Звіт

По проекту системи для проведення олімпіад з програмування Sandra

студента Скляра Павла

Зміст

* Опис документу.......................................................................................3
* Usecase Diagram.......................................................................................3
* Activity Diagrams.......................................................................................3
* Авторизація користувача
* Додавання задач
* Додавання тестів
* Перевірка рішень
* Регістрація користувача
* Формування результатів
* Class Diagrams..........................................................................................7
* Головна діаграма
* Пакет compiling
* Пакет untils
* Пакет structures
* Пакет testing
* Пакет entities
* Пакет java
* Statechart Diagrams..................................................................................10
* Клас OnlineTesterThread
* Клас Tester
* Клас OneTest
* Sequence і Collaboration Diagrams..........................................................10
* Допомога
* Обробка помилок
* Тестування
* Component Diagrams...............................................................................14
* Головна
* Інтерфейси
* Тестуючий комплекс
* Модуль адміністратора
* Модуль користувача
* Пакет обробки помилок
* Пакет бази даних
* Deployment Diagram................................................................................16
* Демонстрація графічного інтерфейсу програми...................................17
* Категорії задач
* Додавання задач
* Тестування
* Користувачі системи

Процес проектування системи тестування Sandra2010

Даний документ описує етапи проектування системи для проведення учнівських олімпіад з програмування ***Sandra 2010.*** Описано поступовий процес проектування системи, що допомагає розробникам краще спланувати свої дії при розробці проекту.

При проектуванні будьякої системи перш за все необхідно виділити фундаментальні сутності системи та можливості, що вони матимуть, чи надаватимуть іншим частинам системи. Для цього у ***Rational Unified Process*** передбачено діаграму діяльностей, з якої починається безпосереднє проектування системи.

***UseCase Diagram***



На цій діаграмі зображено основні можливості та процеси, що проходять у системі тестування. Пізніше на інших діаграмах ці процеси будуть поступово конкретизуватися та уточнюватися до досягнення рівня реалізації та фізичного розташування системи.

Зокрема на діаграмах ***Activity Diagram*** уточнюються етапи роботи системи для виконання задач авторизації, додавання нових задач, додавання тестів, перевірка рішень, регістрація користувача та формування результатів. Ці діаграми надають змогу розробнику системи більш точно зрозуміти, що собою являє той чи інший процес у системі. Така діаграма необхідна для кажного процесу у системі для уникнення непорозумінь у майбутньому при безпосередній розробці проекту системи.

***Activity Diagram / Авторизація користувача***

***Activity Diagram / Додавання задач***



***Use Case View / Додавання тестів*** 

***Activity Diagram View / Перевірка рішень***



***Activity Diagram / Регістрація користувача***



***Activity Diagram / Формування результатів***



Проект тестуючої системи ***Sandra2010*** є повінстю об’єктно-орієнтованим, тому за процесами у системі стоять пакети класів. Кожна сутність системи є окремим класом, що взаємодіє з іншими. Для надання необхідного функціоналу необхідні пакети тестування, а для надання зручного графічного інтерфейсу необхідні пакети графічних класів. Звязок між даними та представленням організовує програміст класів системи. Звязки між пакетами та класами відображено на діаграмах класів, що є вкладненими.

***Class Diagram / Головна діаграма***



Пакети загального призначення ***Compiler*** та ***Utils*** мають лише статичні відокремлені методи, тому вони не мають звязків з іншими пакетами

***Class Diagram*** ***/ Пакет compiling***



***Class Diagram*** ***/ Пакет utils***



Пакет ***structures*** містить класи, що предсавляють певні сутності системи, що допомагають у звязуванні даних з представленням.



Пакет ***testing*** містить усі класи, що відповідають за тестування розвязків учасників. Більшість з класів реалізують інтерфейс потоку для можливості керувати системою під час тестування.



Пакет ***entities*** містить класи оболонки для таблиць бази даних. Для їх функціонування необхідно зробити їх здатними до серіалізації.



Було використано два інтерфейси стандартних для Java тому стандатний пакет ***java*** було включено до діаграми класів.



У підпакетах знаходяться необхідні інтерфейси.





При тестуванні головний клас тестування ***OnlineTesterThread*** має декілька станів своєї роботи. Ці стани класів зображуються на ***Statechart Diagram***



Також клас ***Tester*** - тестування окремого рішення має стани.



Клас ***OneTest*** - тестування окремого тесту також має стани



Основні події системи такі як тестування, обробка помилок, допомога необхідно уточнювати за допомогою ***Sequence Diagrams*** та ***Collaboration Diagrams***, які висвітлять на рівні викликів методів класів взаємодію обєктів для виконання тої чи іншої задачі.

