# Лабораторна робота №2

**Класи**

**Мета роботи**: Використовуючи теоретичне підґрунтя про об’єктно орієнтоване програмування виконати дії що будуть вказано в завданні до лабораторної роботи.

1 Вам необхідно реалізувати метод **intersection** у класі **Line**. Він повинен повертати точку перетину двох ліній (клас **Point**). Якщо лінії збігаються або перетинаються, метод повинен повертати значення null.

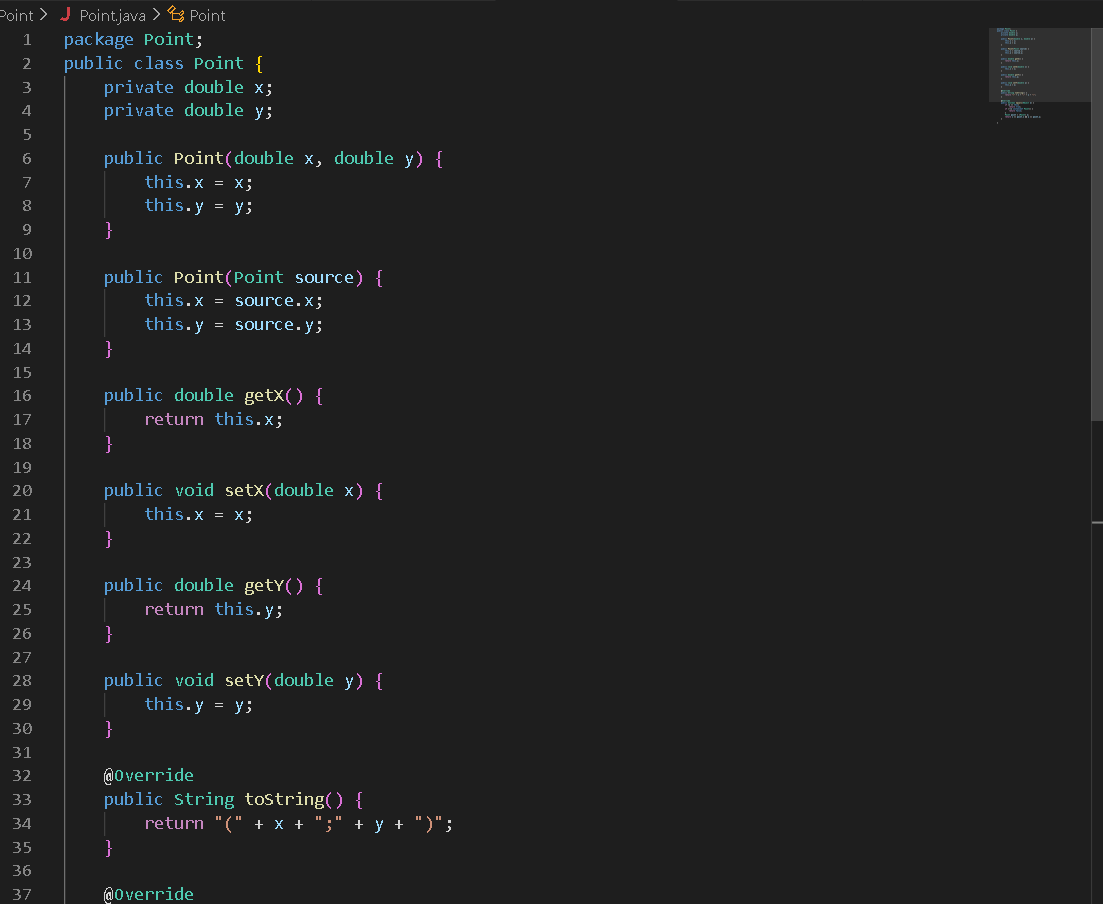
Функція, що описує пряму y = k X + b

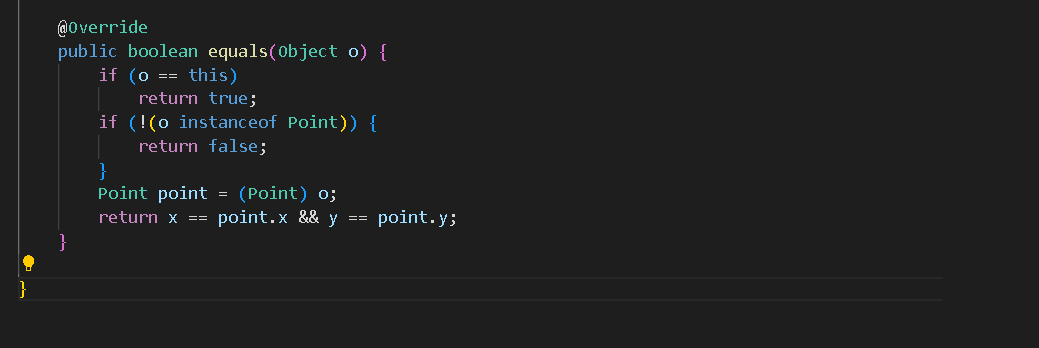
Користувач вводить значення k та b для двох прямих

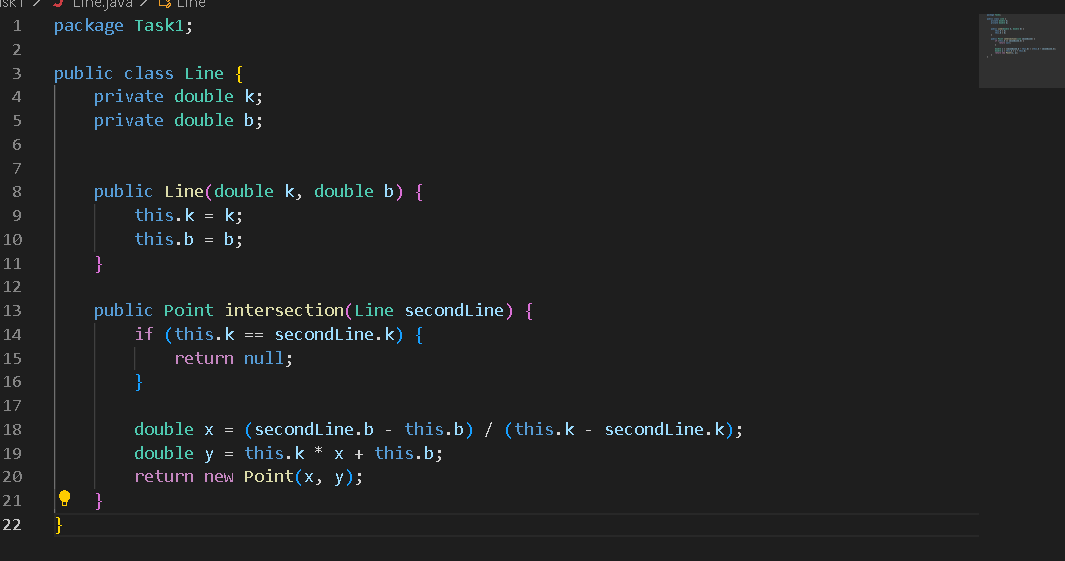
Line line1 = new Line(1,1);

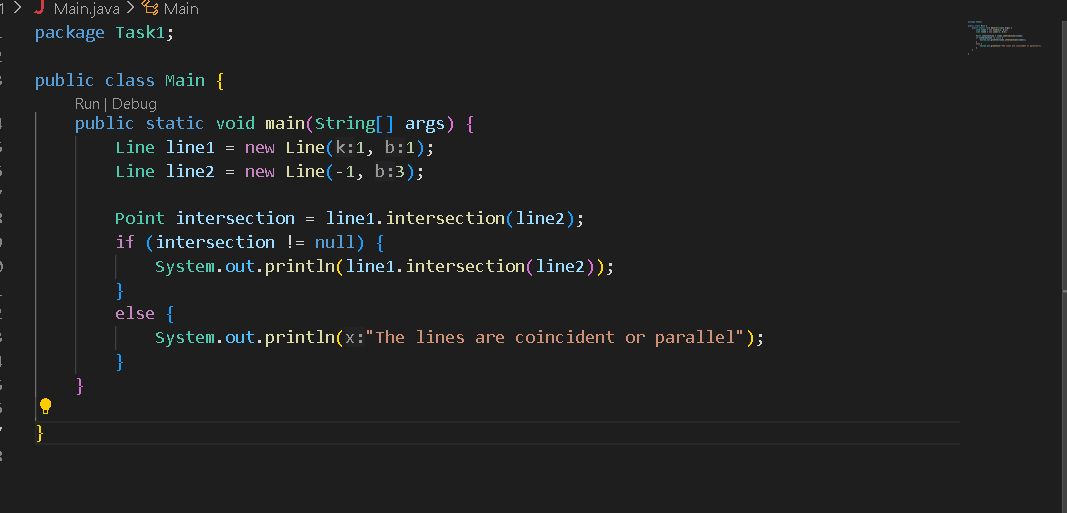
Line line2 = new Line(-1,3);

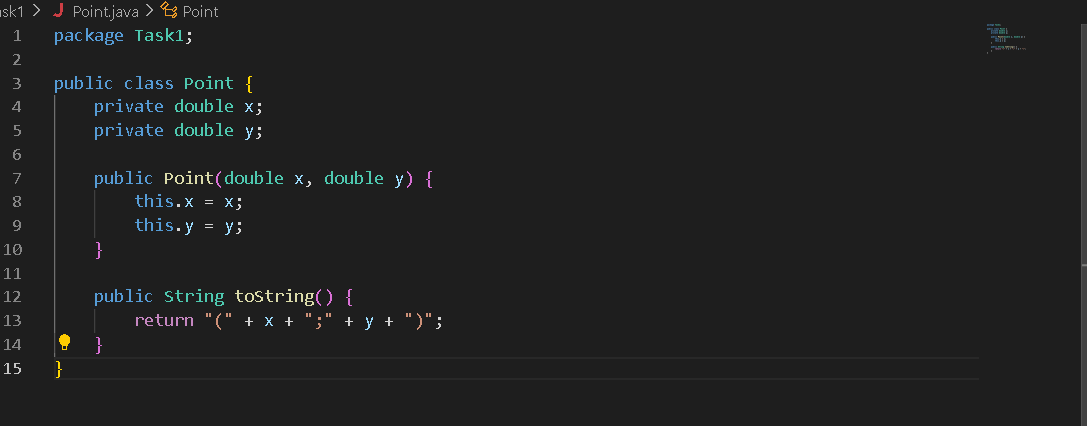
System.out.println(line1.intersection(line2)); // (1;2).











2 Реалізуйте методи класу Segment (відрізок):

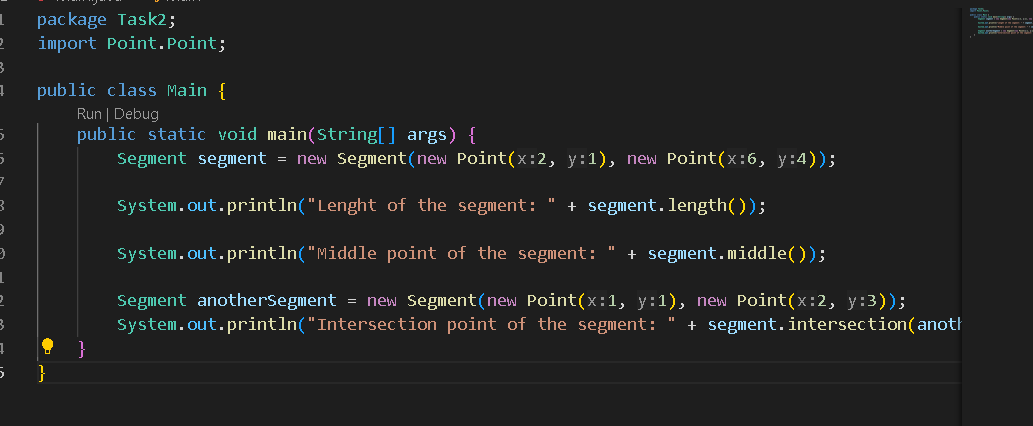
Конструктор, в який як параметри передаються координати точок початку і кінця відрізка (використовуйте клас Point).

