Министерство науки и высшего образования Российской Федерации Федеральное государственное автономное образовательное учреждение высшего образования

«Пермский национальный исследовательский политехнический университет»

Электротехнический факультет

Кафедра «Информационные технологии и автоматизированные системы» направление подготовки: 09.03.04 – «Программная инженерия»

Лабораторная работа №5.

" Наследование. Виртуальные функции. Полиморфизм. "

15 вариант

Выполнил студент гр. РИС-24-2б

Иванова Елена Олеговна

Проверил:

Доц. Каф. ИТАС

Ольга Андреевна Полякова

(оценка) (подпись)

(дата)

г. Пермь, 2025

Цель:

1. Создание консольного приложения, состоящего из нескольких файлов в системе программирования Visual Studio.
2. Создание иерархии классов с использованием простого наследования и абстрактного класса.
3. Изучение полиморфизма и виртуальных методов.

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.

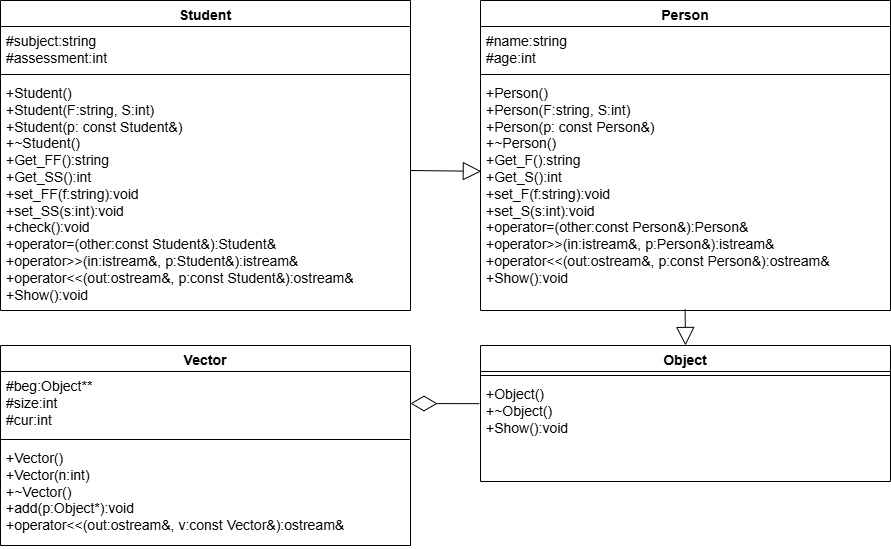
Задача:

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.

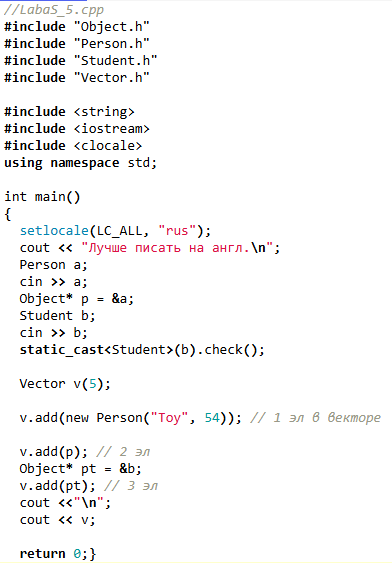
Анализ задачи.

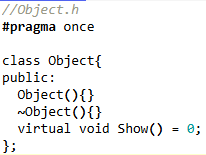
1. Взять классы из лр №4.
2. Определить класс Вектор, элементами которого будут указатели на объекты иерархии классов.
3. Перегрузить для класса Вектор операцию вывода объектов с помощью потоков.

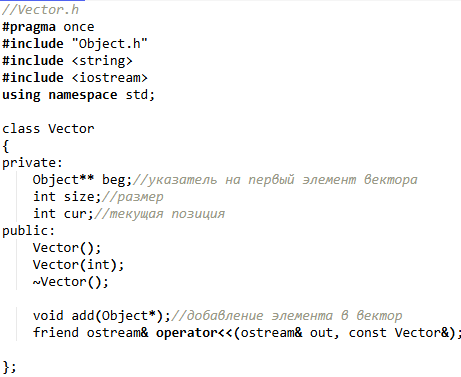
Диаграмма.

