

Задание 4

Разработать MFC-приложение, используя библиотеку классов stl

Общие требования к реализации:

1. Ядро приложения работает с использованием контейнеров и алгоритмов STL
2. Отсутствие C-конструкций в коде (использование вместо них конструкций C++)
3. Удобный интерфейс на базе MFC (QT, GDK...)
4. Система логирования, сохраняющая всю активность приложения и пользователя в файл формата (время, важность, текст сообщения)
5. Обработка исключений, защита от дурака

Варианты задач:

1. Неориентированный граф задан в файле в виде списков инцидентных вершин

количество вершин

вершина 1 список инцидентных вершин

вершина 2 список инцидентных вершин

...

вершина n список инцидентных вершин

Считать его в память. Представление в памяти – в виде массива списков инцидентных вершин. Найти все его компоненты связности и сохранить в файл.

2. Неориентированный граф задан в файле в виде списков инцидентных вершин. Считать его в память. Представление в памяти – также в виде списков инцидентных вершин. Задан номер вершины. Ввести дополнительную нумерацию вершин: заданной вершине присвоить номер 0, всем инцидентным с ней вершинам (слой 1) – номер 1, всем еще не просмотренным вершинам, инцидентным вершинам с номером 1 (слой 2) – номер 2 и т.д. Для этого хранить на каждом шаге список вершин текущего слоя. Вывести в файл вершины по слоям.

3. Неориентированный граф задан в файле в виде списков инцидентных вершин. Считать его в память. Представление в памяти – также в виде списков инцидентных вершин. Найти остовное дерево графа и сохранить в файл в том же формате. Для нахождения остовного дерева можно, например, на каждом шаге выбирать дугу, соединяющую уже просмотренную вершину с еще не просмотренной.

4. В файле записаны слова. Имеется некоторое множество ключевых слов, хранимых в другом файле. Создать таблицу встречаемости для неключевых слов в виде

слово (строка, столбец), ... , (строка, столбец)

и сохранить ее в третьем файле.

5. Разработать структуру данных «словарь, пополняемый словами из файлов». В словаре хранить слово и количество его повторений. Словарь должен уметь себя сохранять в файл, восстанавливать себя из файла и выдавать на экран слова, упорядоченные по алфавиту и по встречаемости.

6. В файле записаны слова. Имеется таблица синонимов, записанная в другом файле (для одного слова может быть несколько синонимов). Разработать структуру данных, поддерживающую таблицу синонимов. Заменить каждое слово в исходном файле на случайный синоним и результат записать в третий файл.

7. Дан список html-файлов, задаваемый в виде строки файлов, перечисляемых через запятую. Найти в них все вхождения URL-адресов в формате *протокол://www.tratata.ppp.ru*, где протокол – http, ftp, gopher, news, telnet, file. Выдать для каждого URL список вхождений (файл, строка, столбец).

8. В файле записаны существительные в разных падежах. Имеется таблица падежей для каждого слова. Разработать структуру данных, поддерживающую таблицу падежей. Выдать в другой файл слова из первого файла в формате

Именительный падеж слова (наименование падежа)

9. Имеется список 40 студентов, изучающих математику, физику, биологию и химию, заданный в файле в виде строк

Фамилия предмет

Каждый предмет изучает около 20 студентов. Разработать структуры данных, позволяющих эффективно отвечать на запросы вида «Выдать всех студентов, изучающих биологию и химию, но не изучающих физику».

10. Разработать контейнер строк `hash_set`, реализованный в виде хеш-таблицы. Контейнер должен содержать методы добавления, удаления и проверки на принадлежность, а также эффективную хеш-функцию. Контейнер должен быть устроен так же как и контейнеры стандартной библиотеки: возвращать свой итератор, который может перемещаться по элементам контейнера в некотором порядке. Заполнить контейнер словами из заданного файла. Сформировать из него два списка слов: содержащих менее 5 букв и являющихся перевертышами. Выдать эти списки отсортированными.

11. Разработать приложение Словарь, имеющий возможность подключения файлов словарей (на разных языках), хранения нескольких значений одного слова.

12. Разработать приложение «Записная книжка». Список функциональности:

- Хранение контактов: ФИО, телефоны, группы контактов
- Хранение напоминаний: Время события, название, пояснение, связь с контактом или группой контактов