]](http://vyuka.pecinovsky.cz/mojj50/index.html#BlueJConfig) (autor Rudolf Pecinovský).

Takto vypadá počeštěné prostředí:

Obrázek 4. BlueJ v češtině

Otevření hotového programu (projektu) v BlueJ

- Každý javový program vytvořený v BlueJ představuje jeden *projekt*. Toto je specificky vlastnost BlueJ, ne Javy jako takové.
- Práce s hotovým programem v BlueJ tedy znamená *otevřít projekt* -> menu **Projekt**  **WIKIPEDIA** The Free Encyclopedia
[<http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=Projekt>]/**Otevřít projekt**  The Free Encyclopedia [<http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=Otevřít projekt>].

Obrázek 5. Otevření hotového projektu v BlueJ

Spuštění hotového programu

- Každý běžný javový program spouštíme (nejen v BlueJ) aktivací metody `main`.
- V BlueJ to znamená kliknout pravým tlačítkem myši na ikoně příslušné třídy a vybrat *void main(String[] args)*.

Obrázek 6. Spuštění hotového projektu v BlueJ

- Při spuštění uvádíme parametry, které metoda `main` dostane. Jelikož zatím žádné specifikovat nepotřebujeme, ponecháme dialog, jak je a stiskneme `Enter` (klikneme `OK`).

Obrázek 7. Parametry spuštění

- Výsledek spuštění programu se projeví v tzv. okně terminálu (konzole) BlueJ a vypadá takto:

Obrázek 8. Výsledek spuštění na konzole BlueJ

Interaktivní vytvoření objektu (instance)

- Objektový přístup znamená především to, že program za běhu vytváří objekty a volá jejich metody. To lze v BlueJ snadno realizovat.

- V demostračním programu Hello máme jednu třídu (Hello), od níž lze vytvořit objekt (instanci) a na ní volat metody.
- Zdrojový kód třídy Hello (dostupný poklepnáním na její ikonu) prozradí, které metody zde máme k dispozici.

Obrázek 9. Editace zdrojového kódu třídy Hello

- Dále postupujeme:
 1. Pravým tlačítkem klikneme na ikonu Hello a vybereme *new Hello()*.
 2. do dialogu uvedeme název odkazu na vytvářený objekt typu Hello - můžeme ponechat *hello1*.

Obrázek 10. Specifikace jména odkazu na novou instanci Hello

3. Vytvořený objekt (instance třídy Hello) je k dispozici jako červená krabička vlevo dole.

Práce s objektem (instancí)

- Vytvořený objekt (instance *hello1* třídy *Hello*) je k dispozici jako červená krabička vlevo dole.

Obrázek 11. Vytvořená instance třídy Hello

- S instancí je nyní možné komunikovat/interagovat - tj. volat metody.
- Jelikož v úvahu (viz zdrojový kód Hello) připadá jen metoda *void go()*, zavoláme právě ji:

Obrázek 12. Volání (spuštění) metody go()

- Metoda proběhne s výsledkem podobným, jako když jsme program spouštěli metodou *main*.

Obrázek 13. Výsledek spuštění go()

Metoda *main* totiž realizovala postupně stejné kroky, jako jsme teď provedli interaktivně (krok po kroku) sami.

Úprava hotového programu v BlueJ

Nejjednodušší úpravou programu je změna těla některé z metod, v našem případě již známé metody *go()*.

- Editor zdrojového kódu třídy Hello aktivujeme pravým tlačítkem - položkou *Otevřít v editoru*.
- Kód metody *go()* změníme přepsáním hlášky "Hello, world" na "Ahoj!".

Po kliknutí na tlačítko Přeložit by se mělo objevit:

Obrázek 14. Metoda *go()* upravena a třída úspěšně přeložena

- Třidu Hello s upravenou metodou spustíme jako v předchozím - např. pravým tlačítkem a metodou *main*.


