

```
/** A constructor with default converter
 * @param width A width of the board
 * @param height a height of the board
 * @param content An optional content taken from the method
 */
```

```
public BoardModel(int width, int height, String content) {
    this(width, height, content, new Base64Converter());
}
```

učebnice jazyka

```
private final Board board;
private final Base64Converter converter;
private String errorMessage;
private boolean gameOver;

/** Constructor with default converter
 * @param width A width of the board
 * @param height a height of the board
 * @param content An optional content taken from the method
 */
public Board(int width, int height, String content) {
    this(width, height, content, new Base64Converter());
}
```

na webových příkladech
pro úplné začátečníky



Učebnice jazyka Java na webových příkladech pro úplné začátečníky

Pavel Ponec

<https://jbook.ponec.net/cs.html>

Jazyková korektura: Ing. Vladimíra Krejčíková

Ilustrace obálky: Marie Brogowski

Vnitřní ilustrace: autor knihy (pokud není uvedeno jinak)

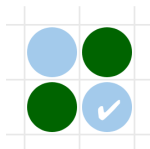
Vydavatel: Pavel Ponec, Brno

ISBN 978-80-270-9286-4

Vydání první, 2021

Poděkování

Rád bych poděkoval své laskavé dceři za mnoho užitečných podnětů a připomínek, bez kterých by tato kniha neměla současnou podobu. Poděkování patří také všem mým dlouhodobým kolegům v týmu za celou řadu užitečných rad, nápadů a konzultací. Kniha byla vytvořena pomocí open-source aplikací včetně ukázkových příkladů, vektorových obrázků a nástrojů pro sestavení výsledku – to vše na svobodném operačním systému Linux. Tímto bych rád vyjádřil také svůj obdiv a poděkování široké komunitě lidí, kteří se starají o vývoj, údržbu a distribuci open-source projektů.



Obsah

Poděkování	1
1 Úvodní informace	6
1.1 Ochranné známky	6
1.2 Vyloučení odpovědnosti	6
1.3 Licence ukázkových příkladů	6
1.4 Typografie	7
1.5 Různé	7
2 Úvod a motivace	8
2.1 Proč vznikla tato kniha?	8
2.2 Co budeme potřebovat?	9
2.3 Proč se učit programovat?	10
2.4 Proč se učit právě jazyk Java?	10
2.5 Proč webové aplikace?	11
2.6 Pár slov o jazyku Java	12
3 Základní pojmy	14
3.1 Data	14
3.2 Algoritmus	14
3.3 Proměnná	16
3.4 Primitivní datové typy	17
3.4.1 Java výrazy a příkazy	18
3.4.2 Přetypování a autoboxing	21
3.5 Texty v jazyce Java	22
3.6 Stromová struktura dat	22
3.7 Model reálného světa	24
3.8 Objekt reálného světa a těleso	24
3.9 Programový objekt	24
3.10 Třída objektů	25
3.10.1 Diagram modelu tříd	25
3.10.2 Název třídy	27
3.10.3 Atributy	27
3.10.4 Metody	28
3.10.5 Konstruktor	28
3.10.6 Třída ve zdrojovém kódu	29
3.10.7 Vytvoření instance třídy a užití	30
3.10.8 Vztahy mezi třídami	33

3.11 Další poznámky ke třídám	37
3.11.1 Viditelnost metod, atributů a tříd	37
3.11.2 Dědičnost	38
3.11.3 Výjimky	41
3.11.4 Balíčky	44
3.11.5 Interface	46
3.11.6 Inline třídy a lambda výrazy	47
3.11.7 Knihovny	48
3.11.8 Komentáře kódu a JavaDoc	48
3.12 Datový typ pole	50
3.12.1 Pole vytvořené metodou třídy	51
3.13 Vybrané třídy standardní knihovny	52
3.13.1 Třída String	52
3.13.2 Rozhraní List	54
Třída ArrayList	54
Generické datové typy	55
List s typem prvku	55
3.13.3 Rozhraní Map	56
3.13.4 Rozhraní Stream	57
3.13.5 Třída BigDecimal	59
3.13.6 Shrnutí kapitoly o třídách	60
3.14 Webové technologie a pojmy	61
3.14.1 XML – rozšiřitelný značkovací jazyk	62
3.14.2 HTML – hypertextový značkovací jazyk	64
3.14.3 CSS – Kaskádové styly	66
3.14.4 URL – Adresa internetové stránky	67
3.15 Shrnutí	68
4 Řešené příklady	70
4.1 Instalace a spuštění	71
4.1.1 Struktura projektu	71
4.1.2 Znakový terminál	71
4.1.3 Získání příkladů	72
4.1.4 Instalace Java a Mavenu	72
4.1.5 Instalace Mavenu na Linux	73
4.1.6 Spouštění příkladů	73
4.1.7 Co dělat při potížích?	74
4.1.8 Virtuální počítač	75

