

Posudek vedoucího diplomové práce

Jméno a příjmení autora posudku: Pavel Parížek

Jméno a příjmení autora práce: Jonáš Klimeš

Název práce: Domain-Specific Language for Learning Programming

Vlastní text:

Cílem práce bylo navrhnout a implementovat systém na výuku programování, který je založen na doménově-specifickém jazyce a vlastním běhové prostředí. Hlavní motivace byla vyhnout se obecným jazykům, které jsou složité pro začátečníky. Jazyk a běhové prostředí měly být implementovány na základě platformy JetBrains MPS.

Autor provedl důkladnou analýzu existujících systémů na výuku programování, jako například Scratch, a na základě této analýzy identifikoval požadavky na nový jazyk, který pojmenoval Eddie. Jazyk Eddie částečně vychází z myšlenek staršího systému Karel, ve kterém je úkolem programu ovládat robota, ale významně je rozvíjí. Implementace jazyka byla vytvořena pomocí mechanismů, které nabízí platforma MPS – konkrétně se jedná zejména o definici syntaxe, editor, a generátor spustitelné reprezentace programu. Součástí běhového prostředí, nazvaného Eddie Studio, je kromě editoru také panel pro vizualizaci běhu programu. Dalším výstupem práce je několik tutoriálů pro základní koncepty jazyka (proměnné, cykly, funkce, a objekty), a také modul pro přípravu vlastních scénářů. Velmi kladně hodnotím úroveň zpracování implementace, a to především strukturu a čitelnost zdrojových kódů, množství testů, a modulární návrh jazyka, který je tak snadno rozšiřitelný.

Text je napsán v anglickém jazyce. Organizace textu je na vysoké úrovni, a po obsahové stránce také nemohu nic zásadního vytknout. Autor podrobně zdůvodňuje všechny zásadní rozhodnutí, které musel učinit během návrhu a implementace jazyka Eddie a běhového prostředí Eddie Studio. Jedinou výhradu mám k množství pravopisných chyb v textu.

Práci celkově hodnotím jako velmi zdařilou. Součástí textu je také dlouhý seznam možných rozšíření, a to včetně návrhu způsobu jejich realizace.

Musím ještě dodat, že autor práce docházel na pravidelné konzultace do firmy JetBrains. Hlavním konzultantem byl Václav Pech, který je členem týmu MPS, a který téma diplomové práce inicioval. Jeho vyjádření přikládám.

Diplomová práce se zaměřila na oblast výuky programování s cílem vytvořit funkční vývojové prostředí pro úplné programátorské začátečníky.

Jedním ze záměrů bylo identifikovat klíčové vlastnosti a schopnosti existujících jazyků a nástrojů a vzájemně je porovnat. Tomu se zdařile věnuje úvodní část práce. Na této analýze je poté vystaven návrh ideálního hypotetického prostředí pro výuku programování, jenž posloužil jako vzor pro vlastní implementovaný nástroj. Tento cíl práce byl tedy dosažen.

Popis MPS, a zvláště pak jeho projekčního editoru, včetně posouzení jeho aplikovatelnosti pro danou problémovou oblast, považují za zdařilý. Práce vyzdvihuje možnosti intuitivní nápovědy, restriktivní syntaxe, barevného odlišení jednotlivých bloků kódu, statické analýzy či interpretace programu. Mnoho z těchto myšlenek je poté uplatněno v implementovaném nástroji.

Implementovaný nástroj v několika ohledech předčil má očekávání. Konstrukce jazyků byly voleny pečlivě jak s ohledem na intuitivní chápání začátečníků, tak s přihlédnutím k trendům v jazycích hlavního proudu. Oceňuji zvláště zaměření na intuitivní nezákladnou syntaxi, která zpřístupní jazyk začínajícím programátorům, preference konstantních hodnot před proměnnými, jež je v souladu s vývojem v mnoha moderních jazycích, a důslednou modularizaci výukových jazyků, díky níž si mohou studenti osvojovat principy programování postupně. Oddělení doménového jazyka (instrukce pro robota) od jadrových jazyků (proměnné, funkce, atd.) umožní snadno změnit problémovou doménu a přizpůsobit tak nástroj zájmům různých skupin studentů.

