ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ПЕРМСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ

ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Лабораторная работа №4

Простое наследование. Принцип подстановки

Вариант №15

Выполнил студент группы РИС-23-3Б

Блинов А. Е.

Проверила доцент кафедры ИТАС

О. А. Полякова

2024 г.

**Постановка задачи**

1. Определить пользовательский класс.

2. Определить в классе следующие конструкторы: без параметров, с параметрами, копирования.

3. Определить в классе деструктор.

4. Определить в классе компоненты-функции для просмотра и установки полей данных (селекторы и модификаторы).

5. Перегрузить операцию присваивания.

6. Перегрузить операции ввода и вывода объектов с помощью потоков.

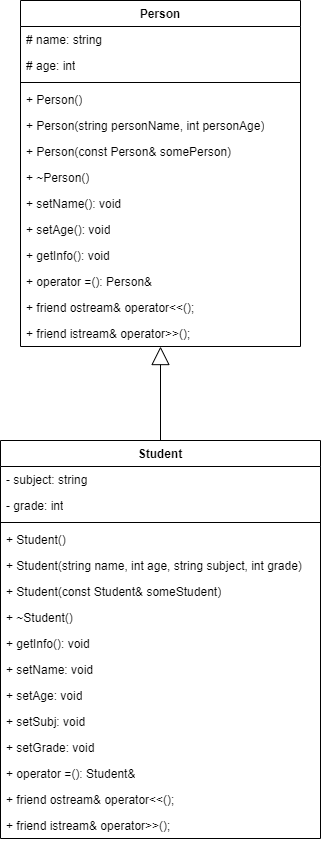
7. Определить производный класс.

8. Написать программу, в которой продемонстрировать создание объектов и работу всех перегруженных операций.

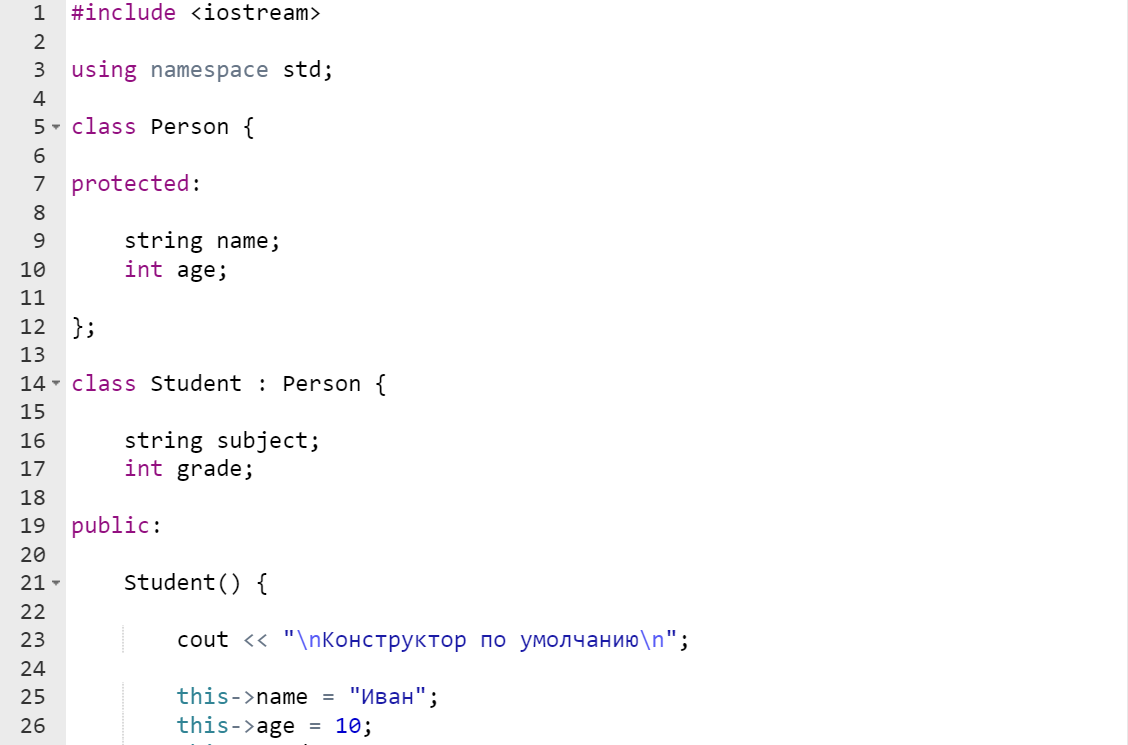
9. Реализовать функции, получающие и возвращающие объект базового класса. Продемонстрировать принцип подстановки.