***Collaboration Diagram / Допомога***

На діаграмі зображено взаємодію основної програми з модулем надання допомоги



***Sequence Diagram / Допомога***

На діаграмі послідовності відображено взаємодію основної програми з модулем допомоги у зрізі часу.



***Collaboration Diagram / Обробка помилок***

На діаграмі зображено взаємодію основної програми з модулем обробки помилок



***Sequence Diagram / Обробка помилок***

На діаграмі послідовності відображено взаємодію основної програми з модулем обробки помилок у зрізі часу.



***Collaboration Diagram / Тестування***

На діаграмі зображено взаємодію класів перевірки в основному циклі перевірки. Ця діаграма надає лише приблизне формулювання того як має проходити перевірка учасників у системі.



***Sequence Diagram / Тестування***

На діаграмі послідовності відображено хід тестування рішень у зрізі часу. Основним класом при перевірці є клас ***OnlineTesterThread*** який циклічно перевіряє наявність неперевірених рішень у базі даних.



На цьому логічне проектвання системи завершено. Основні функціональні питання та питання взаємодії було вирішено. Наступним кроком проектування є розподілення функціональних обовязків за фізичними частинами системи. Необхідно виділити основні фізичні частини системи та вірно розподілити обов’язки між ними. Від вірного розподілу залежить зручність використання системи, її масштабовність та універсальність у використанні. Основним засобом на цьому етапі проектування є ***Component Diagram.***

***Component Diagrams*** призначені для виявлення того, які частини проекту відповідають певним функціональним інтерфейсам, що необхідні користувачу. У системі тестування необхідно надати інтерфейс адміністратора та учасника олімпіади. На загальній діаграмі необхідно задати основні пакети тестуючої системи на всіх рівнях.

***Component Diagram / Головна діаграма***



Пакет ***Інтерфейси*** відповідаю двом основним інтерфейсам,що має надавати система



Пакет ***Тестуючий комплекс*** представляє собою саму тестуючу систему, що складаютсья з двох підпакетів



***Модуль адміністратора*** являє собою реалізацію інтерфейсу адміністратора



***Модуль користувача*** представляє реалізацію інтерфейсу користувача



Пакет ***Обробки помилок*** містить відповідний модуль



Пакет ***Засоби тестування*** містить програми для проведення тестування рішень зокрема таймер, та програма порівняння



