

Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačních technologií



Diplomová práce

**Software pro podporu výuky objektově
orientovaného programování**

Pavel Bory

© 2017 ČZU v Praze

Souhrn:

Tato práce se zabývá návrhem a vývojem prototypu aplikace pro podporu výuky objektově orientovaného programování. V této aplikaci bude zobrazena konfigurovatelná mřížka obsahující obrázek reprezentující aktéra simulace, kterého bude moci uživatel programovat. Tento aktér se bude moci v rámci mřížky pohybovat a interagovat s jinými objekty reprezentovanými obrázky rozmístěnými v této mřížce. Klíčovými technologiemi zvolenými pro realizaci aplikace jsou programovací jazyk Java, technologie pro tvorbu grafického uživatelského rozhraní JavaFX, programovací jazyk Ruby a technologie JRuby pro integraci Java a Ruby. V práci jsou rovněž použity technologie log4j pro logování a JUnit pro tvorbu unit testů. Součástí práce je i ukázkový příklad použití implementovaného prototypu aplikace při výuce programování.

Klíčová slova:

Java, Ruby, JRuby, JavaFX, JUnit, programování, výuka programování, objektově orientované programování, návrh software, implementace software.

Cíl a metodika:

Cílem této práce je navrhnout a vytvořit prototyp software pro podporu výuky objektově orientovaného programování. Software bude navržen tak, aby poskytl studentům integrované vývojové prostředí, ve kterém budou moci tvořit programy, kterými budou ovládat předpřipravené 2D grafické objekty v různých prostředích na základě pokynů a doporučení lektora. Grafické objekty bude možné do software vkládat v běžně rozšířených formátech jako např. PNG. Software dále poskytne možnost definovat různá prostředí pro výuku jednotlivých témat z OOP. Z hlediska technologií bude jako programovací jazyk pro výuku objektově orientovaného programování zvolen jazyk Ruby zejména pro svou jednoduchou syntaxi a možnost integrace s jazykem Java, ve kterém bude software tvořen.

Ke splnění vytyčeného cíle budou nejdříve analyzovány vhodné technologie a možnosti jejich integrace. Dále budou stanoveny základní oblasti z objektově orientovaného programování, které jsou běžně vyučovány a budou podporovány navrženým software. V rámci návrhu budou popsány funkční a nefunkční požadavky na navržený software a navržena softwarová architektura. Ve výsledku bude implementován prototyp software a provedeno jeho otestování.

Teoretická část:

V teoretické části práce jsou analyzovány klíčové technologie pro implementaci aplikace a na základě teoretických východisek je vybrána vhodná strategie integrace Ruby a Java pro tuto práci.

Praktická část:

V úvodu praktické části jsou definovány funkční a nefunkční požadavky na prototyp aplikace kladené. Dále je v praktické části navrženo uživatelské rozhraní, navržena architektura aplikace a jsou navrženy jednotlivé klíčové třídy aplikace. Poté je implementován, zdokumentován a otestován i s použitím unit testů prototyp aplikace. Součástí praktické části práce je i ukázkový příklad použití implementovaného prototypu aplikace při výuce programování.

Závěr:

Tato práce se zabývala návrhem a implementací prototypu software pro podporu výuky objektově orientovaného programování. Software měl být navržen tak, aby umožnil tvořit programy, kterými budou ovládány grafické objekty v předpřipravených prostředích.

V teoretické části práce byly analyzovány pro tuto práci vhodné technologie a možnosti jejich integrace. Analýzována byla technologie JavaFX pro tvorbu uživatelského rozhraní aplikace a programovací jazyky Java a Ruby, přičemž jazyk Ruby byl zvolen jako programovací jazyk, kterým bude moci uživatel programovat chování aktéra v předpřipravených prostředích konfigurovatelných pomocí XML souborů, ve kterých bude moci interagovat s grafickými objekty. Pro integraci Java a Ruby byla zvolena technologie JRuby, která umožní jejich vzájemné propojení, přičemž v teoretické části byly analyzovány možné strategie této integrace. Aplikace bude pro usnadnění ladění a dalšího rozvoje provádět logování pomocí frameworku log4j.

Dále byly definovány funkční a nefunkční požadavky, navržena architektura aplikace, navrženo základní grafické rozhraní, definovány případy užití aplikace a byly navrženy klíčové třídy pro tuto aplikaci. V rámci návrhu tříd byl i na základě teoretických

východisek navržen i způsob integrace Java a Ruby pomocí JRuby a byly popsány klíčové prvky pro tuto integraci.

Poté byl implementován prototyp aplikace, jehož klíčové prvky jsou řádně zdokumentovány a popsány v této práci. Zároveň byly připraveny a zdokumentovány dva příklady použití této aplikace. Aplikace byla na závěr uživatelsky otestována a její klíčové části zdrojového kódu jsou pokryty pomocí unit testů implementovaných za použití frameworku JUnit.

Použitá literatura:

1. **Dave Thomas with Chad Fowler, Andy Hunt.** *Programming Ruby 1.9 & 2.0 The Pragmatic Programmers' Guide.* United States of America : The Pragmatic Programmers, LLC., 2013. 978-1-93778-549-9.
2. **Schildt, Herbert.** *Java 8 Výukový kurz.* Brno : Computer Press, 2016. 978-80-251-4665-1.
3. **Sharan, Kishori.** *Learn Java FX 8 Building User Experience and Interfaces with Java 8.* United States of America : Apress Media, LLC, 2015. 978-1-4842-1143-4.
4. **Charles O Nutter, Thomas Enebo, Nick Sieger, Ola Bini, and Ian Dees.** *Using JRuby Bringing Ruby to Java.* United States of America : The Pragmatic Programmers LLC, 2011. 1-934356-65-4.
5. **Jim Arlow, Ila Neustadt.** *UML 2 a unifikovaný proces vývoje aplikací.* Brno : Computer Press, a.s., 2007. 978-80-251-1503-9.
6. **McConnell, Steve.** *Dokonalý kód.* Brno : Computer Press, a.s., 2006. 80-251-0849-X.
7. **Ruby-Doc.org.** *Dokumentace Ruby.* [Online] 2017. <http://ruby-doc.org/>.
8. **Oracle and/or its affiliates.** *JavaFX 2.2. Oficiální dokumentace.* [Online] 2014. <https://docs.oracle.com/javafx/2/api/overview-summary.html>.
9. **ScriptingContainer.** *Dokumentace JRuby.* [Online] 2015. <http://jruby.org/apidocs/org/jruby/embed/ScriptingContainer.html>.