

Міністерство освіти і науки України  
Національний технічний університет України «Київський  
політехнічний інститут імені Ігоря Сікорського»  
Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Звіт

з лабораторної роботи № 4  
з дисципліни «Основи програмування –  
2. Методології програмування»

«Успадкування та поліморфізм»

Варіант 11

Виконав студент ПІ-13 Дем'янчук Олександр Петрович

(шифр, прізвище, ім'я, по батькові)

Перевірив Вечерковська Анастасія Сергіївна

(прізвище, ім'я, по батькові)

Київ 2022

## Лабораторна робота 4

### Варіант 11

**Мета роботи** - вивчити механізми створення класів та об'єктів.

**Завдання:** Спроекувати клас "Число", який представляє число і містить методи збільшення / зменшення числа у вказану кількість разів та знаходження модуля числа. На основі цього класу створити класи-нащадки "Раціональне число" та "Комплексне число". Створити m об'єктів раціональних чисел та n об'єктів комплексних чисел (дані згенерувати випадковим чином). Раціональні числа зменшити у 2 рази, а комплексні — збільшити у 3 рази. Знайти суму модулів усіх чисел.

### Код програми

**C++**

#### main.cpp

```
#include "func.h"

int main() {
    int m = input();
    int n = input();
    Rational* arrR = RatArray(m);
    Complex* arrC = CompArray(n);
    std::cout << m << " rational numbers: \n";
    displayRat(arrR, m);
    std::cout << n << " complex numbers: \n";
    displayComp(arrC, n);
    std::cout << "rational divided by 2: \n";
    changeRat(arrR, m);
    displayRat(arrR, m);
    std::cout << "complex multiplied by 3: \n";
    changeComp(arrC, n);
    displayComp(arrC, n);
    std::cout << "the sum of abs equals " << findSum(arrR, m, arrC, n);
    delete[] arrR;
    delete[] arrC;
}
```

#### func.h

```
#pragma once
#include <iostream>
#include <cmath>
#include <vector>

class Number {
public:
    Number();
    virtual void multiply(double times) = 0;
    virtual void divide(double times) = 0;
    virtual double getReal() = 0;
    virtual double findAbs() = 0;
};

class Rational : public Number {
private:
    double real;
```

```

public:
    Rational();
    Rational(double Re);
    void multiply(double times);
    void divide(double times);
    double getReal();
    double findAbs();
};

class Complex : public Number {
private:
    double real;
    double imaginary;
public:
    Complex();
    Complex(double Re, double Im);
    void multiply(double times);
    void divide(double times);
    double getReal();
    double getImag();
    double findAbs();
};

int input();

Rational* RatArray(int size);
Complex* CompArray(int size);

void displayRat(Rational* Array, int size);
void displayComp(Complex* Array, int size);

void changeRat(Rational* Array, int size);
void changeComp(Complex* Array, int size);

double findSum(Rational* Arr1, int m, Complex* Arr2, int n);

```

## func.cpp

```

#include "func.h"

Number::Number() {}

Number::Number(double Re, double Im) : real(Re), imaginary(Im)
{ }

Rational::Rational() {}

Rational::Rational(double Re) {
    real = Re;
}

void Rational::multiply(double times) {
    real *= times;
}

void Rational::divide(double times) {
    real /= times;
}

double Rational::getReal() {
    return real;
}

double Rational::findAbs() {

