

Контрольные вопросы:

- ☐ (5 б.) Как организована иерархия классов потоков в библиотеке `IOStream`?
- ☐ (5 б.) Какие состояния потоков реализованы в базовом классе `basic_ios`?
- ☐ (5 б.) В чем разница между манипуляторами и флагами форматирования?
- ☐ (5 б.) Из каких основных элементов состоят пути в файловой системе?
- ☐ (5 б.) Зачем нужны форматы обмена данными, такие как JSON и XML?

Упражнения:

- ☐ (25 б.) Напишите программу, которая удаляет все стандартные комментарии из некоторого файла исходного кода. Комментарии в коде могут быть как однострочными, так и многострочными, но не могут быть вложены друг в друга. Не забудьте также реализовать корректную обработку имеющихся в коде строковых литералов.
- ☐ (25 б.) Создайте текстовый файл и заполните его строками одинаковой длины. Напишите программу, которая выводит в стандартный поток вывода строку с задаваемым пользователем номером из данного файла. Для реализации подобного механизма воспользуйтесь функцией позиционирования в потоках данных [seekg](#).
- ☐ (25 б.) Напишите программу, которая определяет содержимое заданной пользователем существующей на компьютере директории. Для каждого вхождения выполните проверки и выведите в стандартный поток вывода имя, тип и время последнего изменения, которое можно получить из функции [last_write_time](#).
- ☐ (25 б.) Реализуйте структуру для хранения некоторой базовой разнотипной информации о человеке и перегрузите для нее оператор ввода. Создайте несколько экземпляров вашей структуры на основе данных, вводимых пользователем в стандартном потоке ввода. Преобразуйте созданные экземпляры в соответствующий набор JSON объектов. Программно создайте собственный отдельный каталог в рабочей директории проекта и сохраните каждый полученный JSON объект в отдельном текстовом файле в вашем каталоге.