

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 № 3 з дисципліни
«Основи програмування 2. Модульне програмування»

«Перевантаження операторів»

Варіант 15

Виконав студент ІП-13 Карам'ян Варта́н Су́ренович
(шифр, прізвище, ім'я, по батькові)

Перевірів Вечерковська Анастасія Сергіївна
(прізвище, ім'я, по батькові)

Лабораторна робота 3

Перевантаження операторів

Варіант 15

Визначити клас “Коло”, членами якого є радіус кола та координати його центру. Реалізувати для нього декілька конструкторів, геттери, метод обчислення довжини кола. Перевантажити оператори: префіксий “++”/ постфіксий “++” – для інкрементування x-координати і y-координати центру кола відповідно, “*” – для збільшення радіусу кола у вказану кількість разів (ціле число). Створити три кола (C1, C2, C3), використовуючи різні конструктори. Інкрементувати x-координату кола C1 і y-координату кола C2. Збільшити радіус кола C3 у 3 рази. Серед кіл C1, C2, C3 визначити коло найбільшої довжини.

Код програми на мові C++:

lab3 C++.cpp

```
#include "Circle.h"

int main()
{
    Circle C1(12, -29, 3), C2(12, 45), C3(7);
    cout << "=====\n";
    C1.print();
    C2.print();
    C3.print();
    cout << "=====\n";

    ++C1;
    C2++;
    C3 = C3 * 3;

    C1.print();
    C2.print();
    C3.print();
    cout << "=====\n";

    vector<Circle> circles{C1, C2, C3};
    Circle maxCircle = maxCircleLength(circles);
    cout << "Circle with a max length:\n";
    maxCircle.print();
}
```

Circle.h

```
#pragma once
#define _USE_MATH_DEFINES
#include <iostream>
#include <math.h>
#include <iomanip>
#include <vector>

using namespace std;

class Circle
{
    double x;
    double y;
    double radius;
public:
    Circle(double, double, double);

    Circle(double, double);

    Circle(double = 1);

    Circle(const Circle&);

    double getX();

    double getY();

    double getRadius();

    double circleLength();

    void print();

    Circle operator++();

    Circle operator++(int);

    Circle operator*(int);
};

Circle maxCircleLength(vector <Circle>);
```

Circle.cpp

```
#include "Circle.h"

Circle::Circle(double x, double y, double r)
{
    this->x = x;
    this->y = y;
    this->radius = r;
}

Circle::Circle(double x, double y)
{
    this->x = x;
    this->y = y;
    this->radius = 1;
}
```

```

}

Circle::Circle(double r)
{
    this->x = 0;
    this->y = 0;
    this->radius = r;
}

Circle::Circle(const Circle& obj)
{
    this->x = obj.x;
    this->y = obj.y;
    this->radius = obj.radius;
}

double Circle::getX()
{
    return x;
}
double Circle::getY()
{
    return y;
}
double Circle::getRadius()
{
    return radius;
}

double Circle::circleLength()
{
    double temp = 2 * radius * M_PI;
    return temp;
}

void Circle::print()
{
    cout << ">> Circle <<\n";
    cout << "Coordinates of center:\tx = " << x << "\ty = " << y;
    cout << "\nRadius = " << radius;
    cout << "\nCircle length: " << setprecision(4) << circleLength() << "\n\n";
}

Circle Circle::operator++()
{
    ++this->x;
    return *this;
}

Circle Circle::operator++(int unused)
{
    ++this->y;
    return *this;
}

Circle Circle::operator*(int num)
{
    this->radius *= num;
    return *this;
}

Circle maxCircleLength(vector <Circle> circles)
{
    Circle maxCircle = circles[0];
    double maxLength = circles[0].circleLength();
    for (size_t i = 1; i < circles.size(); i++)

```

```

    {
        if (circles[i].circleLength() > maxLength)
        {
            maxCircle = circles[i];
            maxLength = circles[i].circleLength();
        }
    }
    return maxCircle;
}

```

Результати роботи програми:

```

=====
>> Circle <<
Coordinates of center: x = 12 y = -29
Radius = 3
Circle length: 18.85

>> Circle <<
Coordinates of center: x = 12 y = 45
Radius = 1
Circle length: 6.283

>> Circle <<
Coordinates of center: x = 0 y = 0
Radius = 7
Circle length: 43.98

=====
>> Circle <<
Coordinates of center: x = 13 y = -29
Radius = 3
Circle length: 18.85

>> Circle <<
Coordinates of center: x = 12 y = 46
Radius = 1
Circle length: 6.283

>> Circle <<
Coordinates of center: x = 0 y = 0
Radius = 21
Circle length: 131.9
=====

Circle length: 131.9

=====
Circle with a max length:
>> Circle <<
Coordinates of center: x = 0 y = 0
Radius = 21
Circle length: 131.9

D:\Projects\lab3 C++\x64\Debug\lab3 C++.exe (процесс 16108) завершил работу с кодом 0.
Нажмите любую клавишу, чтобы закрыть это окно:

```

Висновок:

У результаті виконання лабораторної роботи ми познайомились з таким механізмом ООП як перевантаження операторів.