**Лабораторна робота №4**

**Абстрактні класи та інтерфейси**

**Мета роботи**: Використовуючи теоретичне підґрунтя абстракті класи та інтерфейси у мові Java, виконати дії що будуть вказано в завданні до лабораторної роботи.

# ЗАВДАННЯ ДО ЛАБОРАТОРНОЇ РОБОТИ

Для виконання лабораторної роботи необхідно використовувати репозиторій на github, що був створений при виконанні завдання до першої лабораторної роботи. Результати роботи необхідно буде завантажити на github для перевірки.

При використанні IDE створити проект під назвою lab04 для розміщення результатів виконання завдань. У випадку використання звичайного текстового редактору створити окрему директорію lab04 для виконання завдань.

Виконайте наступне завдання.

**ЗМІСТ**

[1. ЗАВДАННЯ ДО ЛАБОРАТОРНОЇ РОБОТИ 1](#_Toc166349262)

[2. ВИКОНАННЯ ЗАВДАНЬ ЛАБОРАТОРНОЇ РОБОТИ 3](#_Toc166349263)

[2.1. Реалізація абстрактного класу Figure 3](#_Toc166349264)

[2.2. Реалізація класу Point 4](#_Toc166349265)

[2.3. Реалізація класу Triangle 4](#_Toc166349266)

[2.4. Реалізація класу Quadrilateral 6](#_Toc166349267)

[2.5. Реалізація класу Circle 8](#_Toc166349268)

[2.6. Реалізація класу Main. Перевірка роботи коду завдання 9](#_Toc166349269)

[3. ВИСНОВОК 11](#_Toc166349270)

# ВИКОНАННЯ ЗАВДАНЬ ЛАБОРАТОРНОЇ РОБОТИ

|  |  |
| --- | --- |
| 1 | Створіть три класи Triangle, Quadrilateral, Circle, що розширяють абстрактний клас Figure.  Реалізуйте **конструктор** для кожного з класів.  Для трикутника конструктор у якості параметра приймає три вершини класу Point.  Для чотирикутника конструктор у якості параметра приймає чотири вершини класу Point.  Для кола конструктор у якості параметра приймає координати центу класу Point та радіус типу double.  Конструктор кожної з фігур має перевіряти, що фігура є не виродженою.  Реалізуйте методи **підрахунку площі** для кожної фігури.  Реалізуйте метод **toString()** для кожної з фігур який виводить текст у вказаному вигляді:  Трикутник - Triangle[A(x,y) B(x,y) C(x,y)]  Чотирикутник - Quadrilateral[A(x,y) B(x,y) C(x,y) D(x,y)]  Коло - Circle[(x,y) Radius]  Реалізуйте метод Point centroid()  Повернути центроїд фігури.  Іншими словами, це має бути "центроїд площі". |

## Реалізація абстрактного класу Figure

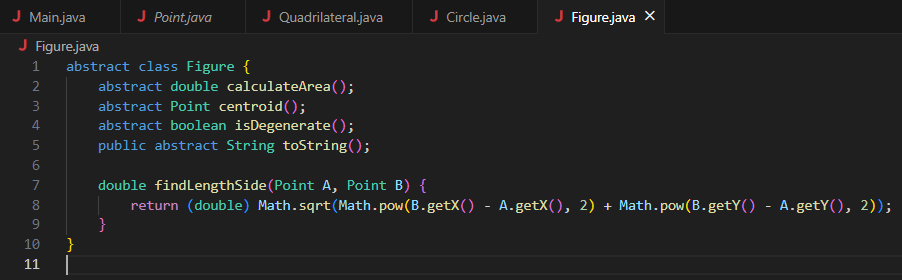


Рис. 2.1. Реалізація абстрактного класу Figure.

Мені потрібно створити клас Figure, який за умовою завдання повинен бути абстрактним. Такий клас створюється за допомогою додавання слова abstract перед словом class та назвою класу. Додаємо абстрактні чотири методи у ньому: calculateArea(), centroid(), isDegenerate(), toString() з відповідними типами та ідентифікаторами. Абстрактний клас зобов’язує класи, що його успадкували, перевизначити і реалізувати його абстрактні методи. Ще додамо дефолт-метод в абстрактному класі findLengthSide(), який буде приймати дві точки і вираховувати відстань між ними, що буде відрізком. Цей метод буде необхідний для знайдень довжин сторін фігур, які я пізніше реалізую.

