

Харківський національний університет імені В. Н. Каразіна
Факультет комп'ютерних наук

РОЗРАХУНКОВО-ГРАФІЧНА РОБОТА №2
ДИСЦИПЛІНА: «КРОС-ПЛАТФОРМНЕ ПРОГРАМУВАННЯ»

Виконав: студент групи КС24
Бондаренко Олег Анатолійович

Перевірив: Споров Олександр
Євгенович

Харків
2024

ЗМІСТ

ЗМІСТ.....	2
Вступ	3
Основна частина.....	4
Класи та їх функціонал.....	4
Висновок	8
Додаток.....	9

Вступ

На цьому занятті ми вивчаємо основи компонентної технології JavaBeans, що є однією з основних технологій для створення перевикористовуваних компонентів у програмуванні на Java. Ми досліджуємо правила створення JavaBeans-компонентів, включаючи їхню структуру та особливості налаштування.

Основна частина

Класи та їх функціонал

У програмі ми маємо 3 пакети:

1. myApplication

2. mybeans

3. xml

1. Клас MyFrame

Цей клас є частиною пакету myApplication і відповідає за створення графічного інтерфейсу користувача у Java. Він розширює клас JFrame, який є контейнером для всіх інших компонентів. У конструкторі встановлюються розміри та розташування вікна, створюються кнопки для відкриття, збереження та очищення даних, а також для виходу з програми.

Також в класі створюються обробники подій для цих кнопок, які виконують відповідні дії. Наприклад, при натисканні кнопки "Open" відкривається файл, а при натисканні "Save" - дані зберігаються у файл. Крім того, виконується оновлення графічних компонентів після змін у даних, щоб забезпечити їх синхронізацію та відображення актуальної інформації користувачу.

2. Інтерфейс Test

Цей клас є точкою входу в програму. Він використовується для запуску та ініціалізації основного вікна програми. Клас розширює JFrame, що

дозволяє йому створювати вікно програми з графічним інтерфейсом користувача. На рисунку 1 зображено результат виконання програми.