```

```

        return sqrt(real * real);
    }

Complex::Complex() {}

Complex::Complex(double Re, double Im){
    real = Re;
    imaginary = Im;
}

void Complex::multiply(double times) {
    real *= times;
    imaginary *= times;
}

void Complex::divide(double times) {
    real /= times;
    imaginary /= times;
}

double Complex::getReal() {
    return real;
}

double Complex::getImag() {
    return imaginary;
}

double Complex::findAbs() {
    return sqrt(real * real + imaginary*imaginary);
}

int input() {
    int num = 0;
    do {
        std::cout << "enter the size: "; std::cin >> num;
    } while (num <= 0);
    return num;
}

Rational* RatArray(int size) {
    Rational* Array = new Rational[size];
    double min = -100.0;
    double max = 100.0;
    for (size_t i = 0; i < size; i++) {
        double Re = (double)(rand()) / RAND_MAX * (max - min) + min;
        Rational R(Re);
        Array[i] = R;
    }
    return Array;
}

Complex* CompArray(int size) {
    Complex* Array = new Complex[size];
    double min = -100.0;
    double max = 100.0;
    for (int i = 0; i < size; i++) {
        double Re = (double)(rand()) / RAND_MAX * (max - min) + min;
        double Im = (double)(rand()) / RAND_MAX * (max - min) + min;
        Complex C(Re, Im);
        Array[i] = C;
    }
    return Array;
}

```

```

void displayRat(Rational* Array, int size) {
    for (int i = 0; i < size; i++)
    {
        printf("%d) %3.3f \n", i+1, Array[i].getReal());
    }
    std::cout << std::endl;
}

void displayComp(Complex* Array, int size) {
    for (int i = 0; i < size; i++)
    {
        printf("%d) %3.3f + %3.3f*i \n", i+1,
Array[i].getReal(), Array[i].getImag());
    }
    std::cout << std::endl;
}

void changeRat(Rational* Array, int size) {
    for (size_t i = 0; i < size; i++)
    {
        Array[i].divide(2);
    }
}

void changeComp(Complex* Array, int size) {
    for (size_t i = 0; i < size; i++)
    {
        Array[i].multiply(3);
    }
}

double findSum(Rational* Arr1, int m, Complex* Arr2, int n) {
    double sum = 0;
    for (size_t i = 0; i < m; i++)
    {
        sum += Arr1[i].findAbs();
    }
    for (size_t i = 0; i < n; i++)
    {
        sum += Arr2[i].findAbs();
    }
    return sum;
}

```

## Python

### func.py

```

import random
import math

class Number:
    def multiply(self, times):
        pass
    def divide(self, times):
        pass
    def findAbs(self):
        pass

class Rational(Number):
    def __init__(self, Re):
        self.real = Re
    def multiply(self, times):
        self.real *= times
    def divide(self, times):
        self.real /= times

```

```

    def findAbs(self):
        return math.sqrt(math.pow(self.real, 2))

class Complex(Number):
    def __init__(self, Re, Im):
        self.real = Re
        self.imaginary = Im
    def multiply(self, times):
        self.real *= times
        self.imaginary *= times
    def divide(self, times):
        self.real /= times
        self.imaginary /= times
    def findAbs(self):
        return math.sqrt(math.pow(self.real, 2) + math.pow(self.imaginary, 2))

def RatList(size):
    return [Rational(round(random.uniform(-100, 100), 3)) for i in range(size)]

def displayRat(Rat):
    for i in range(len(Rat)):
        x = round(Rat[i].real, 3)
        print(f"{i+1} {x}")

def CompList(size):
    return [Complex(round(random.uniform(-100, 100), 3),
                    round(random.uniform(-100, 100), 3)) for i in range(size)]

def displayComp(Comp):
    for i in range(len(Comp)):
        x = round(Comp[i].real, 3)
        y = round(Comp[i].imaginary, 3)
        print(f"{i+1} {x} + {y}*i")

def changeRat(Rat):
    for i in range(len(Rat)):
        Rat[i].divide(2)

def changeComp(Comp):
    for i in range(len(Comp)):
        Comp[i].multiply(3)

def findSum(Rat, Comp):
    sum = 0
    for i in range(len(Rat)):
        sum += Rat[i].findAbs()
    for i in range(len(Comp)):
        sum += Comp[i].findAbs()
    return sum

```

## lab4py.py

```

from func import*

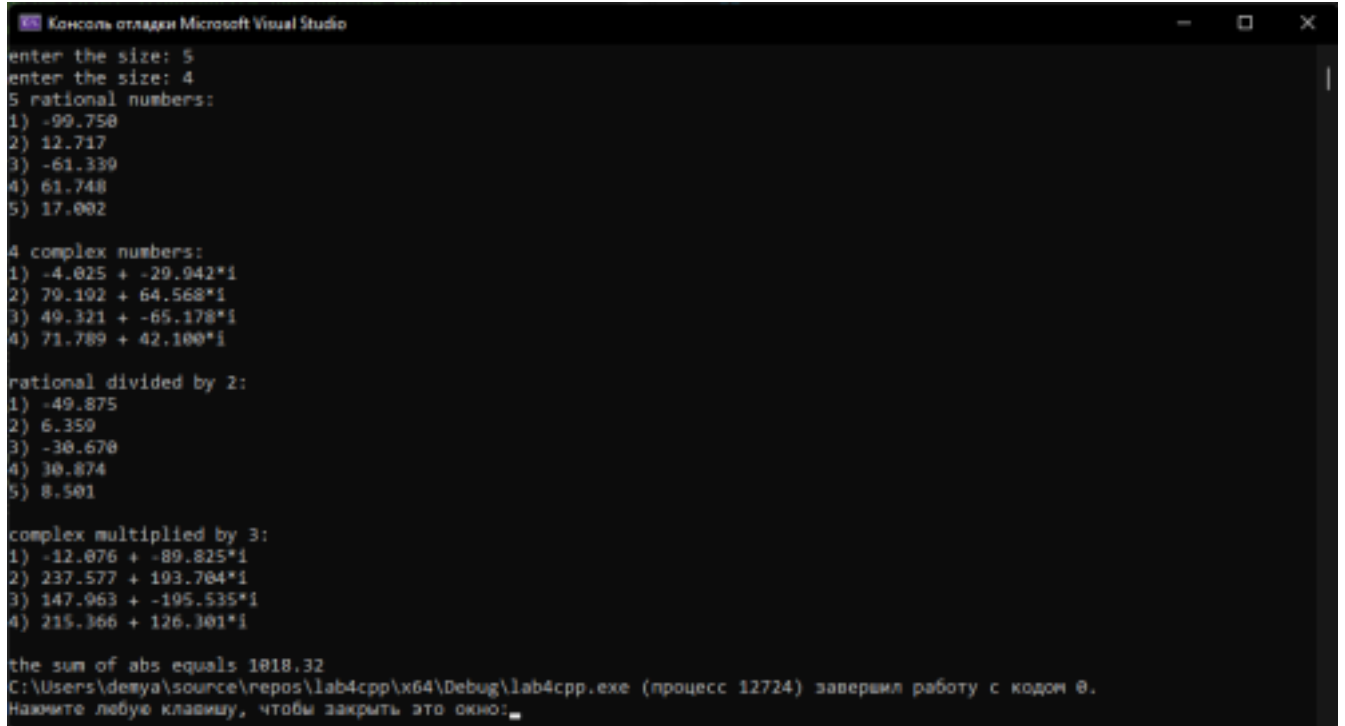
m = int(input("enter size: "))
n = int(input("enter size: "))
arrR = RatList(m);
arrC = CompList(n);
print(m, "rational numbers: ")
displayRat(arrR);
print(n, "complex numbers: ")
displayComp(arrC);
print("rational divided by 2:")
changeRat(arrR)
displayRat(arrR)

```

```
print("complex multiplied by 3: ")
changeComp(arrC)
displayComp(arrC)
print("the sum of abs equals", findSum(arrR, arrC))
```

## Тестування

### C++



```
Консоль отладки Microsoft Visual Studio
enter the size: 5
enter the size: 4
5 rational numbers:
1) -99.750
2) 12.717
3) -61.339
4) 61.748
5) 17.002

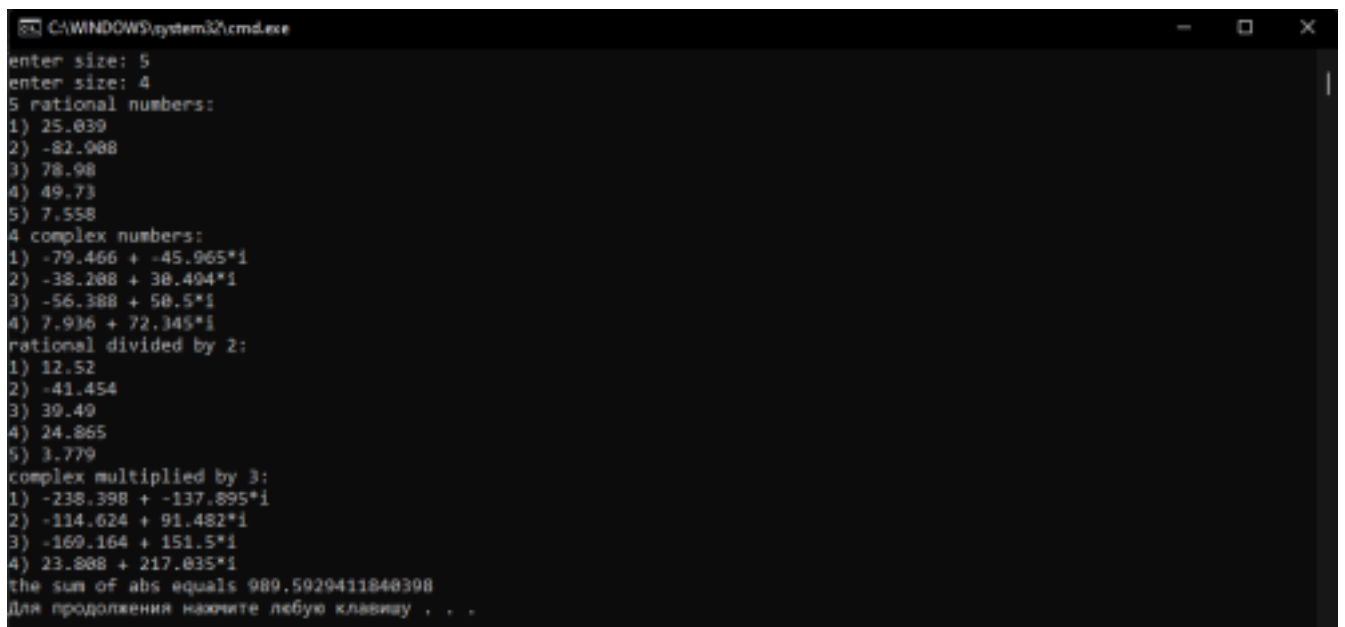
4 complex numbers:
1) -4.025 + -29.942*i
2) 79.192 + 64.568*i
3) 49.321 + -65.178*i
4) 71.789 + 42.100*i

rational divided by 2:
1) -49.875
2) 6.359
3) -30.670
4) 30.874
5) 8.501

complex multiplied by 3:
1) -12.076 + -89.825*i
2) 237.577 + 193.704*i
3) 147.963 + -195.535*i
4) 215.366 + 126.301*i

the sum of abs equals 1018.32
C:\Users\demya\source\repos\lab4cpp\x64\Debug\lab4cpp.exe (процесс 12724) завершил работу с кодом 0.
Нажмите любую клавишу, чтобы закрыть это окно:
```

### Python



```
C:\WINDOWS\system32\cmd.exe
enter size: 5
enter size: 4
5 rational numbers:
1) 25.039
2) -82.908
3) 78.98
4) 49.73
5) 7.558

4 complex numbers:
1) -79.466 + -45.965*i
2) -38.208 + 30.494*i
3) -56.388 + 50.5*i
4) 7.936 + 72.345*i

rational divided by 2:
1) 12.52
2) -41.454
3) 39.49
4) 24.865
5) 3.779

complex multiplied by 3:
1) -238.398 + -137.895*i
2) -114.624 + 91.482*i
3) -169.164 + 151.5*i
4) 23.808 + 217.035*i

the sum of abs equals 989.5929411840398
Для продолжения нажмите любую клавишу . . .
```

### Висновки:

На роботі вивчив і використав на практиці механізми створення і використання класів та механізмів, на практиці використав принципи наслідування та поліморфізму.