Федеральное государственное автономное образовательное учреждение высшего образования «Пермский национальный исследовательский политехнический университет», ПНИПУ

ЛАБОРАТОРНАЯ РАБОТА

КЛАССЫ И ОБЪЕКТЫ. ИНКАПСУЛЯЦИЯ.

Выполнил: студент группы РИС-23-3б

Артем Владимирович Швецов

Проверила: доцент кафедры ИТАС

Ольга Андреевна Полякова

Пермь 2024

**Постановка задачи**

1. Реализовать определение нового класса.
2. Реализовать методы взаимодействия с объектами класса.
3. Реализовать внешнюю функцию

Для варианта 15:

Элемент арифметической прогрессии aj вычисляется по формуле: aj = a0 \* rj. Поле first – дробное число, первый элемент прогрессии a0,поле second – положительное целое число, постоянное отношение r. Реализовать метод element(int j) – вычисление j-ого элемента прогрессии.

**Анализ задачи**

1. Создается новый файл с расширением .h, где определяется новый класс Progression.
2. Создается парный .cpp файл, в котором определяются методы нового класса.
3. В главном .cpp файле создается функция, которая возвращает объект типа Progression в соответствии заданным параметрам.

**Код**

Progression.h

#pragma once

class Progression {

double first;

int second;

double Power(int, int);

public:

void Init(double, int);

void Set1(double);

void Set2(int);

void Show();

double element(int);

};

Progression.cpp

#pragma once

#include <iostream>

#include "Progression.h"

double Progression::Power(int a, int i) {

double res = 1;

for (int j = 0; j < i; j++){

res \*= a;

}

return res;

}

void Progression::Init(double first, int second) {

this->first = first;

this->second = second;

}

void Progression::Set1(double vfirst) {

this->first = vfirst;

}

void Progression::Set2(int vsecond) {

this->second = vsecond;

}

void Progression::Show() {

std::cout << "first: " << this->first << std::endl;

std::cout << "second: " << this->second << std::endl;

}

double Progression::element(int j) {

double res = this->first \* Power(this->second, j);

return res;

}

Main.cpp

#include <iostream>

#include "Progression.h"

using namespace std;

Progression Make\_Progression(double fst, int snd) {

Progression temp;

temp.Init(fst, snd);

return temp;

}

int main() {

Progression prog;

prog.Init(1, 1);

prog.Set1(3.5);

prog.Set2(2);

cout << "Values of prog " << endl;

prog.Show();

cout << "Value of 3rd element: " << prog.element(3) << endl;

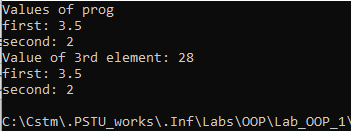
Progression prog2;

prog2 = Make\_Progression(4.5, 2);

prog.Show();

}

**Решение**



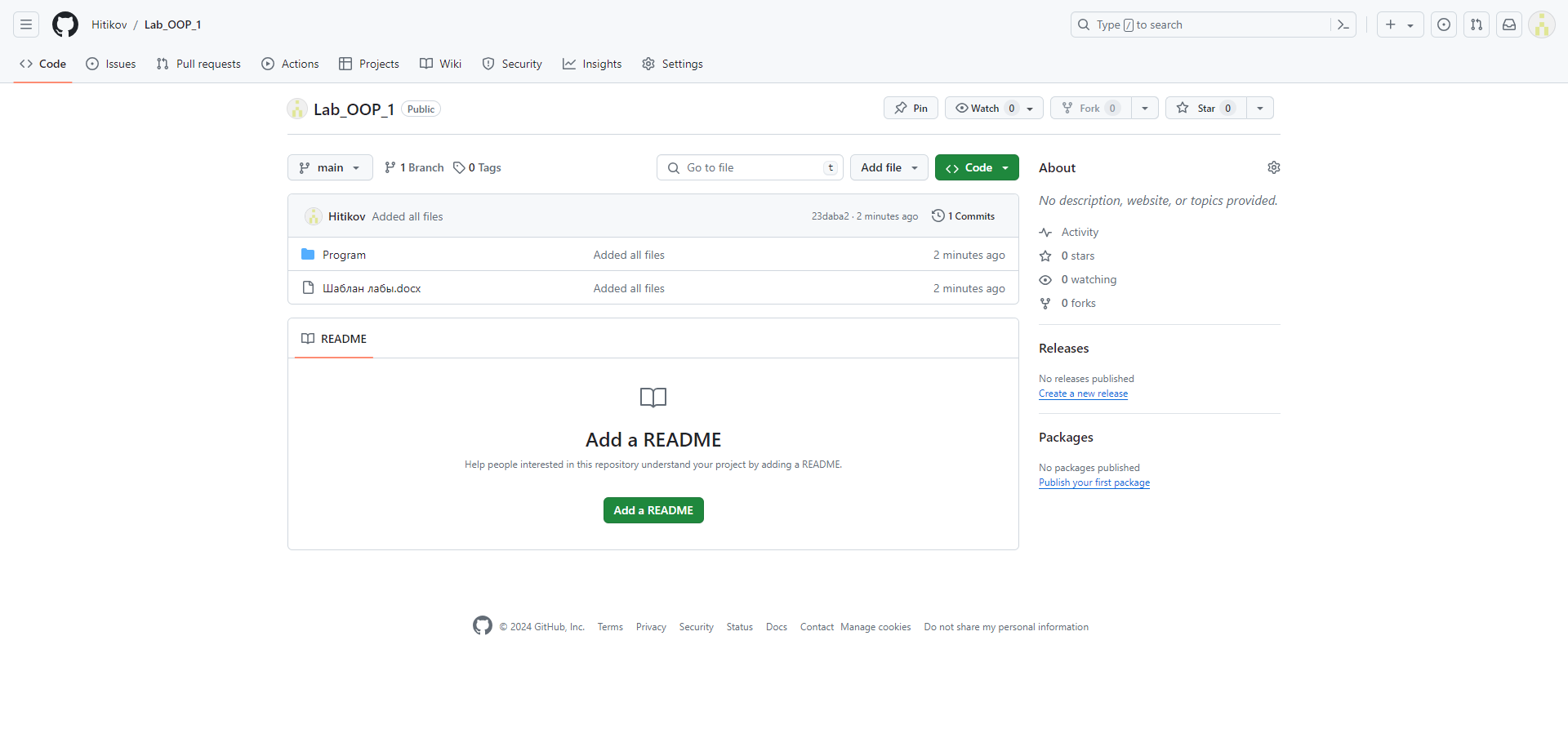
**Выводы**

В ходе выполнения работы были изучены базовые понятия классов и объектов и способы взаимодействия с ними. Также был изучен принцип инкапсуляции.

Ответы на контрольные вопросы:

1. Класс – абстрактный тип данных.
2. Конкретная реализация класса с определенными данными.
3. Атрибуты.
4. Методы.
5. Используются для закрытия или открытия доступа к атрибутам и методам (инкапсуляция)
6. В public сегменте элементы доступны за пределами класса.
7. В private сегменте элементы не доступны за пределами класса.
8. Private
9. Public
10. Public. Внешне с классом взаимодействуют через интерфейс и при его закрытости с классом будет невозможно взаимодействовать.
11. Напрямую, либо через методы модификаторы полей.
12. Напрямую, либо через методы чтения полей.
13. s -> name
14. s.name
15. Необходим геттер (пр. get(){return name}): s->get()
16. Необходим геттер (пр. get(){return name}): s.get()
17. \*(s->name)

**Github**



<https://github.com/Hitikov/Lab_OOP_1>