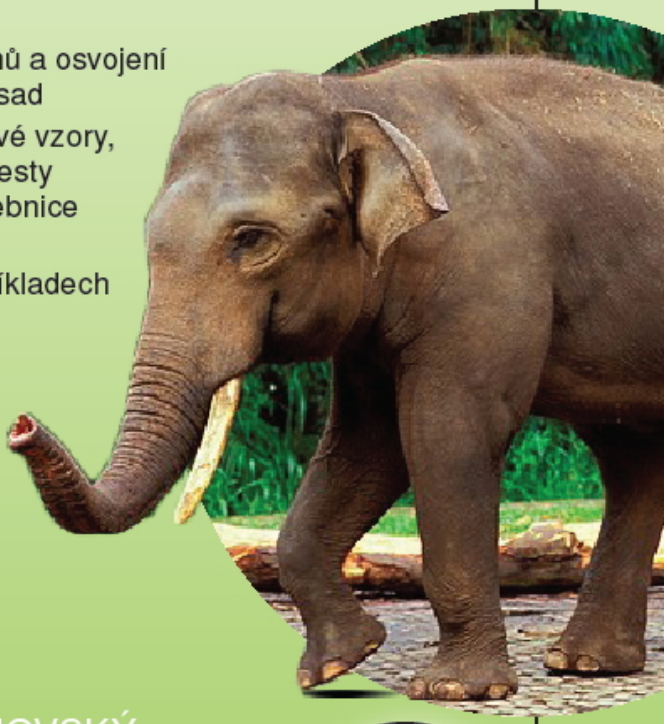
EDICE
myslíme v ...

knihovna programátora

- Učebnice pro ty, kteří nechtějí zůstat obyčejnými kodéry, ale chtějí se stát špičkovými architekty
- Probírá novinky Javy 8, které ovlivňují návrh architektury programu
- Soustředí se na návrh programů a osvojení klíčových architektonických zásad
- Vysvětluje a procvičuje návrhové vzory, refaktoraci kódu, vývoj řízený testy a další oblasti, které běžné učebnice ignorují
- Vše průběžně procvičuje na příkladech řešených spolu se čtenářem
- Doporučená učebnice na řadě středních škol i univerzit



RUDOLF PECINOVSKÝ

Java 8

Úvod do objektové architektury pro mírně pokročilé

GRADA

Java 8

úvod do objektové architektury pro mírně pokročilé

Rudolf Pecinovský
2014

Rudolf Pecinovský

Java 8

Úvod do objektové architektury pro mírně pokročilé

TIRÁŽ TIŠTĚNÉ PUBLIKACE

Vydala Grada Publishing a.s.
U Průhonu 22, Praha 7
jako svoji 5670. publikaci

Odborní lektoři:
doc. Ing. Pavel Herout, Ph.D.,
doc. MUDr. Jiří Kofránek, CSc.,
doc. Ing. Vojtěch Merunka, Ph.D.,
doc. Ing. Miroslav Virius, CSc.

Odpovědný redaktor: Martin Vondráček, Ladislava Soukupová
Návrh vnitřního layoutu: Rudolf Pecinovský
Zlom: Rudolf Pecinovský
Počet stran 656
První vydání, Praha 2014
Vytiskla tiskárna PROTISK, s. r. o.

ISBN 978-80-247-4638-8

TIRÁŽ ELEKTRONICKÉ PUBLIKACE

ISBN 978-80-247-9480-8 (ve formátu PDF)
ISBN 978-80-247-9481-5 (ve formátu EPUB)

Mé ženě Jarušce a dětem
Štěpánce, Pavlínce, Ivance a Michalovi

Stručný obsah

Podrobný obsah

Skrytí spoluautoři

Při tvorbě takto rozsáhlé knihy se autor nemůže spolehnout pouze sám na sebe. Je známou věcí, že když po sobě autor čte svůj rukopis, čte velmi často to, co chtěl napsat, a ne to, co doopravdy napsal. S kontrolou jazykových chyb mu může pomoci redaktor a jazykový korektor. S kontrolou odborných zaškobrtnutí mu však musí pomoci někdo jiný.

S kontrolou odborné správnosti musí pomoci někdo, kdo je odborně na výši a odhalí, že se autor vyjadřuje nepřesně nebo dokonce chybně. S kontrolou srozumitelnosti výkladu musí zase pomoci někdo, kdo se chce látku naučit a je ochoten text podrobně pročíst a upozorňovat na místa, kde výkladu zcela nerozumí nebo se mu zdá, že by bylo vhodné ještě něco doplnit. Úplně dokonalé pak je, když váš text čte zkušený učitel, který umí odhalit nejenom oba výše popsané druhy nepřesností či dokonce chyb, ale také naznačit možná potenciální nedorozumění a dezinterpretace vzniklá z podobnosti vykládané látky s jinými oblastmi, o nichž se výklad nezmiňuje.

Měl jsem to štěstí, že se mi podařilo získat spolupracovníky (a tím i kritiky) ze všech tří zmiňovaných skupin. Všem bych jim chtěl tímto poděkovat. Řekl bych, že díky jejich připomínkám a doporučením je výsledný text kvalitnější.

