

**Міністерство освіти і науки України**  
**Національний університет «Запорізька політехніка»**

**Кафедра програмних засобів**

**Звіт**  
**з лабораторної роботи №1**  
**з дисципліни «Проектування, моделювання та забезпечення якості програмного**  
**забезпечення систем»**  
**«ОСНОВИ РОБОТИ З PYTHON»**

Виконав  
ст. групи КНТ-112м

Іващенко Р.І.

Прийняла  
професор кафедри  
програмних засобів,  
к.т.н., професор

Табунщик Г.В.

2022

## Мета роботи

Вивчити основні можливості мови програмування Python для розробки власних класів.

## Завдання

Створити клас для виконання операцій з комплексними числами. Передбачити операції: складання та віднімання; множення та ділення; обчислення модулю; консольне введення та виведення; ініціалізацію.

### 1. Текст програми

# Variant 7

```
class Complex:
```

```
    def __init__(self, real, imag):
```

```
        self.real = real
```

```
        self.imag = imag
```

```
    def __str__(self):
```

```
        sign = '+' if self.imag >= 0 else "
```

```
        return '{}{}{}'.format(self.real, sign, self.imag)
```

```
    __repr__ = __str__
```

```
class ComplexCalc:
```

```
    def add(self, c1, c2): # складання
```

```
        real = c1.real + c2.real
```

```
imag = c1.imag + c2.imag  
return Complex(real, imag)
```

```
def sub(self, c1, c2): # віднімання  
    real = c1.real - c2.real  
    imag = c1.imag - c2.imag  
    return Complex(real, imag)
```

```
def mul(self, c1, c2): # множення  
    real = c1.real * c2.real - c1.imag * c2.imag  
    imag = c1.imag * c2.real + c1.real * c2.imag  
    return Complex(real, imag)
```

```
def div(self, c1, c2): # ділення  
    real = (c1.real * c2.real + c1.imag * c2.imag) / (c2.real ** 2 + c2.imag ** 2)  
    imag = (c2.real * c1.imag - c1.real * c2.imag) / (c2.real ** 2 + c2.imag ** 2)  
    return Complex(real, imag)
```

```
def abs(self, c): # обчислення модулю  
    return (c.real ** 2 + c.imag ** 2) ** 0.5
```

```
calc = ComplexCalc()
```

```
print("Select operation.")  
print("1.Add")  
print("2.Substract")  
print("3.Multiply")
```

```
print("4.Divide")
print("5.Absolute")
#
while True:
    # введення чисел
    choice = input("Enter choice (1, 2, 3, 4, 5):")

    if choice in ('1', '2', '3', '4', '5'):
        num1 = float(input("Введіть дійсне число першого виразу:"))
        num2 = float(input("Введіть уявне число першого виразу:"))
        num3 = float(input("Введіть дійсне число другого виразу:"))
        num4 = float(input("Введіть уявне число другого виразу:"))

        if choice == '1':
            print(calc.add(Complex(num1, num2), Complex(num3, num4)))

        elif choice == '2':
            print(calc.sub(Complex(num1, num2), Complex(num3, num4)))

        elif choice == '3':
            print(calc.mul(Complex(num1, num2), Complex(num3, num4)))

        elif choice == '4':
            print(calc.div(Complex(num1, num2), Complex(num3, num4)))

        elif choice == '5':
            print('перший вираз - ', calc.abs(Complex(num1, num2)))
            print('другий вираз - ', calc.abs(Complex(num3, num4)))
```

```

s = 'n'
answer = input("Продовжити обчислення? у or n:\n")
if answer in s:
    print('вихід')
    quit()
else:
    continue

if __name__ != "__main__":
    main()

```

## 2. Результат роботи

Варіанти виконання роботи у консолі:

1) Select operation.

