

2. vydání

Pavel Ponec

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/cs2>

Odborná korektura: Ing. Ondřej Guth, Ph.D.

Jazyková korektura: Ing. Vladimíra Krejčíková

Ilustrace obálky: Marie Brogowski

Vnitřní ilustrace: autor knihy (pokud není uvedeno jinak)

Vydavatel: Pavel Ponec, Brno

ISBN 978-80-11-01484-1

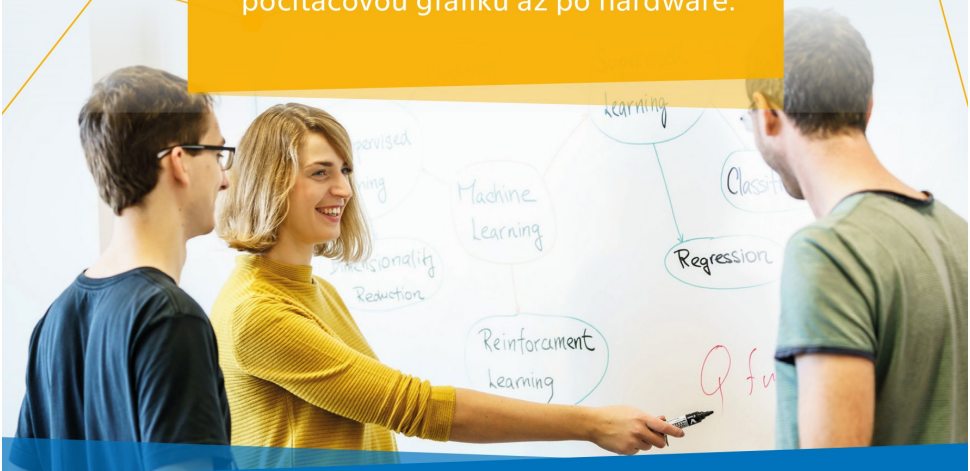
Vydání druhé, 2022



FAKULTA
INFORMAČNÍCH
TECHNologiÍ
ČVUT V PRAZE

PROPOJÍME TĚ SE SVĚTEM IT

Kvalitní vzdělání ve všech oblastech informatiky od informačních systémů a webových technologií přes počítačovou grafiku až po hardware.



PROČ JÍT NA FIT?

- ▶ 100% pokrytí všech oblastí IT
- ▶ více jak 200 firem pro praxi při studiu
- ▶ 100% zaměstnanost na trhu práce
- ▶ 19 špičkově vybavených laboratoří
- ▶ 12 výzkumných skupin

BAKALÁŘSKÝ STUDIJNÍ PROGRAM

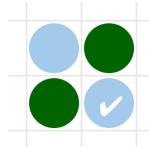
- ▶ titul bakalář (Bc.)
- ▶ standardní doba studia 3 roky
- ▶ 10 specializací

NEPOTŘEBUJEŠ PŘEDCHOZÍ ZNALOST IT. VŠE TĚ NAUČÍME.



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 své uznání a poděkování také široké komunitě lidí, kteří se starají o vývoj, údržbu a distribuci open-source projektů. V neposlední řadě děkuji také doc. RNDr. Josefu Kolářovi, CSc., a Ing. Ondřeji Guthovi, Ph.D., z Katedry teoretické informatiky FIT ČVUT za pomoc při druhém vydání této knihy.



Obsah

Poděkování	1
1 Předmluva	6
2 Úvodní informace	7
2.1 Ochranné známky	7
2.2 Vyloučení odpovědnosti	7
2.3 Licence ukázkových příkladů	8
2.4 Typografie	8
2.5 Různé	8
3 Úvod a motivace	9
3.1 Proč vznikla tato kniha?	9
3.2 Co budeme potřebovat?	10
3.3 Proč se učit programovat?	10
3.4 Proč se učit právě jazyk Java?	11
3.5 Proč webové aplikace?	12
3.6 Pár slov o jazyku Java	12
4 Základní pojmy	14
4.1 Data	14
4.2 Algoritmus	14
4.3 Proměnná	17
4.4 Primitivní datové typy a přetypování	17
4.5 Java výrazy a příkazy	20
4.6 Stromová struktura dat	22
4.7 Model reálného světa	23
4.8 Fyzikální objekt	24
4.9 Programový objekt	24
4.10 Třída objektů	25
4.10.1 Diagram modelu tříd	25
4.10.2 Název třídy	27
4.10.3 Atributy	27
4.10.4 Metody	28
4.10.5 Konstruktor	28
4.10.6 Třída ve zdrojovém kódu	29
4.11 Užití objektu	30
4.11.1 Texty v jazyce Java	30
4.11.2 Vytvoření instance a užití třídy	31

4.11.3 Autoboxing	33
4.11.4 Neplatné objekty null	34
4.11.5 Vztahy mezi třídami	35
4.12 Další poznámky ke třídám	39
4.12.1 Viditelnost metod, atributů a tříd	39
4.12.2 Dědičnost	41
4.12.3 Výjimky	44
4.12.4 Balíčky	47
4.12.5 Komentáře kódu a JavaDoc	50
4.12.6 Interface	52
4.12.7 Inline třídy a lambda výrazy	54
4.12.8 Knihovny	55
4.13 Datový typ pole	55
4.13.1 Pole vytvořené metodou třídy	57
4.14 Vybrané třídy standardní knihovny	57
4.14.1 Třída String	57
4.14.2 Rozhraní List	59
Třída ArrayList	60
Generické datové typy	61
List s typem prvku	61
4.14.3 Rozhraní Map	62
4.14.4 Rozhraní Stream	63
4.14.5 Třída BigDecimal	64
4.14.6 Shrnutí kapitoly o třídách	66
4.15 Webové technologie a pojmy	66
4.15.1 XML – rozšiřitelný značkovací jazyk	67
4.15.2 HTML – hypertextový značkovací jazyk	70
4.15.3 CSS – Kaskádové styly	71
4.15.4 URL – Adresa internetové stránky	74
4.15.5 JSON – úsporný datový formát	75
4.16 Shrnutí	77
5 Řešené příklady	78
5.1 Instalace	79
5.1.1 Hardware a operační systém	79
5.1.2 Apache Maven	80
5.1.3 Znakový terminál	81
5.1.4 Získání příkladů	81

5.1.5 Instalace vývojového prostředí	81
5.1.6 Instalace Mavenu na Linux	82
5.1.7 Spouštění příkladů	82
5.1.8 Co dělat při potížích?	83
5.1.9 Virtuální počítač	84
5.1.10 Virtualizační kontejner	85
5.1.11 Vývojové editory	86
5.2 Servlety	87
5.2.1 Hello, World!	88
První servlet	90
Výsledek servletu	93
5.2.2 Datový model HTML stránky	93
Připojení knihovny do projektu	97
5.2.3 Programové výrazy	97
5.3 Tvorba tabulek	100
5.3.1 Jednoduchá tabulka	100
Příkaz for	101
Inkrementace a dekrementace	102
Řešení úkolu	102
Alternativní procházení pole	104
Podmíněný příkaz if-else	104
Tabulka náhodných čísel	105
5.3.2 Přehled vozidel	107
5.4 Formuláře	109
5.4.1 Formulář s jedním vstupním elementem	109
5.4.2 Formulář s více vstupními elementy	111
Regulární výrazy	112
Formulář osobních údajů	113
Ternární operátor	117
Hodnocení platnosti textů	117
5.4.3 Počítání slov	118
5.5 Zábava	120
5.5.1 Bodové kreslení	120
Pomocná třída Base64Converter	122
Parsování textu	123
Class – třída pro popis tříd	123
Logování událostí	123

Třída Optional	124
5.5.2 Binární podoba textu	125
Řešení	126
5.5.3 Piškvorky s defenzivní strategií hry	127
Diagram tříd	128
Jak to funguje?	129
Enumerátor	130
Příkaz switch	132
Popis tříd diagramu a jejich metod	133
Zpracování dotazu	135
Automatizované testy	136
5.6 Interaktivní aplikace s AJAX	137
5.6.1 Co je to AJAX	137
5.6.2 Testování regulárních výrazů	138
5.6.3 Piškvorky s rychlejší odezvou	141
6 Jak psát dobrý kód	145
Reference a zdroje pro další studium	147
Rejstřík slov	150

1 Předmluva

Kniha, kterou právě čtete, je úvodem do světa tvorby aplikací. Nečekejte však suchý popis programovacího jazyka. Čtenář se na jednoduchých příkladech naučí, jak při tvorbě programů přemýšlet. Pokud vás v této souvislosti straší slovo „algoritmizace“, kniha vás nezklame. Neobsahuje partie z teoretické informatiky, namísto toho je vše vysvětleno formou příkladů pochopitelných i pro naprosté nováčky ve světě tvorby software.

Tato publikace však návodem, jak při programování přemýšlet, nekončí. Čtenář se dozví, jak používat jazyk Java, jeho pravidla, a v neposlední řadě knihovny, které s tvorbou software pomohou. Tento jazyk je již dlouho osvědčenou volbou pro výuku základů programování, jeho znalost zároveň uplatníte i v praxi. Ani jazykem Java záběr knihy nekončí. V knize se dozvíte o základních technologiích webu a naučíte se tvořit webové aplikace.

