

Задачи на работу с классами. Часть 2

4. Класс **StudentGroup**

Создать класс StudentGroup, содержащий в себе информацию о студентах (использовать написанный ранее класс Student). Фактически класс хранит в себе массив данных типа Student.

Реализовать:

- 1) Конструкторы класса
 - a) Конструктор по умолчанию
 - b) Конструктор копирования
- 2) Деструктор класса
- 3) Методы класса
 - a) Добавление элемента в конец
 - b) Удаление элемента по индексу
 - c) Метод, возвращающий индекс элемента по фамилии студента (–1 в случае, если нет такого студента с списке)
 - d) Печать всех элементов списка
- 4) Перегрузить операторы
 - a) Присваивания
 - b) StudentGroup& operator<<(StudentGroup& group, Student const& student); – добавление элемента в список (с помощью внешней функции)
 - c) оператор доступа по индексу (константная и неконстантная версии)

5. Класс **Time**

Создать класс Time, представляющий собой время (часы (0 – 23), минуты (0 – 59), секунды (0 – 59)).

Реализовать:

- 1) Конструкторы класса
 - a) Конструктор по умолчанию, задающий поля нулями
 - b) Конструктор, принимающий строку в формате HH:MM:SS (часы, минуты, секунды, разделенные двоеточием). Если в строке ошибка, вывести на консоль, поля инициализировать нулями
 - c) Конструктор, принимающий число часов (остальные поля заполнить нулями)
 - d) Конструктор копирования
- 2) Методы класса
 - a) Устанавливающие и читающие поля класса (часы, минуты, секунды)
 - b) Добавляющие секунду, минуту, час (три отдельных метода)
 - c) Печатающий время в формате HH:MM:SS
 - d) bool IsAm() const; – возвращает true, если время до полудня
 - e) Вычисляющий кол-во секунд с полуночи
- 3) Перегрузить операторы
 - a) Присваивания
 - b) Сложения (если в результате количество часов будет ≥ 24 , отнять из результата сутки)
 - c) Инкремента и декремента(увеличивает/уменьшает на одну секунду)
 - d) Операторы сравнения (==, !=, <, >, <=, >=)