1.

Класс MultiMap (Мультиотображение), реализация через обычное дерево поиска. Построить параметризированный класс, который реализует мультиотображение, где ключом является строка в стиле Си, а значением – некоторый другой класс. Каждый ключ может быть связан с одним или более значением.

Интерфейс отображения должен поддерживать следующие операции:

добавить пару (ключ, значение);
искать значение по указанному ключу;
искать следующую (предыдущую) пару для найденного по ключу значения;
удалить ключ и соответствующее значение;
получить количество хранящихся ключей;
итератор по множеству ключей и значений;

В качестве тестов рассмотреть отображение строк на целые числа, отображение строк на строки, взять в качестве другого класса один из реализованных в задачах 1-3 добавив, при необходимости, в этот класс некоторые методы. Подробности реализации обсуждаются в рабочем порядке.

2.

Класс Map (Отображение), реализация через сбалансированное дерево поиска. Построить параметризированный класс, который реализует отображение, где уникальным ключом является строка в стиле Си, а значением – некоторый другой класс.

Интерфейс отображения должен поддерживать следующие операции:

добавить пару (ключ, значение);
искать значение по указанному ключу;
искать следующую (предыдущую) пару для найденного по ключу значения;
удалить ключ и соответствующее значение;
получить количество хранящихся ключей;
итератор по множеству ключей и значений;

В качестве тестов рассмотреть отображение строк на целые числа, отображение строк на строки, взять в качестве другого класса один из реализованных в задачах 1-3 добавив, при необходимости, в этот класс некоторые методы. Подробности реализации обсуждаются в рабочем порядке.

3.

Класс Map (Отображение), реализация через красно-черное дерево поиска. Построить параметризированный класс, который реализует отображение, где уникальным ключом является строка в стиле Си, а значением – некоторый другой класс.

Интерфейс отображения должен поддерживать следующие операции:

добавить пару (ключ, значение);
искать значение по указанному ключу;
удалить ключ и соответствующее значение;
получить количество хранящихся ключей;
искать следующую (предыдущую) пару для найденного по ключу значения;
итератор по множеству ключей и значений;

В качестве тестов рассмотреть отображение строк на целые числа, отображение строк на строки, взять в качестве другого класса один из реализованных в задачах 1-3 добавив, при необходимости, в этот класс некоторые методы. Подробности реализации обсуждаются в рабочем порядке.

4.

Класс Map (Отображение), реализация через B-дерево поиска. Построить параметризированный класс, который реализует отображение, где уникальным ключом является строка в стиле Си, а значением – некоторый другой класс.

Интерфейс отображения должен поддерживать следующие операции:

добавить пару (ключ, значение);
искать значение по указанному ключу;
искать следующую (предыдущую) пару для найденного по ключу значения;
удалить ключ и соответствующее значение;
получить количество хранящихся ключей;
итератор по множеству ключей и значений;

В качестве тестов рассмотреть отображение строк на целые числа, отображение строк на строки, взять в качестве другого класса один из реализованных в задачах 1-3 добавив, при необходимости, в этот класс некоторые методы. Подробности реализации обсуждаются в рабочем порядке.

5.

Требуется найти все различия (с точностью до строк) между двумя текстовыми файлами, используя алгоритм наибольшей общей подпоследовательности.

6.

Класс MultiMap (Мультиотображение), реализация через хеш-множество по методу линейных проб. Построить параметризированный класс, который реализует мультиотображение, где ключом является строка в стиле Си, а значением – некоторый другой класс. Каждый ключ может быть связан с одним или более значением.

Интерфейс отображения должен поддерживать следующие операции:

добавить пару (ключ, значение);
искать значение по указанному ключу;
удалить ключ и соответствующее значение;
получить количество хранящихся ключей;
искать следующую (предыдущую) пару для найденного по ключу значения;
итератор по множеству ключей и значений;

В качестве тестов рассмотреть отображение строк на целые числа, отображение строк на строки, взять в качестве другого класса один из реализованных в задачах 1-3 добавив, при необходимости, в этот класс некоторые методы. Подробности реализации обсуждаются в рабочем порядке.

7.

Класс MultiMap (Мультиотображение), реализация через хеш-множество по методу многих списков. Построить параметризированный класс, который реализует мультиотображение, где ключом является строка в стиле Си, а значением – некоторый другой класс. Каждый ключ может быть связан с одним или более значением.

Интерфейс отображения должен поддерживать следующие операции:

добавить пару (ключ, значение);
искать значение по указанному ключу;
удалить ключ и соответствующее значение;
получить количество хранящихся ключей;
искать следующую (предыдущую) пару для найденного по ключу значения;
итератор по множеству ключей и значений;

В качестве тестов рассмотреть отображение строк на целые числа, отображение строк на строки, взять в качестве другого класса один из реализованных в задачах 1-3 добавив, при необходимости, в этот класс некоторые методы. Подробности реализации обсуждаются в рабочем порядке.

