МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

ПЕРМСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Кафедра информационные технологии и автоматизированные системы

**Дисциплина Информатика**

**Классы и объекты**

Выполнил студент ИВТ-22-2б:

Мифтахов Марат Ринатович

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Проверил доцент кафедры ИТАС:

Полякова Ольга Андреевна

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Пермь, 2023

**Постановка задачи**

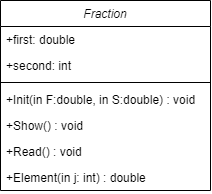
1. Реализовать определение нового класса.
2. Для демонстрации работы с объектами написать главную функцию.
3. Реализовать внешнюю функцию make\_Fraction(), которая получает значения для полей структуры как параметры и возвращает структуру как результат.

**Задание**

1. Элемент арифметической прогрессии aj вычисляется по формуле: a,-a0 r.
2. Поле first - дробное число, первый элемент прогрессии а0, поле second - положительное целое число, постоянное отношение r.
3. Реализовать метод element(int j) -вычисление j-го элемента  
   прогрессии.

**Диаграмма класса**

Ниже представлена UML-диаграмма класса Fraction.



**Код программы**

В файле fraction.h:

struct Fraction {

double first;

int second;

void Init(double, int);

void Show();

void Read();

double Element(int);

};

В файле fraction.cpp:

#include <iostream>

#include <Fraction.h>

#include <cmath>

using namespace std;

void Fraction::Init(double F, int S) {

first = F; second = S;

}

void Fraction::Show() {

cout << "First: " << first << endl;

cout << "Second: " << second << endl;

}

void Fraction::Read() {

cout << "First? " << endl; cin >> first;

cout << "Second? " << endl; cin >> second;

}

double Fraction::Element(int j) {

return first \* pow(second, j);

}

В файле main.cpp:

#include <iostream>

#include <Fraction.h>

#include <Windows.h>

using namespace std;

Fraction makeFraction(double, int);

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

Fraction A;

Fraction B;

A.Init(3.0, 2);

B.Read();

A.Show();

B.Show();

int j1, j2;

cout << "Введите номер какого элемента вы хотите получить " << endl;

cout << "j1: "; cin >> j1;

cout << "j2: "; cin >> j2;

cout << A.Element(5) << endl;

cout << B.Element(6) << endl;

Fraction\* X = new Fraction;

X->Init(2.0, 5);

X->Show();

Fraction mas[3];

for (int i = 0; i < 3; i++) mas[i].Read();

for (int i = 0; i < 3; i++) mas[i].Show();

for (int i = 0; i < 3; i++) {

cout << " Степень? " << endl;

cout << "j:"; cin >> j1;

mas[i].Element(j1);

}

Fraction\* p\_mas = new Fraction[3];

for (int i = 0; i < 3; i++) p\_mas[i].Read();

for (int i = 0; i < 3; i++) p\_mas[i].Show();

for (int i = 0; i < 3; i++) {

cout << " Степень? " << endl;

cout << "j:"; cin >> j1;

cout << mas[i].Element(j1) << endl;

}

double y; int z;

cout << "first?"; cin >> y;

cout << "second?"; cin >> z;

//переменная F формируется с помощью функции make\_fraction()

Fraction F = makeFraction(y, z);

F.Show();

return 0;

}

Fraction makeFraction(double f, int s) {

Fraction t;

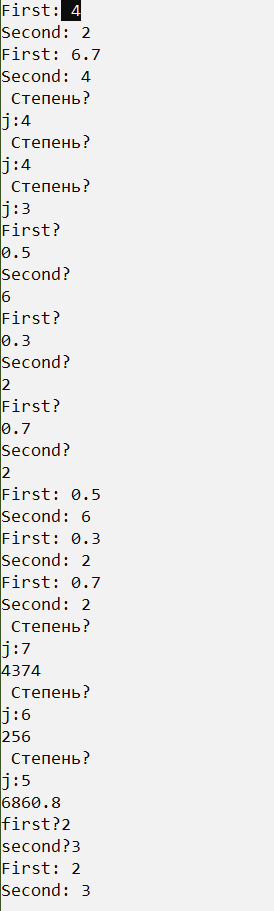
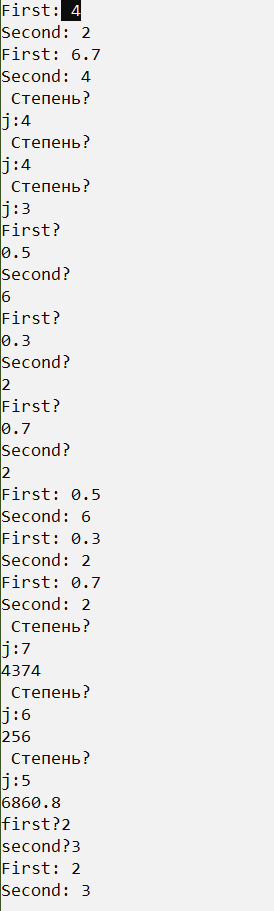
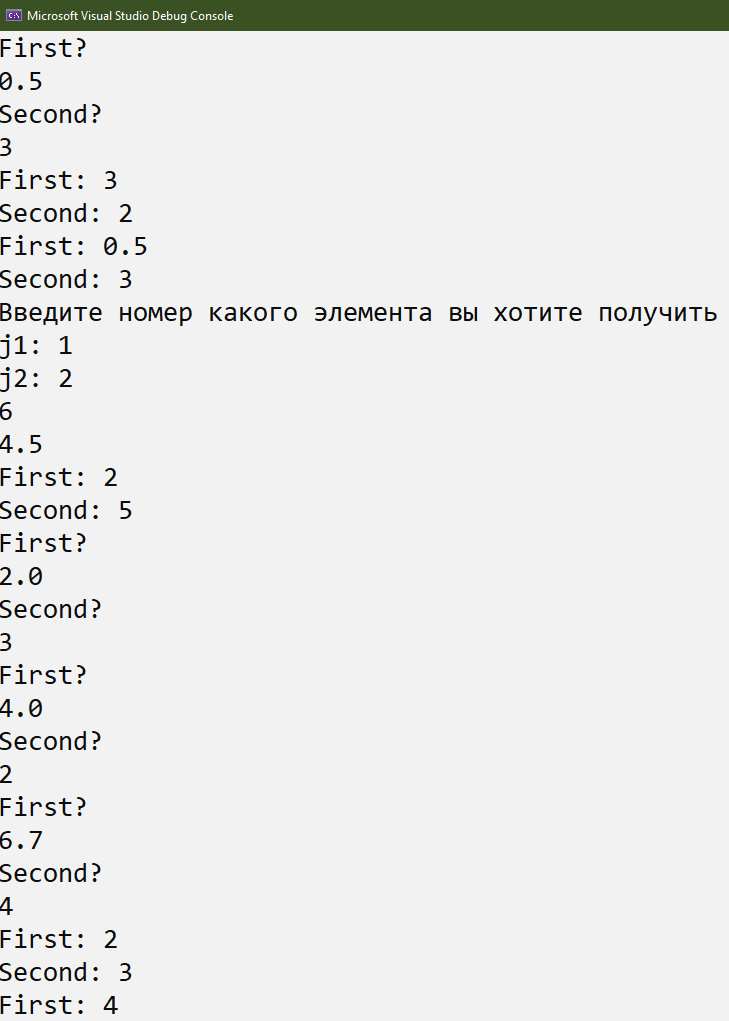
t.Init(f, s);