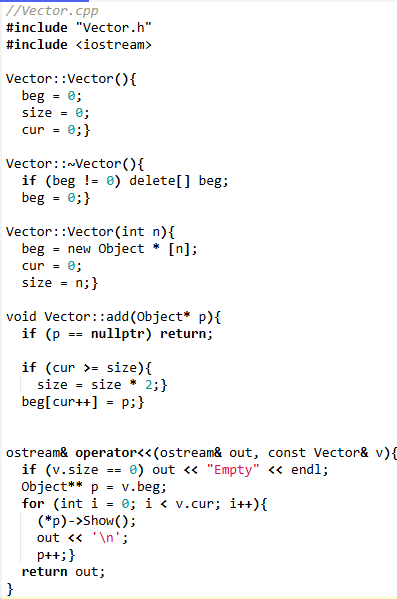


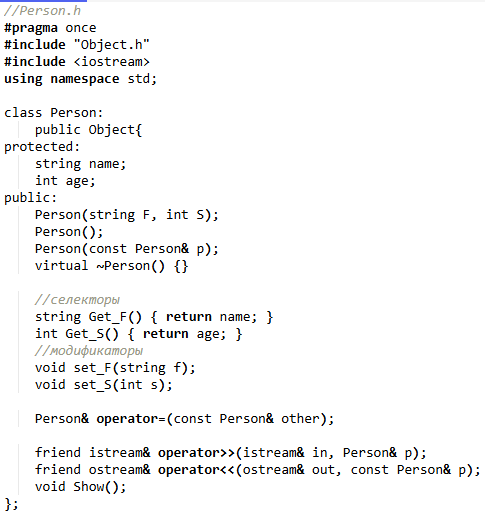
Код.

