Министерство высшего образования и науки Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

ПЕРМСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ (ПНИПУ)

Электротехнический факультет

Дисциплина: основы алгоритмизации и программирования, 2 семестр

**ОТЧЁТ**

Тема: «Классы и объекты. Инкапсуляция»

Выполнил

Студент РИС-22-2б

Зубов Р.А.

Проверил

Доцент кафедры ИТАС

Полякова О. А.

Пермь 2023

# **Содержание**

Постановка задачи**4**

Коды программ**5**

Результаты программ**6**

**Постановка задачи**

Реализовать определение ноого класса. Для дмонстрации работы с объектами написать главную функцию. Продемонстрировать разные способы создания объектов и массивов объектов.

Структура – пара – структура с двумя полями, которые обычно имеют имена first и second. Требуется реализовать тип данных с помощью такой структуры. Во всех заданиях должны присутсвовать :

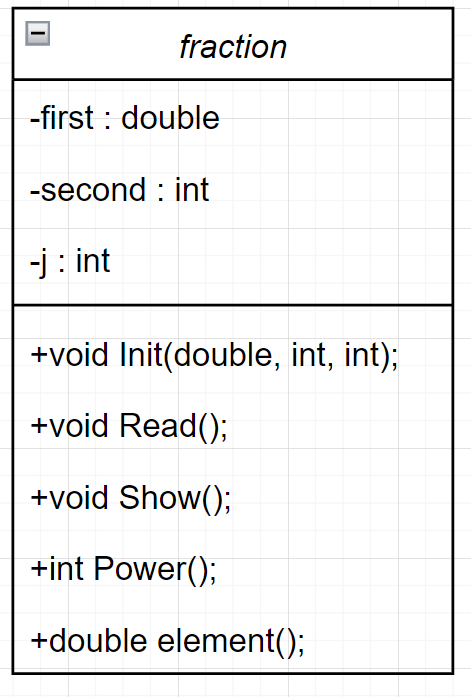
1. Метод инициализации Init (метод должен контролировать значения аргументов на корректность);
2. Ввод с клавиатуры Read;
3. Вывод на экран Show.

Реализовать внешнюю функцию make\_тип(), где тип – тип реализуемой структуры. Функция должна получать значения для полей структуры как параметры функции и возвращаться структуры как результат. При передаче ошибочных параметров следует выводить сообщение и заканчивать работу.

**Задача**

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

**UML диаграмма**

****

**Описание класса**

Класс fraction содержит три поля данных: first, second и j. Первое поле — это число с плавающей точкой двойной точности, второе - целое число, а третье - целое число. Класс также содержит три метода: Init, Read и Show. Метод Init принимает три аргумента и инициализирует поля данных класса. Метод Read считывает значения полей данных с клавиатуры, а метод Show выводит значения полей данных на экран. Метод Power возвращает значение первого поля в степени второго поля, а метод element возвращает значение первого поля.

**Определение компонентных функций**

Компонентные функции — это функции класса, которые используются для работы с полями данных класса. В данном случае, определение компонентных функций класса fraction содержит четыре функции: Init, Read, Power и element.

Функция Init инициализирует поля данных класса значениями, переданными в качестве аргументов.

Функция Read считывает значения полей данных с клавиатуры.

Функция Power возвращает значение второго поля в степени третьего поля.

Функция element возвращает произведение первого поля на результат функции Power.

**Определение функции make\_fraction()**

Функция make\_fraction — это функция, которая создает объект класса fraction и инициализирует его значениями, переданными в качестве аргументов.

**Объяснение результатов работы программы**

В функции main инициализируется объект класса fraction с именем find. Затем вызывается метод Read() класса fraction для объекта find. Далее выводится значение j – го элемента благодаря использования метода element для объекта find. Затем инициализируется объект F класса fraction, которому присваивается значение, возвращаемого функцией make\_fraction. После чего выводится значение j – го элемента благодаря использования метода element для объекта F.