Пакет ***База даних*** представляє таблиці бази даних системи

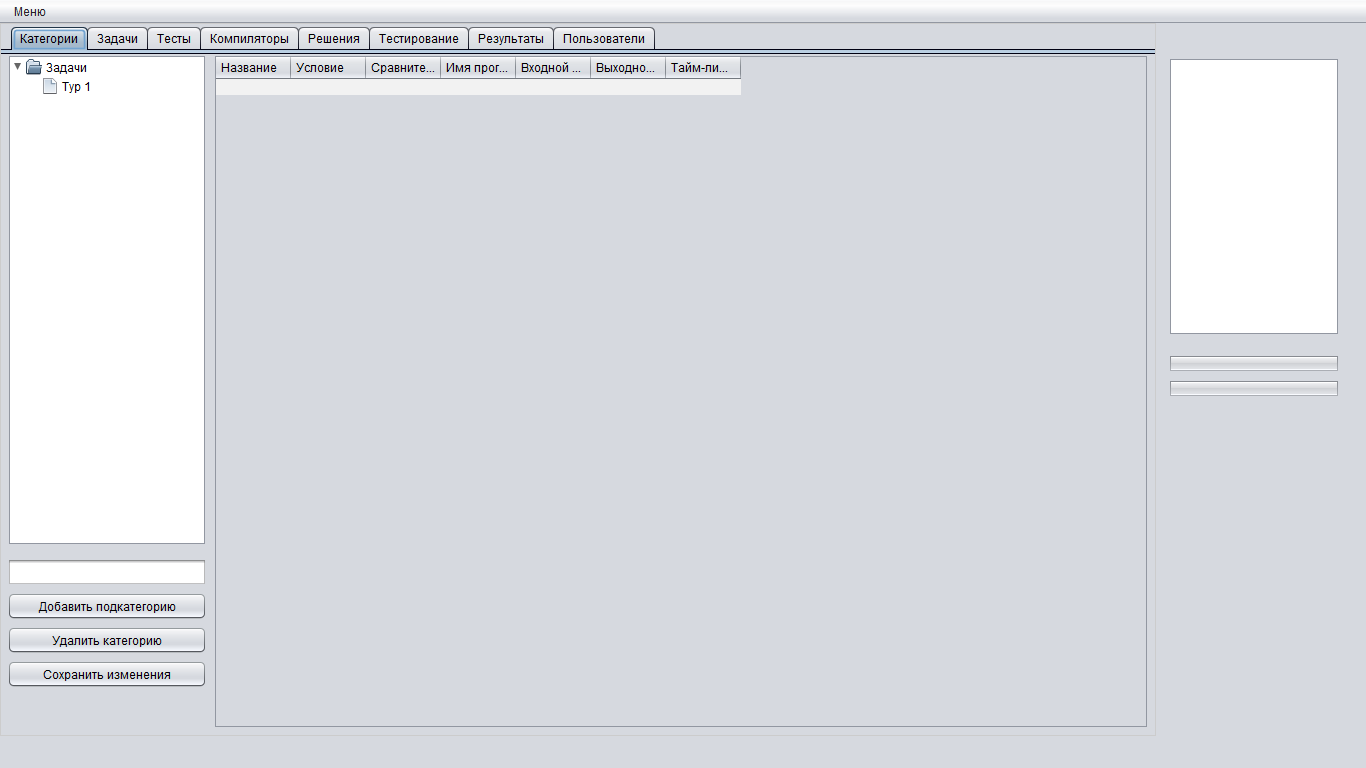


Діаграма ***Deployment Diagram*** завершує етап проектування системи. На ній зображуються фізичні умови розгортання тестуючої системи. Оскільки ***Sandra 2010*** задумувалася, як клієнт-серверна система, необхідно відобразити мережеву взаємодію на цій діаграмі, а також зазначити той факт, що незалежні частини системи – модуль учасника та модуль адміністратора, розділяють спільну базу даних.