4.1.9 Virtualizační kontejner	75
4.1.10 Vývojové prostředí	77
4.2 Servlety	77
4.2.1 Hello, World!	79
Interface prakticky	80
První servlet	80
Výsledek servletu	83
4.2.2 Datový model HTML stránky	83
Rozšířená implementace	87
Připojení knihovny do projektu	88
4.2.3 Programové výrazy	89
4.3 Tvorba tabulek	91
4.3.1 Jednoduchá tabulka	91
Příkaz for	92
Inkrementace a dekrementace	93
Řešení úkolu	94
Alternativní procházení pole	95
Podmíněný příkaz if-else	96
Tabulka náhodných čísel	96
4.3.2 Přehled vozidel	98
4.4 Formuláře	100
4.4.1 Formulář s jedním vstupním elementem	100
4.4.2 Formulář s více vstupními elementy	102
Regulární výrazy	103
Formulář osobních údajů	104
Ternární operátor	108
Hodnocení platnosti textového parametru	108
4.4.3 Počítání slov	110
4.5 Zábava	112
4.5.1 Bodové kreslení	112
Pomocná třída Base64Converter	114
Parsování textu	114
Class – třída pro popis tříd	114
Logování událostí	115
Třída Optional	115
4.5.2 Binární podoba textu	116
Řešení	118

4.5.3 Piškvorky s defenzivní strategií hry	119
Diagram tříd	120
Jak to funguje?	121
Enumerátor	122
Příkaz switch	124
Popis tříd diagramu a jejich metod	125
Zpracování dotazu	127
Automatizované testy	128
Optimalizace datových toků	129
5 Jak psát dobrý kód	130
Reference a zdroje pro další studium	132
Rejstřík slov	134

1 Úvodní informace

V této kapitole jsou uvedeny některé důležité informace, které však nemají přímý vliv na porozumění technickému výkladu.

1.1 Ochranné známky

Oracle, **Java** a **VirtualBox** jsou registrované ochranné známky společnosti Oracle Corporation anebo jejích poboček. Autor knihy nemá žádnou vazbu ke společnosti Oracle. **Apache Maven**, **Apache Tomcat**, **Apache NetBeans** a **NetBeans** jsou ochranné známky neziskové organizace Apache Software Foundation (ASF) [25]. **Ubuntu**, **Xubuntu** a **Canonical** jsou registrované ochranné známky společnosti Canonical Ltd. **Microsoft** a **Windows** jsou registrované ochranné známky společnosti Microsoft Corporation v USA anebo jiných zemích. **Docker** je registrovaná ochranná známka společnosti Docker, Inc. v USA anebo jiných zemích. Docker, Inc., a také další strany mohou mít práva k ochranné známce. **Eclipse** a **Jetty** jsou registrované ochranné známky společnosti Eclipse Foundation, Inc. ve Spojených státech a případně v dalších jiných zemích.

1.2 Vyloučení odpovědnosti

Mějme na paměti, prosím, že vzorové ukázky programového kódu byly vytvořeny za účelem výuky. Naznačují koncept, ale obsahují některá zjednodušení, například důsledné ošetřování vyhozených výjimek je z důvodu stručnosti omezené, a proto ukázky nejsou určeny přímo k produkčnímu použití.

