НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

Факультет прикладної математики

Кафедра прикладної математики

Звіт

лабораторної роботи №3 цикл “Програмування”

із дисципліни «Алгоритмічні основи обчислюваної геометрії та комп’ютерної графіки»

Виконав: Керівник:

Студент групи КМ-03 доцент Сирота С. В.

Куцалаба Н.В.

Київ — 2021

ЗМІСТ

[1 ПОСТАНОВКА ЗАДАЧІ 2](#_Toc87456837)

[1.1 Мета роботи 2](#_Toc87456838)

[1.2 Завдання до лабораторної роботи 2](#_Toc87456839)

[1.3 Індивідуальні завдання згідно з варіантом 2](#_Toc87456840)

[2 ОСНОВНА ЧАСТИНА 3](#_Toc87456841)

[2.1 Опис програми 3](#_Toc87456842)

[ВИСНОВКИ 4](#_Toc87456843)

# ПОСТАНОВКА ЗАДАЧІ

## Мета роботи

Розробити програмний засіб, який знаходить опуклу оболонку множини точок заданих своїми координатами та відображує її на координатній площині і зберігає зображення в одному з графічних форматів. Завдання до лабораторної роботи

Скачати файл з датасетом. Файл в текстовому форматі містить пари цілих чисел які є координатами точок.

Необхідно написати програму будь якою мовою з використанням будь яких бібліотек яка

* Зчитує датасет з файлу;
* Знаходить опуклу оболонку і зберігає її у вигляді окремого датасету;
* Встановлює розміри вікна (полотна – canvas size) **960х540** пкс;
* Відображає опуклу оболонку за допомогою відрізків синього кольору;
* Відображає точки вихідного датасету, разом з опуклою оболонкою;
* Виводить результати у файли будь-якого графічного формату.

## Індивідуальні завдання згідно з варіантом

Завдання з DS6.txt

# ОСНОВНА ЧАСТИНА

## Опис програми

Для написання програми було обрано мову Java. Використано наступні бібліотеки: javax.swing, javax.awt, java.imageio, java.io, java.util, java.nio.

І так в нас є 4 файли в яких описані 5 класів. І так перший файл ReaderDS.java в якому описаний клас ReaderDS, який містить поле-двовимірний масив int. В конструктор об’єкта цього класу буде передаватись шлях до вхідного датасету. Також в цьому класі є функція readsDS, що зчитує вхідний датасет та записує в в двовимірний масив координати x та y, та в масив типу Point.

ConvexHull.java – файл в якому клас Point, що створений для полегшеної праці з координатами, а також клас ConvexHull в якого є статичне поле hull (по-суті це ті ж точки). Цей клас реалізує алгоритм Грехема. Коротко про алгоритм, спершу шукаємо найлівішу точку, тобто з найменшим значенням x, вона гарантовано входить в опуклу оболонку. Далі сортуємо точки в порядку, в залежності від їх розташування від опорної точки(найлівішої). Після цього визначаємо кути, точніше повороти направо та наліво. Якщо кут з повернутий вправо то вилучаємо цю точку, якщо вліво залишаємо. Функції:

orientation(…) – визначає поворот кута(ліворуч, праворуч);

convexHull(…) – реалізовує алгоритм Грехема;

Файл Main.java, в якому описано один клас Main, що наслідує клас JComponent.

Що ж в головному методі, створюється об’єкт того ж класу Main, а також об’єкт типу JFrame, який слугуватиме вікном для зображення, в конструктор якого передаємо назву вікна. Далі ідуть наступні методи з класу JFrame:

setSize(…) – створює вікно заданого розміру;

setVisible(…) – визначає активність вікна;

setLocation(…) – визначає місце на екрані, де розміститься вікно;

add(…) – передає об’єкт, що відповідатиме за вміст вікна;

Тепер можна розглянути метод paint, якому передається об’єкт предка класу Graphics2D. І так в методі створюється полотно розміру 960x540 (для запису графічного файлу) також створюються об’єкти класу Graphics2D для того, щоб зображувати точки й зберегти в графічному форматі це зображення. Отже, функції:

setColor(…) – визначає колір полотна або точок, на якому будуть зображуватись точки;

fillRect(…) - створює полотно для зображення точок;

draw(…) – відображає точки на полотні;

drawline(…) відображає лінію між двома точками;

Також в функції присутній два блоки try-catch. Перший відловлює помилку з проблемою відкриття файлу з датасетом або його відсутності.

А інший з проблемою запису нашого зображення в графічний формат.

Точки у нас досі немає, тому створюємо об’єкт типу Rectangle, якому предаємо початкові координати точки(0,0) та розмірність, в даному випадку 1x1. Іде цикл який перебирає всі точки, що входять в опуклу оболонку та сполучає кожні сусідні синьою лінією. Наступний цикл відображає точки з вхідного датасету. Далі сполучаємо останню точку з опорною та отримуємо опуклу оболонку та за допомогою методу fill(…) відображаємо у вікні, а метод draw(…) малює на полотні, щоб зберегти в графічний формат. За допомогою методу write(…) з класу ImageIO зберігаємо наше намальоване полотно в обраному форматі (.jpeg), відповідно вказуємо шлях для збереження та назву файлу з форматом(для цього слугує константа PATH).

Останній файл MyWriter.java з класом MyWriter, в якому реалізована функція, яка записує в файл координати точок, що входять в опуклу оболонку.

# ВИСНОВКИ

У процесі виконання лабораторної роботи було Розроблено програмний засіб, який відображує  за заданим датасетом точки і зберігає зображення в одному з графічних форматів.

Отримані навички роботи з деякими класами(JFrame, Graphics, Graphiscs2D, JComponent, Vector, ImageIO) з бібліотек Java (javax.swing, java.awt, javax.imageio, java.util).

Результати роботи викладені на хмарному сховищі, текст програми на GIT.

Посилання на Github: <https://github.com/KutsalabaNazariy/VIsualizationDataset>