8.

Класс OrdString (упорядоченное мультимножество строк), реализация через обычное дерево поиска для хранения указателей на строки (строки понимаются в стиле Си). Такое множество позволяет выбирать упорядоченные подмножества и определять следующий и предыдущий элементы для заданной строки. Сравнение строк определяется как перегруженные операторы $<$ для вершин дерева.

Интерфейс класса должен поддерживать следующие операции:

добавить строку в множество;
удалить строку из множества;
искать в множестве данную строку;
получить количество элементов в множестве;

итератор по части множества. Это означает, что после указания некоторой строки мы можем перебирать последовательно (в смысле $<$) строки от указанной в одну или другую сторону.

Для тестов рассмотреть загрузку из файла и автоматическую генерацию множества.

9.

Класс OrdString (упорядоченное множество строк), реализация через сбалансированное дерево поиска для хранения указателей на строки (строки понимаются в стиле Си). Такое множество позволяет выбирать упорядоченные подмножества и определять следующий и предыдущий элементы для заданной строки. Сравнение строк определяется как перегруженные операторы $<, >$ для вершин дерева.

Интерфейс класса должен поддерживать следующие операции:

добавить строку в множество;
удалить строку из множества;
искать в множестве данную строку;
получить количество элементов в множестве;

итератор по части множества. Это означает, что после указания некоторой строки мы можем перебирать последовательно (в смысле $<$) строки от указанной в одну или другую сторону.

Для тестов рассмотреть загрузку из файла и автоматическую генерацию множества.

10.

Класс OrdString (упорядоченное множество строк), реализация через красно-черное дерево поиска для хранения указателей на строки (строки понимаются в стиле Си). Такое множество позволяет выбирать упорядоченные подмножества и определять следующий и предыдущий элементы для заданной строки. Сравнение строк определяется как перегруженные операторы $<, >$ для вершин дерева.

Интерфейс класса должен поддерживать следующие операции:

добавить строку в множество;
удалить строку из множества;
искать в множестве данную строку;
получить количество элементов в множестве;

итератор по части множества. Это означает, что после указания некоторой строки мы можем перебирать последовательно (в смысле $<$) строки от указанной в одну или другую сторону.

Для тестов рассмотреть загрузку из файла и автоматическую генерацию множества.

11.

Класс OrdString (упорядоченное множество строк), реализация через B-дерево поиска для хранения указателей на строки (строки понимаются в стиле Си). Такое множество позволяет выбирать упорядоченные подмножества и определять следующий и предыдущий элементы для заданной строки. Сравнение строк определяется как перегруженные операторы $<, >$ для вершин дерева.

Интерфейс класса должен поддерживать следующие операции:

добавить строку в множество;
удалить строку из множества;
искать в множестве данную строку;
получить количество элементов в множестве;

итератор по части множества. Это означает, что после указания некоторой строки мы можем перебирать последовательно (в смысле $<$) строки от указанной в одну или другую сторону.

Для тестов рассмотреть загрузку из файла и автоматическую генерацию множества.

12.

Класс Rectangle (множество точек в R^2), Такое множество хранит координаты (x, y) точек плоскости, и позволяет выбирать точки, лежащие в некоторой окрестности заданной. Предполагается, что все точки находятся внутри изначально заданного прямоугольника. Диапазон изменения по у делится на равные части, каждой такой части соответствует сбалансированное по х дерево точек, у которых у лежит в данной части.

Интерфейс класса должен поддерживать следующие операции:

добавить точку в множество;
удалить точку из множества;
искать в множестве данную точку;
получить количество элементов в множестве;

итератор по части множества. Это означает, что после указания некоторой точки мы можем перебирать последовательно точки от указанной в одну или другую сторону.

получить список точек, лежащих в данной прямоугольной окрестности заданной точки;

Для тестов рассмотреть загрузку точек из файла и автоматическую генерацию множества.

13.

Требуется реализовать контейнер данных наподобие файловой системы FAT (Win) с возможностью создавать и уничтожать файлы и читать/записывать в них байтовые массивы некоторой длины. Для работы захватывается большой кусок памяти (виртуальный диск), в котором выделяются служебные области модельной системы.

Реализация должна поддерживать следующий интерфейс:

создать файл;

удалить файл;

копировать, переименовать файл;

получить список существующих файлов;

получить длину файла;

прочитать/записать заданное количество байт по указанному смещению от начала файла.

Файловая система FAT построена на основе односвязных списков файловых блоков с хранением ссылок между блоками в отдельной таблице.