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 №1 з дисципліни

«Основи програмування»

Варіант 2

Виконав студент Боровков Іван Ігорович

( прізвище, ім'я, по батькові)

Перевірив викладач Вітковська Ірина Іванівна

( прізвище, ім'я, по батькові)

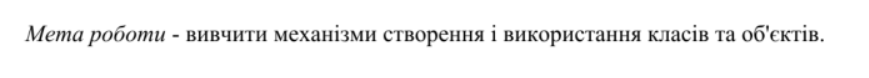
Київ 2022

Лабораторна робота №2

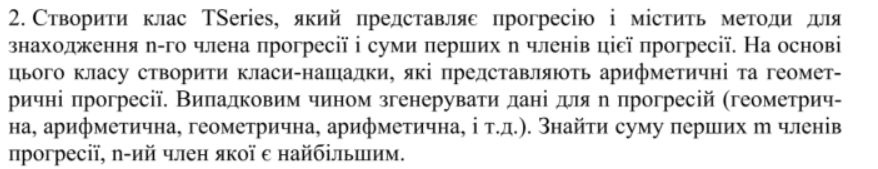
[УСПАДКУВАННЯ ТА ПОЛІМОРФІЗМ](http://ipi.kpi.ua/study/ii-semestr/laboratorni-roboti/laboratorna-robota-4-uspadkuvannya-ta-pooimorfizmn/)

Лабораторна робота №5

[УСПАДКУВАННЯ ТА ПОЛІМОРФІЗМ](http://ipi.kpi.ua/study/ii-semestr/laboratorni-roboti/laboratorna-robota-4-uspadkuvannya-ta-pooimorfizmn/)



Умова задачі:



Код задачі:

Series.hpp

#pragma once

class TSeries

{

protected:

    TSeries(int firstMember, int step);

    ~TSeries();

    int firstMember;

    int step;

public:

    void SetFirstMember(int value);

    void SetStep(int value);

    virtual int FindMember(int memberNum) = 0;

    virtual int FindSum(int num) = 0;

};

Series.cpp

#include "Series.hpp"

void TSeries::SetFirstMember(int value)

{

    firstMember = value;

}

void TSeries::SetStep(int value)

{

    step = value;

}

TSeries::TSeries(int firstMember, int step)

{

    this->firstMember = firstMember;

    this->step = step;

}

TSeries::~TSeries()

{

}

ArithmeticSeries.hpp

#pragma once

#include "Series.hpp"

class ArithmeticSeries: protected TSeries

{

    public:

        ArithmeticSeries(int firstMember, int step);

        int FindMember(int memberNum);

        int FindSum(int num);

};

ArithmeticSeries.cpp

#include "ArithmeticSeries.hpp"

ArithmeticSeries::ArithmeticSeries(int firstMember, int step)

    :TSeries(firstMember, step)

{

}

int ArithmeticSeries::FindMember(int memberNum)

{

    return firstMember + step \* (memberNum - 1);

}

int ArithmeticSeries::FindSum(int num)

{

    return (num/2) \* (2\* firstMember + (num - 1) \* step);

}

GeometricSeries.hpp

#pragma once

#include "Series.hpp"

class GeometricSeries: protected TSeries

{

    public:

        int FindMember(int memberNum);

        int FindSum(int num);

        GeometricSeries(int firstMember, int step);

};

GeometricSeries.cpp

#include "GeometricSeries.hpp"

#include <cmath>

GeometricSeries::GeometricSeries(int firstMember, int step)

    :TSeries(firstMember, step)

{

}

int GeometricSeries::FindMember(int memberNum)

{

    return firstMember \* pow(step, memberNum - 1);

}

int GeometricSeries::FindSum(int num)

{

    return firstMember \* (pow(step, num) - 1) / (step - 1);

}

main.cpp

#include <iostream>

#include "GeometricSeries.hpp"

#include "ArithmeticSeries.hpp"

using namespace std;

int main(int argc, char const \*argv[])

{

    int n, m;

    cout << "Enter n and m:\n";

    cin >> n >> m;

    cout<<'\n';

    int max\_n = -1, max\_sum, max\_i;

    for (size\_t i = 0; i < n; i++)

    {

        int firstMember = rand() % (10 - 1 + 1) + 1;

        int step = rand() % (10 - 2 + 1) + 2;

        cout << i + 1 << "th series first member is: " << firstMember << " and s tep is: " << step <<'\n';

        if (i % 2 == 0)

        {

            GeometricSeries geometricSeries(firstMember, step);

            cout << "Nth member is: " << geometricSeries.FindMember(n) << " and Mth sum is: " << geometricSeries.FindSum(m) << '\n';

            if (geometricSeries.FindMember(n) > max\_n)

            {

                max\_i = i;

                max\_n = geometricSeries.FindMember(n);

                max\_sum = geometricSeries.FindSum(m);

            }

        }

        else

        {

            ArithmeticSeries arithmeticSeries(firstMember, step);

            cout << "Nth member is: " << arithmeticSeries.FindMember(n) << " and Mth sum is: " << arithmeticSeries.FindSum(m) << '\n';

            if (arithmeticSeries.FindMember(n) > max\_n)

            {

                max\_i = i;

                max\_n = arithmeticSeries.FindMember(n);

                max\_sum = arithmeticSeries.FindSum(m);

            }

        }

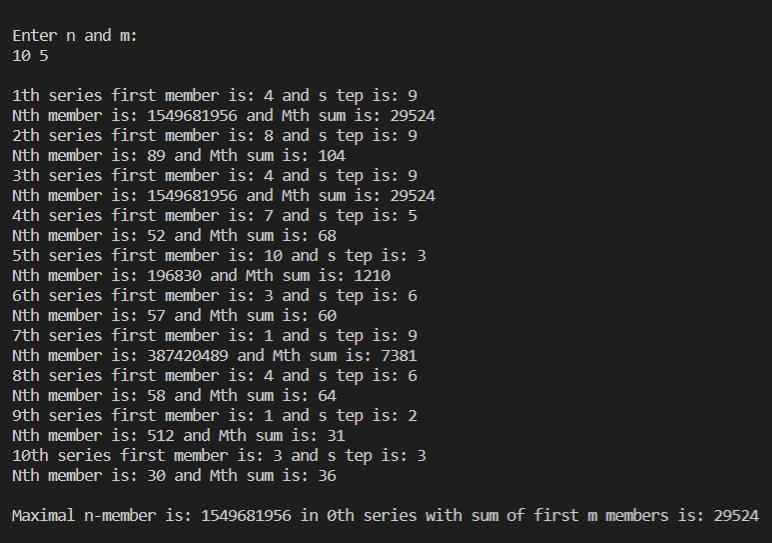
    }

    cout << "\nMaximal n-member is: " << max\_n << " in " << max\_i << "th series with sum of first m members is: " << max\_sum <<"\n\n";

    return 0;

}

Скріншоти виконання програми:



*Висновок*

В ході лабораторної роботи я вивчив механізми створення класів та об’єктів.