Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«**Пермский национальный исследовательский политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

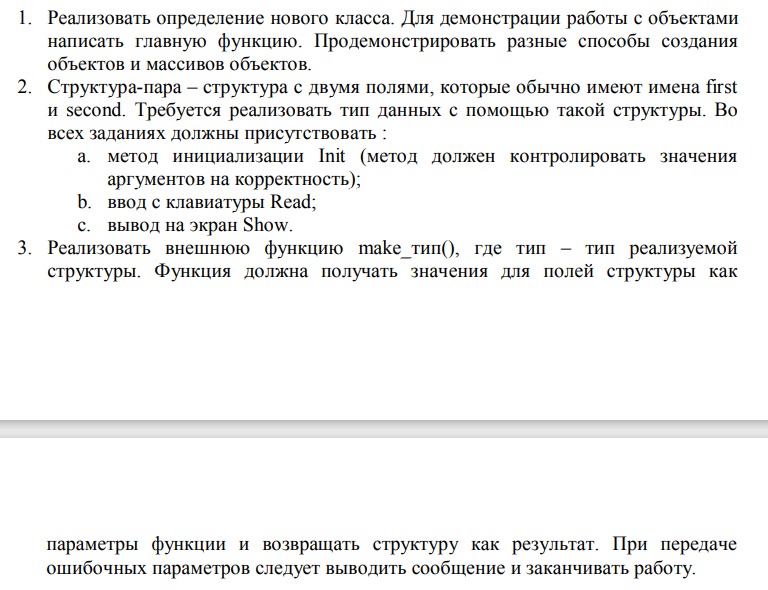
**ОТЧЕТ**

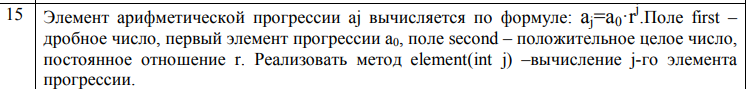
Дисциплина: «Основы алгоритмизации и программирования»

Тема: Лабораторная работа по классам №1

Выполнил работу  
Студент группы РИС-22-1Б  
Ишемцева Мария Андреевна  
Проверил  
Доцент кафедры ИТАС  
Полякова Ольга Андреевна

Постановка задачи:





Вариант 15.

Анализ задачи:

1. Пользовательский класс Progression содержит в себе только public элементы, поля – first типа double, second типа int.
2. Методы класса – Read()(считать значения с клавиатуры, вызвать проверочный метод Init()), метод element()(считает по формуле элемент прогрессии), в последнюю очередь вызывается метод show()(выводит элемент на экран).
3. *Ответы на контрольные вопросы:*

1)Это абстрактный тип данных. Он сочетает в себе два функционала: Первая — это структура, в которой можно хранить различные типы данных: массивы, переменные, функции. Вторая — возможность пользоваться объектно-ориентированным программированием

2)В языке программирования объект – это переменная типа «класс». Класс описывает данные и методы (функции), которые будут использоваться объектом этого класса. Каждый класс описывает логически-завершенную единицу программы.

3)По́ле кла́сса или атрибу́т (переменная-член, data member, class field, instance variable) в объектно-ориентированном программировании — переменная, связанная с классом или объектом.

4)Методы.

5)Чтобы предотвратить изменения «жизненно важных» переменных и методов от других пользователей.

6)Для доступа к методам и полям через ф-ию main().

7)Для работы с этими полями и методами внутри класса.

8)Private.

9)Public.

10)Public, для комфортной работы из любой ф-ии.

11)С помощью конструкторов копирования, с параметрами, без параметров, с помощью сеттеров(метод присваивания), с помощью обычной инициализации в ф-ии main(My.a=5).

12)С помощью методов класса show()(простой cout), либо с использованием геттеров(возвращает значение, обычно полученное от сеттера).

13)S.name.

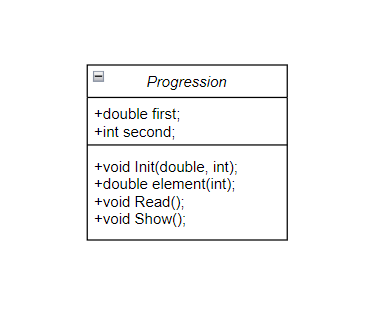
14)S.name.

15)С помощью сеттера void SetName(VolName){name=VolNAme;} из ф-ии main(){s.SetName};

16)С помощью сеттера void SetName(VolName){name=VolNAme;} из ф-ии main(){s.SetName};

17)S.name;

UML диаграмма:



Код:

Основной файл:

#include <iostream>

#include <Windows.h>

#include "Progression.h"

using namespace std;

int maaaain() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

Progression A;

A.Init(0.1, 3);

A.Show();

cout << A.first << "\*" << A.second << "^2 =" << A.element(2) << endl;

Progression B;

B.Read();

B.Show();

int j;

cout << "Введите номер элемента последовательсности: ";

cin >> j;

cout << B.first << "\*" << B.second << "^" << j << "=" << B.element(j) << endl;

return 0;

}  
  
Файл с классом  
  
#pragma once

class Progression {

public:

double first;

int second;

void Init(double, int);

double element(int);

void Read();

void Show();

};

Описание мотодов

#include <iostream>

#include "Progression.h"

using namespace std;

void Progression::Init(double a0, int r) {

first = a0;

second = r;

}

double Progression::element(int j) {

return first \* pow(second, j);

}

void Progression::Read() {

cout << "Введите первый элемент последовательности: ";

cin >> first;

cout << "Введите постоянное отношение: ";

cin >> second;

}

void Progression::Show() {

cout << "Первый элемент последовательности - " << first;

cout << "\nПостоянное отношение - " << second << endl;

}

Вывод программы

