Пермский национальный исследовательский политехнический университет

Кафедра «Информационные технологии и автоматизированные системы»

Лабораторная работа 5

**Наследование. Виртуальные функции. Полиморфизм.**

Выполнил

Студент гр. РИС-24-3б Рунтов Иван Андреевич

Проверил

Доцент кафедры ИТАС Ольга Андреевна Полякова

г. Пермь, 2025

**Постановка задачи**

1. Определить абстрактный класс.

2. Определить иерархию классов, в основе которой будет находиться абстрактный класс (см. лабораторную работу №4).

3. Определить класс Вектор, элементами которого будут указатели на объекты иерархии классов.

4. Перегрузить для класса Вектор операцию вывода объектов с помощью потоков.

5. В основной функции продемонстрировать перегруженные операции и полиморфизм Вектора.

**Вариант 15**

Базовый класс: ЧЕЛОВЕК (PERSON)

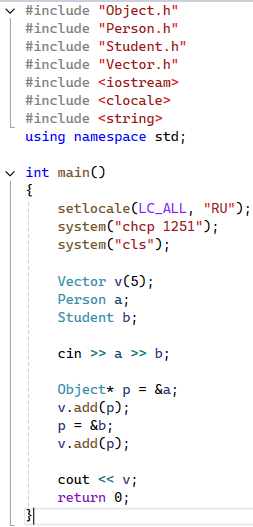
Имя (name) – string Возраст (age) – int

Определить методы изменения полей.

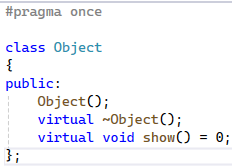
Создать производный класс STUDENT, имеющий поля Предмет – string и Оценка – int.

Определить методы изменения полей и метод, выдающий сообщение о неудовлетворительной оценке.

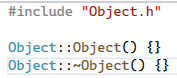
**Исполняемый файл main**



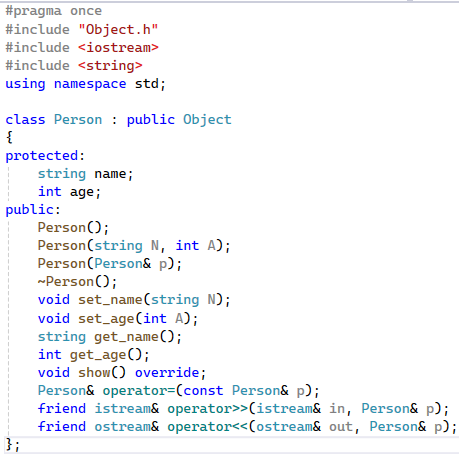
**Определение абстрактного класса Object**



