edice
začínáme s ...

Začínáme programovat v jazyku JAVA

- Nepředpokládá žádné předchozí znalosti programování
- Neomezuje se na výuku kódování, ale učí, čtenáře programovat podle moderních zásad a metodik
- Probírá všechny důležité konstrukce jazyka včetně těch, které začátečnické učebnice přeskakují
- Všechny probírané konstrukce vysvětluje na příkladech
- Pro výklad složitějších programových konstrukcí používá syntaktické diagramy

soubory ke stažení na www.grada.cz

RUDOLF PECINOVSKÝ
JARMILA PAVLÍČKOVÁ

GRADA®

edice
začínáme s ...

Začínáme programovat v jazyku JAVA

RUDOLF PECINOVSKÝ
JARMILA PAVLÍČKOVÁ

GRADA Publishing

Rudolf Pecinovský, Jarmila Pavlíčková

Začínáme programovat v jazyku Java

Vydala Grada Publishing, a.s.
U Průhonu 22, Praha 7
obchod@grada.cz, www.grada.cz
tel.: +420 234 264 401
jako svou 8318. publikaci

Odpovědný redaktor Petr Somogyi
Fotografie na obálce Depositphotos/gdolgikh
Grafická úprava a sazba Rudolf Pecinovský
Počet stran 400
První vydání, Praha 2021
Vytiskla TISKÁRNA V RÁJI, s.r.o, Pardubice

ISBN 978-80-271-4641-3 (ePub)
ISBN 978-80-271-4640-6 (pdf)
ISBN 978-80-271-3062-7 (print)

Všem, kteří se chtějí něco naučit

Stručný obsah

Podrobný obsah

Část B Začínáme programovat 79

Úvod

Otevíráte čtvrté, zcela přepracované vydání knížky, která vás chce naučit programovat moderním, objektově orientovaným stylem. Stylem, jímž se v dnešní době vyvíjí drtivá většina klíčových aplikací. Oproti předchozím vydáním je kniha od základů přepracovaná. Stejně jako v předchozím vydání je výklad rozdělen do dvou dílů.

První díl (ten právě čtete) se soustředí především na výklad základních syntaktických konstrukcí a architektonických principů, které by si měl čtenář osvojit předtím, než se pustí do kódování složitějších projektů. Jeho cílem je, aby čtenáři přešly tyto principy co nejdříve do krve.

Plánovaný druhý díl pak získané návyky prohloubí a seznámí čtenáře s řadou dalších programátorských technik a prohloubí znalosti standardní knihovny. Postupně se v něm navrhuje jednoduchá, ale na druhou stranu netriviální aplikace. V průběhu tohoto návrhu kniha vysvětluje hlubší souvislosti, na něž v běžných učebnicích již nezbývá místo, a doplní čtenářovy znalosti zásad návrhu aplikací.

Kniha je výsledkem mnohaletého experimentování s tím, jak co nejlépe učit *soudobé* programování. Vyzkoušeli jsme si výuku programování snad na všech typech zařízení. Učili jsme v zájmových kroužcích na základní škole, na střední škole, na univerzitách i v rozšiřujících kurzech pro profesionální programátory.

Ve svých kurzech neustále zjišťujeme, že střední i vysoké školy opouští řada programátorů, kteří sice programují v nějakém „moderním“ jazyce, ale neumějí programovat moderně. Absolvované kurzy je sice naučily navržený program zakódovat, ale nenaučily je netriviální program samostatně navrhnout. Naučily je používat nejrůznější frameworky, ale oni tyto frameworky používají většinou mechanicky, bez pochopení základních principů, na jejichž základě jsou navrženy.

Řada vysokých škol se v informatických oborech snaží naučit své studenty zakódovat program v řadě nejrůznějších jazyků. Neučí je však styl programování, ale především syntaxi a sémantiku probíraných jazyků. Důsledkem jsou pak nestabilní a těžko udržovatelné aplikace.

Ve svých učebnicích se to pokoušíme napravit. Reakce čtenářů prozatím naznačují, že se nám to snad alespoň částečně daří.

Tato učebnice je již podle svého názvu určena pro naprosté začátečníky: máme-li dodržet akceptovatelný rozsah, musíme probrat opravdu jen základy. Pro zájemce proto chystáme učebnici pro mírně pokročilé, která se nesoustředí na kód, ale bude se věnovat spíše návrhu. Představili bychom v ní další zásady moderního programování, včetně např. automatizovaného testování, a v závěru bychom navrhli jednoduchou, avšak netriviální aplikaci od úplného počátku. Vše záleží na zájmu potenciálních čtenářů.

Komu je kniha určena

Tato kniha je určena především těm, kteří ještě nikdy neprogramovali, anebo se je to sice někdo snažil naučit, ale oni už vše zase zapomněli. Knihu jsme se snažili napsat tak, aby ji mohl použít bystrý středoškolák. Nepředpokládá žádné předběžné znalosti a dovednosti kromě základů práce s počítačem. Jejím cílem je předat čtenáři základní znalosti a naučit ho dovednosti potřebné k vytváření jednoduchých aplikací. Osvojené základy mu pak umožní, aby v případě hlubšího zájmu o programování v jazyku *Java* pokračoval některou z učebnic určených pro mírně pokročilé programátory – nejlépe samozřejmě chystaným druhým dílem a referenční příručkou [19] či její následovnicí.