Všechny tyto dobrovolné spolupracovníky vnímám jako spoluautory. Proto mi dovolte, abych je tu alespoň abecedně vyjmenoval – byli to: Milan Augustin, Tibor Bako, Jakub Hadam, Michael Charvát, Jiří Kofránek, Jiří Kubala, Petr Kuchař, David Král, Filip Malý, Nikolas Patrik, Jarmila Pavlíčková, Luboš Pavlíček, Josef Svoboda, Petr Vidrman, Martin Vondráček a Martin Žamberský.

Úvod

Otevíráte knížku, která vás chce naučit programovat moderním, objektově orientovaným stylem. Stylem, jímž se v dnešní době vyvíjí drtivá většina klíčových aplikací, ale k jehož výuce ještě řada škol nedospěla. Po nastudování této knížky budou proto mnozí z vás vědět o moderním programování víc než leckterý z vašich učitelů.

Komu je kniha určena

Kniha je určena programátorům, kteří potřebují získat hlubší vhled do problematiky objektově orientovaného programování a návrhu architektury objektově orientovaných programů. K jejímu studiu přitom nepotřebují žádné velké předběžné zkušenosti. Stačí znalosti na úrovni mých začátečnických učebnic.

Kniha je reakcí na nepřetržité nářky vedoucích programátorských týmů, kteří posílají nastoupivší absolventy do mých přeškolovacích kurzů. Stěžují si, že školy opouštějí možná skvělí kodéři, ale neschopní architekti, že tito absolventi znají několik programovacích jazyků a řadu užitečných frameworků, ale mají problém s návrhem kvalitní architektury zadávaného programu.

Většina škol, učebnic a výukových kurzů se soustředí především na výklad syntaxe probíraného programovacího jazyka a probrání hlavních knihoven používané platformy. Jejich autoři se soustřeďují na detaily v bláhové naději, že jejich studenti a čtenáři pak v průběhu následující praxe pochopí vyšší principy. Když se ve výukovém programu objeví kurzy návrhových vzorů, končí často jako defilé technik, jak to či ono šikovně zakódovat.

Koncepce knihy

Ve svých knihách se pokouším tento stereotyp změnit. Soustředím se především na předvádění toho, jak program navrhnout a probrat věci, které považuji za důležité a přitom je v jiných učebnicích většinou nenajdete. Přitom se pokouším vše demonstrovat na příkladech. Na rozdíl od některých učebnic se však nechci omezovat pouze na AHA-příklady, jejichž jediným cílem je, aby po jejich prostudování student prohlásil „Aha! Takto to funguje!“, i když se jim v některých chvílích nevyhnu. Snažím se ale co největší část látky demonstrovat na složitějších příkladech, které však nejsou zahlceny spoustou šumu, za který považuji kód, který se

musí naprogramovat, i když s vykládanou látkou přímo nesouvisí. Dávám přitom přednost příkladům používajícím grafiku, protože jsou pro daný účel většinou nejnázornější.

Kniha je koncipována jako druhý díl knihy *Java 7 – Učebnice objektové architektury pro začátečníky*[1]. Předpokládá proto, že znáte látku zhruba na úrovni prvního dílu. Protože ale očekávám, že knihu bude číst i řada čtenářů, kteří získali své první programátorské zkušenosti z jiných zdrojů, tak pro jistotu na počátku nejdůležitější věci velice stručně zopakuji.

Co se naučíte, uspořádání knihy

Snažil jsem se v knize soustředit na oblasti, které se do jiných knih nevešly, a přitom je jejich osvojení velmi důležité, protože jejich neznalost je v lepším případě příčinou těžkopádných řešení a v horším případě příčinou špatně odhalitelných chyb.

V první části nejprve bleskově zopakuji základní látku prvního dílu a pak vás seznámím s jedním ze tří nejpoužívanějších profesionálních vývojových prostředí: s prostředím *NetBeans*. V minulém dílu jsme pracovali s vývojovým prostředím *BlueJ*. To je sice asi nejlepší prostředí pro získání základních představ a návyků objektově orientovaného programování, ale pro profesionální práci se příliš nehodí. Protože je ale zvládnutí vývojového prostředí zhruba stejně náročné jako zvládnutí programovacího jazyka, je mu věnován celý zbytek první části. Naučíte se je konfigurovat, vytvářet v něm projekty a využívat jeho výhodných vlastností pro zefektivnění své práce.

V druhé části si postupně ukážeme různé programové konstrukce a postupy, které můžete využít při návrhu architektury projektu. Postupně se seznámíte s dalšími návrhovými vzory, osvojíte si definici interfejsů obsahujících definice metod včetně jejich implementace, dozvíte se o možnosti parametrizace datových typů a jejich vzájemném vnořování, naučíte se používat lambda-výrazy, které umožňují pracovat s částmi kódu jako s proměnnými, a získáte první zkušenosti s prací s kontejnery a datovody.

Ve třetí části se ponoříme do výkladu dědění implementace. Začneme podrobným výkladem zavádění objektů a tříd a budeme pokračovat základními vlastnostmi dědění implementace. Dozvíte se, jak vytvářet třídy, jejichž předky jsou jiné třídy, a hlavně na co si dát při konstrukci takovýchto tříd pozor. Pokusím

[1] Kniha je současně slibovaným druhým dílem starší učebnice *OOP – Naučte se myslet a programovat objektově*. Ta sice vyšla v jiném nakladatelství, ale protože dané nakladatelství nedokázalo před několikeré upomínky dostát svým závazkům, tak jsem z tvorby druhého dílu vycouval.

se vám demonstrovat některé vlastnosti dědění na třídách, které dědění emulují využitím návrhového vzoru *Dekorátor* a předvedu vám, jak definovat společného rodiče skupině tříd, které mají některé společné vlastnosti.

Ve čtvrté, závěrečné **části**, vám představím algoritmické konstrukce, jejichž používání je sice postupně nahrazováno jinými technikami (ty probírají předchozí části učebnice), ale prozatím se bez nich stále ještě neobejdeme. Navíc se ve starších programech s některými novějšími konstrukcemi ani nesetkáte.

Programovací jazyk

Přestože má kniha ve svém názvu uveden jazyk Java, tak musím dopředu oznámit, že se nejedná o učebnici jazyka Java. Těch je k dispozici více než dost. Jedná se o učebnici objektově orientovaného programování a jazyk Java je zde používán jako jazyk, v němž jsou zapsány programy demonstrující probíranou látku. Tento jazyk jsem zvolil z několika důvodů:

- Je to stále s odstupem nejpoužívanější objektově orientovaný jazyk. Programátoři v Javě jsou stále nejžádanější (a mají i jedny z nejvyšších platů).
- Je to jazyk poskytující všechny klíčové konstrukce používané v moderním programování.
- Vytvořené programy nejsou omezeny na jediný operační systém, ale můžete je přenášet mezi různými operačními systémy.
- A vlastnost neocenitelná pro studium a začátečnické experimenty: nástroje pro vývoj všech druhů aplikací od čipových karet až po rozsáhlé aplikace běžící na několika počítačích můžete sehnat zdarma.

Potřebné vybavení

Pro úspěšné studium této knihy budete potřebovat:

- chuť naučit se objektově programovat a výdrž v situacích, kdy se vám nebude zcela dařit,
- rozumně výkonný počítač,
- základní vývojovou sadu Javy (JDK) ve verzi 8 a vyšší, kterou si můžete stáhnout se stránek http://www.oracle.com/technetwork/java/javase/downloads.
- vývojové prostředí *NetBeans* ve verzi 8 a vyšší, které si můžete stáhnout ze stránek https://netbeans.org.
- doprovodné příklady, které si můžete stáhnout ze stránky knihy http://knihy.pecinovsky.cz/uoa2_j8.

Doprovodné projekty

Na stránce knihy na adrese http://knihy.pecinovsky.cz/uoa2_j8 najdete také soubor s generátorem projektů použitých v knize. Téměř každá kapitola má svůj doprovodný projekt. Některé kapitoly jich mají dokonce několik. Možná bude někomu připadat zbytečné vytvářet takové množství projektů, když jsou jednotlivé sekce umísťovány do samostatných balíčků a řada projektů ve svém obsahu vychází z projektů předchozích, k nimž buď něco přidá a/nebo něco jiného drobně upraví. Důvodem tohoto uspořádání je snaha vyjít začátečníkům vstříc a umožnit jim v každé kapitole začít znovu s funkčním projektem nezávisle na tom, do jakého stavu přivedli projekt z kapitoly přechozí.

První díl byl určen pro naprosté začátečníky, kterým jsem se snažil vyjít vstříc, a protože vím, jaký problém s angličtinou má značná část studentů, používal jsem pro větší názornost české identifikátory. Pořízením této učebnice jste se přihlásili mezi pokročilejší programátory, a ti se bez znalosti angličtiny neobejdou. V této branži platí: *programátor musí umět anglicky, anebo musí změnit zaměstnání*. V tomto dílu jsou proto už všechny identifikátory anglicky. České zůstávají pouze dokumentační komentáře.

Doplňková literatura

Kniha je tlustá, ale ani tak se do ní nevešlo vše, co by bylo potřeba. Navíc se mi nechtělo podrobně rozebírat témata, která už jsem vysvětlil v publikaci, kterou si můžete zdarma stáhnout. Hovořím konkrétně o knize *Java 5.0 – Novinky jazyka a upgrade aplikací*, kterou si můžete zdarma stáhnout na mých webových stránkách na adrese http://knihy.pecinovsky.cz/java5novinky.

Kromě toho se na svých stránkách http://vyuka.pecinovsky.cz chystám postupně zveřejňovat materiály, které budu připravovat pro studenty a případné polotovary budoucích publikací. Když si uvědomím, že při psaní knih vychází honorář asi tak na 7 korun za hodinu práce, tak se domnívám, že nic neztratím, když budu budoucí texty poskytovat zdarma a ponechám na rozhodnutí čtenářů, zda si jich cení tak, že jsou za ně ochotni něco zaplatit.