Objeví se:

Obrázek 15. Výsledek běhu upravené třídy Hello

Sestavení a spuštění programu bez BlueJ

- Cílem je porozumět obecné struktuře javového programu
- a být schopni vytvořit a spustit jej i mimo prostředí BlueJ,
- - zatím bez umístění třídy do balíku

Základní životní cyklus javového programu - bez BlueJ

- Program sestává z jedné (ale obvykle více) **tříd** (class). Ukážeme na příkladu třídy s názvem *NazevTridy*:
- Zdrojový kód každé (veřejně přístupné) třídy je umístěn v jednom souboru (*NazevTridy.java*)

[<http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=NazevTridy.java>]
- Postup:
 - vytvoření zdrojového textu (libovolným editorem čistého textu) -> **NazevTridy.java**


[<http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=NazevTridy.java>]
]


- překlad (nástrojem javac) 
[<http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=javac>] **NazevTri-
dy.java** 
[<http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=NazevTridy.java>]
] -> **NazevTridy.class** 
[<http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=NazevTridy.class>]
]
- spuštění, např. **java NazevTridy** 
[<http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=java> **NazevTri-
dy**]
]
- překládá se javac NazevTridy.java 
[<http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=javac> **NazevTridy.java**]
(název souboru se třídou včetně přípony **.java** 
[<http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=.java>]!!!)
- spouští se vždy udáním **java NazevTridy** 
[<http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=java> **NazevTridy**] (název
třídy bez přípony **.class** 
[<http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=.class>]!!!)

Demo "Ahoj!"

- Půjde o podobný jednoduchý demoprogram, jaký jsme spouštěli v BlueJ pod názvem *Hello*.
- Zdrojový kód bude vypadat takto:





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

Zdrojový kód (tj. **Hello.java** 
[<http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=Hello.java>]) bude umístěn do




našeho pracovního adresáře, tj. `c:\devel\pb162` 
[\[http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=c:\devel\pb162\]](http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=c:\devel\pb162).

Překlad a spuštění "Ahoj!"

Překlad

1. Máme nainstalován J2SDK 6
2. Jsme v adresáři `c:\devel\pb162` 
[\[http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=c:\devel\pb162\]](http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=c:\devel\pb162), v něm je soubor `Hello.java` 
[\[http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=Hello.java\]](http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=Hello.java)
3. Spustíme *překlad* `javac` `Hello.java` 
[\[http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=javac Hello.java\]](http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=javac Hello.java)
4. Je-li program správně napsán, přeloží se "mlčky"
5. (výsledný `.class` 
[\[http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=.class\]](http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=.class) soubor bude v témže adresáři jako zdroj)

Spuštění

1. Poté spustíme *program* `Hello: java -classpath . Hello` 
[\[http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=java -classpath . Hello\]](http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=java -classpath . Hello)
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` 
[\[http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=.class\]](http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=.class)) se budou hledat v odpovídajících podadresářích aktuálního adresáře (adresáře `.` 
[\[http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=.\]](http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=.))
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 [\[http://pspad.zde.cz\]](http://pspad.zde.cz) (dostupný zdarma, instalovaný na všech Win strojích v učebnách na FI)

Překlad "Ahoj!" ("for dummies")

Překlad

překladačem

javac 
The Free Encyclopedia

[<http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=javac>] (úspěšný, bez hlášení překladače)

Spuštění "Ahoj!" ("for dummies")

Spuštění

voláním

java 
The Free Encyclopedia

[<http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=java>]

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` 
The Free Encyclopedia

[<http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=String>] (řetězec)


- předávají se při spuštění z příkazového řádku do pole `String[] args`


The Free Encyclopedia

[[http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=String\[\] args](http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=String[]%20args)]

Metoda

`main` 
The Free Encyclopedia

[<http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=main>] nevrací žádnou hodnotu - návratový typ je vždy(!) `void` 
The Free Encyclopedia

[<http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=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)


The Free Encyclopedia





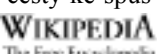
[[http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=\(PATH\)](http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=(PATH))] musejí obsahovat i adresář

`JAVA_HOME\bin` 
The Free Encyclopedia

[[http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=JAVA_HOME\bin](http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=JAVA_HOME%5Cbin)]

Praktické informace (aneb co je vhodné udělat)

Systémové proměnné by měly obsahovat:

- `JAVA_HOME=kořenový adresář instalace` 
[\[http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=JAVA_HOME=kořenový_adresář_instalace\]](http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=JAVA_HOME=kořenový_adresář_instalace) Javy, např. `JAVA_HOME=c:\jdk6.0` 
[\[http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=JAVA_HOME=c:\jdk6.0\]](http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=JAVA_HOME=c:\jdk6.0)
- možné je nastavit proměnnou `CLASSPATH=cesty ke třídám` 
[\[http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=CLASSPATH=cesty ke třídám\]](http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=CLASSPATH=cesty_ke_třídám) (podobně jako v `PATH` 
[\[http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=PATH\]](http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=PATH) jsou cesty ke spustitelným souborům), např. `CLASSPATH=c:\devel\pb162` 
[\[http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=CLASSPATH=c:\devel\pb162\]](http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=CLASSPATH=c:\devel\pb162)



Poznámka

Místo nastavení proměnné `CLASSPATH` je vhodnější volat spuštění programu (podle potřeby i překlad) s volbou `-classpath`.

Sestavení a spuštění programu s třídami v balících




- Cílem je pochopit a umět psát javové programy s třídami umístěnými do balíků

Demo "Ahoj!"

- Půjde o podobný jednoduchý demoprogram, jaký jsme spouštěli v předchozí ukázce, tj. *Hello*.
- Navíc se zde objeví použití *balíků* - třída bude umístěna do balíku `cz.muni.fi.pb162.hello`.
- Zdrojový kód bude vypadat takto:






```
package cz.muni.fi.pb162.hello;
public class Hello {
    // Program spouštíme aktivací funkce "main"
    public static void main(String[] args) {
        System.out.println("Ahoj!");
    }
}
```

}


Pozn: Jelikož třída Hello  je umístěna do balíku [\[http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=Hello\]](http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=Hello) je umístěna do balíku `cz.muni.fi.pb162.hello`, její zdrojový soubor musí být uložen v podadresáři `cz\muni\fi\pb162\hello` 
[\[http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=cz\muni\fi\pb162\hello\]](http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=cz\muni\fi\pb162\hello) našeho pracovního adresáře, tj. `c:\devel\pb162` 
[\[http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=c:\devel\pb162\]](http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=c:\devel\pb162).



Překlad a spuštění "Ahoj!"

Překlad

1. Máme nainstalován J2SDK 6
2. Jsme v adresáři `c:\devel\pb162` 
[\[http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=c:\devel\pb162\]](http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=c:\devel\pb162), v něm je podadresář `cz\muni\fi\pb162\hello` 
[\[http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=cz\muni\fi\pb162\hello\]](http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=cz\muni\fi\pb162\hello), v něm je soubor `Hello.java` 
[\[http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=Hello.java\]](http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=Hello.java)
3. Spustíme *překlad* `javac cz\muni\fi\pb162\hello\Hello.java` 
[\[http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=javac cz\muni\fi\pb162\hello\Hello.java\]](http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=javac cz\muni\fi\pb162\hello\Hello.java)
4. Je-li program správně napsán, přeloží se "mlčky"
5. (výsledný `.class` 
[\[http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=.class\]](http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=.class) soubor bude v téže adresáři jako zdroj)

Spuštění


1. Poté spustíme *program* `Hello: java -classpath . cz.muni.fi.pb162.hello.Hello`  [\[http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=java -classpath . cz.muni.fi.pb162.hello.Hello\]](http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=java -classpath . cz.muni.fi.pb162.hello.Hello)
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 `.` )
[\[http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=.class\]](http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=.class))
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 [\[http://pspad.zde.cz\]](http://pspad.zde.cz) (dostupný zdarma, instalovaný na všech Win strojích v učebnách na FI)

Překlad "Ahoj!" ("for dummies")

Překlad `javac`  překladačem [\[http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=javac\]](http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=javac) (úspěšný, bez hlášení překladače)

Spuštění "Ahoj!" ("for dummies")




Spuštění `java`  voláním [\[http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=java\]](http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=java)


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 `String`  typu `String[] args`  (řetězec)
[\[http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=String\]](http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=String)
- předávají se při spuštění z příkazového řádku do pole `String[] args` 
[\[http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=String\[\] args\]](http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=String%5B%5D)

Metoda `main`  [\[http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=main\]](http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=main) nevrací žádnou hodnotu

WIKIPEDIA [http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=void]
The Free Encyclopedia

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

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>