

Курс: Объектно-ориентированное программирование на C++

Встреча №24

ТЕМА: ПОТОКИ ЧАСТЬ 2

Задания для самостоятельной работы:

Задание №1

К уже существующему проекту класса *Student* добавьте механизмы сохранения/загрузки данных класса. Используйте механизм потоков для файловых операций.

Задание №2

К уже существующему проекту класса *Дробь* добавьте механизмы сохранения/загрузки данных класса. Используйте механизм потоков для файловых операций.

Курс: Объектно-ориентированное программирование на C++

Встреча №25

ТЕМА: СТРОКИ И УМНЫЕ УКАЗАТЕЛИ

Задания для самостоятельной работы:

Задание №1

Реализуйте программу, которая предоставляет функциональность по работе с файлами. Приложение получает у пользователя путь к файлу, после чего программа отображает интерфейс по анализу содержимого файла.

Возможности интерфейса:

- Поиск заданной строки (по итогам показывать статистику поиска строки по всему файлу);
- Замена строки на новую строку (по итогам показать статистику по замене строки);
- Отображение файла на экран;
- Переворот содержимого файла.

Используйте для работы с файлами потоки, а для работы со строками класс *string*.

Задание №2

Создайте приложение «Валидатор HTML-файлов». Приложение запрашивает у пользователя путь к файлу, после чего проверяет файл на валидность. Критерием валидности для вашего проекта являются правила: все открытые теги должны закрываться, если был знак < его должен закрыть знак >. По итогам валидации

нужно отобразить результат проверки на экран. Используйте для работы с файлами потоки, а для работы со строками класс *string*.

Примеры:

Содержимое валидного файла

```
<html>  
<body>To be or not to be</body>  
</html>
```

Содержимое невалидного файла

```
<html>  
<body To be or not to be</body>  
</html>
```

**Курс: Объектно-ориентированное
программирование на C++****Встреча №26****ТЕМА: STL. ЧАСТЬ 1**

Задания для самостоятельной работы:

Задание №1

Написать программу «Автоматизированная информационная система ЖД вокзала». Система содержит: сведения об отправлении поездов дальнего следования. Для каждого поезда указываем: номер, время отправления, станцию назначения. Обеспечить:

- Ввод данных в систему;
- Вывод информации о всех поездах;
- Вывод информации о запрашиваемом поезде.

Использовать контейнерный класс *vector*

Задание №2

Используя контейнер *map* реализовать программу – словарь. Основные возможности:

- поиск слова;
- добавление, удаление, редактирование слов;
- чтение и запись словаря в файл.