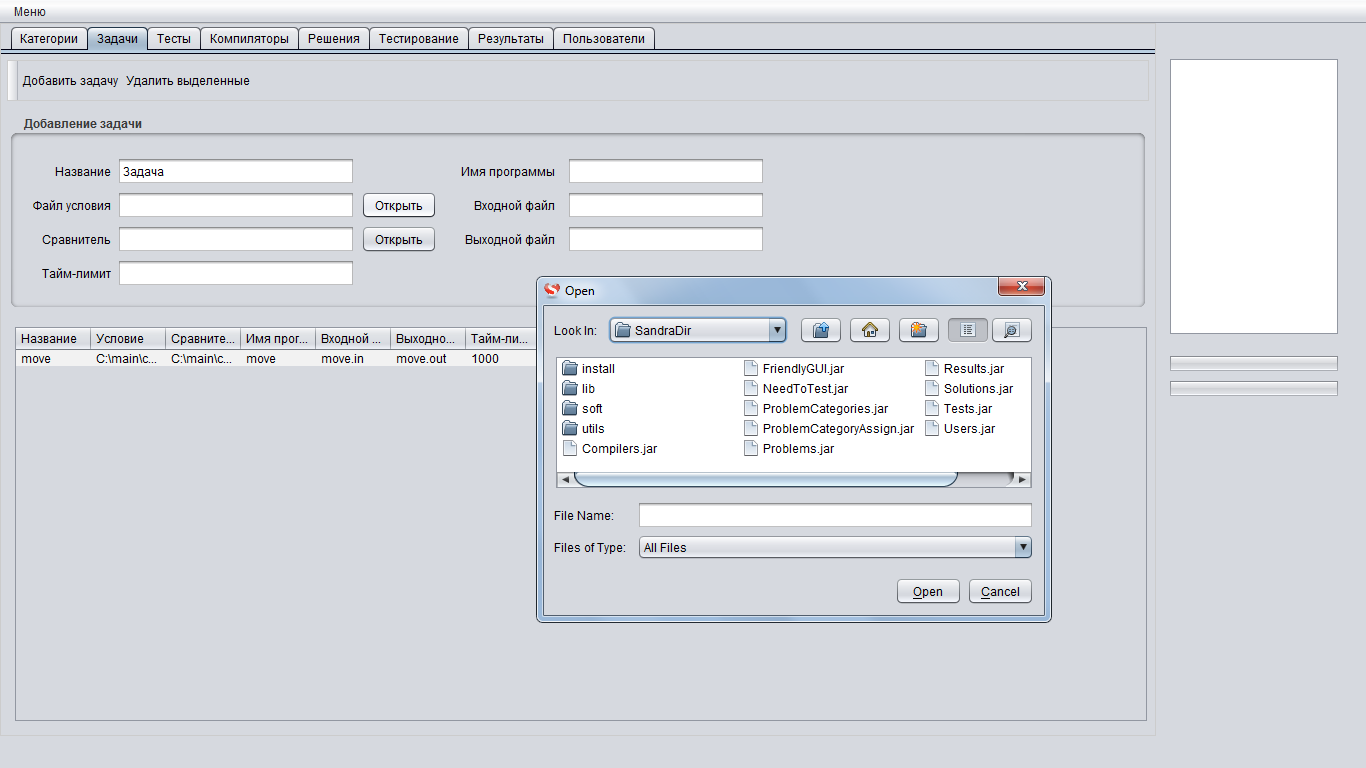


Демонстрація графічного інтерфейсу програми

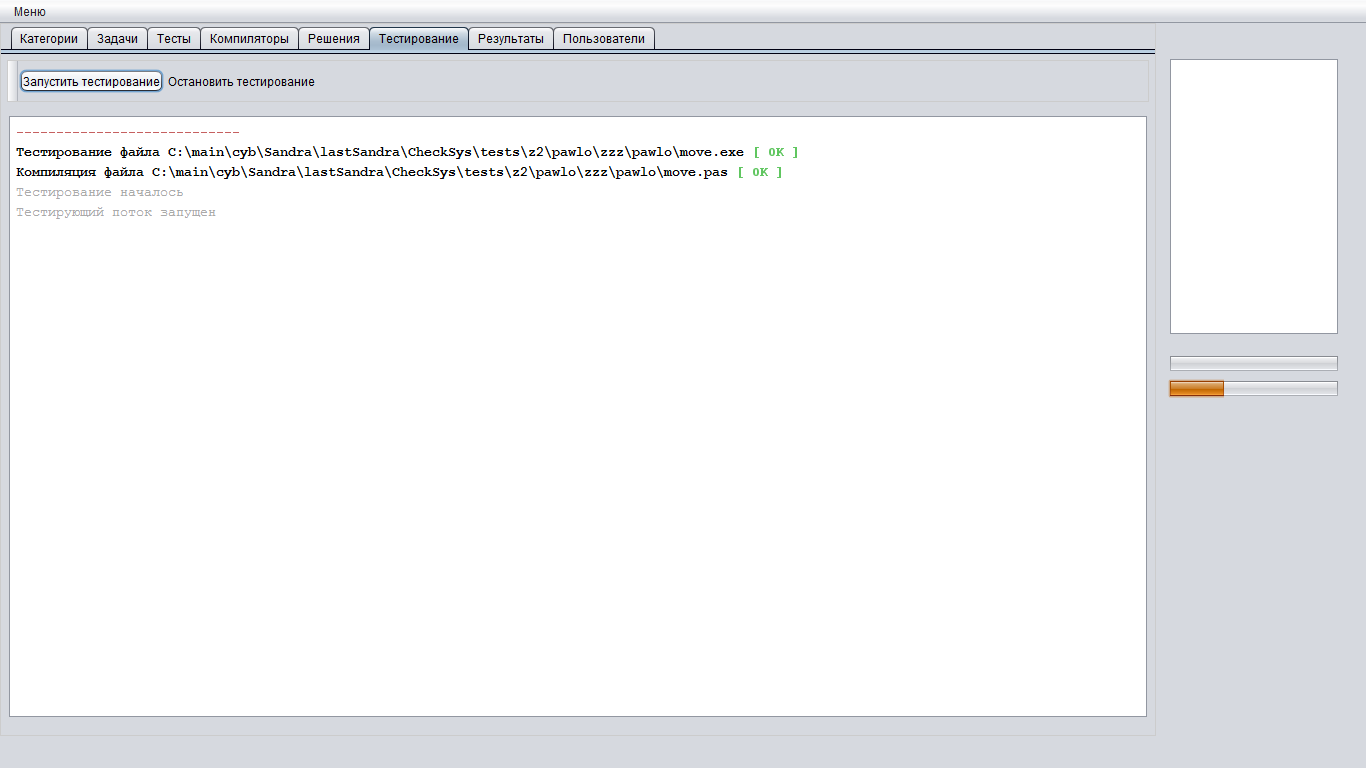
-***Категорії задач*** : розділ програми, де задачі подаються в ієрархічному вигляді розбиті на категорії



-***Задачі, Тести та Компілятори*** : в цих розділіах програми додаються задачі, тести та компілятори відповідно. Після створення задачі можна розбивати по категоріях, зводити за ними рейтинги та перевіряти їх рішення.



-***Перевірка рішень*** : головний процес системи – тестування рішень учасників олімпіади проводиться у цьому розділі.



-***Користувачі*** : в цьому розділі адміністратор може додавати чи видаляти користувачів