## Реалізація класу Point

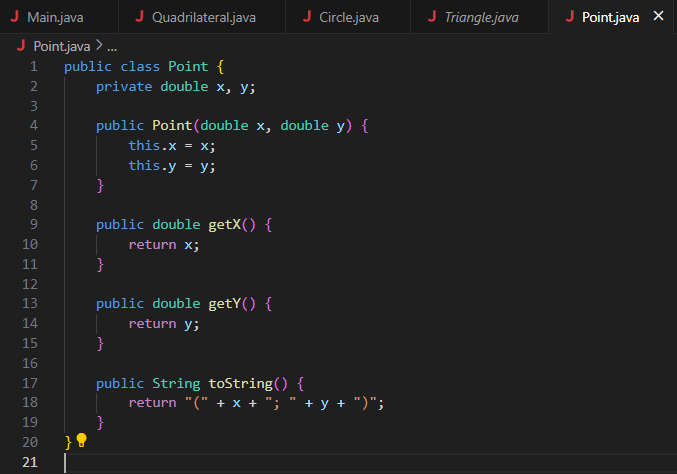


Рис. 2.2. Реалізація класу Point.

Цей клас я ще реалізував у своїй другій лабораторній роботі з цієї дисципліни. То ж просто переніс код цього класу в окремий файл проєкту(директорії) четвертої лаби.

