|  |
| --- |
| OSTRAVSKÁ UNIVERZITA  PŘÍRODOVĚDECKÁ FAKULTA  KATEDRA INFORMATIKY A POČÍTAČŮ |
| Plugin pro IDEA pro generování diagramu tříd do PlantUML  BAKALÁŘSKÁ PRÁCE |
| Autor práce: Marie Šrámková  Vedoucí práce: RNDr. Marek Vajgl, Ph.D. |
| 2022 |

|  |
| --- |
| UNIVERSITY OF OSTRAVA  FACULTY OF SCIENCE  DEPARTMENT OF INFORMATICS AND COMPUTERS |
| A plugin for IDEA to generate class diagram into PlantUML  BACHELOR THESIS |
| Author:  Marie Šrámková  Supervisor:  RNDr. Marek Vajgl, Ph.D. |
| 2022 |

(Zadání vysokoškolské kvalifikační práce)

Cílem práce je realizovat konfigurovatelný plugin pro vývojové prostředí IDEA, který bude umět převádět třídy definované v projektu do diagramu tříd jazyka PlantUML.

Řešení bude moci u otevřeného projektu vybrat oblast, která se má zahrnout do generování, a další parametry generování (vybraní členové, detailnost atd). Následně bude umět nástroj vygenerovat diagram do jazyka PlantUML. Konfigurace generování bude možno uložit.

Přínosem práce je vytvoření vlastního generovacího přístupu ze zdrojových kódu Java s možností poměrně podrobné specifikace generovaného obsahu.

Postup práce:

1) Definice cílů a analýza současného stavu

2) Definice metodiky řešení práce

3) Vytvoření vlastního řešení

4) Shrnutí vytvořeného řešení, přínosy

5) Shrnutí cílů práce, závěr

ABSTRAKT

Cílem této bakalářské práce je vytvořit konfigurovatelný plugin v prostředí IntelliJ IDEA, který bude do složky projektu generovat class diagram podle zvolených parametrů a který do složky projektu také uloží konfiguraci pro daný projekt pro příští generování. Plugin po kliknutí na kořenovou složku projektu zobrazí vstupní formulář, který umožní uživateli zvolit v otevřeném projektu atributy, třídy, metody a další parametry pro generování PlantUML class diagramu. Diagram se po vygenerování uloží a zobrazí. Zvolené hodnoty se uloží pro příští generování a při dalším generování se zobrazí možnost využití poslední zvolené konfigurace. Plugin také umožní uživateli spravovat již vzniklé konfigurace (smazání, změna názvu konfigurace, apod.).

*Klíčová slova:*

*(klíčová slova vypsaná na řádku, oddělená od sebe čárkami)*

**ABSTRACT**

The text of the abstract.

*Keywords:*

čestné prohlášení

Já, níže podepsaný/á student/ka, tímto čestně prohlašuji, že text mnou odevzdané závěrečné práce v písemné podobě je totožný s textem závěrečné práce vloženým v databázi DIPL2.

Ostrava dne

………………………………

podpis studenta/ky

|  |
| --- |
| Poděkování |
| Prohlašuji, že předložená práce je mým původním autorským dílem, které jsem vypracoval/a samostatně. Veškerou literaturu a další zdroje, z nichž jsem při zpracování čerpal/a, v práci řádně cituji a jsou uvedeny v seznamu použité literatury.  V Ostravě dne . . . . . . . . . . . .  . . . . . . . . . . . . . . . . . .  (podpis) |

OBSAH

[ÚVOD 3](#_Toc440839236)

[1 NADPIS 3](#_Toc440839237)

[1.1 Podnadpis 3](#_Toc440839238)

[1.2 Podnadpis 3](#_Toc440839239)

[1.2.1 Podpodnadpis 3](#_Toc440839240)

[2 NADPIS 3](#_Toc440839241)

[2.1 Podnadpis 3](#_Toc440839242)

[2.1.1 Podpodnadpis 3](#_Toc440839243)

[3 NADPIS 3](#_Toc440839244)

[3.1 Podnadpis 3](#_Toc440839245)

[3.1.1 Podpodnadpis 3](#_Toc440839246)

[4 NADPIS 3](#_Toc440839247)

[4.1 Podnadpis 3](#_Toc440839248)

[4.1.1 Podpodnadpis 3](#_Toc440839249)

[ZÁVĚR 3](#_Toc440839250)

[RESUMÉ 3](#_Toc440839251)

[SUMMARY 3](#_Toc440839252)

[SEZNAM POUŽITÉ LITERATURY 3](#_Toc440839253)

[SEZNAM POUŽITÝCH SYMBOLŮ 3](#_Toc440839254)

[SEZNAM OBRÁZKŮ 3](#_Toc440839255)

[SEZNAM TABULEK 3](#_Toc440839256)

[SEZNAM PŘÍLOH 3](#_Toc440839257)

ÚVOD

Text

1. NADPIS

Text (1)

* 1. Podnadpis

Text (2)

* 1. Podnadpis

Text

* + 1. Podpodnadpis

Text

1. Rešerše

Technologie se již několik let šíří tak, že zasahují téměř do každého odvětví. S rostoucím počtem použití roste také jejich komplexita. Abychom se my či vývojáři těchto technologií v tak komplexních technologiích vyznali, používají se různé způsoby, jak popsat problematiku (textové znázornění, grafické znázornění a další způsoby). Vhodným a jednoduchým způsobem vyjádření problematiky je pomocí grafického znázornění – diagramů neboli grafů. Diagramy jsou kombinací textu a obrázků, což nám pomáhá se v popisu určité problematiky lépe vyznat.

S problémem orientace v komplexních projektech se setkává většina vývojářů a programátorů. Popis projektu může napomoci jak programátorům vyznat se ve svém vlastním projektu a předejít tak možným chybám, tak i velkým firmám, uvnitř kterých kolaboruje více lidí na jednom projektu. V tomto případě může vhodný diagram ulehčit workflow mezi jednotlivými zaměstnanci, odděleními apod., a tak zlepšit chod firmy a její prosperitu. Michal Obluk se touto problematikou okrajově zabývá v práci [1].

Pro svou práci jsem zvolila vytvoření generátoru diagramu tříd PlantUML pro programovací jazyk Java v prostředí IntelliJ IDEA. Programovací jazyk Java je objektově orientovaný (více o OOP např. v práci [2]), tzn. že všechny objekty reálného světa můžeme reprezentovat jako objekty (třídy) a jednotlivé výskyty objektů reálného světa jako jejich instance. K popisu projektu můžeme využít více typů diagramů (diagram aktivit, diagram tříd, které jsou popsány v práci [1], sekvenční diagram a další typy diagramů). Pro svou práci jsem zvolila diagram tříd [3], který popisuje základní stavební prvky objektů programovacího jazyka Java.

V rámci průzkumu aktuálního stavu v problematice generování diagramu tříd do plantUML jsem se zaměřila na dva základní okruhy – analýzu již existujícího komplexního řešení a analýzu obecných metod, přístupů a nástrojů.

Z první části (analýzy existujících řešení) jsem zjistila, že již existuje nástroj pro tvorbu diagramu PlantUML ve vývojovém prostředí IntelliJ IDEA. Plugin PlantUML integration [4] umožňuje převést textový soubor puml do diagramu PlantUML a zobrazit jej v grafické podobě. Diagram Jelikož je tato funkcionalita potřebná i v mé aplikaci, rozhodla jsem se ji využít.