Sympatický je i fakt, že se práce v několika místech zabývá myšlenkou postupného zvykání studentů na konstrukce obvyklé v profesionálních programovacích jazycích a řeší jejich zařazení do implementovaného nástroje.

Autor věnoval rovněž velkou pozornost vizuální podobě jazyka a využil tak mnoho vlastností nabízených projekčním editorem - barvou a stylem odlišené různé syntaktické elementy, barvy pozadí znázorňující bloky různých typů, adaptivní označení hranic bloků kódu s vysvětlujícím komentářem u konce bloku, položky v menu při doplňování kódu s barevným označením a detailním popisem apod.

Za velmi zásadní prvek implementovaného nástroje považují mechanismus tutoriálů, které studentovi předloží úkol a poté automaticky vyhodnotí jeho splnění. Sympatická je i pozornost věnovaná možnosti lokalizace nástroje. Implementovaná pravidla pro statickou analýzu kódu pomůžou studentům snížit počet chyb, které by je jinak mohly snadno odradit od dalšího experimentování.

Nastíněné možnosti pro budoucí rozvoj nástroje naznačují, že tato oblast skrývá potenciál pro další vylepšení. Zvláště neúplnou implementaci editování výrazů (expressions) a závorkování považují za citelné omezení. Chybějící podpora pro kolektivní datové typy a dědičnost vlastností tříd komplikuje využití nástroje pro výuku pokročilejších konceptů. Postrádám i schopnost interpretace kódu, díky které by bylo možno průběžně vizualizovat akce tvořeného programu a poskytovat tak studentům zpětnou vazbu. Upozadění těchto vlastností z časových důvodů je ovšem pochopitelné.

Kromě opakujících se drobných jazykových problémů nemám k textu práce žádné námítky. Text je srozumitelný, dobře strukturovaný a logicky řazený. Vložené obrázky jsou dobře voleny a vhodně doplňují text.

Autor musel během své práce zvládnout netriviální principy návrhu programovacích jazyků a IDE funkcionality v MPS a v určitých ohledech překročit limity běžného použití MPS (barevná pozadí, interpreter, vlastní nástrojový panel s grafickým výstupem atd.). Autor prokázal schopnost překonat omezení nástroje při implementaci potřebných řešení (např. statické typy). Vytvořil tak dobrý základ, na kterém je možné stavět další rozšíření a vylepšení, z nichž mnohá jsou v práci konkrétně navrhována.

Práce splnila stanovené cíle a úroveň jejího zpracování hodnotím velmi pozitivně. Student pracoval samostatně, systematicky a dobře komunikoval. Prokázal tak svoje schopnosti v oboru návrhu jazyků, implementace IDE podpory a softwarového inženýrství.

Doporučení k obhajobě:

Z výše uvedených důvodů práci doporučuji k obhajobě.

Vynikající práce vhodná pro soutěž studentských prací	ANO <input type="checkbox"/>
---	------------------------------

Seznam soutěží studentských prací, viz <http://www.mff.cuni.cz/studium/bcmgr/prace/>

Pokud jste výše zaškrtnli ANO, zdůvodněte prosím svůj návrh, případně uveďte konkrétní soutěž, pro kterou je práce vhodná (rámeček lze nechat prázdný, pokud za dostatečné zdůvodnění považujete text posudku):

--

V Praze dne: 2.6.2016

Podpis:

** nehodící se škrtněte (vymažte)*

*** do SISu vkládejte formulář nepodepsaný (ve formátu PDF), podpis je potřeba doplnit až na vytištěný posudek.*