KМинистерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учереждение высшего образования

**«Пермский национальный исследовательский политехнический университет»**

Электротехнический факультет

Кафедра «Информационные технологии и автоматизированные системы»

Направление подготовки: Разработка информационных систем (РИС)

**Лабораторная работа 5**

Выполнил студент гр. РИС-24-3б

Караваев Артем Андреевич

Проверил:

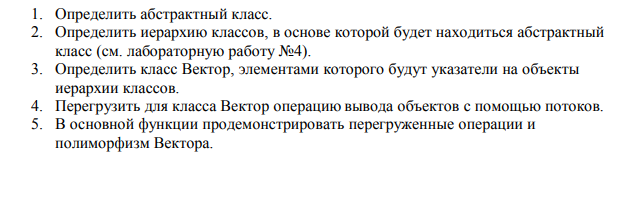
Доц. каф. ИТАС

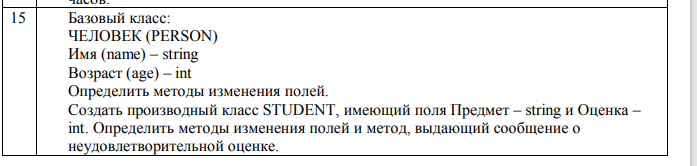
Ольга Андреевна Полякова

г. Пермь, 2024

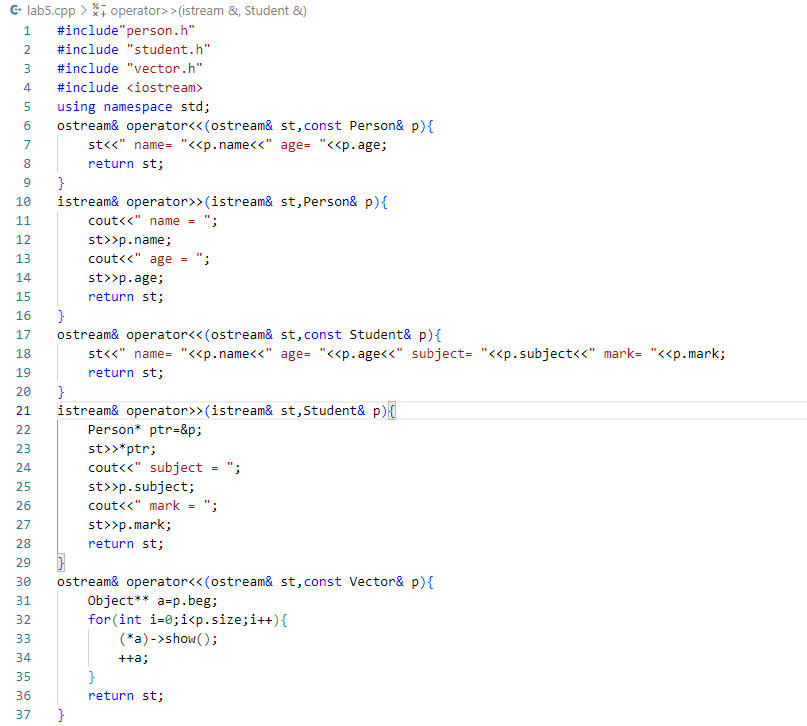
**Постановка задачи.**

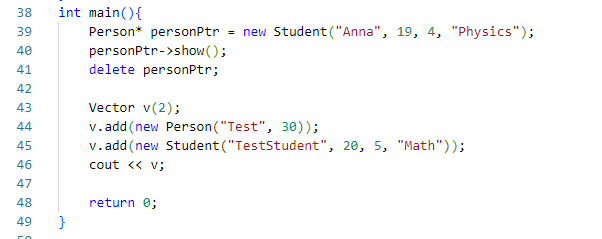
15 вариант:

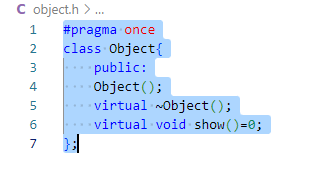


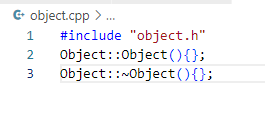


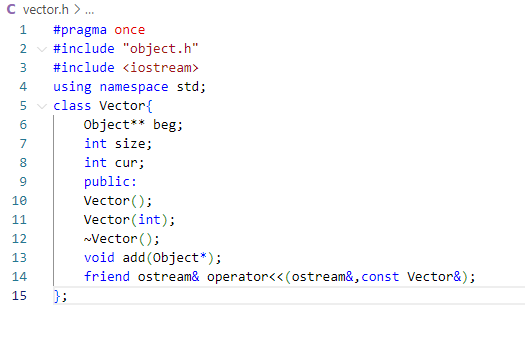
Программа на с++:

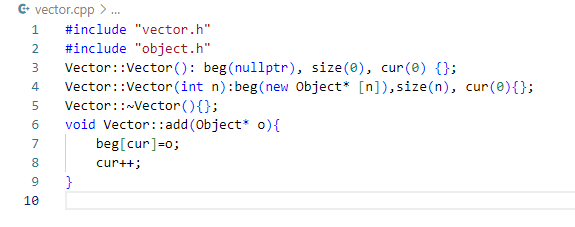


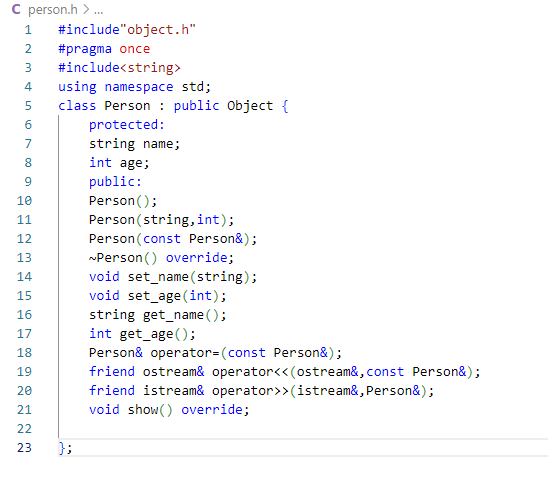


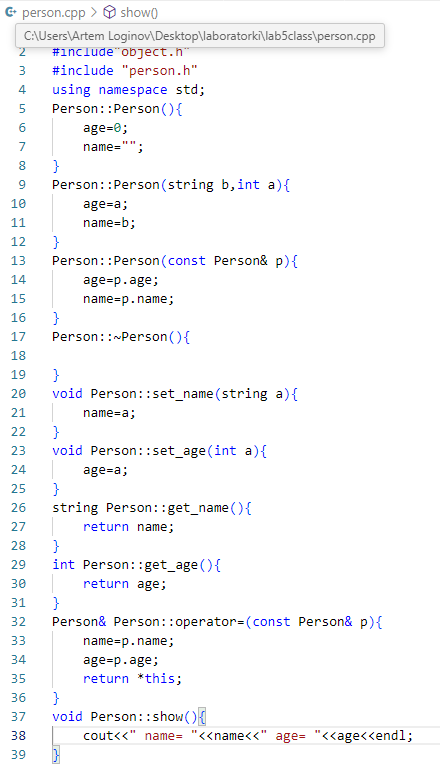


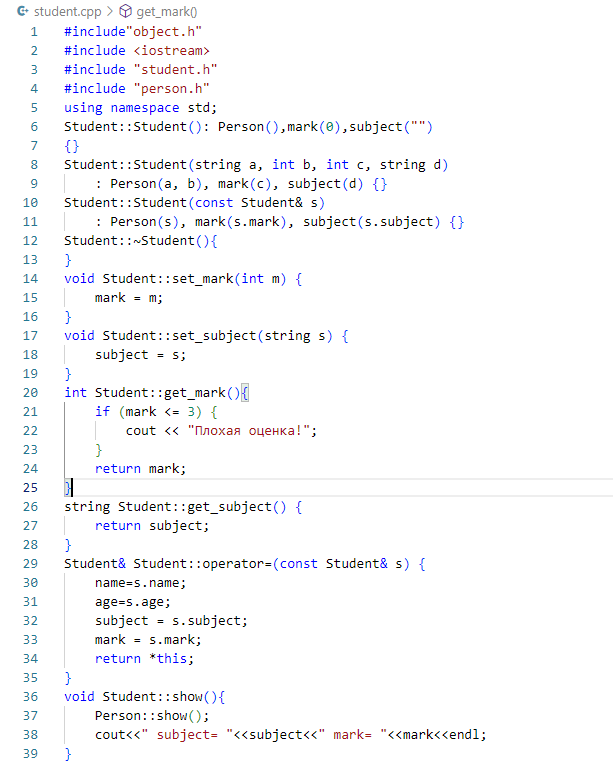
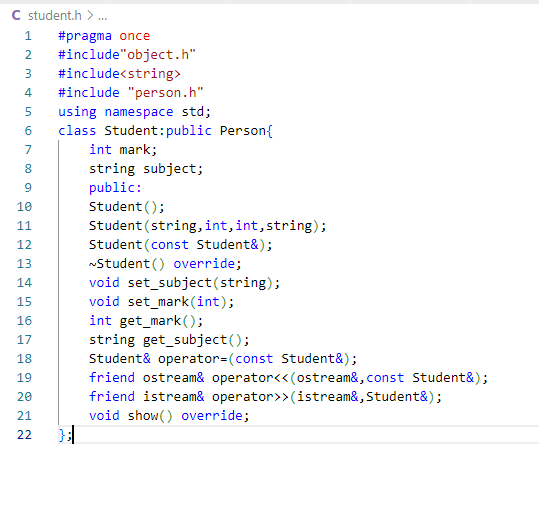




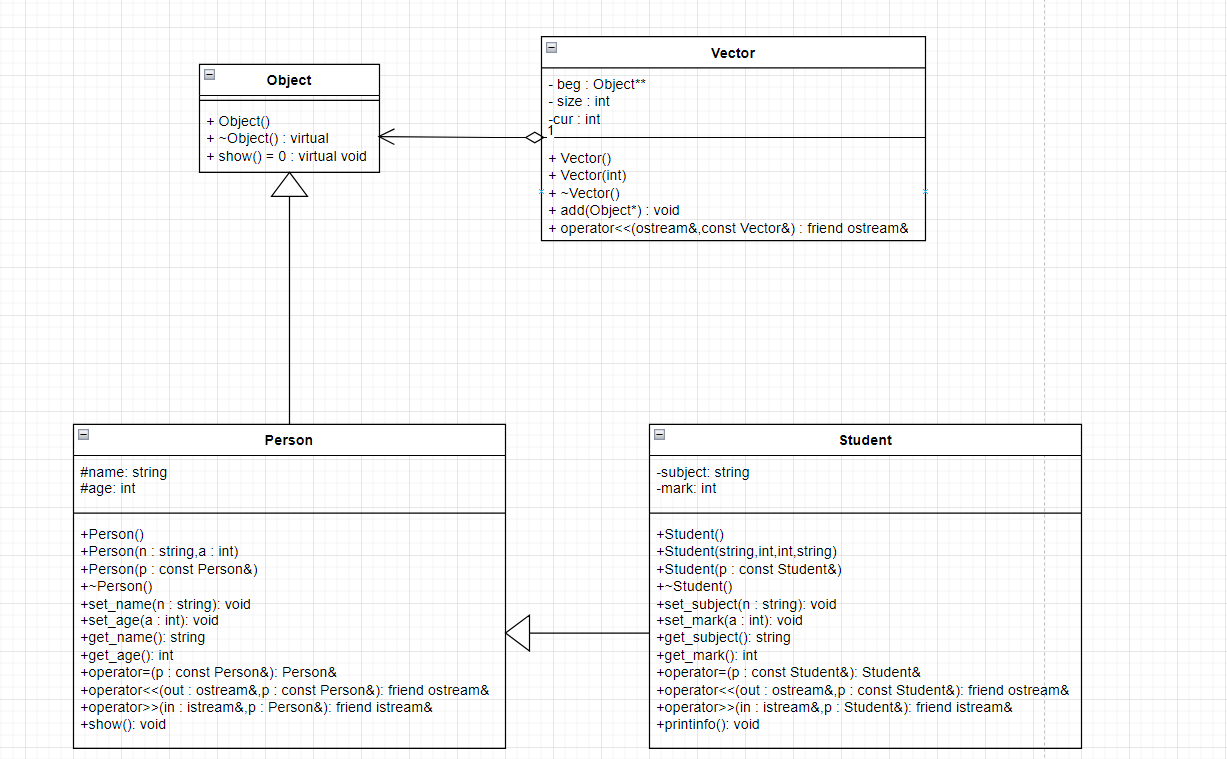




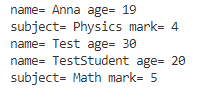


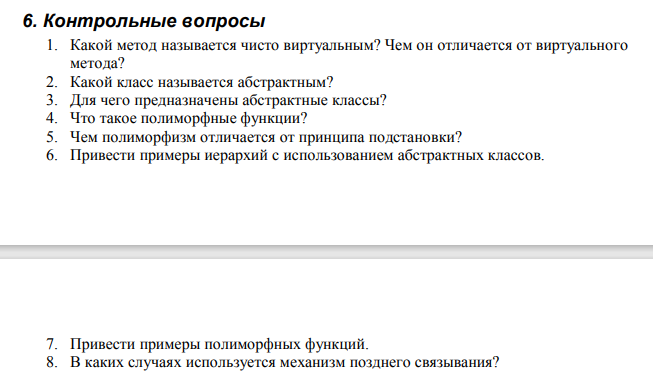


UML-диаграмма:



Входные и выходные данные:





1. Виртуальный метод: Имеет реализацию в базовом классе, но может быть переопределен в производных классах

Чисто виртуальный метод: Не имеет реализации в базовом классе должен быть реализован в производных классах

1. Класс, содержащий хотя бы один чисто виртуальный метод.
2. Создание интерфейсов, организация полиморфного поведения
3. Функции, которые могут работать с объектами разных типов через указатели/ссылки на базовый класс.
4. Полиморфизм: Техническая возможность вызывать разные реализации методов через единый интерфейс, Принцип подстановки*:* Смысловое требование, что производные классы должны корректно работать везде, где используется базовый класс
5. // Абстрактный класс

class Shape {

public:

virtual double area() const = 0; // чисто виртуальный

virtual ~Shape() {}

};

class Circle : public Shape {

double radius;

public:

Circle(double r) : radius(r) {}

double area() const override { return 3.14 \* radius \* radius; }

};

class Square : public Shape {

double side;

public:

Square(double s) : side(s) {}

double area() const override { return side \* side; }

};

1. void printArea(const Shape& shape) {

cout << "Area: " << shape.area(); // Вызовется правильная реализация

}

// Использование:

Circle c(5);

Square s(4);

printArea(c); // Выведет площадь круга

printArea(s); // Выведет площадь квадрата

1. При вызове виртуальных методов через указатель/ссылку на базовый класс, в полиморфных контейнерах (хранящих указатели на базовый класс)

https://github.com/Prefix008/lab.git