EDICE
myslíme v ...

knihovna programátora

- Učebnice pro ty, kteří nechtějí zůstat obyčejnými kodéry, ale chtějí se stát špičkovými architekty
- Postupuje podle metodiky Architecture First
- 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 7

učebnice objektové architektury pro začátečníky

GRADA

EDICE
myslíme v ...

O autorovi

Rudolf Pecinovský patří ke špičkovým odborníkům na výuku programování. Publikoval již 43 učebnic, které byly přeloženy do pěti jazyků, a nepřeberné množství článků a příspěvků na odborných konferencích. Je autorem metodiky výuky programování *Karel*, navazující metodiky *Baltík* a moderní metodiky výuky objektově orientovaného programování známé pod anglickým názvem *Architecture First*. Učí programování na VŠE a FIT ČVUT. Současně pracuje jako Senior EDU Expert ve firmě ICZ a.s., kde má na starosti doškolování profesionálních programátorů a organizaci kurzů, které si objednávají jiné firmy.

O knize

Tato kniha je třetím vydáním populární učebnice programování, která je na našem trhu zcela ojedinělá. Na rozdíl od ostatních učebnic, které se soustředí na výuku syntaxe jazyka a práce s knihovnami, se tato kniha soustředí především na výklad architektonických konstrukcí. Neučí čtenáře kódovat, ale snaží se jej naučit, jak programy navrhovat. Učí jej, jak má při programování myslet. Reaguje tak na známou skutečnost, že kodérů je hodně, ale dobrých softwarových architektů je proklatě málo (proto také mají několikanásobně vyšší platy).

Kniha je sice primárně určena začátečníkům, ale ohlasy na předchozí vydání ukázaly, že v ní najdou poučení i zkušení programátoři. Většina učebnic a kurzů programování totiž vyvolává falešnou představu, že objektově programovat znamená používat třídy a dědění. Tato kniha je první, která ukazuje, že objektově orientované programování přináší především jiný způsob myšlení. Jak výstižně napsal jeden čtenář: „*Myslel jsem si, že nejsem žádné programátorské ucho. Když jsem ale přečetl vaši učebnici, otevřel jsem oči a hubu. Konečně jsem pochopil věci, které mi ostatní učebnice nedokázaly vysvětlit.*“

Kniha vznikla na základě dlouholetých autorových zkušeností se školením profesionálních programátorů, výukou programování na univerzitě i vedením žákovských programátorských kroužků. Autor v ní uvádí čtenáře krok za krokem do tajů objektově orientovaného programování a ukazuje mu, jak možnosti této moderní technologie co nejlépe využít a kde si dát naopak pozor na její úskalí.

Výklad je postaven na příkladech, které autor spolu s čtenářem postupně řeší a přitom čtenáře učí nejenom základním programátorským návykům a dovednostem, ale předvede mu i nejrůznější užitečné triky, z nichž mnohé nikde jinde vysvětlené nenajdete. Současně upozorňuje na nejčastější začátečnické chyby, které před svými čtenáři ostatní učebnice většinou tají. Navíc probírá i řadu témat (např. návrhové vzory), která jsou většinou probírána až v pokročilých, nebo dokonce nadstavbových kurzech, přestože patří do základní výbavy objektového programátora.

knihovna programátora

RUDOLF PECINOVSKÝ

Java 7

učebnice objektové architektury pro začátečníky

GRADA
Publishing

Java 7
učebnice objektové architektury pro začátečníky

Rudolf Pecinovský

Vydala Grada Publishing, a.s. U Průhonu 22, Praha 7
jako svou 4971. publikaci

Odborní lektoři:
doc ing. Alena Buchalcevová, Ph.D., doc. ing. Pavel Herout, Ph.D.,
doc. MUDr. Jiří Kofránek, CSc., doc. ing. Vojtěch Merunka,
Ph.D., prof. RNDr. PhDr. Antonín Slabý, CSc., doc ing. Miroslav Virius, CSc.

Odpovědný redaktor: Martin Vondráček
Návrh vnitřního layoutu: Rudolf Pecinovský
Zlom: Rudolf Pecinovský
Počet stran 496
Vydání 1., 2012

Vytiskla Tiskárna PROTISK, s.r.o., České Budějovice

ISBN 978-80-247-3665-5 (tištěná verze)
ISBN 978-80-247-8325-3 (elektronická verze ve formátu PDF)
ISBN 978-80-247-8326-0 (elektronická verze ve formátu EPUB)

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

Stručný obsah

Podrobný obsah

Skrytí spoluautoři

Knihu bych nemohl dokončit v předepsaném čase, kdyby mi s ní nepomohli moji studenti, kteří se podíleli především na přípravě doprovodných programů a dalšího podpůrného programového vybavení a přispěli i řadou připomínek k obsahu a stylu výkladu. Dovolte mi proto abecedně uvést alespoň ty nejzasloužilejší.

Daniel Bartoň se podílel na odladění některých knihovních programů, převodu knihoven do angličtiny pro druhý díl učebnice a úpravách speciální třídy umožňující definici jediné testovací třídy pro celou skupinu tříd se společným rodičem.

Sergej Bobuskyy měl hlavní podíl na vývoji pluginu BJ2NB, který doplňuje možnosti prostředí *NetBeans*, v němž budeme pracovat v druhém dílu, o schopnost průběžného zobrazování diagramu tříd a jeho provázání se zdrojovým kódem, na které si zvyknete u prostředí *BlueJ*, jež se používá v dílu prvním.

Martin Fiala vyvinul pro plugin BJ2NB editor kopenogramů.

David Král pomáhal s přípravou podkladů pro generátor projektů a s úpravami tohoto generátoru v rámci jeho průběžného zdokonalování.

Filip Malý doplnil a upravil sadu maker, která automatizují některé činnosti spojené s přípravou rukopisu a jeho následného převodu do finální podoby. Podílel se i na vývoji knihovny pro automatizované testování vyvinutých programů.

Vladimír Oraný v rámci své diplomové práce vyvinul (a před vydáním knihy na poslední chvíli ještě upravil) speciální knihovnu umožňující výrazně rozšířit možnosti deklarace požadavků, kterým musejí vyhovovat testované programy. Setkáte se s ní v druhém dílu.

Jarmila Pavlíčková a **Luboš Pavlíček** nejsou mí studenti, ale kolegové na VŠE. Mnohé z formulací použitých ve výkladu se vytříbily na základě našich četných (a mnohdy i vášnivých) debat o výuce programování. Jarmile bych chtěl navíc poděkovat za to, že mne osvobozuje od řady administrativních úkonů, které jsou pro mne téměř nezvládnutelnou překážkou.

Martin Vondráček podrobně pročetl celý rukopis a přispěl řadou poznámek k zvýšení jeho čitelnosti, srozumitelnosti a odborné přesnosti.

Na závěr pak musím vyjádřit svůj velký dík firmě ICZ a veškerému osazenstvu oddělení Realizace. Bez jejich podpory by kniha nevznikla.

Předmluva k prvnímu vydání

Rudu Pecinovského jsem poprvé potkal v době, kdy jsme oba studovali na Jaderné fakultě ČVUT v Praze. Doopravdy jsme se ale poznali až mnohem později, když jsme na počátku devadesátých let spolupracovali na překladu manuálů k jistému dodnes populárnímu programovému prostředí. Brzy jsme zjistili, že máme jeden společný zájem – učit lidi, jak kvalitně psát programy.

V současné době dominuje při tvorbě aplikací objektově orientované programování. Moderní vývojové nástroje, které jsou na trhu k dispozici, jeho znalost předpokládají, aplikační knihovny z něj vycházejí, softwarové firmy ho vyžadují, nově vznikající programovací jazyky jsou čistě objektové. A když už jsme u těch jazyků: Java je dnes asi nejpoužívanější jazyk pro vývoj nových aplikací a zcela určitě to je jazyk, který se nejdynamičtěji rozvíjí. Přesto téměř všechny učebnice Javy, které na trhu najdete, začínají procedurálním programováním a k objektově orientovanému programování se dostanou až ke konci. Objekty pak často vypadají jako nepříliš pohodlná nadstavba nad procedurálním programováním.

Řekl jsem, že tak vypadají *téměř* všechny knihy. Kniha Rudy Pecinovského je totiž velmi příjemnou výjimkou. Je to učebnice, která objekty opravdu začíná a prvních několik kapitol se ani ničím jiným nezabývá. Teprve poté, co zvládnete základní pojmy a dovednosti objektově orientovaného programování, se začne zabývat konstrukcemi, jako je cyklus nebo podmínka. Tento postup, který si autor vyzkoušel na začínajících programátorech v programátorských kroužcích a který používá při výuce profesionálů, vás naučí od počátku myslet objektově. Ukazuje objekty jako něco opravdu přirozeného, jako něco, co výrazně usnadňuje přemýšlení o řešené úloze.