I když autor věnoval značné úsilí odstraňování chyb a překlepů v textu, nelze vyloučit jejich přehlédnutí. Opravy, které byly zaregistrovány po vydání knihy, najdeme na její domovské stránce [1], kde lze nahlásit případné nesrovnalosti ve výkladu či jiné připomínky. Děkuji za všechny konstruktivní podněty.

1.3 Licence ukázkových příkladů

Po zakoupení knihy si prosím uschovejte doklad o nákupu, protože vás opravňuje používat ukázkové příklady k soukromým i komerčním účelům. Distribuce zdrojového kódu k této knize je možná pouze při zachování původního komentáře v hlavičce. Třetí strana je oprávněna použít zdrojový kód pouze pro své vlastní studium, jinak je třeba získat vlastní licenci zakoupením knihy nebo e-booku. V případě nejasností se obraťte na vydavatele knihy nebo přímo na autora, kontakty najdete na internetové stránce knihy [1].

Zvažte, prosím, že výroba knihy není jedinou nákladovou položkou, další výdaje souvisí s propagací a distribucí. Děkuji tímto za podporu i pochopení.

1.4 Typografie

Slova psaná *kurzívou* budou označovat názvy produktů a nové technické termíny. Kvůli jejich vzrůstajícímu počtu však tyto termíny nebudou vyznačovány v celém rozsahu knihy, ale jen ve vybraných částech pro zdůraznění jejich specifického významu. Slova psaná **neproporcionálním** písmem (znaky s pevnou šířkou) značí programové výrazy a příkazy, klávesy nebo názvy souborů. V ukázkách zdrojového kódu jsou klíčová slova jazyka *Java* vyznačena modře (●), třídy objektů zeleně (●) a literály červeně (●).

1.5 Různé

Jména osob, uvedená v programových ukázkách v této knize, jsou smyšlená za účelem názorného výkladu a jejich případná podobnost s reálnými osobami je čistě náhodná.

2 Úvod a motivace

Write once, run anywhere.

— Sun Microsystems, americká společnost

Od roku 1991 usilovala firma *Sun Microsystems* o vytvoření jednoduchého, robustního, objektově orientovaného programovacího jazyka, jehož aplikace by byly nezávislé na platformě zařízení, hlavní myšlenku reprezentoval slogan v záhlaví této kapitoly. Vedoucím projektu a původním autorem byl kanadský softwarový programátor James Gosling. Roku 1996 vydala firma první oficiální verzi jazyka pod názvem *Java* a o jedenáct let později (roku 2007) uvolnila většinu zdrojových kódů pod licencí **open-source**. Počátkem roku 2010 proběhla akvizice firmy *Sun Microsystems* společností *Oracle*, která se tak stala vlastníkem oficiální implementace Java SE platformy.

2.1 Proč vznikla tato kniha?

Když jsem začínal kdysi s jazykem *Java*, kladl jsem si otázku, zdali není možné napsat učebnici nějakým více atraktivním způsobem. Důležité zkušenosti a nadhled jsem začal nabývat až praxí. Po letech praxe jsem zveřejnil krátký článek o sestavování webových stránek v jazyce *Java*, postavených na objektovém modelu HTML elementů, a přitom mě napadlo, že tohle by mohl být skvělý prostředek pro tvorbu interaktivních učebnicových příkladů při studiu jazyka. V nejbližším knihkupectví jsem otevřel jednu z nových učebnic tohoto jazyka a spatřil opět formální popis s černobílými obrázky, akademickými příklady a několika málo grafickými ukázkami postavenými na dnes už zacházející knihovně Swing. Tehdy jsem se rozhodl vykročit na nejistou dráhu inovativního spisovatele učebnice o programování a výsledek držíte teď v ruce.

