МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение

высшего образования

«ТЮМЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Практическое занятие № 13\_14.

ОСНОВНЫЕ ПРИНЦИПЫ ООП

**Выполнил** студент:

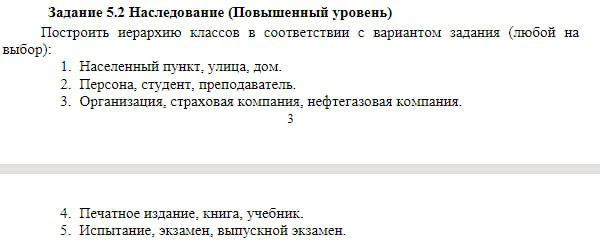
Герасимов Константин Сергеевич

Тюмень-2023

**Работа в аудитории**

**1. Решение задачи 1**

***1.1 Постановка задачи***



***1.2 Текст программы***

#include <iostream>  
#include <string>  
#include <utility>  
  
using namespace std;  
  
class Test {  
protected:  
 string subject;  
 int questions;  
 int time;  
public:  
 Test(string subj, int quest, int t) {  
 subject = std::move(subj);  
 questions = quest;  
 time = t;  
 }  
 virtual void printInfo() {  
 cout << "Subject: " << subject << endl;  
 cout << "Number of questions: " << questions << endl;  
 cout << "Time limit: " << time << " minutes" << endl;  
 }  
};  
  
class Exam : public Test {  
private:  
 int difficulty;  
public:  
 Exam(string subj, int quest, int t, int diff) : Test(std::move(subj), quest, t) {  
 difficulty = diff;  
 }  
 void printInfo() override {  
 cout << "Exam information: " << endl;  
 Test::printInfo();  
 cout << "Difficulty level: " << difficulty << endl;  
 }  
};  
  
class FinalExam : public Exam {  
private:  
 bool isOral;  
public:  
 FinalExam(string subj, int quest, int t, int diff, bool oral) : Exam(std::move(subj), quest, t, diff) {  
 isOral = oral;  
 }  
 void printInfo() override {  
 cout << "Final Exam information: " << endl;  
 Exam::printInfo();  
 if (isOral) {  
 cout << "Oral Exam" << endl;  
 }  
 else {  
 cout << "Written Exam" << endl;  
 }  
 }  
};  
  
int main() {  
 Test\* test = new Test("Mathematics", 50, 120);  
 Exam\* exam = new Exam("Physics", 100, 180, 3);  
 auto\* finalExam = new FinalExam("Computer Science", 150, 240, 4, true);  
  
 test->printInfo();  
 cout << endl;  
 exam->printInfo();  
 cout << endl;  
 finalExam->printInfo();  
}

***1.3 Результат тестирования программы***

***Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание***

**Домашняя работа**

**2. Тестовое задание 1**

***2.1 Задание теста***

Как объявить класс "Автомобиль" с приватными свойствами "марка" и "год выпуска", а также публичными методами для доступа к этим свойствам?

a) class Car { private: string brand; int year; public: ... };

b) class Car { private string brand; int year; public: ... };

c) class Car { private: string brand; int year; public: ... };

d) class Car { private; string brand; int year; public: ... };

***Ответ: a)***

**3. Тестовое задание 2**

***3.1 Задание теста***

Какой из следующих операторов используется для доступа к свойствам и методам объекта класса?

a) .

b) ::

c) ->

d) :

***Ответ: a)***

**4. Тестовое задание 3 (практическая задача)**

***4.1 Формулировка задачи***

Напишите класс Point в котором необходимо задать две координаты точки, а также реализуйте метод вычисления расстояния между двумя этими точками.

***4.4 Оригинальный листинг программы (решение задачи):***

#include <iostream>  
#include <cmath>  
  
class Point {  
private:  
 double xCoordinate;  
 double yCoordinate;  
  
public:  
 Point(double x, double y) {  
 xCoordinate = x;  
 yCoordinate = y;  
 }  
  
 double getDistanceTo(Point otherPoint) {  
 double deltaX = xCoordinate - otherPoint.xCoordinate;  
 double deltaY = yCoordinate - otherPoint.yCoordinate;  
 return std::sqrt(deltaX \* deltaX + deltaY \* deltaY);  
 }  
};

***4.5 Набор тестовых данных:***

int main() {  
 Point point1(2.0, 3.0);  
 Point point2(-1.5, 2.5);  
 Point point3(0.0, -1.0);  
  
 double distance1to2 = point1.getDistanceTo(point2);  
 double distance1to3 = point1.getDistanceTo(point3);  
 double distance2to3 = point2.getDistanceTo(point3);  
  
 std::cout << "Расстояния между точками:" << std::endl;  
 std::cout << "Расстояние между точкой 1 и точкой 2: " << distance1to2 << std::endl;  
 std::cout << "Расстояние между точкой 1 и точкой 3: " << distance1to3 << std::endl;  
 std::cout << "Расстояние между точкой 2 и точкой 3: " << distance2to3 << std::endl;  
  
 return 0;  
}

**4.6 Пример работы:**

**Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание**