

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»

«Класи та об'єкти»

Варіант 14

Виконав студент ІП-13, Замковий Дмитро Володимирович

(шифр, прізвище, ім'я, по батькові)

Перевірів Вечерковська Анастасія Сергіївна

(прізвище, ім'я, по батькові)

Лабораторна робота 3

Класи та об'єкти

Мета заняття: вивчити механізм створення і використання класів та об'єктів

Завдання: Розробити клас «Геометрична прогресія», який задається першим членом a_0 , знаменником q та кількістю членів n . Створити масив об'єктів даного класу. Визначити прогресію (її номер), що має найбільший останній член

Код програми:

Програма реалізована на мові програмування C++

Lab3.cpp

```
#include "Geometric_progression.h"
#include "help_modul.h"

using namespace std;

int main()
{
    vector<Geometric_progression> progressions;
    int leng, num;

    cout << "Enter how many progressions you want to specify: ";
    cin >> leng;

    progressions = input_geo_progressions(leng);
    num = find_max(progressions);

    for (int i = 0; i < leng; i++)
    {
        progressions[i].output_str(i + 1);
    }
    cout << endl << endl << "Geometric progression with largest last term: " << num + 1 <<
endl;
}
```

Geometric_progression.h

```
#pragma once
#include <string>
#include <iostream>
using namespace std;

class Geometric_progression
{
private:
    double a;
    double q;
    unsigned int n;

public:
    Geometric_progression(double set_a, double set_q, int set_n) : a(set_a), q(set_q),
n(set_n) {};
    double get_a();
```

```
double get_q();
unsigned int get_n();
void output_str(int);
};
```

help_modul.h

```
#pragma once
#include <iostream>
#include "Geometric_progression.h"
#include <vector>
using namespace std;

vector<Geometric_progression> input_geo_progressions(int length);
int find_max(vector<Geometric_progression> progressions);
double power(double a, int n);
```

Geometric_progression.cpp

```
#include "Geometric_progression.h"

double Geometric_progression::get_a()
{
    return a;
}

double Geometric_progression::get_q()
{
    return q;
}

unsigned int Geometric_progression::get_n()
{
    return n;
}

void Geometric_progression::output_str(int num)
{
    cout << endl << endl << "Progression " << num << " :";
    cout << endl << "\tFirst member: " << a << endl
        << "\tDifference: " << q << endl
        << "\tNumber: " << n;
}
```

help_modul.cpp

```
#include "help_modul.h"

vector<Geometric_progression> input_geo_progressions(int length)
{
    vector<Geometric_progression> geo_progressions;
    double a, q;
    unsigned int n;

    for (int i = 0; i < length; i++)
    {
        cout << endl << "Progression " << i + 1 << " : " << endl;

        cout << "Enter a: ";
```

```

        cin >> a;

        cout << "Enter q: ";
        cin >> q;

        cout << "Enter n: ";
        cin >> n;

        Geometric_progression progres(a, q, n);
        geo_progressions.push_back(progres);
    }
    return geo_progressions;
}

int find_max(vector<Geometric_progression> progressions)
{
    double max_value = progressions[0].get_a() * (power(progressions[0].get_q(),
progressions[0].get_n() - 1));
    int max_pos = 0;
    double temp;
    for (int i = 0; i < progressions.size(); i++)
    {
        temp = progressions[i].get_a() * (power(progressions[i].get_q(),
progressions[i].get_n() - 1));
        if (temp > max_value)
        {
            max_value = temp;
            max_pos = i;
        }
    }
    return max_pos;
}

double power(double a, int n)
{
    double res = 1;
    for (int i = 0; i < n; i++)
    {
        res = res * a;
    }
    return res;
}

```

Результат:

```
C:\Windows\system32\cmd.exe
Enter how many progressions you want to specify: 3

Progression 1 :
Enter a: 5
Enter q: 7
Enter n: 3

Progression 2 :
Enter a: 4
Enter q: 7
Enter n: 3

Progression 3 :
Enter a: 4
Enter q: 2
Enter n: 3

Progression 1 :
    First member: 5
    Difference: 7
    Number: 3

Progression 2 :
    First member: 4
    Difference: 7
    Number: 3

Progression 3 :
    First member: 4
    Difference: 2
    Number: 3

Geometric progression with largest last term: 1
Press any key to continue . . .
```