return t;

}

**Вывод программы**

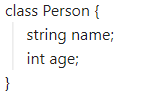
На рисунке 2 представлен вывод программы на консоль.

*Рисунок 2 - Вывод программы*

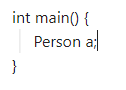
**Контрольные вопросы**

1. Что такое класс?

Класс - это абстрактный пользовательский тип данных, который представляет собой шаблон для экземпляров объектов.

2. Что такое объект класса?

Объект класса — это конкретный экземпляр данного класса.

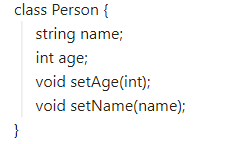


3. Как называются поля класса?

Поля класса называются атрибутами.

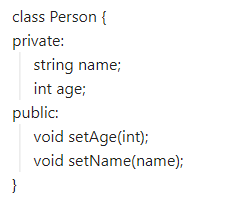
4. Как называются функции класса?

Функции класса называются методами.

5. Для чего используются спецификаторы доступа?

Спецификаторы доступа используются для сокрытия доступа к данным класса извне  
6. Для чего используется спецификатор public?

Спецификатор public используется для разрешения использования членов класса извне.

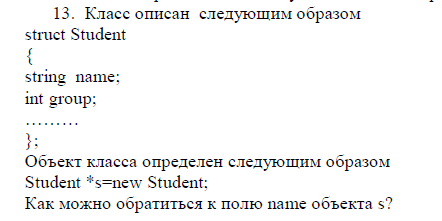
7. Для чего используется спецификатор private?

Спецификатор private используется для запрета использования членов класса извне.  
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?

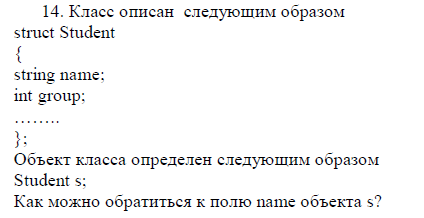
Ответ: public  
10. Какой спецификатор доступа должен использоваться при описании интерфейса класса? Почему?

Ответ: protected, потому что позволяет наследовать данные класса.  
11. Каким образом можно изменить значения атрибутов экземпляра класса?

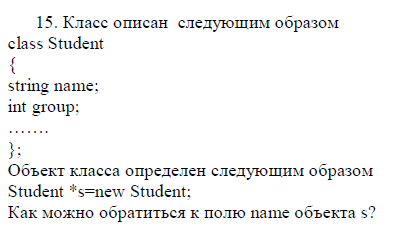
Ответ: через объект и ссылку на объект (.), а также через указатель на объект (->)  
12. Каким образом можно получить значения атрибутов экземпляра класса?

Ответ: через геттеры и сеттеры либо напрямую   
\_

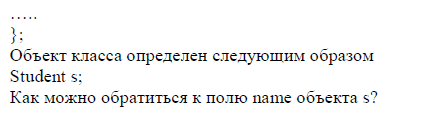
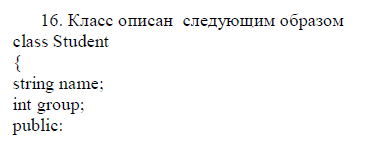
Ответ: через s→name

\_

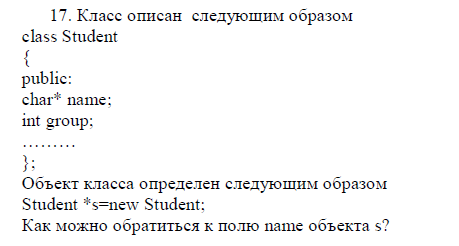
Ответ: через s.name

\_

Ответ: через геттер

\_

Ответ: через геттер

\_

Ответ: через s→name