Министерство образования и науки Российской Федерации

Федеральное государственное образовательное автономное учреждение высшего образования

"Пермский национальный исследовательский политехнический университет"

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ №4

Дисциплина: Основы алгоритмизации и программирования

Тема: Простое наследование. Принцип подстановки.

Вариант: 13

|  |  |
| --- | --- |
| Выполнил работу | |
| Студент группы РИС-22-1б | |
| Кирпичников Илья | |
|  | |
| Проверил работу | |
| Доцент кафедры ИТАС | |
| Полякова О.А. | |
|  | |

Пермь – 2023

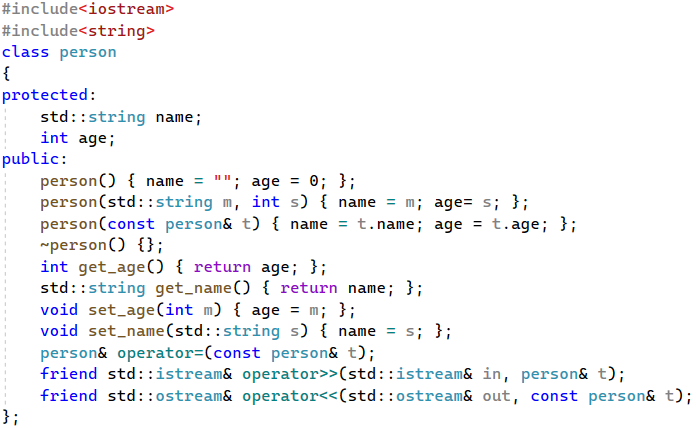
**Постановка задачи**

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.
10. Продемонстрировать принцип подстановки.

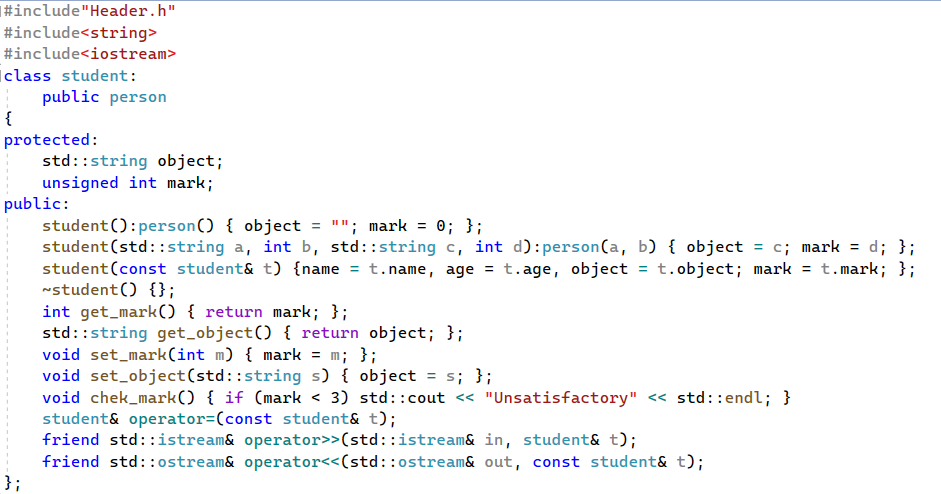
Вариант 15:

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

**Описание классов**

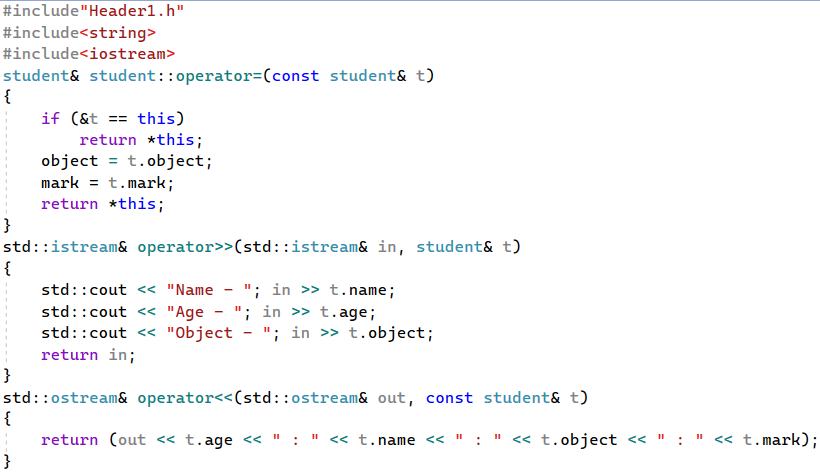
****

*Приложение 1 – Описание классов*

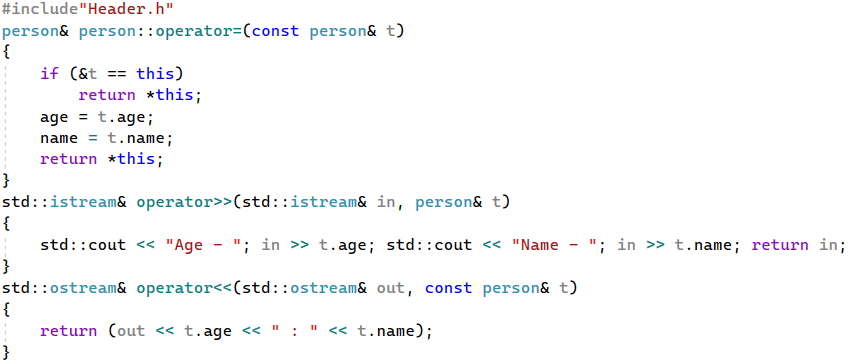
****

*Приложение 2 – Описание классов*

**Определение компонентных функций**

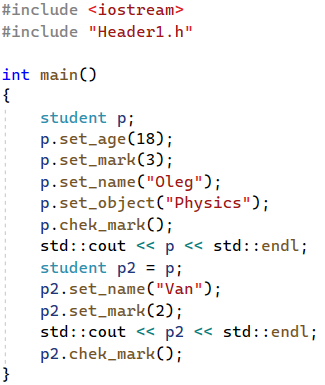
****

*Приложение 3 – Определение компонентных функций*

****

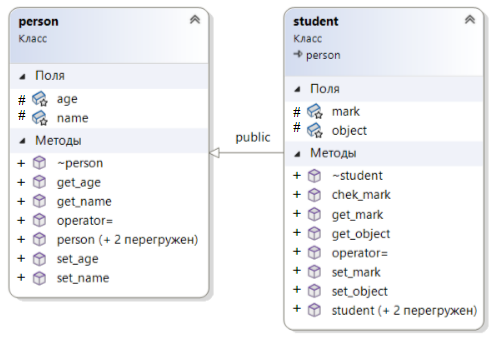
*Приложение 4 – Определение компонентных функций*

**Функция main()**

****

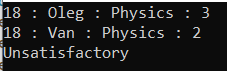
*Приложение 5 – Главная функция*

**UML-диаграмма**



*Приложение 6 – UML-диаграмма*

**Работа программы**



*Приложение 7 – Работа программы*

**Ответы на контрольные вопросы**

1. Механизм наследования используется для создания новых классов на основе уже существующих классов, путем наследования их свойств и методов.

2. Компоненты класса, описанные со спецификатором public, наследуются также со спецификатором public в производном классе.

3. Компоненты класса, описанные со спецификатором private, не наследуются в производном классе и не могут быть доступны из него.

4. Компоненты класса, описанные со спецификатором protected, наследуются также со спецификатором protected в производном классе.

5. Производный класс описывается с помощью ключевого слова class, за которым следует имя производного класса, затем двоеточие и имя базового класса или список базовых классов через запятую.

6. Конструкторы наследуются, но могут быть переопределены в производном классе.

7. Деструкторы наследуются, но могут быть переопределены в производном классе.

8. Сначала конструируется базовый класс, затем производный класс.

9. Сначала уничтожаются объекты производных классов, затем базового класса.