## Реалізація класу Triangle

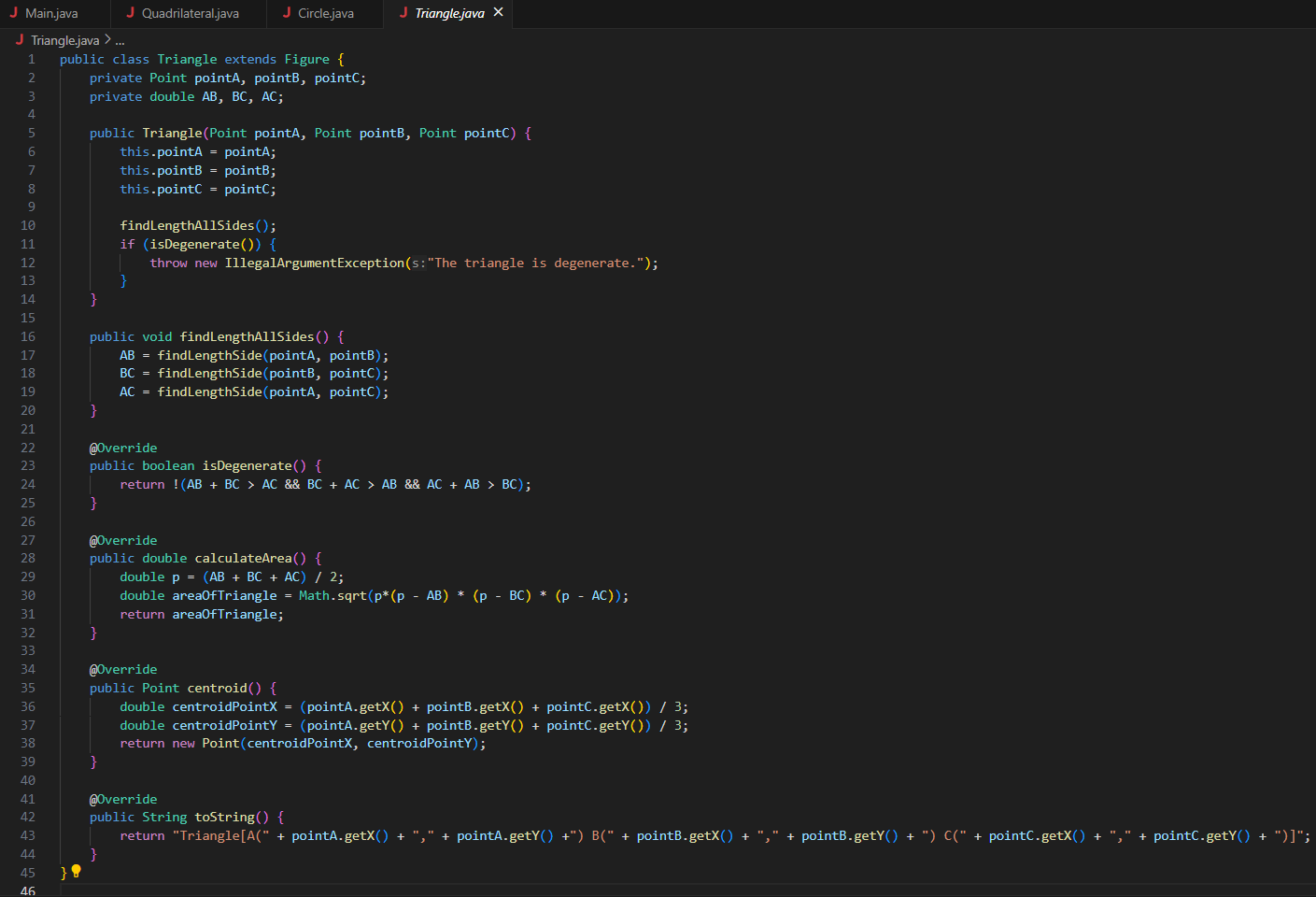


Рис. 2.3. Реалізація класу Triangle.

Практично більшу частину функціоналу цього класу я реалізував ще в другій лабораторній роботі цієї дисципліни, тому просто переніс код у окремий файл до директорії четвертої лабораторної роботи. Я лише зробив декілька змін в коді цього класу, а саме: клас Triangle успадкував абстрактний клас Figure, додав змінні AB, BC, AC, що відповідатимуть за довжини сторін і далі замість методів-гетерів для довжин сторін я створив метод findLengthAllSides(), у якому я присвою виклики методу findLengthSide, передаючи відповідні дві точки для кожної із сторін. Потім я викликав у конструкторі класу метод findLengthAllSides() і зробив перевірку в конструкторі, чи не є трикутник виродженим за допомогою методу isDegenerate(), якщо він вироджений, то викинути виключення із відповідним повідомленням.

Потім я перейменував інші методи, щоб вони сходились у назвах з методами абстрактного класу, утім логіку в них я залишаю, якою вона і була. Потім ще перед цими методами я додав анотацію @Override, це означає, що метод перевизначається з абстрактного класу-батька Figure. Ще у деяких місцях змінив код для більшої читабельності.