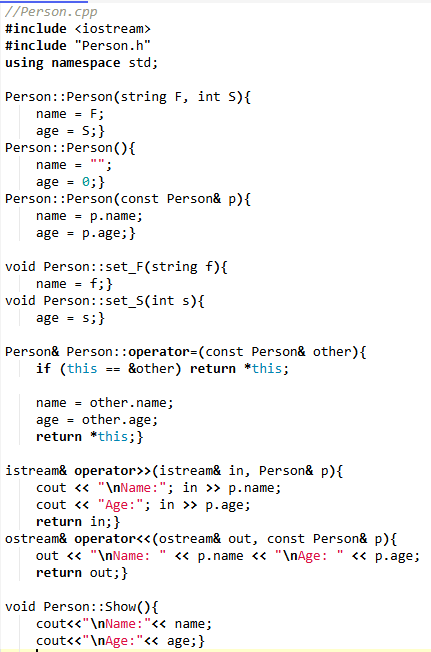


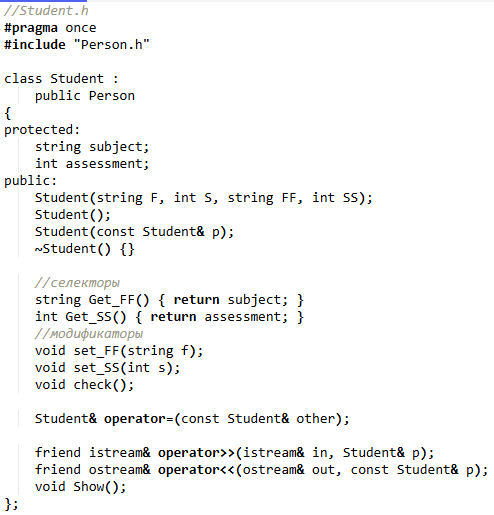


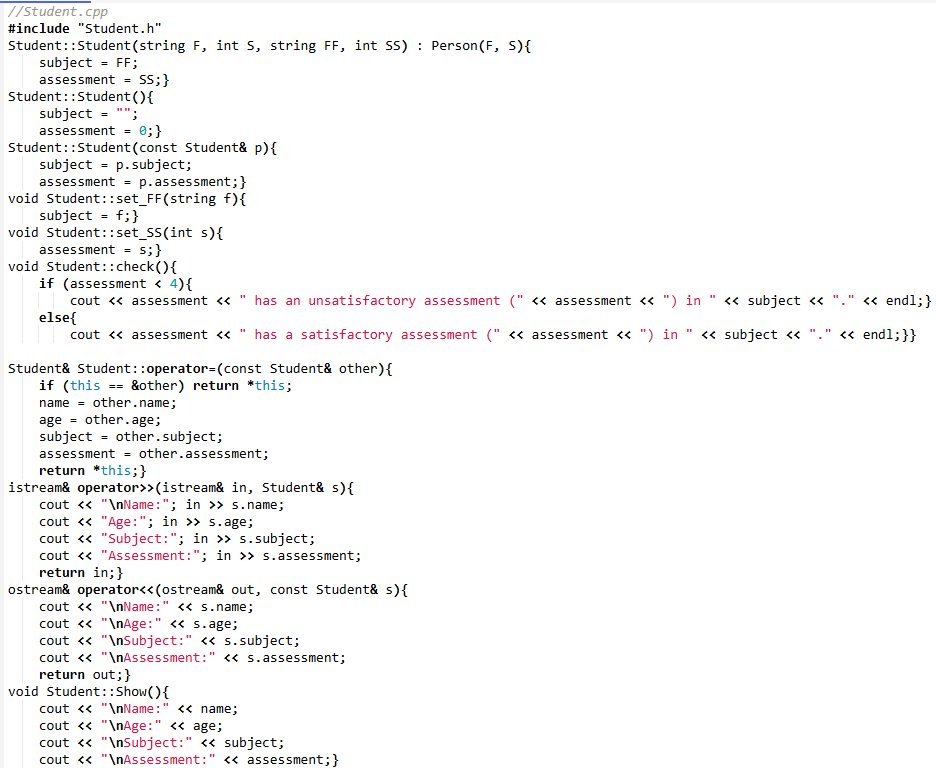




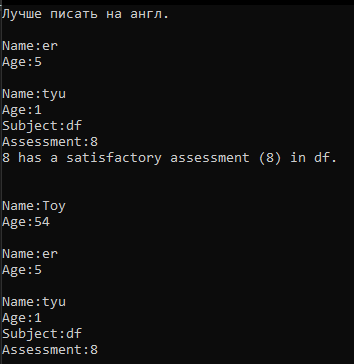








Вывод.



Контрольные вопросы:

1. Какой метод называется чисто виртуальным? Чем он отличается от виртуального метода?

Чисто виртуальный метод — это метод без реализации, который обязан быть переопределен в производном классе. Он делает класс абстрактным.

Виртуальный метод — это метод с реализацией по умолчанию, который может быть переопределен в производном классе. Он позволяет использовать полиморфизм.

1. Какой класс называется абстрактным?

Абстрактный класс — это класс, который содержит хотя бы один чисто виртуальный метод.

1. Для чего предназначены абстрактные классы?

Абстрактные классы предназначены для определения общего интерфейса и базовой функциональности, которую должны реализовать производные (конкретные) классы.

1. Что такое полиморфные функции?

Полиморфные функции — это функции, которые могут работать с объектами разных типов, предоставляя единый интерфейс для вызова.

1. Чем полиморфизм отличается от принципа подстановки?

Полиморфизм дает вызывать методы разных классов через один интерфейс.

Принцип подстановки говорит, что эти методы должны работать так, как ожидается, чтобы программа не сломалась.

1. Привести примеры иерархий с использованием абстрактных классов.

class Device {

protected:

string name;

bool isOn;

public:

Device(const string& n) : name(n), isOn(false) {}

virtual void use() = 0;

void turnOn() {

isOn = true;

cout << name << " is turned ON." << endl;}

void turnOff() {

isOn = false;

cout << name << " is turned OFF." << endl;}

virtual ~Device() {}};

class Lamp : public Device {

public:

Lamp(const string& n) : Device(n) {}

void use() override {

if (isOn) {

cout << "The lamp is shining brightly!" << endl;

} else {

cout << "Turn the lamp ON to use it." << endl;}}};

1. Привести примеры полиморфных функций.

class Animal {

public:

virtual void makeSound() const = 0;

virtual ~Animal() {}};

class Dog : public Animal {

public:

void makeSound() const override {

cout << "Woof!" << endl;}

};

class Cat : public Animal {

public:

void makeSound() const override {

cout << "Meow!" << endl;}

};

void makeAnimalsTalk(Animal\* animals[], int count) {

for (int i = 0; i < count; ++i) {

animals[i]->makeSound();}}

1. В каких случаях используется механизм позднего связывания?
2. Если функция базового класса объявлена virtual, то при вызове через указатель/ссылку на базовый класс будет вызвана реализация производного класса.
3. Если базовый класс содержит чисто виртуальные функции, то механизм позднего связывания гарантирует, что вызывается реализация из конкретного класса.
4. Позднее связывание позволяет модифицировать логику базового класса без изменения кода, который его использует.
5. Динамический полиморфизм лежит в основе многих паттернов.
6. Если класс предназначен для наследования, деструктор должен быть виртуальным, чтобы при удалении объекта через указатель на базовый класс вызывался правильный деструктор.