Laboratory work 2
Working with classes in Python

Gain basic skill in creating and working with classes in the Python programming language.

Variant 1

Tokiya Hiruma
Group kh221-ibe

Working with classes in Python

Objective: gain basic skill in creating and working with classes in Python programming language.

1. Creating a title and calculating a variant.

Laboratory work 2

Tokiya Hiruma KH-221i6.e
Working with lasses in Python

入力 [1]:
```
print(f'Variant: {ord("T") % 3 + 1}')
```

```
Variant: 1
```

2. Task 1
    a. Create the class Point and implement the following methods:Constructor with x and y coordinates as parameters;
    b. __str__ method to represent the class objects as a string.

Task 1
Create the class Point and implement the following methods:
- Constructor with x and y coordinates as parameters;
- __str__ method to represents the class objects as a string.

入力 [9]:
```
class Point:
    def __init__(self, x, y):
        self.x = x
        self.y = y

    def __str__(self):
        return f'x = {self.x}; y = {self.y}'
```

3. Task 2
    a. Create the class Line and implement the following methods:
    b. constructor that takes k and b coefficients as parameters. Note that lines are defined by linear equations: y = k * x + b.
    c. __str__ method to represents the class objects as a string in the form y = k * x + b
    d. method intersection(Line), that must return a Point of intersection of two lines. If lines coincide or do not intersect, the method must return None.

入力 [15]:
```
class Line:
    def __init__(self, x, b):
        self.x = x
        self.b = b

    def __str__(self):
        return f'y = k * {self.x} + {self.b}'

    def intersection(self, line):
        if self.x == line.x:
            return None
        else:
            intersect_x = (line.b - self.b) / (self.x - line.x)
            intersect_y = (self.x * line.b - line.x * self.b) / (self.x - line.x)
        return Point(intersect_x, intersect_y)
```

4. Testing and checking function on different values.

入力 [9]:
```
class Point:
    def __init__(self, x, y):
        self.x = x
        self.y = y

    def __str__(self):
        return f'x = {self.x}; y = {self.y}'
```

入力 [24]:
```
p1 = Point(5, 12)
print(p1.x, p1.y)
print(p1)
```

```
5 12
x = 5; y = 12
```

入力 [25]:
```
p1 = Point(7, -10)
print(p1.x, p1.y)
print(p1)
```

```
7 -10
x = 7; y = -10
```

入力 [15]:
```
class Line:
    def __init__(self, x, b):
        self.x = x
        self.b = b

    def __str__(self):
        return f'y = k * {self.x} + {self.b}'

    def intersection(self, line):
        if self.x == line.x:
            return None
        else:
            intersect_x = (line.b - self.b) / (self.x - line.x)
            intersect_y = (self.x * line.b - line.x * self.b) / (self.x - line.x)
        return Point(intersect_x, intersect_y)
```

入力 [16]:
```
line1 = Line(3,3)
line2 = Line(2,5)
```

入力 [17]:
```
print(line1)
print(line2)
print(line1.intersection(line2))
```

```
y = k * 3 + 3
y = k * 2 + 5
x = 2.0; y = 9.0
```

入力 [18]:
```
line1 = Line(2,3)
line2 = Line(2,5)
```

入力 [19]:
```
print(line1)
print(line2)
print(line1.intersection(line2))
```

```
y = k * 2 + 3
y = k * 2 + 5
None
```

入力 [22]:
```
line1 = Line(-1,0)
line2 = Line(0,-1)
```

入力 [23]:
```
print(line1)
print(line2)
print(line1.intersection(line2))
```

```
y = k * -1 + 0
y = k * 0 + -1
```

Conclusion: after completing laboratory work I gained basic skills in creating and working with classes in Python programming language.