## Реалізація класу Quadrilateral

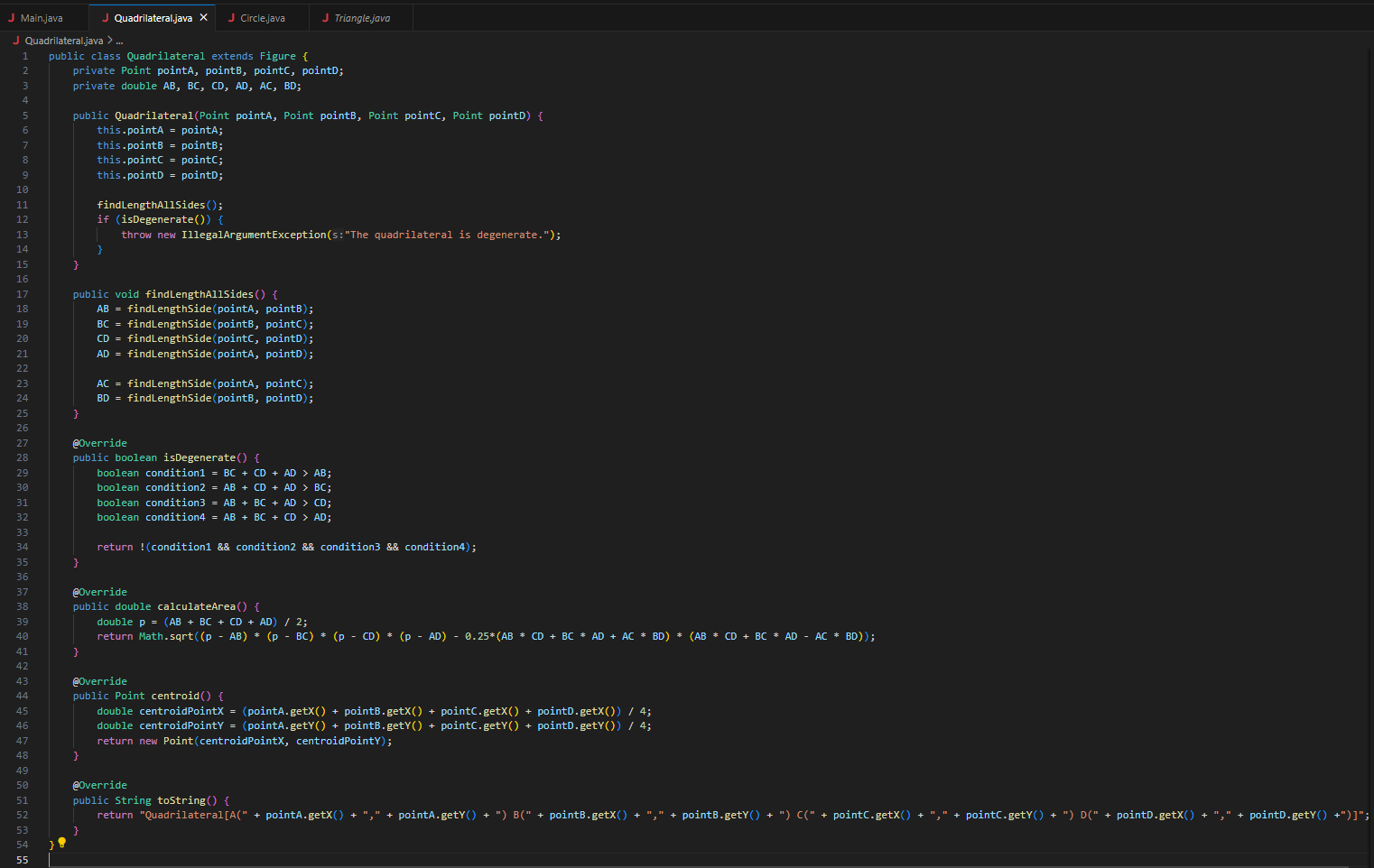


Рис. 2.4. Реалізація класу Quadrilateral.

Реалізація класу Quadrilateral cхожа на реалізацію класу Triangle. Quadrilateral успадковує абстрактний клас Figure. Потім копіюємо змінні та конструктор класу, метод findLengthAllSides() в наш клас Quadrilateral з класу Triangle. Додаємо AD(сторона чотирикутника) та BD(діагональ чотирикутника), pointD(точка четвертої вершини чотирикутника). Змінна АС буде вважатись не стороною, а діагоналлю чотирикутника. Відповідно до цих змін змінюємо конструктор, тепер він приймає чотири точки, які і будуть атрибутами. Виклик метод findLengthAllSides() залишаємо у конструкторі і перевірку за допомогою методу isDegenerate() теж.

У методі findLengthAllSides() знаходимо довжини сторін та діагоналей чотирикутника за допомогою методу findLengthSide(), передавши відповідні дві точки в якості аргументу.

Перепишемо реалізацію перевизначення методів isDegenerate(), calculateArea(), centroid(), toString().

Для того, щоб визначити, чи є чотирикутника виродженим, є таке правило, що сума будь-яких трьох сторін чотирикутника повинна бути більше за четверту. Правило знаходиться за посиланням: <https://profmeter.com.ua/communication/learning/course/course25/chapter01049/?LESSON_PATH=939.1049> - Властивості довжин сторін чотирикутника. Так як у нас чотири сторони, то потрібно чотири перевірки, помістимо їх в чотири змінни. І метод isDegenerate() повертає результат цього виразу !(condition1 && condition2 && condition3 && condition4). Тобто, якщо хоча одна умова дасть false, то це перетворяється у true за допомогою знаку «!», що означає, що чотирикутника вироджений, а якщо все true, то тоді перетворюється у false, тобто чотирикутник невироджений.

Далі перепишемо реалізацію перевизначення методу обрахунку площі фігури. Щоб обрахувати площу будь-якого довільного чотирикутника скористаємося формулою формулою Брамагупти, вона знаходиться за посиланням: <https://uk.wikipedia.org/wiki/%D0%A4%D0%BE%D1%80%D0%BC%D1%83%D0%BB%D0%B0_%D0%91%D1%80%D0%B0%D0%BC%D0%B0%D0%B3%D1%83%D0%BF%D1%82%D0%B8> - Формула Брамагупти. Реалізуємо цей вираз і повертаємо його результат.

Тепер перепишемо реалізацію перевизначення методу пошуку центроїда фігури. За визначенням у математиці це буде точка, координати якої це сума значень по осі Х всіх вершин поділено на 4 та сума значень по осі У всіх вершин поділено на 4, тобто С. Де х1, х2, х3, х4 та у1, у2, у3, у4 – це абсциси та ординати точок вершин чотирикутника відповідно. Це визначення пояснюється за посиланням: <https://www.youtube.com/watch?v=FymF4PXLI2c> - How to Find the Centroid of a Quadrilateral : Algebra, Geometry & Other Math Tips. Знаходимо абсцису та ординату центроїда і повертаємо об’єкт класу Point, передавши в якості аргументів абсцису та ординату.

Переписуємо реалізацію перевизначення методу toString(), що відповідає за рядкове представлення об’єкта відповідним чином, як нам указано в завданні.

## Реалізація класу Circle

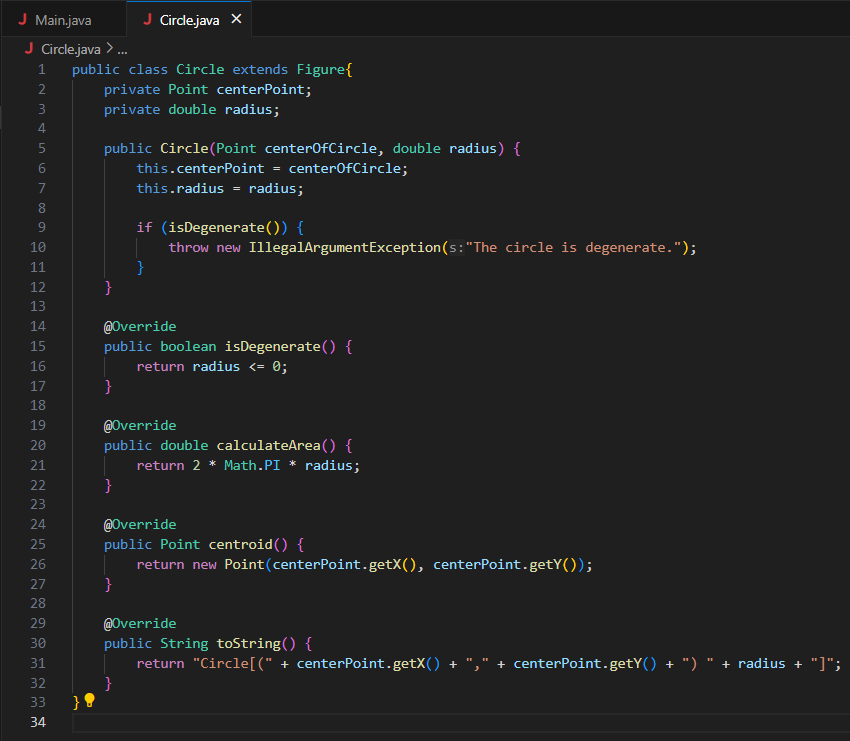


Рис. 2.5. Реалізація класу Circle.

Створюємо клас Circle, який успадковує абстрактний клас Figure. Створюємо змінні centerPoint (центральна точка кола), radius (радіус кола). Конструктор класу буде приймати два аргументи – центральну точку кола і радіус. Визначаємо атрибути центральну точку кола та радіус кола. У конструкторі ще також робимо перевірку на виродженість фігури за допомогою методу isDegenerate(), перевірка подібна за схемою як у попередніх класах. Метод isDegenerate() буде реалізовано далі.

За визначенням у математиці коло вважається виродженим, коли його радіус дорівнює нулю, тобто, коли коло є точкою. Це означає, що метод isDegenerate() просто повертає порівняння, чи радіус менше нуля чи дорівнює нулю, якщо так, то коло вироджене, якщо ні – то коло невироджене.

Далі перевизначимо метод calculateArea(). Але є нюанс, за математичними визначеннями коло немає площі, площу має круг, то ж нехай цей метод обраховуватиме довжину кола. Формула довжини кола є такою: С = 2R, де С – довжина кола, R – радіус кола, – число «Пі»(приблизно 3.14). Повертаємо к результат обрахунок за цією формулою.

Перевизначимо метод centroid(). За математичним визначенням центроїдом кола є центральна точка кола, тобто, повертаємо об’єкт класу Point, передавши в якості аргументів абсцису та ординату центральної точки кола.

Перевизначаємо метод toString(), що використовується для рядкового представлення об’єкта. Змінимо представлення на таке, як вказано в завданні.

## Реалізація класу Main. Перевірка роботи коду завдання

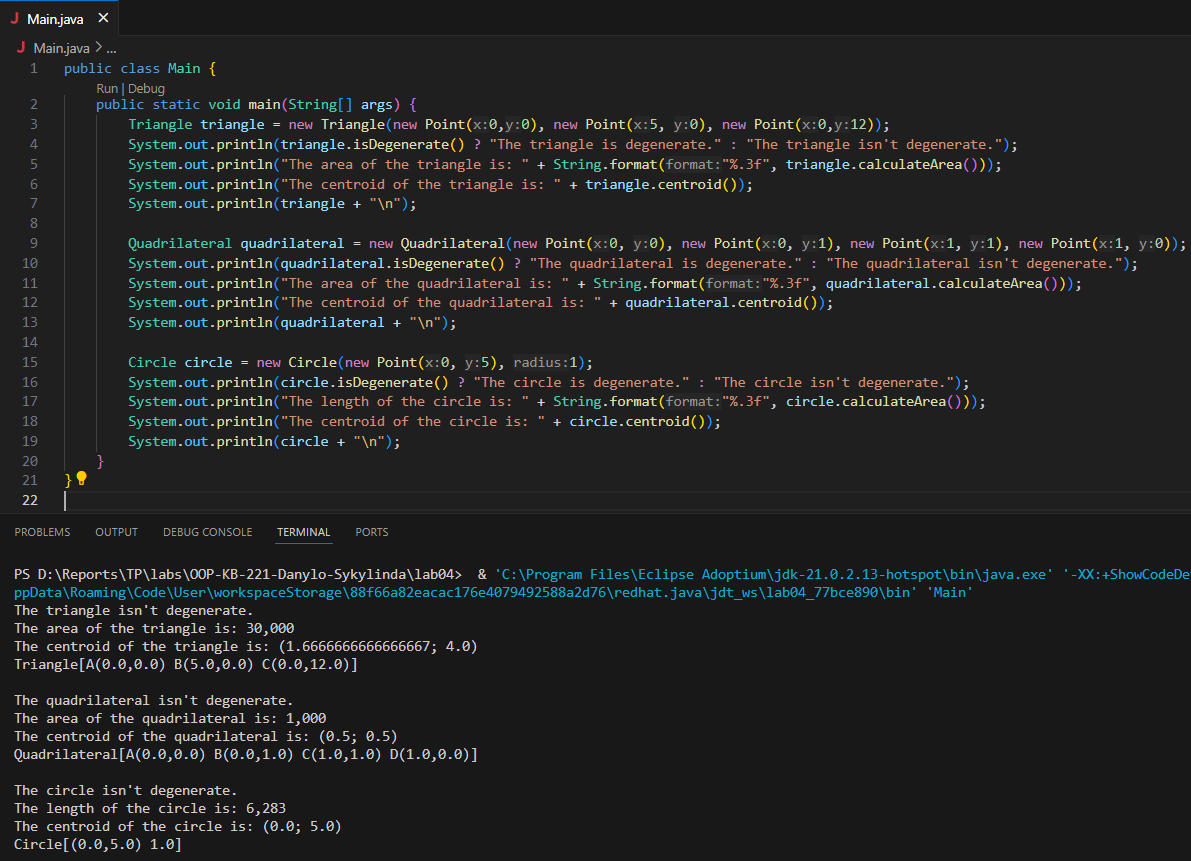


Рис. 2.6. Реалізація класу Main. На фото нижче перевірка роботи коду завдання.

Створюємо об’єкти класів Triangle, Quadrilateral та Circle. Для кожного об’єкту викликаємо його методи isDegenerate(), calculateArea(), centroid() та для кожного з об’єктів виводимо рядкове представлення об’єкта. Як бачимо нижче на Рис. 2.6. код всього завдання написаним мною все обраховує та виводить правильно, як і вимагалося.

# ВИСНОВОК

Я ознайомився з матеріалом, що було подано до цієї лабораторної роботи. Я дізнався про абстрактні класи та інтерфейси, про їх відмінності та про те, де краще їх застосувати в коді. Також я виконав завдання лабораторної роботи, засвоїв теоретичні навички на практиці, набув практичних навичок при вирішенні подібних завдань. Покращив своє розуміння в застосуванні ООП. Завантажив усі виконані завдання та звіт лабораторної роботи на Github та Moodle.