

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования

**Пермский национальный исследовательский
политехнический университет**

Факультет электротехнический

Кафедра ИТАС

ОТЧЁТ

о лабораторной работе №1 (классы)

Выполнил:
Студент группы ИВТ-23-1Б
Пискунов Д. А.

Проверил:
Доцент кафедры ИТАС
Яруллин Д.В.

Пермь 2024

Вариант 15

Элемент арифметической прогрессии a_j вычисляется по формуле: $a_j = a_0 \cdot r^j$. Поле `first` – дробное число, первый элемент прогрессии a_0 , поле `second` – положительное целое число, постоянное отношение r . Реализовать метод `element(int j)` – вычисление j -го элемента прогрессии.

Текст программы

Файл classes1.h

```
1  #pragma once
2
3  class Progression {
4  public:
5      double first;
6      int second;
7      void init(double, int);
8      void show();
9      void read();
10     double element(int);
11 };
```

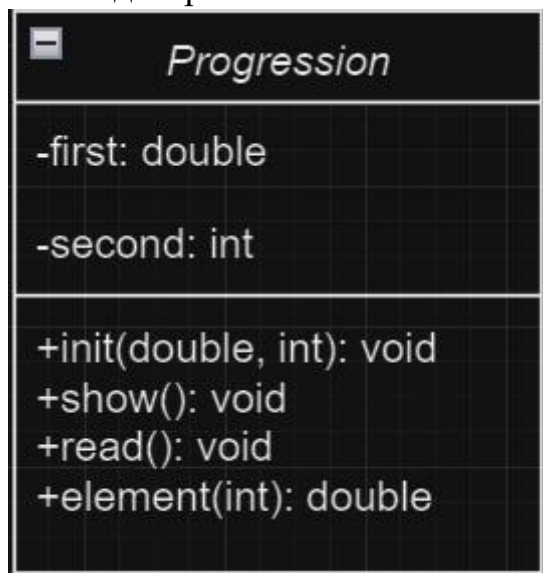
Файл classes1.cpp

```
1  #include <iostream>
2  #include "classes1.h"
3  #include <math.h>
4
5  void Progression::init(double f, int s) {
6      first = f;
7      second = s;
8  }
9  void Progression::show() {
10     std::cout << "first: " << first << "\n";
11     std::cout << "second: " << second << "\n";
12 }
13 void Progression::read() {
14     std::cout << "first: \n";
15     std::cin >> first;
16     std::cout << "second: \n";
17     std::cin >> second;
18 }
19
20 double Progression::element(int j) {
21     double j_elem = first * pow(second, j);
22     return j_elem;
23 }
```

Файл Classes_1.cpp

```
1  #include<iostream>
2  #include<math.h>
3  #include "classes1.h"
4
5  using namespace std;
6
7  Progression make_progression(double f, int s) {
8      Progression t;
9      t.init(f, s);
10     return t;
11 }
12
13 int main() {
14     system("chcp 1251>null");
15
16     double a;
17     int r;
18     int c;
19     double result;
20
21     cout << "Введите дробное число - первый элемент последовательности\n";
22     cin >> a;
23
24     cout << "\nВведите целое число - постоянное отношение\n";
25     cin >> r;
26
27     cout << "\nВведите целое число - номер элемента последовательности\n";
28     cin >> c;
29
30     Progression t = make_progression(a, r);
31     t.show();
32     result = t.element(c);
33     cout << endl << c << " элемент последовательности = ";
34     cout << result;
35
36
37     return 0;
38 }
```

UML- диаграмма:



Тест

Введите дробное число - первый элемент последовательности
13.135

Введите целое число - постоянное отношение
10

Введите целое число - номер элемента последовательности
4
first: 13.135
second: 10

4 элемент последовательности = 131350

C:\Users\M0kASiH\Desktop\Classes_1\x64\Debug\Classes_1.exe (процесс 17104) завершил работу с кодом 0.
Нажмите любую клавишу, чтобы закрыть это окно..._

Ответы на вопросы:

1. Что такое класс?

Ответ: класс - абстрактный пользовательский тип данных.

2. Что такое объект (экземпляр) класса?

Ответ: объектом называется отдельный элемент класса, который характеризуется данными его полей. Второе название объекта класса - экземпляр класса

3. Как называются поля класса?

Ответ: атрибуты.

4. Как называются функции класса?

Ответ: методы.

5. Для чего используются спецификаторы доступа?

Ответ: управление видимостью элементов класса.

6. Для чего используется спецификатор public?

Ответ: для общедоступных элементов класса

7. Для чего используется спецификатор private?

Ответ: для закрытых элементов класса.

8. Если описание класса начинается со спецификатора class, то какой спецификатор доступа будет использоваться по умолчанию?

Ответ: private

9. Если описание класса начинается со спецификатора struct, то какой спецификатор доступа будет использоваться по умолчанию?

Ответ: public

10. Какой спецификатор доступа должен использоваться при описании интерфейса класса? Почему?

Ответ: public.

11. Каким образом можно изменить значения атрибутов экземпляра класса?

Ответ: через сеттер или напрямую.

12. Каким образом можно получить значения атрибутов экземпляра класса?

Ответ: через геттер или напрямую.

13. Класс описан следующим образом

```
struct Student
{
    string name;
    int group;
    .....
};
```

Объект класса определен следующим образом

```
Student *s=new Student;
```

Как можно обратиться к полю name объекта s?

Ответ: s→name;

14. Класс описан следующим образом

```
struct Student
{
    string name;
    int group;
    .....
};
```

Объект класса определен следующим образом

```
Student s;
```

Как можно обратиться к полю name объекта s?

Ответ: s.name;

15. Класс описан следующим образом

```
class Student
{
    string name;
    int group;
    .....
};
```

Объект класса определен следующим образом

```
Student *s=new Student;
```

Как можно обратиться к полю name объекта s?

Ответ: с помощью геттера.

16. Класс описан следующим образом

```
class Student
{
    string name;
    int group;
public:
```

```
    .....
};
```

Объект класса определен следующим образом

Student s;

Как можно обратиться к полю name объекта s?

Ответ: с помощью геттера.

17. Класс описан следующим образом

```
class Student
{
public:
    char* name;
    int group;
```

```
    .....
};
```

Объект класса определен следующим образом

Student *s=new Student;

Как можно обратиться к полю name объекта s?

Ответ: s→name.