Титульный лист материалов по дисциплине

ДИСЦИПЛИНА **Структуры и алгоритмы обработки данных (ч. 2)**
(полное наименование дисциплины без сокращений)

ИНСТИТУТ **ИТ**

КАФЕДРА **Математического обеспечения и стандартизации информационных технологий**
полное наименование кафедры)

ВИД УЧЕБНОГО МАТЕРИАЛА **Практические работы**
(в соответствии с пп.1-11)

ПРЕПОДАВАТЕЛЬ **Красников С.А., Рысин М.Л., Скворцова Л.А., Туманова М.Б., Макеева О.В., Сартаков М.В.**
(фамилия, имя, отчество)

СЕМЕСТР **3 семестр, 2023-2024 уч. год**
(указать семестр обучения, учебный год)

.

СОДЕРЖАНИЕ

1 ПРАКТИЧЕСКАЯ РАБОТА 2

Тема: Внешние структуры данных: текстовый и двоичный файлы.

Цель: Получить навыки применения файловых потоков языка C++ (или файлов языка Си) по управлению текстовым и двоичным файлами.

2 ЗАДАНИЕ 1

2.1 Формулировка

Разработать программу, управления текстовым файлом

2.2 Требования

1) Реализация ввода-вывода на основе файловых потоков C++: ofstream, ifstrim.
2) Имя физического файла вводится пользователем и передается в функции обработки через параметр.
3) При открытии файла выполнять контроль его существования и открытия. Примечание. При отладке программы можете имя физического файла определить через константу.
4) Разработать функции для выполнения операций над текстовым файлом.

— создание текстового файла средствами текстового редактора кодировки ASCII, содержащего десятичные числа по несколько чисел на строке;
— вывод содержимого текстового файла;
— добавление новой записи в конец файла;
— прочитать значение числа, указав его порядковый номер в файле, и вернуть его значение;
— определить количество чисел в файле.

5) Разработать приложение и выполнить тестирование всех функций. Приложение должно содержать диалоговый интерфейс на основе текстового меню.
6) Контроль открытия и существования файла выполнить в основной программе перед вызовом функции. Перед закрытием файла, проверить отсутствие ошибок ввода и вывода (метод good)/

7) Создать модуль и перенести в него все отлаженные функции. Исключить функции из приложения. Отладить приложение, подключив к нему модуль с функциями.
8) Разработать функции для реализации дополнительных операций, определенных вариантом и сохранить их в модуле с остальными функциями.
9) Выполнить тестирование приложения в полном объеме.

Дополнительное задание варианта:

Номер: Задание: Дополнительные операции

2.3 Тестовый пример

Тестовый пример файла приведён в листинге 2.1.

Листинг 2.1 – Копия содержания текстового файла на примере 20 записей

2.4 Реализация приложения

2.4.1 Функционал приложения

Прототипы функций, реализующих операции задания приведены в листингах (2.2-2.6) [Нумерация листингов была сбита, после 1 следует 3] #

2.4.1.1 Функция

Реализует: Вывод содержимого текстового файла Предисловие:

Листинг 2.2 – Функция _НАЗВАНИЕ_ФУНКЦИИ_

Постусловие

Тестирование функции

Исходный файл: Приведён в листинге 2.1. Ожидаемый результат:

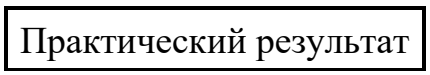

Рисунок 2.1 – Практический результат

2.4.1.2 Функция

Реализует: Добавление новой записи в конец файла Предисловие:

Листинг 2.3 – Функция _НАЗВАНИЕ_ФУНКЦИИ_

Постусловие

Тестирование функции

Исходный файл: Приведён в листинге 2.1. Ожидаемый результат:

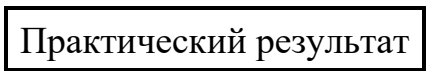

Рисунок 2.2 – Практический результат

2.4.1.3 Функция

Реализует: Чтение значения числа согласно его порядковому номеру Предисловие:

Листинг 2.4 – Функция _НАЗВАНИЕ_ФУНКЦИИ_

Постусловие

Тестирование функции

Исходный файл: Приведён в листинге 2.1. Входные данные: Ожидаемый результат:

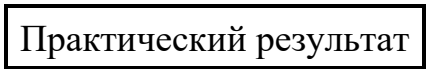

Рисунок 2.3 – Практический результат

2.4.1.4 Функция

Реализует: Определение количества чисел в файле Предисловие:

Листинг 2.5 – Функция _НАЗВАНИЕ_ФУНКЦИИ_

Постусловие

Тестирование функции

Исходный файл: Приведён в листинге 2.1. Ожидаемый результат:

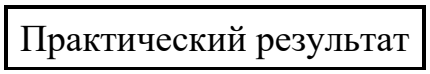

Рисунок 2.4 – Практический результат

2.4.1.5 Функция

Реализует: Определение количества чисел в файле Предисловие:

Листинг 2.6 – Функция _НАЗВАНИЕ_ФУНКЦИИ_

Постусловие

Тестирование функции

Исходный файл: Приведён в листинге 2.1. Ожидаемый результат:

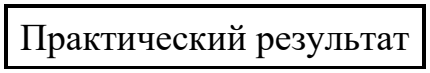

Рисунок 2.5 – Практический результат

2.4.2 Код основной программы

Код основной программы приведён в листинге 2.7

Листинг 2.7 – Код основной программы

2.5 Результаты тестирования

Скриншоты тестов каждой функции приведены на рисунках (2.1 - 2.5) Все тесты

3 ЗАДАНИЕ 2

3.1 Условие задания

3.1.1 Формулировка условия

Разработать программу управление двоичными файлами с записями фиксированной длины.