Text zabíhá i mimo téma základů nutných pro vytvoření fungujícího programu. Dozvíte se, jak zdrojový kód organizovat, aby byl přehledný a pochopitelný, nebo proč a jak zdrojový kód obohatit o dokumentaci. Jistě oceníte popis aplikací, které s programováním pomohou, například editoru – vývojového prostředí. Seznámíte se i s dlouhodobě osvědčenou a stále používanou technologií, díky které je možné své softwarové projekty efektivně spravovat nebo sdílet s dalšími vývojáři. Kromě příkladů částí programu v textu je součástí knihy i hotová aplikace spolu s návodem na její zprovoznění. Vedle možnosti prohlédnutí všech detailů je tak možné se učit pomocí zkoušení si vlastních úprav.

Přeji příjemné vykročení do oblasti programování při čtení této knihy.

Ing. Ondřej Guth, Ph.D.

2 Úvodní informace

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

2.1 Ochranné známky

Oracle, **Java** a **VirtualBox** jsou registrované ochranné známky společnosti Oracle Corporation anebo jejich 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. **ECMAScript** je ochranná známka skriptovacího jazyka specifikovaná standardem ECMA international.

2.2 Vyloučení odpovědnosti

Mějme na paměti, prosím, že vzorové ukázky programového kódu byly vytvořeny za účelem výuky. Naznačují koncept, ale obsahují některá zjednodušení, například 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.

2.3 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. Licence tak umožňuje pedagogům použití ukázkového projektu při výuce. 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.

2.4 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 (●).

2.5 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.

3 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.

3.1 Proč vznikla tato kniha?

Když jsem začínal s jazykem *Java*, kladl jsem si otázku, zdali není možné napsat učebnici nějakým více atraktivním způsobem. Po mnoha letech 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 máte před sebou právě teď.

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 ukázkou praktického použití, vyjasnění souvislostí a užitečných vlastností. 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,

kteřé jsou podle specifikace jazyka platné, ale jejich použití je považováno za nešikovné či nedoporučované. Doplňujícím zdrojem 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.

3.2 Co budeme potřebovat?

Knihá 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 uložených *souborů* na disku. První kapitola se zaměřuje na motivaci výuky programovacího jazyka *Java* a zmiňuje také výhody webových aplikací.

Pro lepší pochopení obsahu knihy doporučuji zkusit 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 17, pro sestavení a spuštění příkladů.

Další podrobnosti lze najít v kapitole [O instalaci](#).

3.3 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 neobejde bez vývojářů softwaru (SW), 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 získá zároveň důvěru zákazníka, může dodávat svoji práci třeba na druhý konec světa z pohodlí svého domova.

3.4 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 *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. Standardní *Java* obsahuje také sofistikovanou knihovnu na tvorbu plnohodnotných desktopových aplikací pro podporované platformy.
3. Nástroje nezbytné pro vývoj a provoz standardních komerčních aplikací lze získat zdarma, včetně pokročilých editorů zaměřených na vývoj aplikací (dále jen IDE z anglického *Integrated Development Environment*), podporujících i sestavení projektu a jeho spuštění. Referenční implementace *Javy* s otevřenou licencí má název *OpenJDK* [4]. Při instalaci *Javy* však nejsme vázani na jediného dodavatele, její implementaci nabízí několik dalších společností.
4. Vývojáři *Javy* mají k dispozici rozsáhlou nabídku různých **knihoven** s otevřeným zdrojovým kódem (anglicky *open-source*), které pokrývají široké spektrum nejrůznějších témat, což šetří vývojářům drahý čas či zbytečnou práci. Jedním tématem se někdy zabývá hned několik knihoven, které se liší svým programovým rozhraním či implementací.
5. Kolem jazyka se vytvořila silná komunita, která přispívá ke včasnému odhalení případných chyb či komplikací.
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 prodloužení spojené s uvolněním paměti představovat handicap, který lze ovšem 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.

3.5 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.

3.6 Pár slov o jazyku Java

Java je objektový programovací jazyk se silnou typovou kontrolou. Objektově orientované programování (dále *OOP*) je specifický způsob zápisu zdrojového kódu do textových souborů popisujících *třídy*, které mají svůj *název*, *datovou složku* a skupinu *algoritmů*, jež jsou obsaženy v pojmenovaných útvarech zvaných *metody*. Vlastnostmi *OOP* se budeme zabývat později, předtím si však ještě ujasníme některé základní pojmy z informatiky.

Zdrojový kód jazyka *Java* se zapisuje do textových souborů (s příponou *java*) podle pravidel, kterým se říká *specifikace jazyka* [3]. Jak se programovací jazyk vyvíjí, vznikají nové verze specifikace, které korespondují s určitým *Java* prostředím