ФГАОУ ВО Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и технологий

Высшая школа киберфизических систем и управления

**ОТЧЕТ**

**по лабораторным работам №11,12**

**по темам «Замещение функций базового класса, конструкторы и деструкторы в производном классе»**

по дисциплине «Практикум по программированию»

Выполнил

студент гр.23533/2 А.И. Шибалов

Проверил

Ассистент В.Э. Ковалевский

Санкт-Петербург

2019

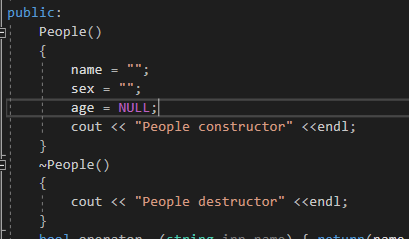
Задание

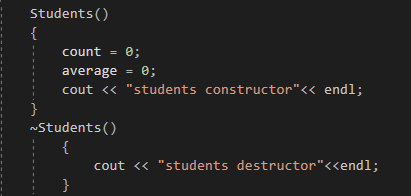
На примере лабораторной работы по работе со списками продемонстрировать замещение функций базового класса, а также конструкторы и деструкторы производных классов в консольном приложении на языке программирования C++.

Разработка программы

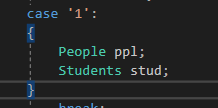
*Конструкторы и деструкторы*

Обратимся к классу People и Students, его наследующему. Их конструкторы и деструкторы выглядят так:





Инициализируем объект класса People и Student.



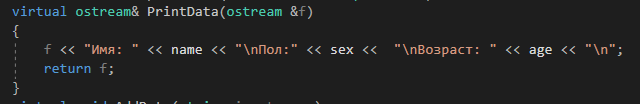
В консоли увидим, что сначала выполняется конструктор класса People для объекта ppl, затем конструктор People и конструктор Student для объекта stud. Деструкторы вызываются в обратном порядке.

*Замещение функций базового класса*

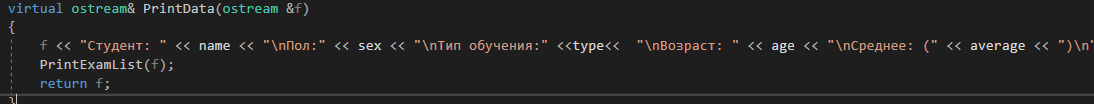
Обратимся опять же к классу People и классам Students и Academic, его наследующим.

В классах мы имеем функции PrintData с одинаковыми сигнатурами, но с разным содержанием. Функции выглядят следующим образом:

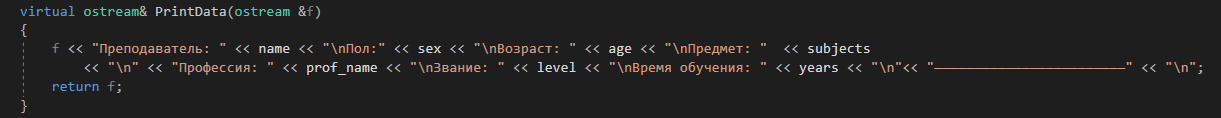
Класс People:



Класс Students:



Класс Academic:



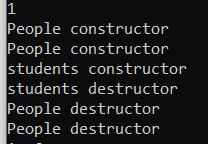
Дело в том, что в наследующем классе Students присутствуют дополнительные аргументы type и average, которые нужно вывести на экран, и соответственно вывод объектов этих классов будет отличаться.

Класс Academic так же имеет параметры, присущие только ему.

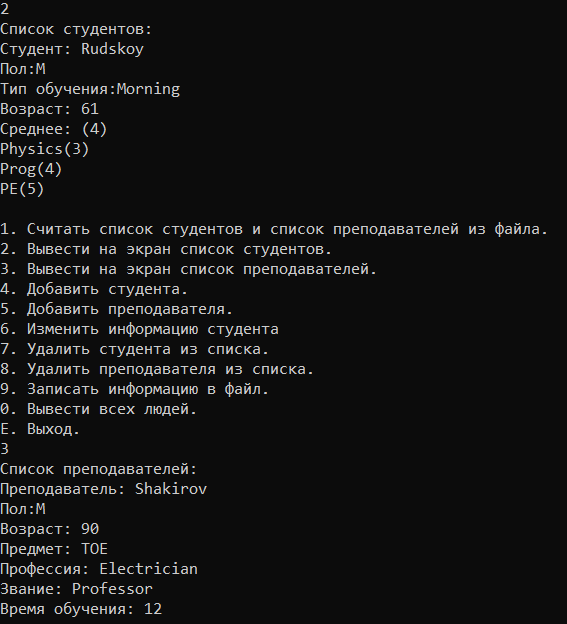
Выбор функции, которая будет выполняться, зависит от класса объекта, ее вызывающего.

Демонстрация работы

*Конструкторы и деструкторы*



*Замещение функций базового класса*

**

Заключение

В процессе выполнения лабораторной работы было продемонстрировано применение конструкторов и деструкторов классов, а также использование замещения функций базового класса.