1.Add

2.Subtract

3.Multiply

4.Divide

5.Absolute

Enter choice (1, 2, 3, 4, 5):2

Введіть дійсне число першого виразу:5

Введіть уявне число першого виразу:6

Введіть дійсне число другого виразу:7

Введіть уявне число другого виразу:8

-2.0-2.0i

Продовжити обчислення? у or n:

у

Enter choice (1, 2, 3, 4, 5):

2) Select operation.

1.Add

2.Subtract

3.Multiply

4.Divide

5.Absolute

Enter choice (1, 2, 3, 4, 5):1

Введіть дійсне число першого виразу:2

Введіть уявне число першого виразу:3

Введіть дійсне число другого виразу:4

Введіть уявне число другого виразу:5

6.0+8.0i

Продовжити обчислення? y or n:

n

вихід

Process finished with exit code 0

### **3. Висновки**

В цій лабораторній роботі були розглянуті та вивчені основні можливості мови програмування Python для розробки власних класів.

### **4. Контрольні запитання**

1. Чим відрізняються поняття класу у C++ та Python? Дайте загальне визначення класу.

Поняття класу у C++ та Python нічим не відрізняються.

Клас – це проект об'єкта. Сам по собі клас нічого не робить і не означає, він

лише описує як щось зробити чи створити.

2. Наведіть загальну структуру класу.

```
class class_name:
    variable = value
    ...
    def method_name (self, ...):
        self.class_variable = value
    ...
    ...
```

3. Що таке інтерфейс класу та його реалізація?

Інтерфейсом класу називають опис найменувань і типів, його властивостей і методів. Знаючи інтерфейс можна отримати ту функціональність, яка закладена в класі.

Реалізація класу складається з програмного коду методів.

4. У чому полягає роль конструкторів та деструкторів у класі?

Роль конструктора — встановити початковий стан об'єкта шляхом ініціалізації атрибутів об'єкта та визначити інваріант класу.

Роль деструктора — знищення екземпляра об'єкта і повернення виділених ресурсів ОС.

5. Які способи можна застосувати для ініціалізації об'єкта класу? Як для цього застосовуються конструктори?

Кожен клас повинен мати конструктор, навіть якщо не створюється конструктор явно, він сам створить конструктор за замовчуванням.

Конструктор у Python - це спеціальний метод, який використовується для ініціалізації членів класу під час виконання та під час створення об'єкта.

У Python кілька вбудованих методів класів, які починаються з подвійного підкреслення (`__`), і вони мають особливе значення в Python.

Ім'я конструктора завжди буде `__init__()`.

Метод `__init__` викликається щоразу під час створення нового екземпляра класу (об'єкта). Екземпляр класу передається як аргумент параметру `self`, а значення аргументів – параметрам методу.

6. Чим відрізняються функції – члени класу від функцій, не пов'язаних з будь-яким класом?

Функція яка використовується в класі називається методом, та може використовувати лише один аргумент `self`.

7. Що означає об'єкт класу?

Об'єкт класу це програмна сутність, яка містить дані та процедури.

8. Назвіть основні принципи керування пам'яттю у Python.

а) для керування пам'яттю використовується спеціальний механізм, який називається аллокатор і необхідний для оптимізації кількості системних визовів. Аллокатор складається з:

- Арена – великий безперервний шматок пам'яті (зазвичай 256 кілобайт), що містить кілька сторінок віртуальної пам'яті операційної системи.

- Пул - одна сторінка віртуальної пам'яті (зазвичай 4 кілобайти).

- Блок - маленький шматочок пам'яті, який використовується для зберігання одного об'єкта.

б) керування об'єктами відбувається за допомогою системи лічильників посилок (`reference count`), і пам'ять вивільняється, коли лічильник падає до нуля.

в) механізм очищення пам'яті - це збирач сміття (`garbage collector`), заснований на ідеї поколінь.



## Список бібліографічних посилань

1. Лекції з дисципліни «Проектування, моделювання та забезпечення якості програмного забезпечення систем». Викладач Табунщик Г.В.