

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ  
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Кафедра програмної інженерії та інтелектуальних технологій управління

Звіт з лабораторної роботи №2

З дисципліни “Поглиблений курс програмування Python”

Виконала:

студентка групи КН-221Д

Хорешко К.О.

Перевірила:

Доцент: Козуля М. М.

Харків 2023

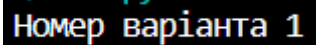
**Мета:** Отримати базові навички створення та роботи з класами на мові програмування Python.

**Примітка:**

Щоб вибрати номер варіанта, запустіть оператор

```
N = ord("К") % 3 + 1  
  
print("Номер варіанта", N)
```

Рисунок 1 - Лістинг до виконання коду для визначення варіанту



Номер варіанта 1

Рисунок 2 - Результат виконання коду

**Варіант 1. Створення класу для опрацювання геометричної прямої**

1) Створити клас Point и реалізувати наступні методи:

- Конструктор, що приймає x та y координати в якості параметрів;
- `__str__` метод для представлення об'єктів класу в якості

2) Створити клас Line для визначення геометричних прямих і реалізувати наступні методи:

- Конструктор, який приймає k та b коефіцієнти як параметри. Зверніть увагу, що прямі визначаються лінійними рівняннями:  $y = k * x + b$ .
- `__str__` метод для представлення об'єктів класу у вигляді рядку у формі  $y = k * x + b$
- метод `intersection(Line)`, що повинен повернути об'єкт класу

Point точки перетину двох прямих.

- Якщо лінії збігаються або не перетинаються, метод повинен повернути None.

### Лістинг для виконання завдання 1 та 2:

```
class Point:

    def __init__(self, x, y):

        self.x = x

        self.y = y

    def __str__(self):

        return "x = {}, y = {}".format(self.x, self.y)

class Line:

    def __init__(self, k, b):

        self.k = k

        self.b = b

    def __str__(self):

        return "y = {} * x + {}".format(self.k, self.b)

    def intersection(self, other):

        if self.b == other.b:

            print("прямі збігаються")

            return None
```

```
        elif self.k == other.k:

            print("лінії паралельні")

            return None

        else:

            x = (other.b - self.b) / (self.k - other.k)

            y = self.k * x + self.b

            return Point(x, y)
```

Рисунок 3 - Лістинг коду до завдання 1 та 2

### Сніпети:

```
print("Code snippet 1:")

a = Point(2,3)

print(a.x, a.y)

print(a)

print("")

print("Code snippet 2:")

line1 = Line(3,3)

line2 = Line(2,5)

print(line1)

print(line2)

print("")

print("Code snippet 3:")

line1 = Line(3,3)
```

```

line2 = Line(2,5)

print(line1.intersection(line2))

print("")

print("Code snippet 4:")

line1 = Line(2,3)

line2 = Line(2,5)

print(line1.intersection(line2))

print(line1)

print(line2)

```

Рисунок 4 - Лістинг до сніпетів

```

Code snippet 1:
2 3
x = 2, y = 3

Code snippet 2:
y = 3 * x + 3
y = 2 * x + 5

Code snippet 3:
x = 2.0, y = 9.0

Code snippet 4:
лінії паралельні
None
y = 2 * x + 3
y = 2 * x + 5

```

Рисунок 5 - Результат виконання коду

### **Висновок:**

Під час виконання лабораторної роботи, було отримано базові навички створення та роботи з класами на мові програмування Python шляхом вирішення математичної задачі.