Koncepce výkladu

Musíme vás upozornit na to, že kniha, kterou držíte v ruce, se od běžných učebnic poněkud liší. Ostatní učebnice jsou totiž většinou především učebnicemi nějakého programovacího jazyka. Jejich autoři se proto ve svém výkladu soustředí hlavně na výklad vlastností popisovaného jazyka a jeho knihoven. Bohužel se v nich ale nedozvíte skoro nic o tom, jak při návrhu programů přemýšlet, aby vás nezaskočily náhlé změny zadání, kterými je současné programování pověstné. Takovéto učebnice proto nevychovávají návrháře, kteří by uměli program navrhnout, ale pouze kodéry, kteří umějí zanalyzované zadání zakódovat.

Tady si neodpustíme vzpomínku na jednoho studenta, který na konci semestru vysvětloval, že už profesionálně programuje, takže si myslel, že kurz hravě zvládne. Když jsme na začátku semestru probírali naprosté základy, řekl si, že to je pouhé hraní a že si počká, až začneme doopravdy programovat, a pak chybějící body rychle dožene. Když se k nám však v polovině semestru připojil, zjistil, že řadu konstrukcí, které běžně používáme, vůbec nechápe. Nezbylo nám, než mu zopakovat, že mezi prostým používáním objektových konstrukcí a návrhem objektových programů je velký rozdíl a že řada účastníků přichází do našich kurzů právě proto, aby se tento jiný způsob programátorského myšlení naučila.

Vypadá to, jako když autoři předpokládají, že se při čtení jejich knihy naučíte programovat nějak sami od sebe – obdobně, jako se to museli naučit oni. Zkušenosti s programátory, kteří navštěvují naše kurzy ve firmě či na univerzitě, však ukazují, že tohoto výsledku bývá dosaženo jen zřídka.[1] Většina z nich zná poměrně dobře konstrukce nějakého objektově

[1] Výzkum z přelomu století ukázal, že pouze 10 % programů psaných v objektově orientovaných jazycích je navrženo opravdu objektově. (Goddard, D. 1994. Is it really object oriented? *Data Based Advis.* 12, 12 (Dec. 1994), 120-123.) Od té doby se situace trochu zlepšila, nicméně na svých školeních stále pozoruji, že strukturovaně navržené programy psané v objektových jazycích v mnoha softwarových firmách převažují.

orientovaného programovacího jazyka a základy často používaných frameworků. Bohužel, skoro nikdo z nich v něm neumí objektově programovat, neumí přepnout na objektový způsob uvažování.

Pro jistotu proto varuji: toto není učebnice jazyka *Java*, toto je učebnice objektového programování v *Javě*. My se primárně nebudeme učit, jak program zapsat, ale jak jej navrhnout. Cílem není vychovat z čtenářů kodéry v Javě, ale připravit je na to, aby z nich mohli vyrůst schopní architekti. Nebudeme vás proto učit specialitám použitého programovacího jazyka (ty jsou podrobně popsány v knize [19]), ale pokusíme se vás naučit efektivně navrhovat a vytvářet spolehlivé a snadno udržovatelné programy. Jinými slovy: chceme vás naučit dovednostem, které budete používat, ať už budete programovat v jakémkoliv objektově orientovaném jazyku. Jazyky přicházejí a odcházejí. Základní programátorské techniky a způsob myšlení však žijí daleko déle než jazyky, se kterými byly zavedeny.

Díky tomu, že se místo na programovací jazyk soustředíme spíše na vlastní programování, vznikl v knize prostor pro výklad řady programátorských zásad a technik, o nichž se klasické učebnice vůbec nezmiňují, a to často ani učebnice pro zkušené programátory. Tyto zásady a techniky se většinou přednášejí až v nadstavbových kurzech, které učí vytvářet programy tak, aby s vámi mohly růst a aby vás nezaskočily rychle se měnící požadavky zákazníků (a připravte se na to, že tyto měnící se požadavky vás potkají i v případě, kdy jste zákazníkem sami sobě). Přitom vůbec nejde o techniky složité, které by začátečník nezvládl pochopit. Ostatní učebnice je pomíjejí pouze proto, že nesouvisejí přímo se syntaxí programovacího jazyka, ale týkají se obecného programování.

Tato učebnice je naopak vysvětluje od samého počátku výkladu, protože víme, že byste si je měli osvojit co nejdříve. Ti, kteří se nejprve učí kódovat, a teprve následně se dozvídají, jak lze návrh programu zefektivnit, bývají příliš v zajetí svých zkušeností s kódováním a při návrhu programů se pak zbytečně silně soustředí na některé nepodstatné detaily.

Potřebné vybavení

K vývoji programů budete potřebovat vývojovou sadu JDK, kterou můžete stáhnout na adrese https://www.oracle.com/java/technologies/javase-downloads.html. Potřebujete ale sadu JDK 17 nebo mladší. Na starších verzích *Javy* by naše programy nemusely pracovat. Počítejte s tím, že instalační soubory zabírají zhruba 160 MB a po instalaci bude sada zabírat přes 300 MB.

K vývojové sadě je vhodné si stáhnout a nainstalovat i dokumentaci. Pokud ale neopouštíte web, tak můžete využívat i její trvalou instalaci na webu. Musíte se však smířit s tím, že ji seženete pouze anglicky. ZIP soubor s dokumentací zabírá asi 50 MB a po rozbalení bude zabírat zhruba 460 MB.