Министерство высшего образования и науки Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

**«Пермский национальный исследовательский политехнический университет» (ПНИПУ)**

Электротехнический факультет

Кафедра «Информационные технологии и автоматизированные системы»

ОТЧЁТ

по лабораторной работе №1

Тема: «Классы и объекты. Инкапсуляция.»

Выполнил

Студент группы РИС-22-2б

Дудкин Александр Васильевич

Проверил доц. Кафедры ИТАС

Полякова Ольга Андреевна

Пермь 2023

# Основа для выполнения задачи

1. Реализовать определение нового класса. Для демонстрации работы с объектами написать главную функцию. Продемонстрировать разные способы создания объектов и массивов объектов.
2. 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 присутствовать:
   1. метод инициализации Init (метод должен контролировать значения аргументов на корректность);
   2. ввод с клавиатуры Read;
   3. вывод на экран Show.
3. Реализовать внешнюю функцию make\_type(), где type – тип реализуемой структуры. Функция должна получать значения для полей структуры как параметры функции и возвращать структуру как результат. При передаче ошибочных параметров следует выводить сообщение и заканчивать работу.

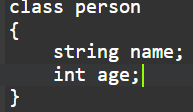
ВАРИАНТ 15:

Поле first – дробное положительное число, катет *а* прямоугольного треугольника, поле second – дробное положительное число, катет *b* прямоугольного треугольника. Реализовать метод hypotenuse() – вычисление гипотенузы.

# Контрольные вопросы

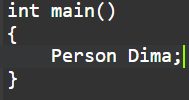
* 1. Что такое класс?

*Класс* - абстрактный тип данных, определяемым пользователем. Представляет собой модель реального объекта в виде данных и функций для работы с ними.



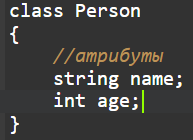
* 1. Что такое объект (экземпляр) класса?

*Экземпляр класса –* это описание конкретного объекта класса. Допустим: Класс – человек, объект – Дима



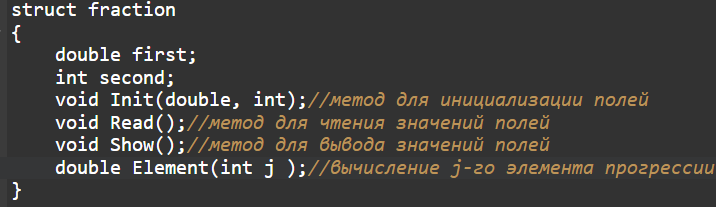
* 1. Как называются поля класса?

*Атрибуты.*

**

* 1. Как называются функции класса?

*Методы.*

**

* 1. Для чего используются спецификаторы доступа?

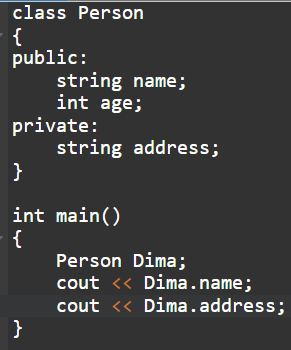
Управление видимостью элементов класса (чтение/запись).

* 1. Для чего используется спецификатор public?

Для общедоступных элементов класса

* 1. Для чего используется спецификатор private?

Для закрытых элементов класса.

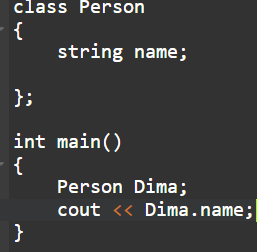


* 1. Если описание класса начинается со спецификатора class, то какой спецификатор доступа будет использоваться по умолчанию?

Private

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

Public



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

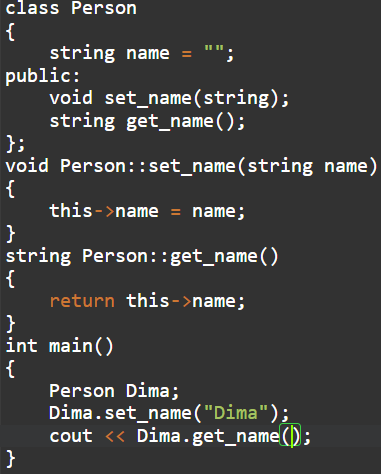
Public, так как нужно иметь доступ к методу, чтобы его описать в дальнейшем.

* 1. Каким образом можно изменить значения атрибутов экземпляра класса?

С помощью метода – модификатора (сеттер).

* 1. Каким образом можно получить значения атрибутов экземпляра класса?

С помощью метода – селектор (геттер).





class Student

{

string name;

int group;

………

};

Объект класса определен следующим образом

Student \*s=new Student;

Как можно обратиться к полю name объекта s?

s->name;



class Student

{

Public:

string name;

int group;

……..

};

Объект класса определен следующим образом

Student s;

s.name;

Как можно обратиться к полю name объекта s?

s.name;



class Student

{

string name;

int group;

…….

};

Объект класса определен следующим образом

Student \*s=new Student;

Как можно обратиться к полю name объекта s?

С помощью геттера



class Student

{

string name;

int group;

public:

Объект класса определен следующим образом:

Student s;

Как можно обратиться к полю name объекта s?

С помощью геттера



class Student

{

public:

char\* name;

int group;

………

};

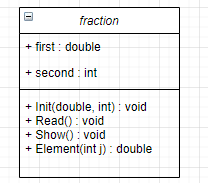
Объект класса определен следующим образом:

Student \*s=new Student;

Как можно обратиться к полю name объекта s?

s->name.

UML-диаграмма



Код программы:

Trev.h:

#pragma once

#include <iostream>

#include <cmath>

using namespace std;

class Trev

{

double first;

double second;

public:

Trev();

int getFirst() const;

void setFirst(double i);

void setSevond(double i);

void hipotenuse();

};

Trev.cpp:

#include "Trev.h"

Trev::Trev()

{

first = 0;

second = 0;

}

int Trev::getFirst() const

{

return first;

};

void Trev::setFirst(double a)

{

first = a;

};

void Trev:: setSecond(double a)

{

second = a;

}

void Trev::hipotenuse()

{

double a;

a = sqrt(pow(first, 2) + pow(second, 2));

cout << a << endl;

}

Sourse.cpp:

#include <iostream>

#include <cmath>

#include "Trev.h"

using namespace std;

int main()

{

setlocale(0, "RUS");

Trev trev;

cout << "Укажите размеры катетов" << endl;

double x, y;

cin >> x >> y;

trev.setFirst(x);

trev.setSevond(y);

trev.hipotenuse();

}

UML:

