

Лабораторная работа №6

«Программирование с использованием деревьев. Хэширование»

Задание 1.

Исходная информация в виде массива находится в компоненте `StringGrid` или его аналоге. Каждый элемент массива содержит строку текста и целочисленный ключ (например, Ф.И.О. и номер паспорта).

Разработать класс для работы с деревом поиска, содержащий следующие методы

(не использовать шаблоны STL и boost):

- внести информацию из массива в дерево поиска;
- сбалансировать дерево поиска;
- добавить в дерево поиска новую запись;
- по заданному ключу найти информацию в дереве поиска и отобразить ее;
- удалить из дерева поиска информацию с заданным ключом;
- распечатать информацию прямым, обратным обходом и в порядке возрастания ключа.

На основе родительского класса создать производный класс для решения задачи выбранного варианта.

Написать программу, иллюстрирующую все методы работы с деревом поиска. Результат формирования и преобразования дерева отображать в компонентах `TreeView`, Метод или аналогах. Написать обработчик события, реализующий работу с методом решения своего варианта.

Индивидуальные задания

1. Поменять местами информацию, содержащую максимальный и минимальный ключи.
2. Подсчитать число листьев в дереве. (Лист – это узел, из которого нет ссылок на другие узлы дерева.)
3. Удалить из дерева ветвь с вершиной, имеющей заданный ключ.
4. Определить максимальную глубину дерева, т.е. число узлов в самом длинном пути от корня дерева до листьев.
5. Определить число узлов на каждом уровне дерева.
6. Удалить из левой ветви дерева узел с максимальным значением ключа и все связанные с ним узлы.
7. Определить количество символов во всех строках, находящихся в узлах дерева.
8. Определить число листьев на каждом уровне дерева.
9. Определить число узлов в дереве, в которых есть указатель только на одну ветвь.
10. Определить число узлов в дереве, у которых есть две дочери.

11. Определить количество записей в дереве, начинающихся с определенной буквы (например а).
12. Найти среднее значение всех ключей дерева и найти узел, имеющий ближайший к этому значению ключ.
13. Найти запись с ключом, ближайшим к среднему значению между максимальным и минимальным значениями ключей.
14. Определить количество узлов в левой ветви дерева.
15. Определить количество узлов в правой ветви дерева.

Задание 2.

Разработать приложение, в котором содержатся следующие классы:

Родительский класс, реализующий методы работы с хеш-таблицей на основе массива стеков (не использовать шаблоны STL и boost).

Производный класс, созданный на базе родительского и реализующий метод решения своего варианта.

В приложении продемонстрировать работу всех методов работы с хеш-таблицей. Результат формирования и преобразования хеш-таблицы показывать в компоненте Мемо методом Print(Memo) или его аналогах в виде строк, отображающих стеки.

Написать обработчик события, реализующий вызов метода решения своего варианта.

Индивидуальные задания

1. Создать хеш-таблицу со случайными целыми ключами в диапазоне от -50 до +50 и преобразовать ее в две таблицы. Первая должна содержать только положительные ключи, а вторая – отрицательные.
2. Создать хеш-таблицу со случайными целыми ключами и удалить из него записи с четными ключами.
3. Создать хеш-таблицу со случайными целыми ключами в диапазоне от -10 до 10 и удалить из него записи с отрицательными ключами.
4. Создать хеш-таблицу со случайными целыми ключами и найти запись с минимальным ключом.
5. Создать хеш-таблицу со случайными целыми ключами и найти запись с максимальным ключом.
6. Подсчитать, сколько элементов хеш-таблицы со случайными ключами превышает среднее значение от всех ключей.
7. Создать хеш-таблицу из случайных целых чисел и найти в ней номер стека, содержащего минимальное значение ключа.
8. Создать хеш-таблицу из случайных целых чисел и найти в ней номер стека, содержащего максимальное значение ключа.