3.1.2 Требования задания

Общие требования: файл состоит из записей определенной структуры, согласно варианту. Записи имеют ключ, уникальный в пределах файла.

Требования к подготовке и выполнению задания

1) Разработать структуру записи двоичного файла согласно варианту задания.
2) Подготовить тестовые данные в текстовом файле с кодировкой ASCII, в соответствии со структурой записи варианта. При открытии файла выполнить контроль его существования и открытия. Примечание. Реализация операций по чтению данных из файла будет проще, если значение для каждого поля записи размещать на отдельной строке текстового редактора.
3) Имя файла вводит пользователь.
4) При открытии файла обеспечить контроль существования и открытия файла.
5) При применении механизма прямого доступа к записи файла выполнить контроль присутствия записи с заданным номером в файле.
6) Разработать функции для выполнения операций:

— преобразование тестовых данных из текстового файла в двоичный файл;

— сохранение данных двоичного файла в текстовом, так, чтобы используя их можно было восстановить двоичный файл;
— вывод всех записей двоичного файла;
— доступ к записи по ее порядковому номеру в файле, используя механизм прямого доступа к записи в двоичном файле;
— удаление записи с заданным значением ключа, выполнить путем замены на последнюю запись.
— манипулирование записями в двоичном файле согласно дополнительным операциям, определенным в варианте;

7) Сохраните функции в новом модуле.
8) Разработать приложение, демонстрирующее выполнение всех операций, подключив к нему модуль с функциями.
9) Выполнить тестирование приложения, продемонстрировав выполнение всех операций.

3.1.3 Формулировка задания варианта

Дополнительное задание варианта:

Вариант: Структура записи: Дополнительные операции: 1. 2.

3.2 Тестовый пример

Копия содержания текстового файла на примере 5 записей для преобразования в двоичный файл приведена в листинге 3.1.

Листинг 3.1 – Содержание текстовый файл

Содержание двоичного файла (результат операции вывода двоичного файла на экран), полученного из данных текстового файла приведено в листинге 3.2.

Листинг 3.2 – Содержание двоичного файла

3.3 Реализация приложения

3.3.1 Структура записи двоичного файла (из кода) и ее размер в байтах (ручной расчет и системный)

Структура приведена в листинге 3.3

Листинг 3.3 – Структура записи двоичного файла

Ручной расчёт размера файла в байтах:

Системный расчёт размера файла в байтах:

3.3.2 Изображение структуры двоичного файла с записями фиксированной длины.

Структура изображена в листинге 3.4

Листинг 3.4 – Изображение структуры двоичного файла

3.3.3 Функционал приложения

Прототипы функций, реализующих операции задания приведены в листингах (3.5-3.11)

3.3.3.1 Функция

Реализует: Преобразование тестовых данных из текстового файла в двоичный файл Предисловие:

Листинг 3.5 – Функция _НАЗВАНИЕ_ФУНКЦИИ_

Постусловие

Тестирование функции

Исходный файл: Приведён в листинге 3.1. Ожидаемый результат: Приведён в листинге 3.2

Практический результат

Рисунок 3.1 – Практический результат

3.3.3.2 Функция

Реализует: Сохранение данных двоичного файла в текстовом, так, чтобы используя их можно было восстановить двоичный файл Предисловие:

Листинг 3.6 – Функция _НАЗВАНИЕ_ФУНКЦИИ_

Постусловие

Тестирование функции

Исходный файл: Приведён в листинге 3.2. Ожидаемый результат: Приведён в листинге 3.1

Практический результат

Рисунок 3.2 – Практический результат

3.3.3.3 Функция

Реализует: Вывод всех записей двоичного файла Предисловие:

Листинг 3.7 – Функция _НАЗВАНИЕ_ФУНКЦИИ_

Постусловие

Тестирование функции

Исходный файл: Приведён в листинге 3.2. Ожидаемый результат:

Практический результат

Рисунок 3.3 – Практический результат

3.3.3.4 Функция

Реализует: Доступ к записи по её порядковому номеру в файле, используя механизм прямого доступа к записи в двоичном файле Предисловие:

Листинг 3.8 – Функция _НАЗВАНИЕ_ФУНКЦИИ_

Постусловие

Тестирование функции

Исходный файл: Приведён в листинге 3.2. Входные данные: Ожидаемый результат:

Практический результат

Рисунок 3.4 – Практический результат

3.3.3.5 Функция

Реализует: Удаление записи с заданным значением ключа, путем замены на последнюю запись Предисловие:

Листинг 3.9 – Функция _НАЗВАНИЕ_ФУНКЦИИ_

Постусловие

Тестирование функции

Исходный файл: Приведён в листинге 3.2. Входные данные: Ожидаемый результат:

Практический результат

Рисунок 3.5 – Практический результат

[Первая_функция_варианта] # #### Функция Реализует: Предисловие:

Листинг 3.10 – Функция _НАЗВАНИЕ_ФУНКЦИИ_

Постусловие

Тестирование функции

Исходный файл: Приведён в листинге 3.2. Входные данные: Ожидаемый результат:

Практический результат

Рисунок 3.6 – Практический результат

[Вторая_функция_варианта] # #### Функция Реализует: Предисловие:

Листинг 3.11 – Функция _НАЗВАНИЕ_ФУНКЦИИ_

Постусловие

Тестирование функции

Исходный файл: Приведён в листинге 3.2. Входные данные: Ожидаемый результат:

Практический результат

Рисунок 3.7 – Практический результат

3.3.4 Код основной программы

Код основной программы приведён в листинге 3.12

Листинг 3.12 – Код основной программы

3.4 Результаты тестирования

Скриншоты тестов каждой функции приведены на рисунках (3.1 - 3.7) Все тесты