10. Виртуальные функции позволяют переопределять методы в производном классе, а механизм позднего связывания обеспечивает вызов правильного метода во время выполнения программы, даже если объект имеет тип базового класса.

11. Конструкторы не могут быть виртуальными, но могут иметь виртуальные методы. Деструкторы могут быть виртуальными.

12. Спецификатор virtual не наследуется.

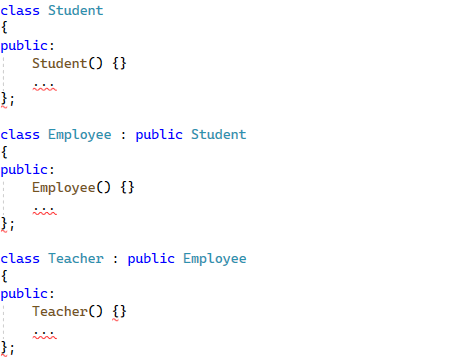
13. Открытое наследование устанавливает отношение "является", где производный класс является расширением базового класса.

14. Закрытое наследование устанавливает отношение "реализуется в терминах", где производный класс реализуется в терминах базового класса, но не является его расширением.

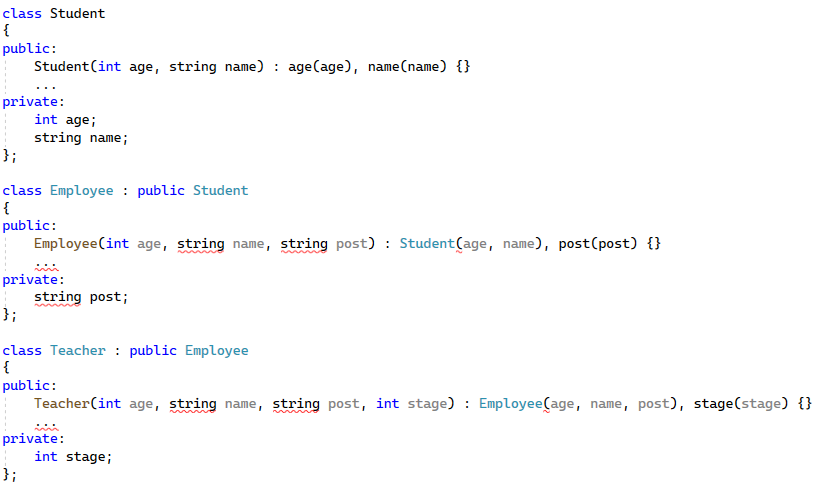
15. Принцип подстановки заключается в том, что объекты производного класса могут использоваться везде, где ожидается объект базового класса, без изменения поведения программы.

16. Объект х будет иметь компонентные данные: age (взятый от класса Student), name (взятый от класса Student), post (взятый от класса Employee) и stage (взятый от класса Teacher).

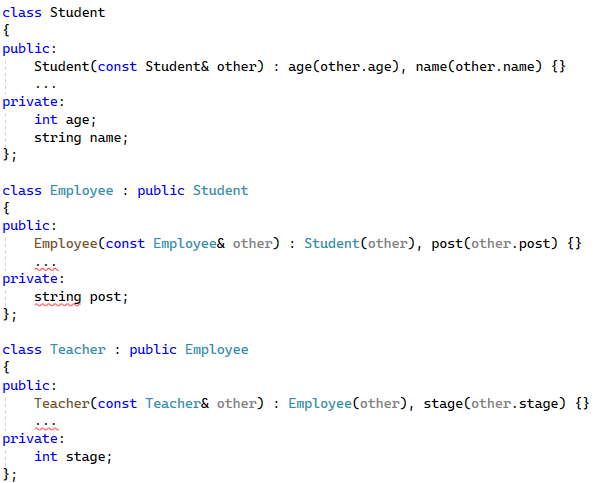
17. Конструкторы без параметров для классов:



18. Конструкторы с параметрами для классов:



19. Конструкторы копирования для классов:



20. Операция присваивания для классов:

