Съдържание

Съдържание……………………………………………………………………………………………………………………………………….I

Списък на таблиците и фигурите…………………………………………………………………………………………………….III

Списък на речниковите съответствия……………………………………………………………………………………………..VI

Увод………………………………………………………………………………………………………………………..………………………...1

Мотивация…..………………………………………………………………………………………………………………………..………….2

Цел…...………………………………………………………………………………………………………………………..……………………..3

Обзор и сравнителен анализ на съществуващи решения…..………………………………………………………….4

Eclipse Orion…..……………………………………………………………………………………………………………………..5

Eclipse Flux…….………………………………………………………………………………………………………………………7

Eclipse Che…..………………………………………………………………………………………………………………………..8

Eclipse Dirigible……………………………………………………………………………………………………………………10

Cloud9…...……………………………………………………………………………………………………………………………12

Архитектура на среда за разработка в облака…...…………………………………………………………….14

All-In-One………………………………………………………………………………………………………………..16

Production..…………………………………………………………………………………………………………….17

RCP………………………………………………………………………………………………………………………….18

Multi-tenant……………………………………………………………………………………………………………19

Извод………………………………………………………………………………………………………………………..………..20

Проектиране на модул за In-memory Dynamic Java Compilation…………………………………………………..21

Реализиране на модул за In-memory Dynamic Java Compilation…………………………………………………..22

Проектиране на модул за Git Integration……………………………………………………………………………………….28

Реализиране на модул за Git Integration……………………………………………………………………………………….29

Проектиране на модул за Debugger……….……………………………………………………………………………………..37

Реализиране на модул за Debugger……………………………………………………………………………………………….38

Създаване на облачно приложение с избрания инструмент…………………………..…………………………..46

База от данни……………………………………………………………………………………………………………………..47

Архитектура………………………………………………………………………………………………………………………..48

Използвани технологии……………………………………………………………………………………………………..51

Основни функции……………………………………………………………………………………………………………….53

Заключение……………………………………………………………………………………………………………………………………..55

Използвана литература…………………………………………………………………………………………………………………..57

Приложение…………………………………………………………………………………………………………..……………………….61

Списък на таблиците и фигурите

Таблица 1: Критерийно оценяване на Orion, Flux, Che, Dirigible и Cloud9..………………………………...№

Фиг. 1 Работен плот на проекти в Eclipse Orion……………………………………………………………………………..№

Фиг. 2 Редактор изходен код на Eclipse Orion……………………………………………………………………………….№

Фиг. 3 Синхронизиране на проект от интегрираната среда за разработване Eclipse в Eclipse Flux…………………………………………………………………………………………………………………………………………………..№

Фиг. 4 Начална страница на Eclipse Che…………………………………………………………………………………………№

Фиг. 5 Работен плот на проект в Eclipse Che………………………………………………………………………………….№

Фиг. 6 Начален екран на Eclipse Dirigible……………………………………………………………………………………….№

Фиг. 7 Работен плот на Eclipse Dirigible………………………………………………………………………………………….№

Фиг. 8 Tабло за управление на проекти в Cloud9………………………………………………………………………….№

Фиг. 9 Работна среда за проект в Cloud9……………………………………………………………………………………….№

Фиг. 10 Основни компоненти на архитектурата на Eclipse Dirigible…………………………………………....№

Фиг. 11 Eclipse Dirigible - опция за разполагане All-In-One...............................................................№

Фиг. 12 Eclipse Dirigible - опция за разполагане Production..............................................................№

Фиг. 13 Eclipse Dirigible - опция за разполагане RCP.........................................................................№

Фиг. 14 Eclipse Dirigible - опция за разполагане Multi-tenant...........................................................№

Фиг. 15 Място на модулa за In-memory Dynamic Java Compilation в архитектурата на Eclipse Dirigible……………………………………………………………………………………………………………………………………………№

Фиг. 16 ООП модел на модул за In-memory Dynamic Java Compilation в Eclipse Dirigible…………...№

Фиг. 17 Структура на класа ClassFileManager.....................................................................................№

Фиг. 18 Структура на класа JavaClassObject.......................................................................................№

Фиг. 19 Структура на класа JavaExecutor...........................................................................................№

Фиг. 20 Структура на класа JavaServlet..............................................................................................№

Фиг. 21 Място на модулa за Git Integration в архитектурата на Eclipse Dirigible………………………...№

Фиг. 22 ООП модел на модул за Git Integration в Eclipse Dirigible част 1……………………………………..№

Фиг. 23 ООП модел на модул за Git Integration в Eclipse Dirigible част 2……………………………………..№

Фиг. 24 ООП модел на модул за Git Integration в Eclipse Dirigible част 3……………………………………..№

Фиг. 25 Структура на класа JGitConnector.................................................……………………………………..№

Фиг. 26 Структура на класа BaseCommandDialog..............................................................................№

Фиг. 27 CloneCommandDialog - графичен интерфейс……………………………………………………………………№

Фиг. 28 Структура на класа CloneCommandHandler..........................................................................№

Фиг.29 Събитие от потребителския интерфейс, което задейства CloneCommandHandler............№

Фиг. 30 Място на модулa за Debugger в архитектурата на Eclipse Dirigible………………………………….№