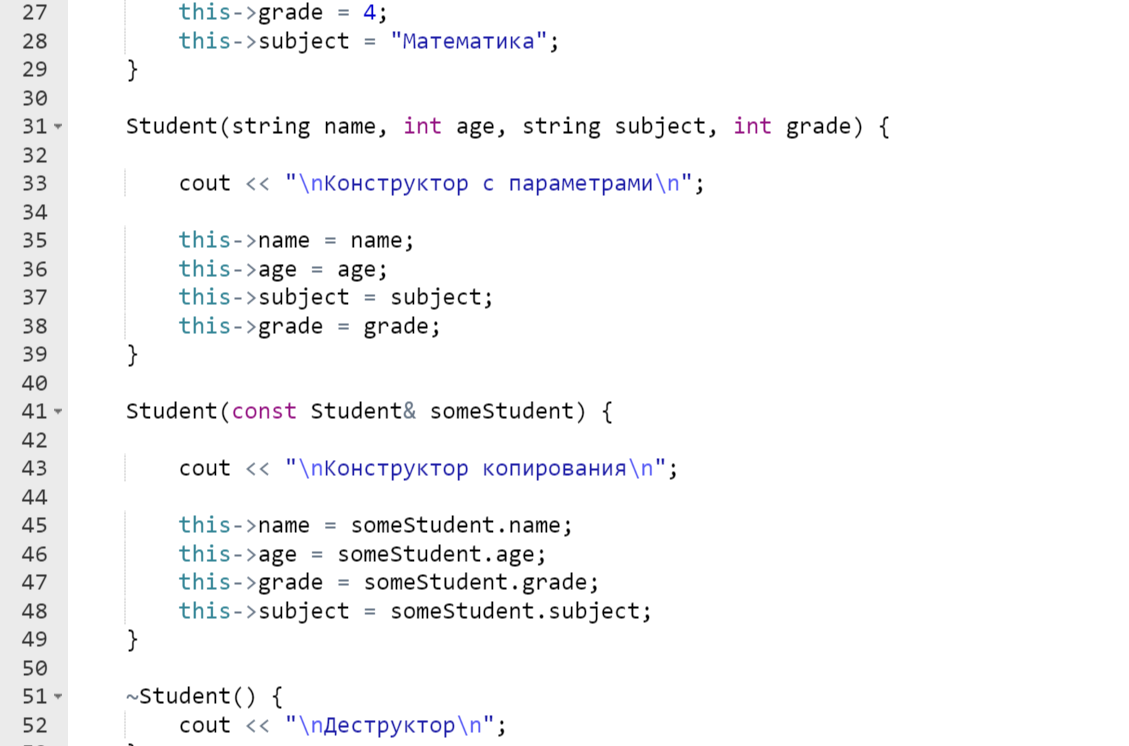
Базовый класс: ЧЕЛОВЕК (PERSON) Имя (name) – string Возраст (age) – int; Определить методы изменения полей. Создать производный класс STUDENT, имеющий поля: Предмет – string и Оценка – int. Определить методы изменения полей и метод, выдающий сообщение о неудовлетворительной оценке.