V roce 2020 byl navržen a vydán plugin PlantUML Parser [5], jenž umožňuje generovat diagram PlantUML za pomoci výše zmíněného pluginu. Nyní už však ne na základě manuálně napsaného puml souboru, ale už přímo ve vývojovém prostředí. Plugin lze najít ve vývojovém prostředí při kliknutí na projekt. Zde najdeme možnost PluntUML Parser, jež nám otevře dialogové okno, ve kterém konkretizujeme podmínky generování. Tento plugin však nenabízí potřebné funkcionality a je nedostačující. Chybějícími funkcionalitami (chybějící funkcionality) z výše zmíněného pluginu jsou a ty tvoří také požadavky:

* volba umístění vygenerovaného diagramu a konfiguračního souboru a jejich následné uložení
* volba názvu vygenerovaného diagramu a konfiguračního souboru
* uložení konfigurace generování (parametry generování) pro jednotlivé složky/soubory/moduly/projekty
* načtení uložených konfigurací
* možnost výběru generování tříd/rozhraní
* možnost výběru generování vnitřních tříd
* definice, v jakém jazyce bude výsledný soubor vygenerován (formát „Java“, nebo formát diagramu tříd UML)
* možnost generování všech vybraných projektů najednou
* možnost generování všech projektů pro jednotlivé moduly
* možnost definovat komentář
* možnost definovat orientaci zobrazení tříd (levá/pravá)

Teoretická část

Text

* 1. Požadavky

Text

* + 1. Podpodnadpis

Text

1. Praktická část

Text

* 1. Podnadpis

Text

* + 1. Podpodnadpis

Text

ZÁVĚR

RESUMÉ

SUMMARY

SEZNAM POUŽITÉ LITERATURY

1. **Příjmení, Jméno.** *Název knihy.* Město vydání: Vydavatelství, 2003. 123-4-56-789123-4.

2. **Příjmení1, Jméno1 a Příjmení2, Jméno2.** Název webové stránky. *Název webu.* [Online] Produkční společnost, 23. Září 2006. [Citace: 19. Září 2008.] http://www.urladresa.cz. 12-3456-789-12.

[1] OBLUK, Michal. *Modelování Business procesů s pomocí UML: BUSINESS PROCESS MODELING WITH UML*. Ostrava, 2006. Bakalářská. Ostravská univerzita. Vedoucí práce Lukasík Petr.

[2] Stuart Hirshfield and Raimund K. Ege. 1996. Object-oriented programming. ACM Comput. Surv. 28, 1 (March 1996), 253–255. DOI:https://doi.org/10.1145/234313.234415

[3] Berardi, Daniela & Calvanese, Diego & De Giacomo, Giuseppe. (2005). Reasoning on UML class diagrams. Artificial Intelligence. 168. 70-118. 10.1016/j.artint.2005.05.003.

[4] STEINBERG, Eugene, MAMONTOV, Ivan, Henady ZAKALUSKY, Max GORBUNOV, Vojtěch KRÁSA a Andrew KOROLEV, ed. PlantUML integration [online]. In: . [cit. 2021-11-16]. Dostupné z: https://plugins.jetbrains.com/plugin/7017-plantuml-integration/

[5] SHUZIJUN. *PlantUML Parser* [online]. In: . [cit. 2021-11-16]. Dostupné z: https://plugins.jetbrains.com/plugin/15524-plantuml-parser

[6] A GroupLayout Example. *Oracle Java Documentation: The Java™ Tutorials* [online]. [cit. 2022-01-03]. Dostupné z: <https://docs.oracle.com/javase/tutorial/uiswing/layout/groupExample.html>

[7] How to Use GroupLayout. *Oracle Java Documentation: The Java™ Tutorials* [online]. [cit. 2022-01-03]. Dostupné z: https://docs.oracle.com/javase/tutorial/uiswing/layout/group.html

SEZNAM POUŽITÝCH SYMBOLŮ

|  |  |  |
| --- | --- | --- |
| ABC |  | Význam první zkratky. |
| B |  | Význam druhé zkratky. |
| C |  | Význam třetí zkratky. |
|  |  |  |

SEZNAM OBRÁZKŮ

SEZNAM TABULEK

SEZNAM PŘÍLOH