Abych přiblížil učebnicové příklady co možná nejvíce reálným aplikacím, rozhodl jsem se názvy programových konstrukcí (jako jsou *třídy* objektů, jejich metody či komentáře kódu) ponechat v angličtině. Pro výklad principů webových aplikací bylo třeba zmínit také stručné základy značkovacího jazyka HTML (z anglického *Hypertext Markup Language*) pro zápis internetových stránek a některé další technologie. Kniha však má také určitý přesah do oblasti návrhu vývoje a životního cyklu informačních systémů. Vzhledem k širšímu záběru témat se kniha nebude zabývat všemi aspekty jazyka do úplných detailů; namísto toho klade větší důraz na vysvětlení ukázek reálného použití, vyjasnění souvislostí a zajímavých vlastností jazyka. Jsem si vědom, že ve snaze udržet dobrou čtivost textu jsem se občas dopustil určitého zjednodušení. Kniha tedy není úplnou specifikací jazyka, spíše se snaží přinést čtenáři nadhled umožňující samostatné získávání potřebných

informací z internetových zdrojů. Výklad se záměrně vyhýbá programovým konstrukcím, které sice jsou podle specifikace jazyka platné, ale jejich použití je považováno za nešikovné či nedoporučované. Výchozím zdrojem dalších informací by mohla být dokumentace standardní knihovny jazyka zvaná *JavaDoc* [5]. Užitečné tipy implementačních postupů lze najít třeba na stránkách serveru *StackOverflow* [15]. Pokud se podaří touto knihou probudit ve čtenáři zájem o vývoj programových aplikací, pak tahle kniha splní jeden ze svých cílů.

2.2 Co budeme potřebovat?

Knihla předpokládá, že čtenář je běžným uživatelem internetu, zvládne instalaci počítačových programů (například hry nebo internetového prohlížeče), orientuje se v základních pojmech, jako jsou například *pevný disk* či *soubor* na disku, a rozumí organizaci ukládaných dat na disku. První kapitola se zaměřuje na motivaci výuky programovacího jazyka *Java* a zmiňuje také výhody webových aplikací.

Pro lepší pochopení obsahu knihy doporučuji zkoušet si jednotlivé příklady na počítači a následně zkoumat vliv vlastních úprav zdrojového kódu na chování spuštěné aplikace. V případě potíží je vždy možné aktuální projekt nahradit původní verzí jejím rozbalením z archivu.

Co tedy budeme potřebovat pro spuštění ukázkových příkladů?

1. Počítač (či notebook) s volnou kapacitou RAM minimálně 1 GB a 1 GB volného místa na pevném disku s operačním systémem *Windows*, *Linux*, *MacOS*, případně i jiným, na který lze nainstalovat *Java Development Kit*.
2. Připojení k internetu nejen pro instalaci potřebných programů, ale i k automatickému stažení nezbytných knihoven během spuštění přiložených příkladů.
3. Prostředí *Java Development Kit*, verze 11 (minimálně 8), pro sestavení a spuštění příkladů.

Knihovna *Maven* (potřebná pro sestavení a spuštění příkladů) je součástí archivu s příklady. Volba operačního systému bude na vás, hlavními kandidáty na výběr jsou *Windows*, *Linux* nebo *MacOS*. Testování příkladů probíhalo na operačním systému *Xubuntu* [17], a nelze vyloučit, že u jiných operačních systémů budou nároky vyšší. Systém *Xubuntu* je lehká linuxová distribuce s grafickým prostředím *Xfce*, podporovaná firmou *Canonical*, která vychází z distribuce *Ubuntu* založené na *Debian GNU/Linux*. Instalace *Javy* a dalších nezbytných knihoven ukrojí z pevného disku asi 500 MB.

2.3 Proč se učit programovat?

Pokud se vám dostala tato kniha do ruky, zřejmě už jste nějakou motivaci našli. Pracovní trh se potýká s nedostatkem odborníků, a to nejen v oboru informačních technologií (dále jen IT). V mnoha oborech může být řešením nasazení automatizace či umělé inteligence, nic z toho se však bez vývojářů softwaru (SW) neobejde, což ještě více posílí poptávku po IT odbornících. Běžným obchodníkům pomůže někdy i jednoduchá prezentace zboží (či služeb) na internetu s možností sestavení objednávky a následné platby.

