Санкт-Петербургский исследовательский университет

информационных технологий, механики и оптики

**Практическая работа №6.**

**Объявление и реализация класса. Реализация инкапсуляции. Конструкторы и деструкторы.**

Выполнила:

Анисимова Ксения Алексеевна

Группа: К3120

Проверила: Филимонова И. А.

Санкт-Петербург

2019

**Цель работы:** научиться объявлять и реализовывать классы, реализовывать инкапсуляцию и узнать о конструкторах и деструкторах.

**Задачи:**

1) Научиться реализовывать сущность студент в виде класса.

2) Научиться разделять реализацию и представление.

3) Научиться объявлять и удалять объект.

4) Научиться использовать конструктор.

5) Научиться сохранять данные в файл после окончания работы программы.

**Ход работы:**

1) Создана программа, которая занимается учетом успеваемости студентов. Каждый студент является объектом класса со свойствами – имя (name), фамилия (last\_name), средний балл(average\_score). Для каждого свойства в классе прописаны методы set и get, пример показан на рисунке 1.

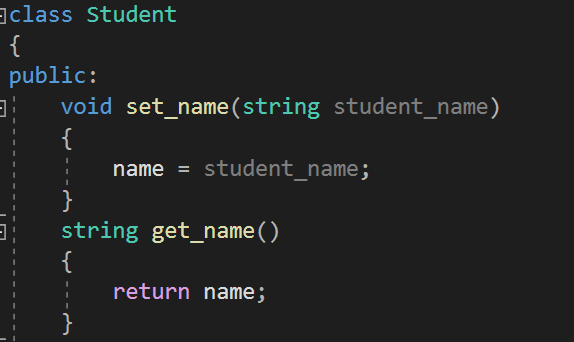


Рисунок 1 – Методы set и get.

Также студент имеет промежуточные оценки, хранящиеся в массиве из пяти элементов, по которым высчитывается средний балл студента.

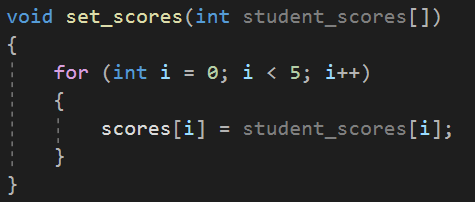


Рисунок 2 – Реализация метода.

В функции main () создан объект класса Student.



Рисунок 3 – Объявление переменной класса Student.

Организован ввод имени, фамилии студента и его промежуточных оценок с клавиатуры, после чего происходит передача введённых данных set-функциям.



Рисунок 4 – Пример передачи данных set-функции.

Отдельно происходит подсчёт среднего балла студента и вывод на экран данных о студенте.

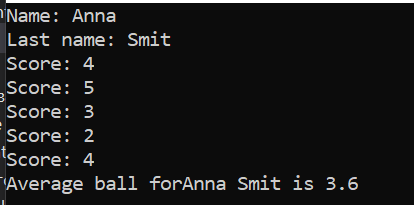


Рисунок 5 – Результат работы программы.

2) Разделена реализация класса от представления. Для этого все методы класса Student перенесены в отдельный файл student.cpp. После в проект добавлен заголовочный файл student.h, в котором указаны только прототипы этих методов. Также создан новый файл main.cpp, в котором указан код, создающий конкретного студента.

3) Создан объект класса Student, для которого выделена память в куче с помощью указателя, причём для доступа к методам и свойствам использовался оператор косвенного обращения ->



Рисунок 6– Выделение памяти в куче для объекта класса с помощью указателя.

После необходимо освободить память, занимаемую объектом.



Рисунок 7 – Освобождение памяти.

4) В класс Student добавлен конструктор, который принимает имя и фамилию ученика, и сохраняет эти значения в соответствующих переменных класса.

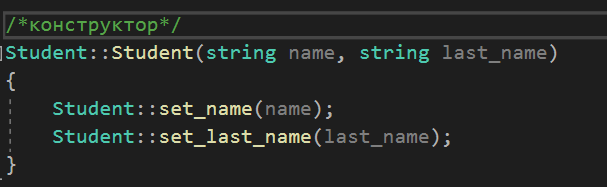


Рисунок 6 – Создание конструктора класса.

Теперь при создании объекта необходимо сразу передавать конструктору параметры.

5) Реализована возможность сохранения данных в текстовый файл в заданном формате после окончания работы программы. Для этого прописан отдельный метод save, вызываемый в деструкторе.

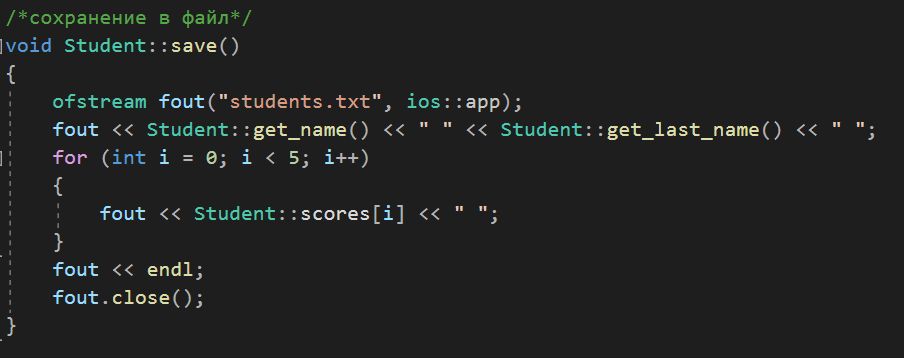


Рисунок 7 – Реализация метода save.

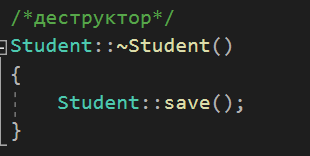


Рисунок 8 – Реализация деструктора.

**Вывод:** в ходе работы изучены механизмы реализации класса, конструктора и деструктора, разделения реализации и представления, объявления и удаления объекта и сохранения данных в файл.