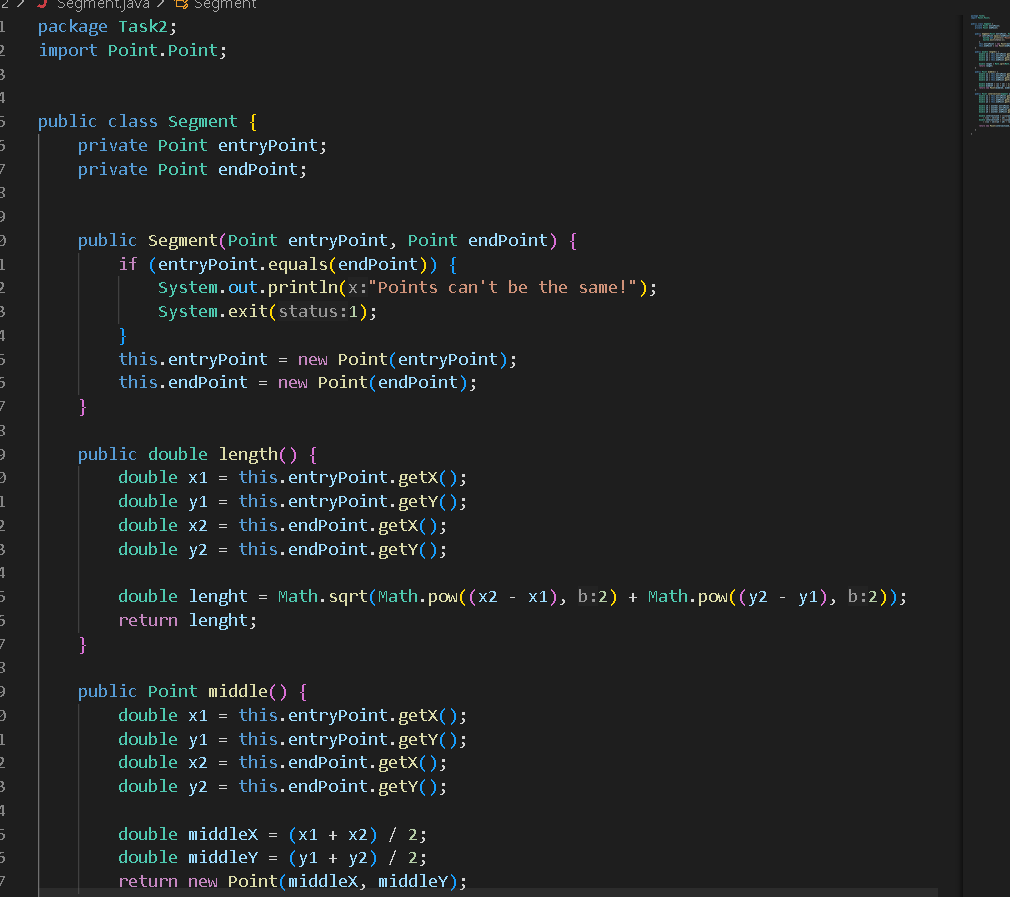
Переконайтеся, що створений відрізок існує і не є виродженим, що означає, що початок і кінець відрізка не є однією точкою.

Реалізуйте метод double **length**() – повертає довжину сегмента.

Реалізуйте метода Point **middle**() – повертає середню точку сегмента.

Реалізуйте метод Point **intersection**(Segment another) – повертає точку перетину поточного відрізка з іншим.







3 Реалізуйте методи класу **Triangle**:

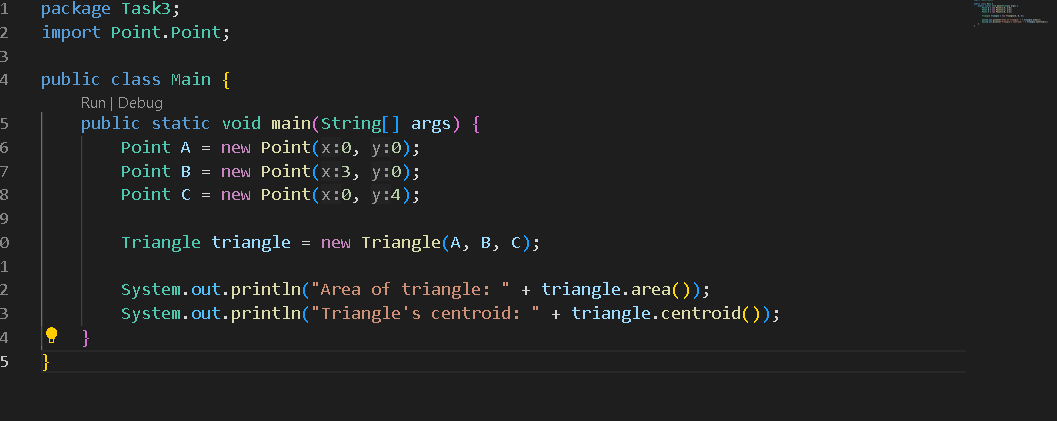
Конструктор, що має як параметри координати трьох вершин (клас Point).

Переконайтеся, що ці точки належать до вершин трикутника.

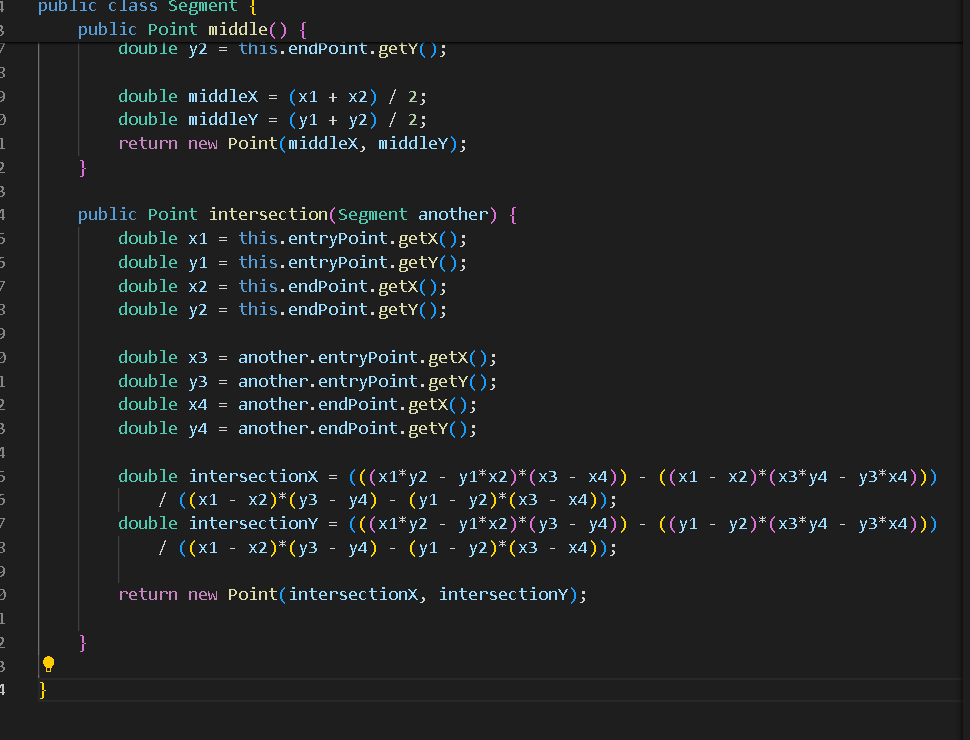
Перевірте, що створений трикутник існує і не вироджений.

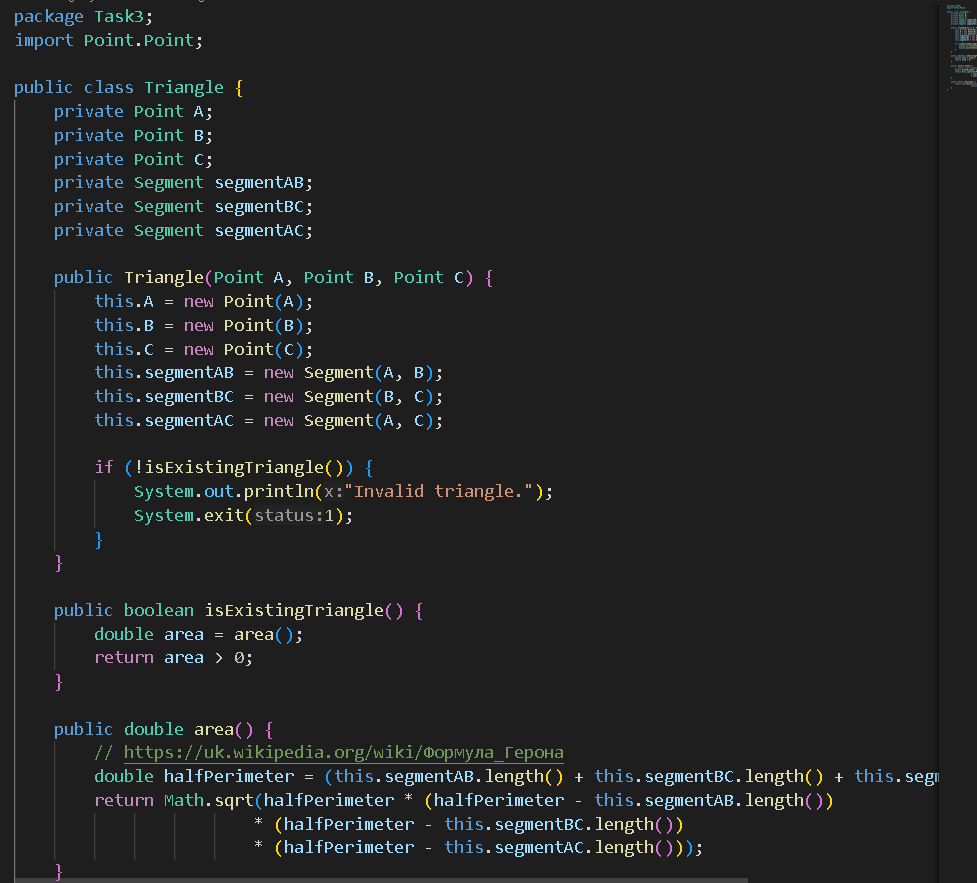
Реалізуйте метод double **area**() – повертає площу трикутника.

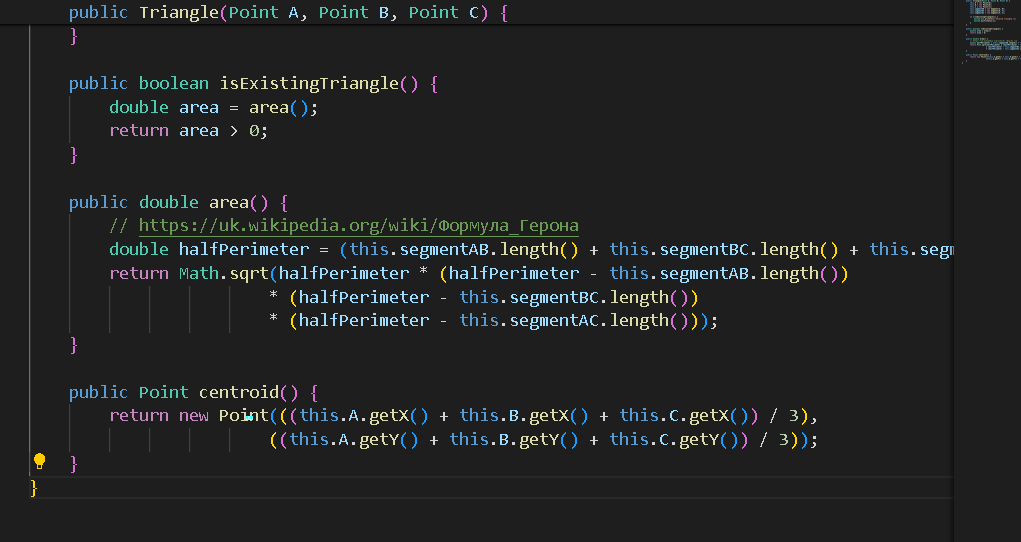
Реалізуйте метод Point centroid() – повертає центроїд трикутника.





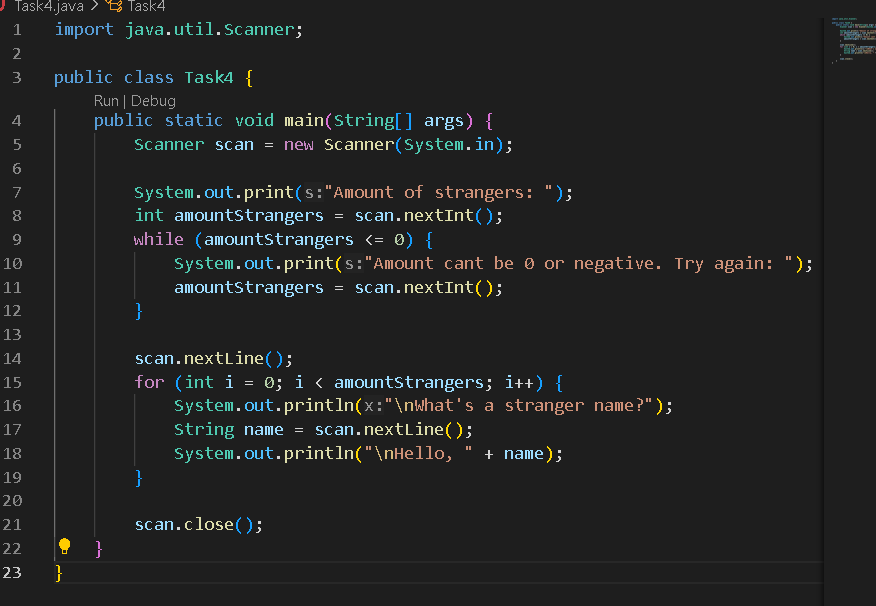


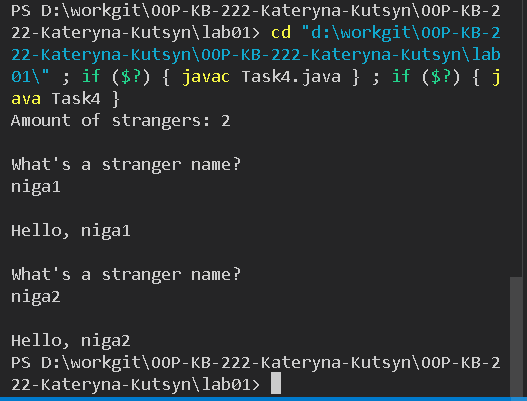




4 Напишіть просту програму, яка запитує кількість незнайомців, з якими потрібно зустрітися, рядково читає імена незнайомців, по рядках виводить: "Hello, (ім'я незнайомця)" для кожного незнайомця.

Доповнити програму перевірками на кількість, що є від’ємною, та рівна 0.



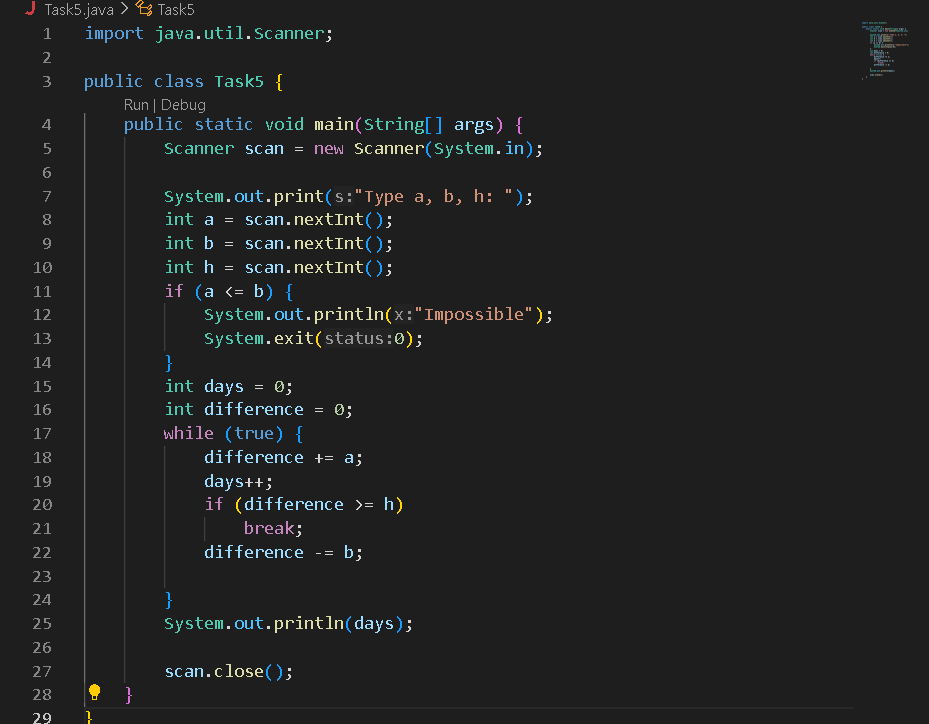


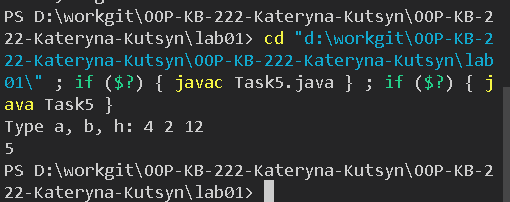
5 Наведено таку умову: равлик піднімається по дереву на a футів на день. Потім щоночі равлик сповзає вниз на b футів. Висота дерева — h футів.

Напишіть програму з підрахунком кількості днів, які знадобляться равлику, щоб дістатися до вершини дерева.

Програма читає a, b, h рядково. Вхідні значення гарантовано є додатніми цілими числами.

Якщо равлик не може дістатися до вершини дерева, має виводитися повідомлення: Impossible

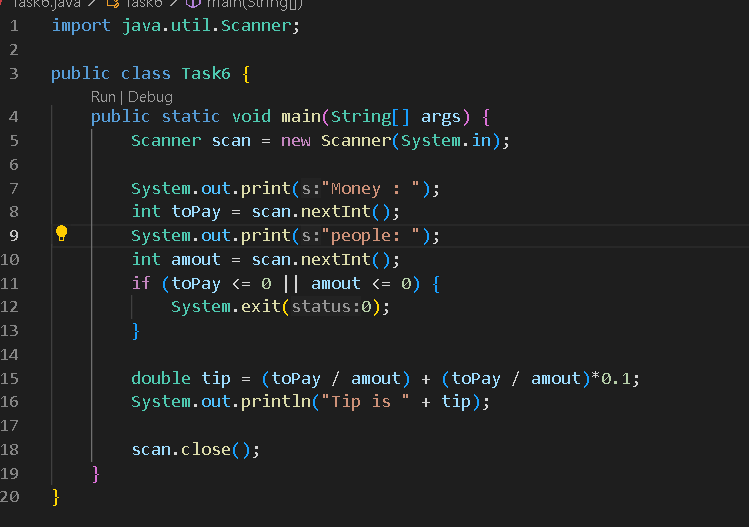


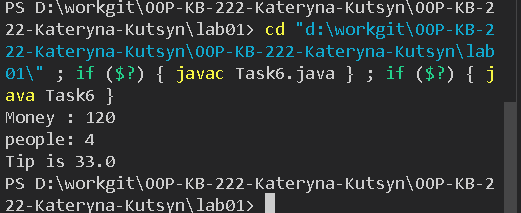


6 Дано така умова: компанія друзів відвідує ресторан. Вони вирішили розділити рахунок порівну та додати 10 відсотків від загальної суми рахунку як чайові. Далі друзі покривають рівними частинами загальну суму платежу.

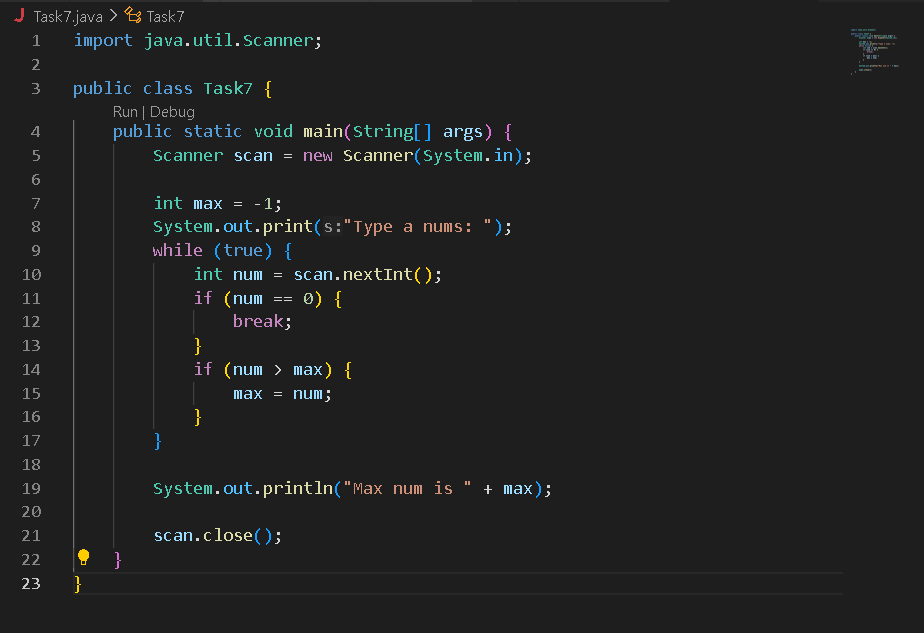
Напишіть програму, яка зчитує загальну суму рахунку та кількість друзів, а потім виводить розмір частини оплати.

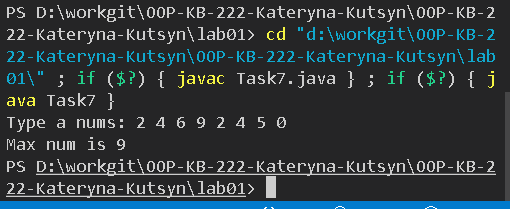
Доповнити програму перевірками на від’ємну суму рахунку та нульову кількість друзів.





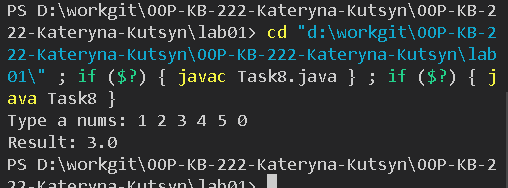
7 Напишіть програму, яка зчитує послідовність цілих значень зі стандартного введення і знаходить максимальне з них. Ви повинні зчитувати значення послідовності, доки наступне не стане 0. Нульове значення означає кінець вхідної послідовності та не є її елементом. Послідовність гарантовано містить хоча б одне значення.



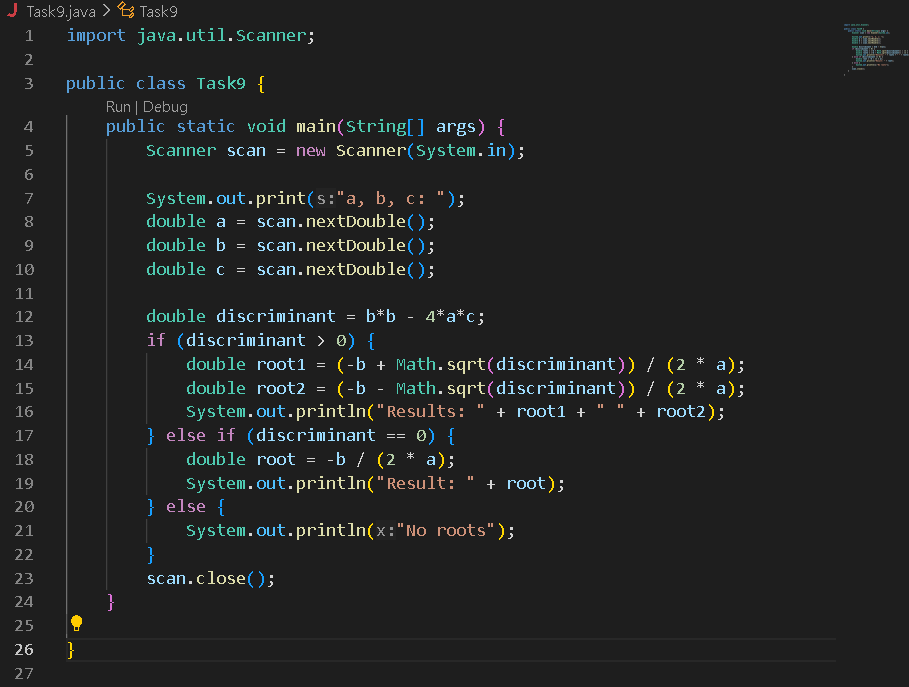


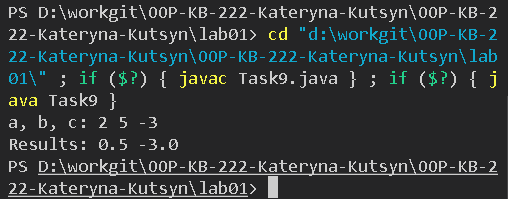
8 Напишіть програму, яка зчитує послідовність цілих значень зі стандартного введення та знаходить середнє значення. Ви повинні зчитувати значення послідовності, доки наступне не стане 0. Нульове значення означає кінець вхідної послідовності та не є її частиною. Послідовність гарантовано містить хоча б одне значення.



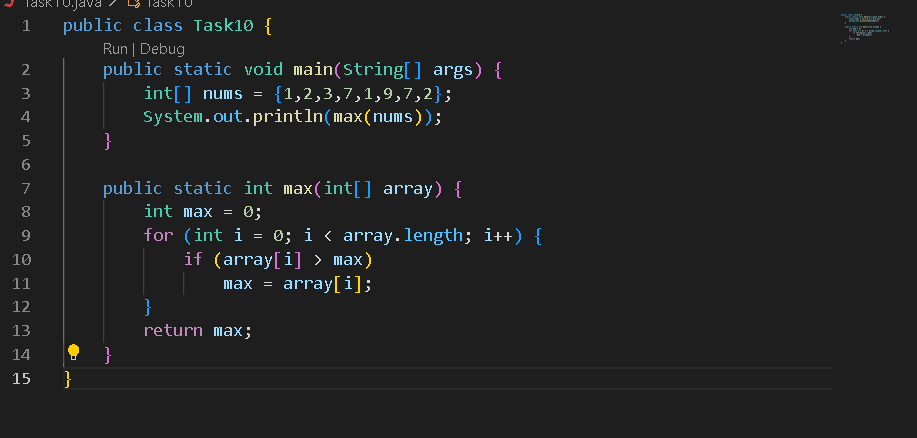


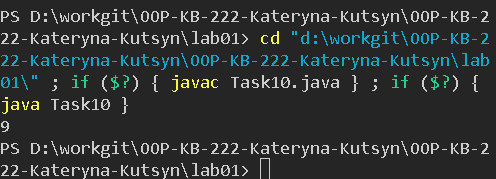
9 Напишіть програму для вирішення квадратних рівнянь. При виконанні врахуйте залежність значення дискримінанту до кількості результатів



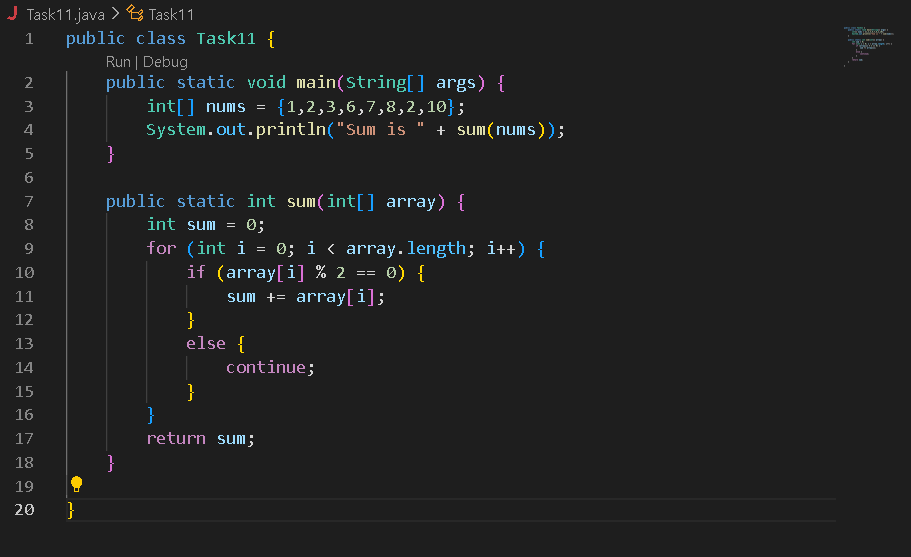
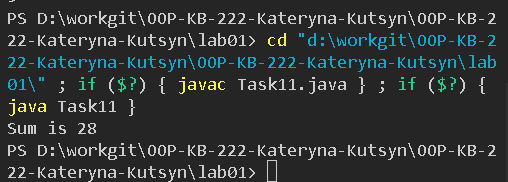


10 Напишіть програму, що містить метод max(…). Правильна реалізація має отримати масив значень типу int та повернути його максимальне значення. Вхідний масив гарантовано не буде порожнім або null. Метод max не повинен змінювати масив.





11 Напишіть програму, що містить метод sum. Правильна реалізація має отримати масив значень int та повернути суму парних чисел. Якщо цей масив дорівнює нулю або порожній, метод повертає 0. Метод sum не повинен змінювати масив.

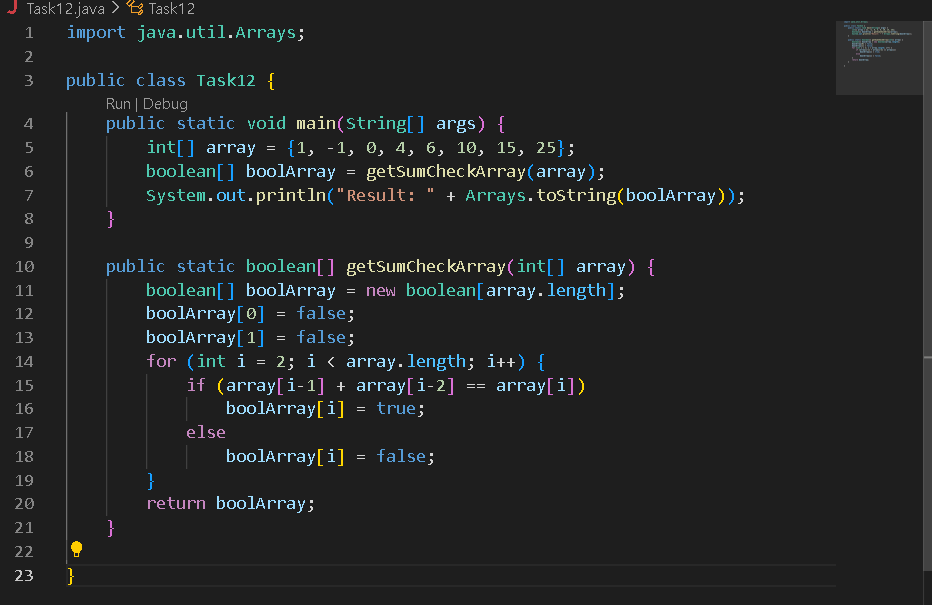
12 Напишіть програму, що містить метод getSumCheckArray. Правильна реалізація повинна отримати масив значень int і повернути масив логічних значень, де кожен елемент є результатом перевірки: чи є вихідний елемент сумою двох попередніх елементів у даному масиві.

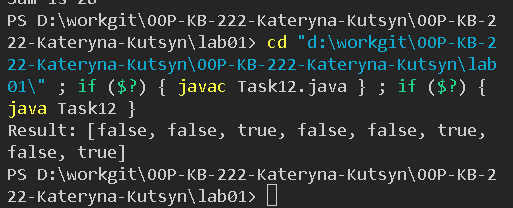
Гарантується, що довжина масиву становить 2 або більше.

Цей масив гарантовано не дорівнює null.

Метод повертає масив логічних значень, де кожен елемент є результатом для відповідного елемента в даному масиві.

Перші два елементи логічного масиву завжди false.

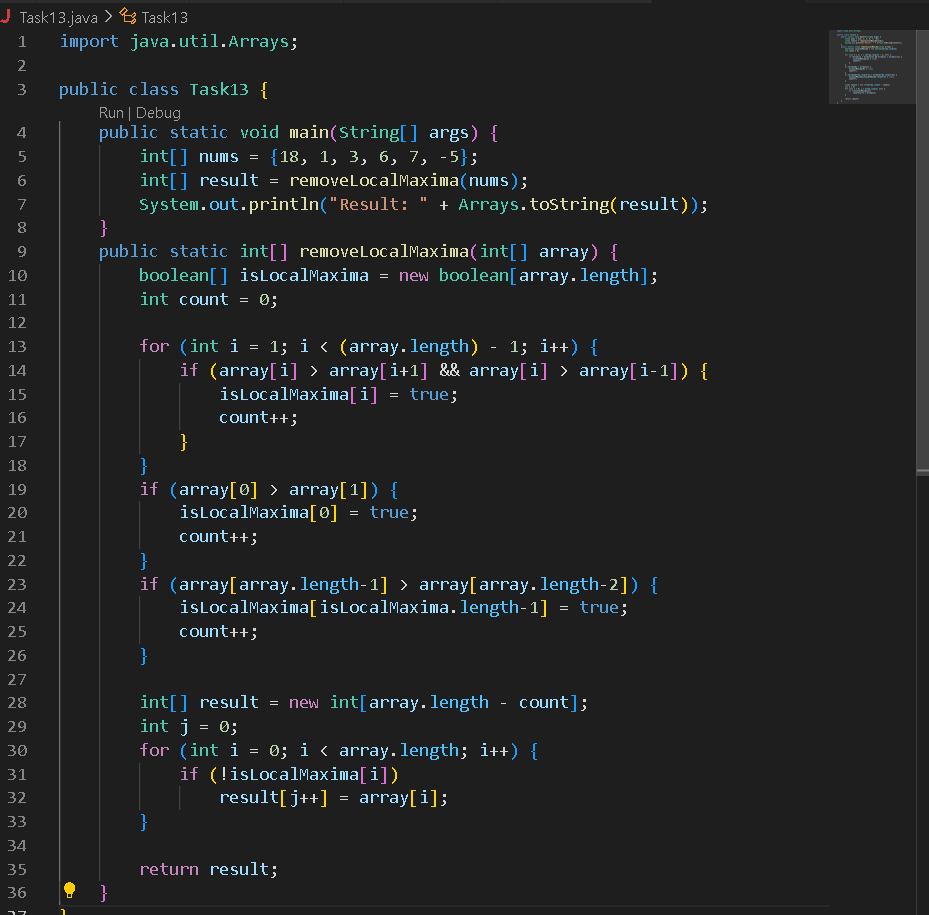


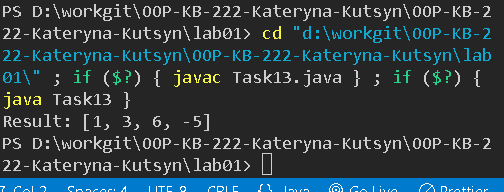


13 Напишіть програму, що містить метод removeLocalMaxima(…). Правильна реалізація має отримати масив значень типу int і повернути копію даного масиву з усіма видаленими локальними максимумами. Вихідний масив не можна змінювати.

**Локальний максимум** — це елемент, який більший за будь-який з його сусідніх елементів. Вам необхідно видалити елементи, які є локальними максимумами у вихідному масиві.

* Розмір цього масиву гарантовано більше 1.
* Розмір цього масиву гарантовано більше 1.
* Якщо масив не має локальних максимумів, ви повинні повернути його копію без змін.
* Ви можете використовувати методи java.util.Arrays.\*.





14 Напишіть програму що містить два методи, що реалізують наступний функціонал:

void cycleSwap(int[] array) - Зсуває всі елементи в даному масиві вправо на 1 позицію. В цьому випадку останній елемент масиву стає першим. Наприклад, 1 3 2 7 4 стає 4 1 3 2 7.

void cycleSwap(int[] array, int shift) - Зсуває всі елементи в заданому масиві вправо на shift позицій. Гарантується, що значення зсуву невід'ємне і не більше за довжину масиву. Наприклад, 1 3 2 7 4 зі зсувом 3 стає 2 7 4 1 3.