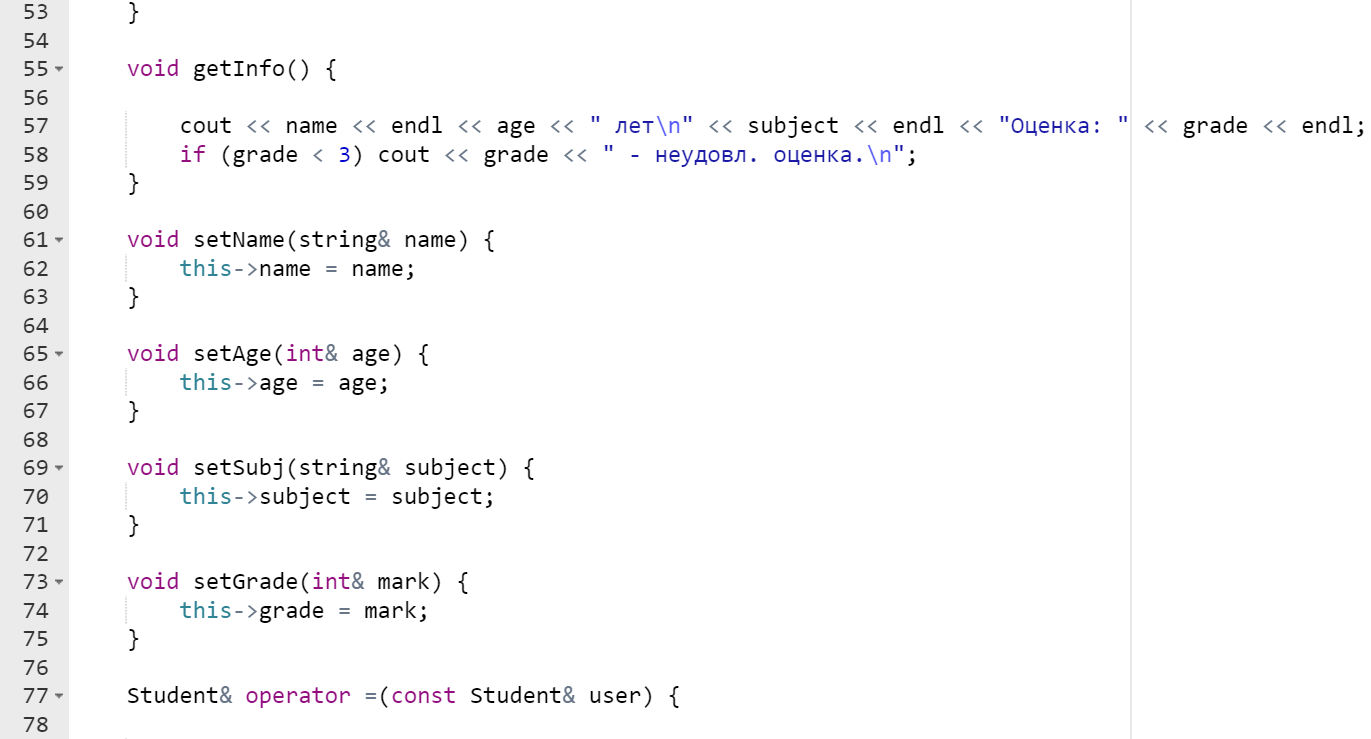
**UML диаграмма**

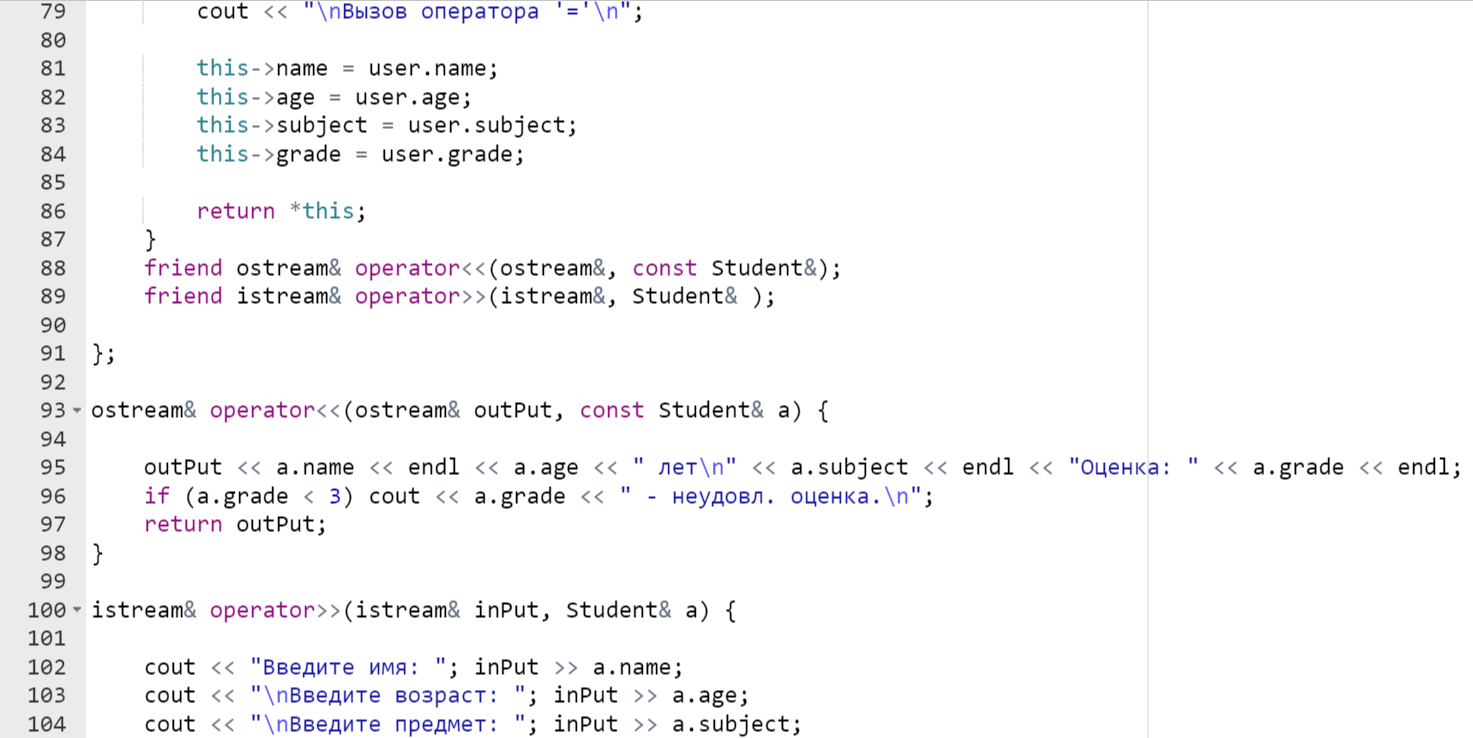
****

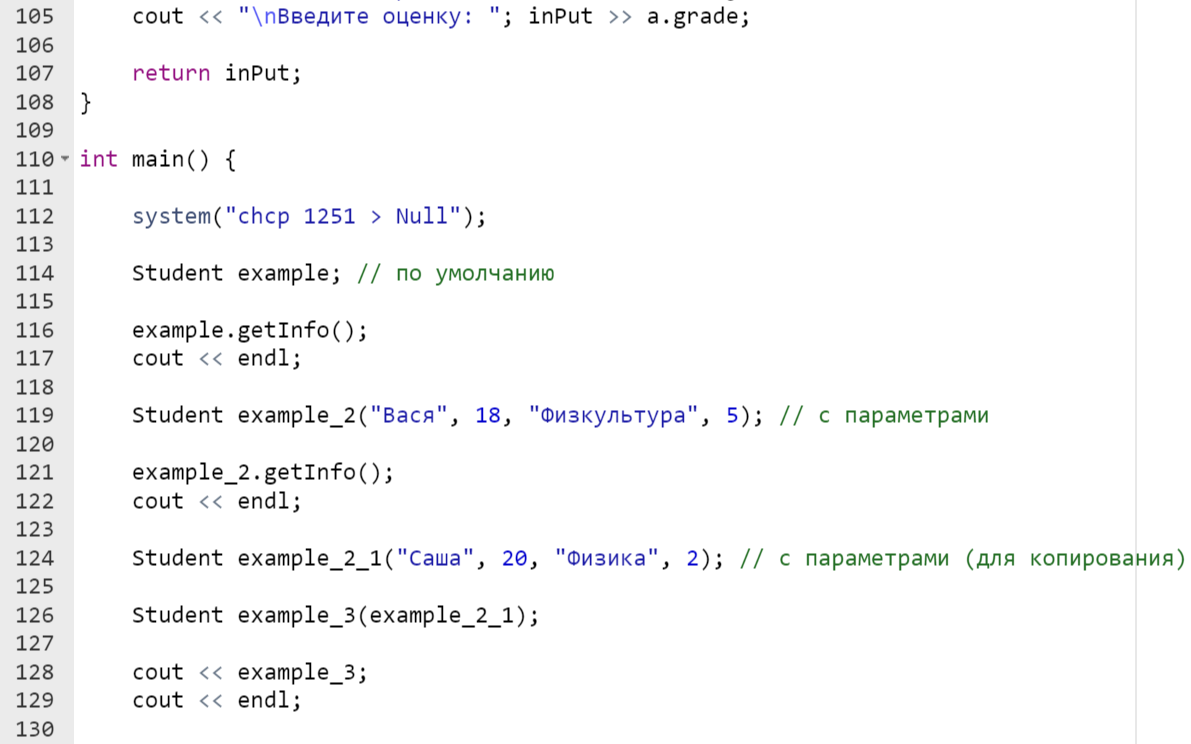
**Код на языке C++**

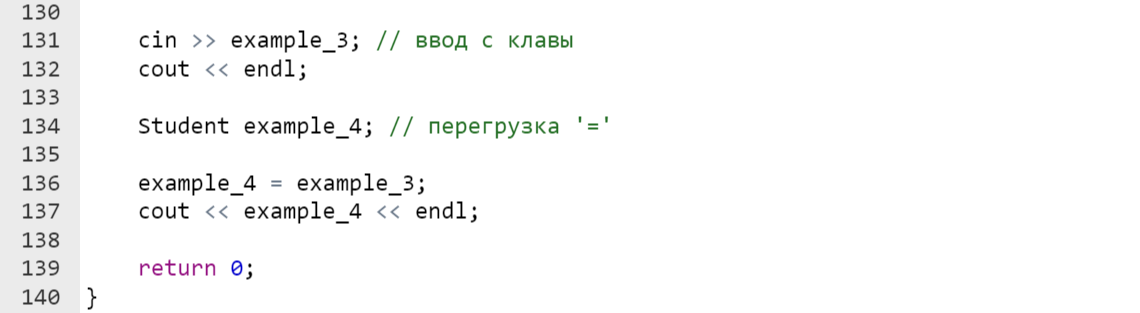
****

****

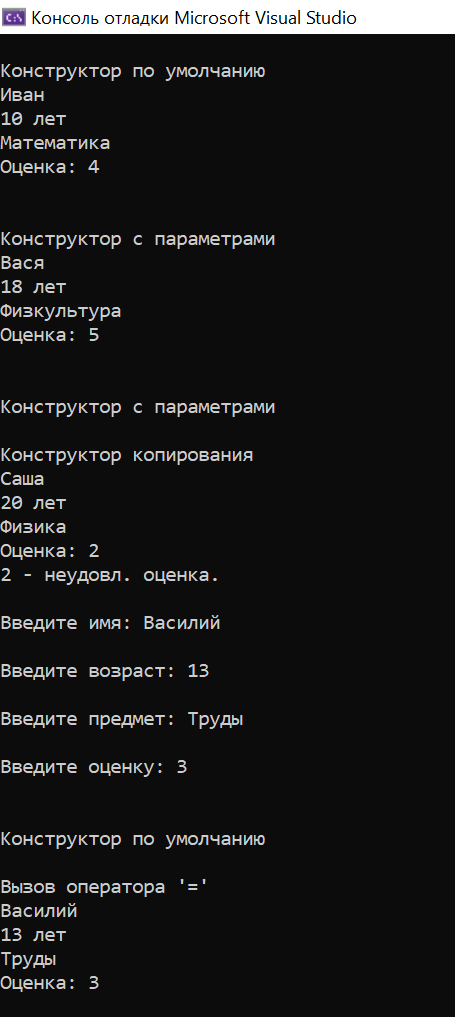
****

****

****

****

**Результаты работы программы**

****

**Выводы**

Программа работает успешно.