Фиг. 31 ООП модел на модул за Debugger (Runtime) в Eclipse Dirigible част 1…………………………….№

Фиг. 32 ООП модел на модул за Debugger (Runtime) в Eclipse Dirigible част 2…………………………….№

Фиг. 33 ООП модел на модул за Debugger (Runtime) в Eclipse Dirigible част 3…………………………….№

Фиг. 34 ООП модел на модул за Debugger (Repository) в Eclipse Dirigible…………………………………...№

Фиг. 35 ООП модел на модул за Debugger (IDE) в Eclipse Dirigible……………………………………………….№

Фиг. 36 Структура на класа JavaScriptDebugFrame............................................................................№

Фиг. 37 Структура на класа BreakpointMetadata...............................................................................№

Фиг. 38 Структура на класа DebugView..............................................................................................№

Фиг. 39 DebugView - графичен интерфейс…………………………………………………………………………………....№

Фиг. 40 Диаграма на базата от данни…………………………………………………………………………………………...№

Фиг. 41 Архитектура на облачно приложение в Eclipse Dirigible част 1……………………………………….№

Фиг. 42 Архитектура на облачно приложение в Eclipse Dirigible част 2……………………………………….№

Фиг. 43 Облачно приложение за търг - начален екран………………………………………………………………..№

Фиг. 44 Облачно приложение за търг - екран за купуване/наддаване……………………………………….№

Фиг. 45 Облачно приложение за търг - потребителски интерфейс за администратори…………….№

Фиг. 46 Облачно приложение за търг - структура на проекта в Eclipse Dirigible………………………….№

Фиг. 47 Зареждане на библиотеката на Bootstrap в html файл……………………………………………………№

Фиг. 48 Съдържание на файла index.html…………………………..………………………………………………………..№

Фиг. 49 Съдържание на файла app.js - AngularJS контролер………………………………………………………..№

Фиг. 50 Част от изходния код на файла /ScriptingServices/user/hard.js………………………………………..№

Фиг. 51 Част от изходния код на файла /ScriptingServices/user/bid.js………………………………………….№

Фиг. 52 Съдържание на файла main.access…………………………………………………………………………………..№

Списък на речниковите съответствия

**[1] Интегрирана среда за разработване като услуга (Integrated Development Environment as as Service IDEaaS)** - част от номенклатурата на облачните услуги. Услугата предлага уеб интегрирана среда за разработка.

**[2] Интегрирана среда за разработка (Integrated Development Environment - IDE)** - софтуерно приложение, което предлага инструменти за улесняване и автоматизиране на разработването на софтуер (Ref. 19).

**[3] Интелигентно довършване на код (Intelligent Code Completion)** - набор от инструменти, които служат за автоматично и интелигентно предлагане на предложения за дописване на недовършен код (Ref. 35).

**[4] Локален софтуер (On-Premises Software)** - софтуер, който е инсталиран върху локалните компютри на организацията (Ref. 27).

**[5] Микро услуги (Microservices)** - софтуерен архитектурен стил, чрез който сложни приложения се разделят на малко независими модули, които си комуникират помежду си без специфики на езика, на които са написани (Ref. 24).

**[6] Облачни услуги (Cloud Computing)** - предоставяне на изчислителни услуги, а не на готов продукт. Най-често се определят като използване на споделени ресурси, софтуер и информация през интернет (Ref. 17).

**[7] Помощник (Wizard)** - целта му е да улесни и автоматизира задачите, които подлежат на автоматизация. Пример за това е генерирането на Java клас и други (Ref. 25).

**[8] Приложно-програмен интерфейс (Application Programming Interface - API)** - интерфейс който предлага набор от протоколи, шаблони и инструменти за създаване на софтуерни приложения. (Ref. 33).

**[9] Разработване върху система (In-System Development)** - техника, която наподобява прогамирането върху работеща платка In-System Programming (Ref. 26), приложена в разработването на софтуер.

**[10] Система за контрол на кода (Source Code Control System - SCCS)** - система за контрол и ревизия на изходен код, първоначално разработена от IBM (Ref. 20).

**[11] Софтуер като услуга (Software as a Service - SaaS)** - терминът е част от номенклатурата за видовете облачни услуги. Използва се за абонаментно предоставяне на софтуер за потребителите (Ref. 18).

**[12] Софтуер с отворен код (Open Source Software)** - софтуерен продукт, кодът на който може да бъде видян, редактиран и преизползван в зависимост от лиценза, под който се разпространява (Ref. 22).

**[13]** **Уеб интегрирана среда за разработка (Web Integrated Development Environment - WebIDE, WIDE или cloud IDE)** - уеб базирано софтуерно приложение, което предоставя инструменти за улесняване и автоматизиране на процесa по разработване на софтуер (Ref. 21).

**[14] Ускорено разработване на приложение (Rapid Application Development - RAD)** - термин използван за описване на подходи за разработване на софтуер, които се различават от стандартния модел - “водопад”. Целта на тези техники е софтуерните разработчици да се съсредоточат повече върху разработването, а не върху планирането на задачите (Ref. 23).

**[15] Kаквото виждате, това получавате (What You See Is What You Get - WYSIWYG)** - графичен редактор (drag and drop) за потребителски интерфейс.