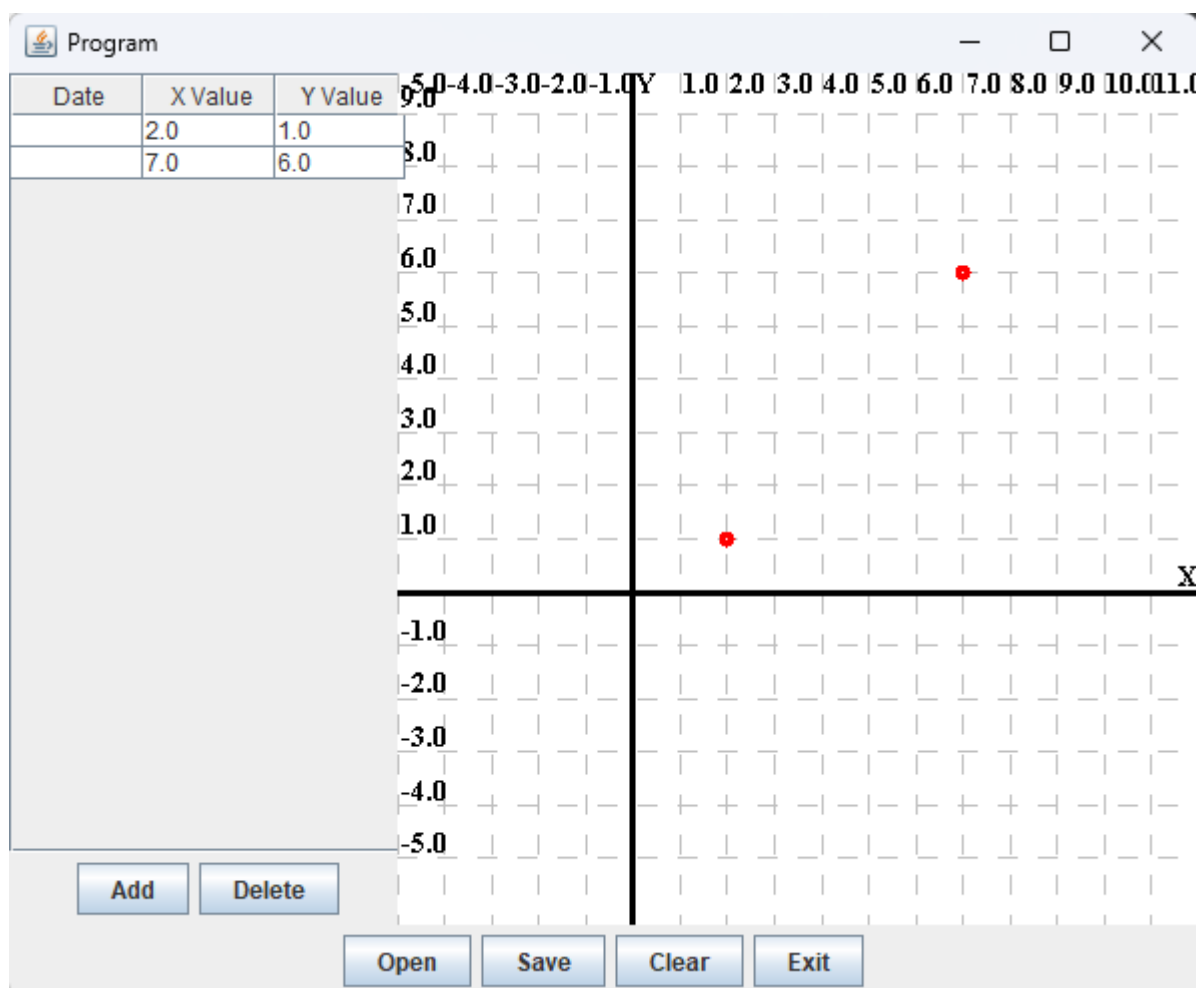


Рисунок 1. Виконання класу Test

3.Клас Data

Цей клас представляє об'єкт, який містить дані про певну точку або значення. Він містить три основні поля: date (дата), x і y, які відповідають вісьем координат або значенням.

Клас має два конструктори: один без параметрів, який створює об'єкт з пустими значеннями, і другий конструктор з параметрами, який дозволяє встановити значення x, y та date при створенні об'єкта.

4. Клас DataSheet

Цей клас є контейнером для об'єктів класу Data. Він використовується для зберігання та управління набором даних. Основна його функція - це забезпечення можливості додавання, видалення та доступу до об'єктів Data.

5. Клас DataSheetGraph

Цей клас є підкласом JPanel, який використовується для відображення графіку на основі даних з класу DataSheet. Він реалізує метод paintComponent, щоб намалювати графік у вікні.

6. Клас DataSheetTable

Цей клас є підкласом JPanel і використовується для відображення даних у вигляді таблиці. Він містить об'єкт JTable, який відображає дані, і об'єкт DataSheetTableModel, який визначає модель даних для цієї таблиці.

7. Клас DataSheetTableModel

Цей клас є підкласом AbstractTableModel і використовується для визначення моделі даних для таблиці у GUI. Він представляє собою посередника між відображенням даних та їхнім джерелом. Клас визначає, як дані відображаються у таблиці, як вони зчитуються, змінюються та оновлюються.

8. Клас MySAXParser

Цей клас використовується для розбору XML-файлів з даними, які містять точки (x, y) з датою, та перетворення їх на об'єкти типу Data та DataSheet. Для цього використовується SAX парсер. Цей клас є частиною функціоналу програми для обробки та візуалізації даних із XML-файлів. Він дозволяє ефективно виконувати розбір XML та інтегрувати ці дані у вже існуючий або новий DataSheet для подальшого використання.

9. Клас NumMethods

Цей клас WriteToXML є абстрактним класом, який містить статичний метод saveXMLDoc, призначений для збереження об'єктів типу DataSheet у вигляді XML-файлу. Клас використовує API для обробки XML документів і відтворює передані дані у вигляді відповідного XML-документу.

Висновок

Ми розробили програму, яка надає зручний інтерфейс для роботи з даними у вигляді точок. Використовуючи цю програму, користувач може додавати дані у таблицю, представлену в графічному інтерфейсі, і бачити їх візуалізовані на графіку у вигляді точок.

Завдяки використанню XML, програма може зберігати ці дані у структурованому форматі, що дозволяє зберегти структуру та інтегрованість даних. Крім того, вона дозволяє завантажувати раніше збережені дані для подальшого використання.

Цей функціонал дозволяє користувачам легко працювати зі своїми даними, маніпулюючи ними через зручний інтерфейс програми та забезпечуючи можливість зберігати та знову використовувати їх у майбутньому.

Додаток

Репозиторій: https://github.com/Oleg23332/CrossPlat/tree/main/RGR_2