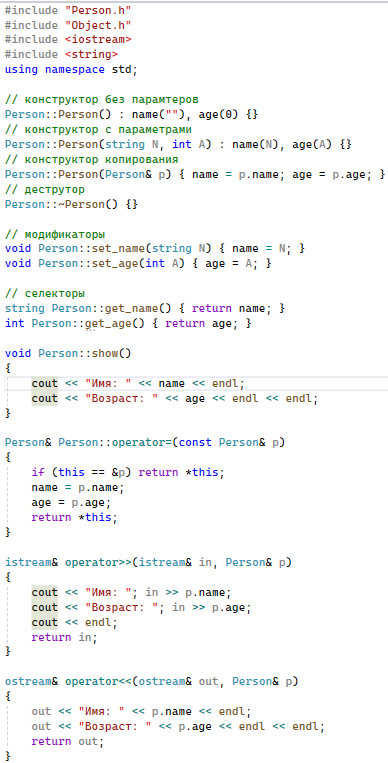
**Описание класса Object**



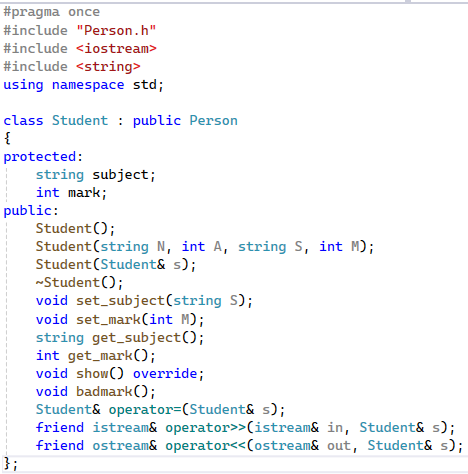
Определение класса Person



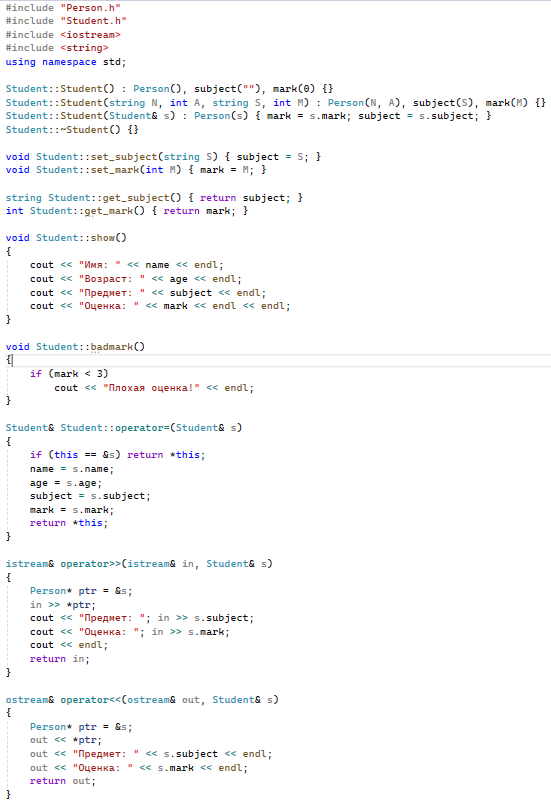
**Описание класса Person**



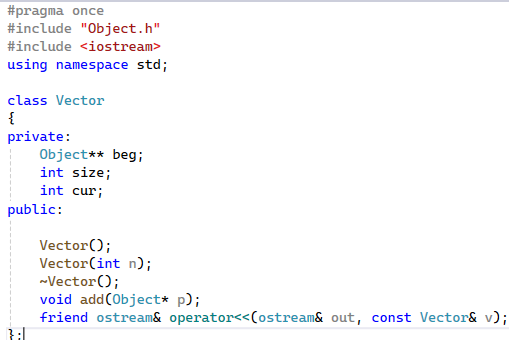
**Определение класса Student**



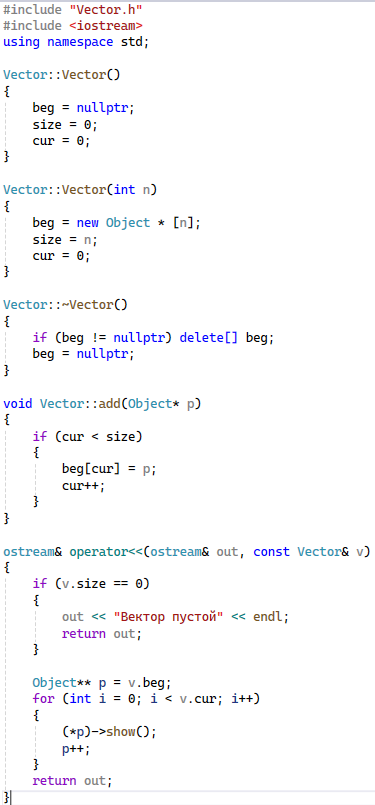
**Описание класса Student**



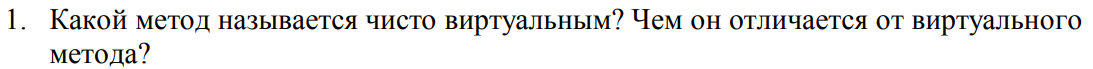
**Описание класса Vector**



**Определение класса Vector**



**Ответы на контрольные вопросы**



Чисто виртуальным методом называется метод, который не определяется в базовом классе. Реализация этого метода выполняется только в производных классах.



Абстрактным называется класс, который содержит хотя бы одну чисто виртуальную функцию.



Абстрактные классы предназначены для представления общих понятий, которые предполагается конкретизировать в производных классах.



Полиморфные функции предназначены для работы с объектом любого типа в пределах одной иерархии.



Полиморфизм предполагает разную реализацию методов базового класса. Принцип подстановки предполагает, что объекты разных типов в пределах одной иерархии можно подставлять в коде вместо друг друга.



class Object

{

public:

virtual void show() = 0;

}

class Person : public Object

{

public:

void show() {}

}

class Student : public Person

{

public:

void show() {}

}



Person:

virtual void show()

{

cout << “It’s person”;

}

Student : public Object

void show()

{

Cout << “It’s student”;

}



Механизм позднего связывания используется в тех случаях, когда необходимо обращаться через указатель не по типу указателя, а по типу значения, на которое указывает указатель.

Ссылка на github: <https://github.com/RuntovIvan/Informatika>