

## Лабораторная работа 5, часть 2

Черновик 0.8

Целью работы является знакомство студентов со структурами и их использование.

Студенты должны получить и закрепить на практике следующие знания и умения:

1. Описание структурного типа.
2. Обработка массивов.
3. Обработка строк.
4. Обработка текстовых и двоичных файлов.
5. Организация корректной работы с ресурсами (файловые описатели).
6. Использование в программе аргументов командной строки.

### Общее задание

1. Исходный код лабораторной работы располагается в ветке lab\_52. В этой ветке создается папка lab\_52\_1, в которой располагается решение для варианта 1, и папка lab\_52\_2, в которой располагается решение для варианта 2.
2. Исходный код должен соответствовать правилам оформления исходного кода.
3. Для каждого варианта создается отдельный проект в *QT Creator*. Для каждого проекта должно быть два варианта сборки: Debug (с отладочной информацией) и Release (без отладочной информации).

#### *Замечание*

По согласованию с преподавателем, проводящим практические занятия, вы можете использовать другую среду разработки (например, Microsoft Visual Studio Code), но два варианта сборки проекта нужно предусмотреть в любом случае.

4. Созданный проект обязательно должен быть многофайловым.
5. Вариант, полнота его реализации (количество задач), типы обрабатываемых файлов (текстовый/двоичный) определяются преподавателем.
6. При работе с текстовым файлом данные сначала загружаются в массив, затем обрабатываются, выводятся на экран и при необходимости сохраняются в файл. При работе с двоичным файлом данные обрабатываются непосредственно в файле.
7. В текстовом файле на каждой строке располагается значение одного поля структурного типа.
8. Поля в структурном типе описываются в том порядке, в котором указано в задании. Структурный тип никак не упаковывается.
9. При вводе информации для формирования переменной структурного типа значения полей вводятся в том порядке, в котором указано в задании.
10. При выводе информации из переменной структурного типа на экран или в текстовый файл поля выводятся в том порядке, в котором указано в задании. Каждое поле располагается на отдельной строке.

11. В случае если ваше приложение запущено с неправильными или незнакомыми параметрами командной строки, приложение должно возвращать код ошибки 53.
12. Для каждой реализованной задачи подготавливаются тестовые данные, которые демонстрируют правильность работы программы. Входные данные должны располагаться в файлах in\_z.txt(bin), выходные out\_z.txt(bin), где z – номер тестового случая. Тестовые данные готовятся и помещаются под версионный контроль еще до того, как появится реализация задачи.
13. Реализовав очередную задачу и проверив правильность ее работы, оцените полноту подготовленных тестовых данных на основе процента покрытия кода этими данными. Добейтесь 100% покрытия кода тестовыми данными. Если это невозможно, необходимо это обосновать.

## Индивидуальное задание

### Вариант 1

Задана следующая информация о студентах:

- фамилия (максимум 25 символов);
- имя (максимум 10 символов);
- оценки по четырем предметам (массив, элементы которого имеют тип “целое беззнаковое длиной 32 бита”).

Написать программу, которая реализует следующие функции:

1. Упорядочивает информацию о студентах по фамилиям. При этом информация о студентах с одинаковой фамилией должна быть упорядочена по имени. Результат выводится на экран. Запуск программы: «app.exe st file\_in» в случае текстового файла или «app.exe sb file\_in» в случае двоичного файла.
2. Выводит информацию о студентах, фамилия которых начинается с заданной подстроки, в другой файл. Запуск программы: «app.exe ft file\_in file\_out substr» в случае текстовых файлов или «app.exe fb file\_in file\_out substr» в случае бинарных файлов.
3. Для каждого студента вычисляет средний балл. Затем вычисляет среднее значение среднего балла и удаляет студентов, средний балл которых меньше среднего значения. Результат записывается в тот же файл. Запуск программы: «app.exe dt file\_in\_out» в случае текстового файла или «app.exe db file\_in\_out» в случае двоичного файла.

### Вариант 2

Задана следующая информация о товаре:

- наименование (максимум 30 символов);
- изготовитель (максимум 15 символов);
- цена единица товара (целое беззнаковое длиной 32 бита);
- количество единиц товара (целое беззнаковое длиной 32 бита).

Написать программу, которая реализует следующие функции:

1. Упорядочивает информацию о товаре по убыванию цены за единицу товара. При этом информация о товарах с одинаковой ценой должна быть упорядочена по количеству единиц (сначала должны идти товары, которых больше). Результат сохраняется в другой файл. Запуск программы: «app.exe st file\_in file\_out» в случае текстовых файлов или «app.exe sb file\_in file\_out» в случае бинарных файлов.

2. Выводит на экран информацию о товарах, наименование которых заканчивается на заданную подстроку. Запуск программы: «app.exe ft file\_in substr» в случае текстового файла или «app.exe fb file\_in substr» в случае бинарного файла.
3. Добавляет информацию о новом товаре в уже упорядоченную последовательность товаров так, чтобы упорядоченность не нарушилась. Информация о новом товаре запрашивается у пользователя. Результат записывается в тот же файл. Запуск программы: «app.exe at file\_in\_out» в случае текстового файла или «app.exe at file\_in\_out» в случае бинарного файла. Повторное использование сортировки не считается правильным решением задачи.