GitHub: <https://github.com/alexblin123/Laba-4-OOP.git>

**Контрольные вопросы**

**1. Для чего используется механизм наследования?**

Наследование - это механизм получения нового класса на основе уже существующего. Существующий класс может быть дополнен или изменен для создания нового класса.

Существующие классы называются **базовыми**, а новые – **производными**.

**2. Каким образом наследуются компоненты класса, описанные со спецификатором public?**

**public** – член класса может использоваться любой функцией, которая является членом данного или производного класса, а также к **public** - членам возможен доступ извне через имя объекта.

**3. Каким образом наследуются компоненты класса, описанные со спецификатором private?**

**private** – член класса может использоваться только функциями – членами данного класса и функциями – “друзьями” своего класса. В производном классе он недоступен.

**4**. **Каким образом наследуются компоненты класса, описанные со спецификатором protected?**

**protected**– то же, что и **private*,*** но дополнительно член класса с данным атрибутом доступа может использоваться функциями-членами и функциями – “друзьями” классов, производных от данного.

**5. Каким образом описывается производный класс?**

class имя\_класса : список\_базовых\_классов {список\_компонентов\_класса};

В производном классе унаследованные компоненты получают статус доступа private, если новый класс определен с помощью ключевого слова class, и статус public, если с помощью struct. (данные наследуются без изменения уровня доступа к ним)

**6. Наследуются ли конструкторы?**

Поскольку конструкторы не наследуются, при создании производного класса наследуемые им данные-члены должны инициализироваться конструктором базового класса.

Конструктор базового класса вызывается автоматически и выполняется до конструктора производного класса

**7. Наследуются ли деструкторы?**

В C ++ конструкторы и деструкторы не наследуются. Однако они вызываются, когда дочерний класс инициализирует свой объект. Конструкторы вызываются один за другим иерархически, начиная с базового класса и заканчивая последним производным классом. Деструкторы вызываются в обратном порядке.

**8. В каком порядке конструируются объекты производных классов?**

Объекты класса конструируются снизу-вверх: сначала базовый, потом компоненты-объекты (если они имеются), а потом сам производный класс.

Таким образом объект производного класса содержит в качестве подобъекта объект базового класса.

**9. В каком порядке уничтожаются объекты производных классов?**

Уничтожаются объекты в обратном порядке: сначала производный,

Потом его компоненты-объекты, а потом базовый объект

Таким образом, порядок уничтожения объекта противоположен по отношению к порядку его конструирования.

**10. Что представляют собой виртуальные функции и механизм позднего связывания?**

Виртуальная функция — это функция-член, которую предполагается переопределить в производных классах. К механизму виртуальных функций обращаются в тех случаях, когда в каждом производном классе необходимо, чтобы функции-компоненты работали иначе, чем базовом. Позднее связывание означает, что объект связывается с вызовом функции только во время исполнения программы, а не раньше.

**11. Могут ли быть виртуальными конструкторы? Деструкторы?**

Конструкторы не могут быть виртуальными в отличие от деструкторов. Практически каждый класс, имеющий виртуальную функцию, должен иметь виртуальный деструктор.

**12. Наследуется ли спецификатор virtual?**

Виртуальность также наследуется, метод объявленный как виртуальный в базовом классе, неявно становиться виртуальным в производном

**13. Какое отношение устанавливает между классами открытое наследование?**

Не понял вопрос

**14**. **Какое отношение устанавливает между классами закрытое наследование?**

Не понял вопрос

**15. В чем заключается принцип подстановки?**

Принцип подстановки можно сформулировать так: класс S может считаться подклассом T, если замена объектов T на объекты S не приведет к изменению работы программы.

Производные классы должны дополнять, а не заменять поведение базовых классов.