Při čtení Rudovy knihy jsem občas litoval, že už umím programovat, a tak jen doufám, že slibované další díly budou stejně dobré.

M. Virius

Úvod

Otevíráte třetí vydání knížky, 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.

Co je nového ve 3. vydání

Oproti předchozímu vydání je kniha od základů přepracovaná. Kniha je nyní rozdělena do dvou dílů. První díl se soustředí především na výklad základních architektonických principů, které by si měl čtenář osvojit předtím, než se pustí do kódování složitějších projektů. Jeho cílem je, aby čtenáři přešly tyto principy do krve dřív, než se začne soustředit na kód.

Druhý díl pak získané návyky prohloubí a seznámí čtenáře s řadou dalších programátorských technik. Vysvětlí hlubší souvislosti, na něž v běžných učebnicích již nezbývá místo, a doplní čtenářovy znalosti syntaxe jazyka. Současně učí práci s profesionálním vývojovým nástrojem a seznámí čtenáře s jeho základními vlastnostmi.

Druhý díl je možno použít jako samostatnou učebnici pro ty, kteří již zvládli základní architektonické principy návrhu objektových aplikací z jiných učebnic.

Komu je kniha určena

Tato knížka je určena těm, kteří to se svojí touhou naučit se moderně programovat myslí vážně a chtějí se naučit programovat dobře. Zaměřuje se na ty, kteří nechtějí zůstat obyčejnými kodéry, jejichž práci postupně přebírají nejrůznější nástroje, ale chtějí se propracovat mezi špičkové architekty, kteří umějí navrhnout optimální řešení splňující požadavky zákazníka (a jsou podle toho také ohodnoceni).

Knížka je sice primárně určena těm, kteří ještě nikdy neprogramovali, ale ukázalo se, že se z ní dozvědí hodně nového i poměrně zkušení programátoři. První vydání jsem psal na základě zkušeností z mnoha kurzů. Při vedení těchto kurzů jsem si uvědomil, že to, co děti (a částečně i dospělí, kteří s programováním teprve začínají) pochopí poměrně snadno, zvládají programátoři s předchozími zkušenostmi z neobjektového programování obtížně. Spoustu úsilí totiž musí věnovat

tomu, aby se nejprve odnaučili mnohé z toho, co si před tím pracně osvojili. Teprve pak začnou pomalu vstřebávat jiný způsob programátorského myšlení, jejž objektově orientované programování vyžaduje[1].

Od čtenářů předchozích vydání jsem dostal řadu mailů, v nichž mi psali, že jsou sice zkušenými programátory, ale teprve zde pochopili některé věci, které jim ostatní učebnice a kurzy nedokázaly vysvětlit. Uvědomil jsem si přitom, že knihu mohou s výhodou použít i ti, kteří již nějakou dobu programují a potřebují se přeškolit z klasického, strukturovaného programování na programování objektově orientované (80 % účastníků mých kurzů).

Tady bych ale chtěl případné zkušenější čtenáře upozornit na jedno úskalí: dospělý člověk již, na rozdíl od dětí, nedokáže přijmout nové informace, aniž by je okamžitě nespojoval se svými dosavadními znalostmi a zkušenostmi. Dozví-li se proto informaci, která neodpovídá zcela jeho dosavadním zkušenostem, tak ji podvědomě „ohne“, aby se s nimi dostala do souladu. To je nejčastějším zdrojem problémů zkušenějších programátorů při přechodu na nové paradigma, protože to, co si zapamatují, bývá v důsledku výše zmíněného „ohnutí“ něco trochu jiného, než co se jim snažil přednášející sdělit.

Co se naučíte

Musím 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.

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 moje kurzy ve firmě či na univerzitě, však ukazují, že tohoto výsledku bývá dosaženo jen zřídka. Většina účastníků mých kurzů zná poměrně dobře konstrukce nějakého objektově orientovaného programovacího jazyka. Bohužel, skoro nikdo z nich v něm neumí objektově programovat[2], neumí přepnout na objektový způsob uvažování.

---

[1] Statistiky uvádějí, že typická doba přechodu na objektové paradigma je 12 až 18 měsíců, přičemž čím je programátor zkušenější, tím déle mu přerod trvá.

[2] 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?

Pokračování na další stránce