Je dobré vědět, že zadání projektu i výsledek práce SW vývojáře se sdílí relativně snadno přes internet, a tak pokud někdo má dostatečné jazykové znalosti, a zároveň získá důvěru zákazníka, může dodávat svoji práci třeba na druhý konec zeměkoule z pohodlí svého domova.

2.4 Proč se učit právě jazyk Java?

Říká se s nadsázkou: „Kolik právníků, tolik názorů.“ U programovacích jazyků je to podobné: diskuze o výhodách různých programovacích jazyků bývají velmi emotivní, přitom každý argument může mít svůj díl pravdy. V této nepřehledné situaci může být pro začátečníka obtížné vybrat správný směr. Proč tedy právě *Java*?

1. Programovací jazyk *Java* patří na trhu práce dlouhodobě k nejžádanějším, je tedy pravděpodobné, že čas strávený studiem se podaří brzy zhodnotit [18].
2. Hotové aplikace lze spouštět (při dodržení určitých pravidel) na různých operačních systémech (například: *Windows*, *Linux*, *MacOS*), pokud je na nich k dispozici běhové prostředí zvané *Java Runtime Environment* (dále jen *JRE*). Takové schopnosti programovacího jazyka se označují výrazem multiplatformní.
3. Nástroje nezbytné pro vývoj a provoz standardních komerčních aplikací lze získat zdarma, včetně sofistikovaných editorů zaměřených na vývoj aplikací (dále jen IDE), podporujících i sestavení projektu a jeho spuštění. Referenční implementace *Javy* s otevřenou licencí má název *OpenJDK* [4], existují však implementace i jiných dodavatelů.
4. *Java* patří mezi jazyky s mimořádně rozsáhlou nabídkou různých **knihoven** s otevřeným zdrojovým kódem (anglicky *open-source*), které mohou vývojářům ušetřit drahý čas či zbytečnou práci.
5. Kolem jazyka se vytvořila silná komunita, která přispívá ke včasnému odhalení případných problémů.

6. Vývojáři jazyka *Java* dosud věnovali značnou pozornost zpětné kompatibilitě jazyka při vydávání jeho nových verzí. V praxi to znamená, že uživatel má velkou naději, že spustí starší *Java* projekt i na poslední verzi prostředí (anglicky *run-time*), a vývojáři zase mohou používat (kompilovat) starý zdrojový kód bez úprav v poslední verzi překladače.
7. Jazyk nabízí silnou typovou kontrolu, která urychluje odhalení některých chyb zápisu zdrojového kódu, a tím šetří čas potřebný na testování aplikace.
8. *Java* se řadí k programovacím jazykům, které za běhu aplikace uvolňují automaticky paměť po nepotřebných objektech, což urychluje vývoj a eliminuje některé problémy s vyčerpáním paměti. Uvolňování paměti má na starost algoritmus označovaný anglickým termínem *Garbage collector*. V určitých situacích (například během řízení časově kritických procesů) však může krátké časové prodlení spojené s uvolněním paměti představovat handicap, který však lze eliminovat výběrem vhodné strategie algoritmu *Garbage collector*.
9. Programovací jazyk *Java* je k dispozici vývojářům relativně dlouhou dobu a dnes už je standardem pro výuku objektového programování (OOP). Znalost principů objektového programování jazyka *Java* může usnadnit pochopení objektového přístupu také v jiných jazycích.

2.5 Proč webové aplikace?

Webové aplikace pro internetové prohlížeče se těší stále větší oblibě koncových uživatelů i provozovatelů a důvodů je hned několik.

1. Internetové připojení je dnes už snadno dostupné pro většinu uživatelů osobních počítačů i mobilních zařízení.
2. Nové internetové technologie umožňují vývojářům stále zlepšovat ergonomii webového uživatelského rozhraní, které se pomalu blíží komfortu desktopových aplikací.
3. Webové aplikace jsou na operačním systému prakticky nezávislé, klienti tedy mohou používat libovolné zařízení, které poskytuje internetový prohlížeč.
4. Distribuce nových verzí SW produktů k zákazníkovi má minimální náklady.
5. Pro útočníka je obtížné webovou aplikaci odcizit, nebo prolomit její licenční omezení.