

1.

Класс Map (Отображение), реализация через обычное дерево поиска. Построить параметризованный класс, который реализует отображение, где уникальным ключом является строка в стиле Си, а значением – некоторый другой класс. В методах класса использовать рекурсию.

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;

печать отображения;

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 обычное дерево поиска. Построить параметризованный класс, который реализует отображение, где уникальным ключом является строка в стиле Си, а значением – некоторый другой класс. В методах класса не использовать рекурсию.

Интерфейс отображения должен поддерживать следующие операции:

добавить пару (ключ, значение);

искать значение по указанному ключу;

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 MultiMap (Мультиотображение), реализация через обычное дерево поиска. Построить параметризованный 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 значениями. В методах класса использовать рекурсию.

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;

печать отображения;

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 MultiMap (Мультиотображение), реализация через обычное дерево поиска. Построить параметризованный 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 значениями. В методах класса не использовать рекурсию.

Интерфейс отображения должен поддерживать следующие операции:

добавить пару (ключ, значение);

искать значение по указанному ключу;

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.

Класс Map (Отображение), реализация через обычное дерево поиска. Построить параметризованный класс, который реализует отображение, где уникальным ключом является некоторый другой класс, а значением – строка в стиле Си. В методах класса использовать рекурсию.

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;
- печатать отображения;

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

6.

Класс MultiMap (Мультиотображение), реализация через обычное дерево поиска. Построить параметризованный 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. В методах класса не использовать рекурсию.

Интерфейс отображения должен поддерживать следующие операции:

- добавить пару (ключ, значение);
- искать значение по указанному ключу;
- удалить ключ и соответствующее значение;
- получить количество хранящихся ключей;

итератор по множеству ключей и значений; В качестве тестов рассмотреть отображение целых чисел на строки, отображение строк на строки, взять в качестве ключа один из реализованных в задачах 1-3, добавив, при необходимости, в этот класс некоторые методы. Подробности реализации обсуждаются в рабочем порядке.

7.

Класс Map (Отображение), реализация через сбалансированное дерево поиска. Построить параметризованный 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;
- 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 Map (Отображение), реализация через красно-черное дерево поиска. Построить параметризованный 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;
- 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.

9.

Класс Map (Отображение), реализация через B-дерево поиска. Построить параметризованный 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;
- 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.

10.

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

Интерфейс отображения должен поддерживать следующие операции:

- добавить пару (ключ, значение);
- искать значение по указанному ключу;
- удалить ключ и соответствующее значение;
- получить количество хранящихся ключей;
- итератор по множеству ключей и значений;

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

11.

Класс MultiMap (Мультиотображение), реализация через Хэш-множество по методу линейных проб. Построить параметризованный 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. В методах класса не использовать рекурсию.

Интерфейс отображения должен поддерживать следующие операции:

- добавить пару (ключ, значение);
- искать значение по указанному ключу;
- 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.

12.

Класс MultiMap (Мультиотображение), реализация через Хэш-множество по методу многих списков. Построить параметризованный 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. В методах класса не использовать рекурсию.

Интерфейс отображения должен поддерживать следующие операции:

- добавить пару (ключ, значение);
- искать значение по указанному ключу;
- 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.

13.

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.

14.

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.

15.

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.

16.

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.

17.

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 y делится на равные части, каждой такой части соответствует сбалансированное по x дерево точек, у которых y 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;
получить список точек, лежащих в данной прямоугольной окрестности заданной точки;

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

18.

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.

Корень дерева будет соответствовать исходному прямоугольнику. Поделив прямоугольник на 4 равные части, выберем те из них, где есть точки и получим поддеревья первого уровня. И так продолжим, пока в прямоугольниках содержатся точки.

Интерфейс класса должен поддерживать следующие операции:

добавить точку в множество;
удалить точку из множества;
искать в множестве данную точку;
получить количество элементов в множестве;
получить список точек, лежащих в данной прямоугольной окрестности заданной точки;

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