**Контрольные вопросы**

1. Что такое класс?

Класс – абстрактный тип данных, определяемый пользователем, и представляет собой модель реального объекта в виде данных и функций для работы с ним.

class MyClass {

public:

int myNum;

string myString;

};

1. Что такое объект класса?

[Объект класса - это экземпляр класса, созданный на основе определения класса в коде](https://metanit.com/cpp/tutorial/5.2.php)

class Dog {

public:

void bark() {

cout << "Гав-гав!" <<endl;

}

};

1. Как называются поля класса?

Поля класса - это переменные, которые определены внутри класса и хранят данные объекта. [Поля класса могут быть публичными, приватными или защищенными](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%BB%D0%B5_%D0%BA%D0%BB%D0%B0%D1%81%D1%81%D0%B0).

class Dog {

public:

string name; // Публичное поле класса

private:

int age; // Приватное поле класса

};

1. Как называются функции класса?

Функции класса – это методы.

class Dog {

public:

void bark() {

cout << "Гав-гав!" << endl;

}

private:

int age; // Приватное поле класса

public:

void setAge(int a) {

age = a;

}

};

1. 6. 7. Спецификаторы доступа private и рublic управляют видимостью элементов

класса.

Элементы, описанные после служебного слова private, видимы только внутри класса. Этот вид доступа принят в классе по умолчанию. Интерфейс класса описывается после спецификатора public. Действие любого спецификатора распространяется до следующего спецификатора или до конца класса. Можно задавать несколько секций private n public, порядок их следования значения не имеет.

8. Если описание класса начинается с Class, то по умолчанию будет использоваться спецификатор доступа private.

9. Если описание класса начинается с struct, то по умолчанию будет использоваться спецификатор доступа public.

10. Спецификаторы доступа дают автору класса возможность определить, какие члены класса доступны пользователям класса (т.е. интерфейс) и какие члены предназначены для внутреннего использования класса. В C++ есть три спецификатора доступа: public, private и protected.

[Спецификатор public делает члены класса - поля и функции открытыми, доступными из любой части программы](https://metanit.com/cpp/tutorial/5.4.php). [Спецификатор private делает члены класса закрытыми, доступными только внутри класса](https://metanit.com/cpp/tutorial/5.4.php). [Спецификатор protected делает члены класса доступными только внутри класса и его наследников](https://learn.microsoft.com/ru-ru/cpp/cpp/member-access-control-cpp?view=msvc-170).

11. [В C++ значения атрибутов экземпляра класса можно изменить с помощью методов класса](https://ru.stackoverflow.com/questions/459448/%D0%92-%D1%87%D0%B5%D0%BC-%D0%BE%D1%82%D0%BB%D0%B8%D1%87%D0%B8%D0%B5-%D0%B2-%D0%BF%D1%80%D0%B8%D1%81%D0%B2%D0%BE%D0%B5%D0%BD%D0%B8%D0%B8-%D0%B0%D1%82%D1%80%D0%B8%D0%B1%D1%83%D1%82%D0%B0-%D0%BA%D0%BB%D0%B0%D1%81%D1%81%D0%B0-%D0%B2-%D0%BA%D0%BE%D0%BD%D1%81%D1%82%D1%80%D1%83%D0%BA%D1%82%D0%BE%D1%80%D0%B5-%D0%B8-%D0%BD%D0%B5%D1%82).

class Person {

public:

void setName(string newName) {

name = newName;

}

private:

string name;

};

12. для изменения значения атрибута name

Person person;

person.setName("John");

13. Для обращения к полю name объекта s нужно использовать оператор “->”. s->name = "John";

14. Для обращения к полю name объекта s нужно использовать оператор “.” s.name = "John";

15. 16. К private полю класса Student нельзя получить доступ извне.

17. Для обращения к полю name объекта s нужно использовать оператор “->”. s->name = "John";s?