Лабораторная работа №3

по курсу «Языки программирования и методы программирования» (информатика, 3 семестр)

Техническое задание

|  |  |  |
| --- | --- | --- |
| Дата | 23.12.2016 |  |
| Выполнила | Елагина Д. |  |

Используемые термины и сокращения

|  |  |
| --- | --- |
| АТД | Абстрактный тип данных |
|  |  |

# Постановка задачи

Написать программу на C++ для сравнения алгоритмов поиска по ключу: с помощью хеш-таблиц и других аргументов. Написать краткое техническое задание (ТЗ). Выполнить реализацию. Написать для нее тесты.

# Функциональные требования

* 1. АТД должен позволять хранить элементы любых типов. Для этого класс должен быть объявлен как шаблонный.
  2. Последовательность можно создать вручную или автоматически, предварительно выбрав вид создаваемой последовательности
* Произвольная
* Отсортированная по возрастанию
* Отсортированная по убыванию
  1. Из созданной последовательности можно построить хеш-таблицу (IDictionary<TKey,TElement>), которая обрабатывает коллизии 2 различными способами:
* Перемешивание сложением
* Перемешивание сцеплением
  1. Хеш-функция должна удовлетворять следующему обязательному условию: если значения этой функции для некоторых двух элементов различаются, то и сами элементы различны.

# Требования к структурам данных и алгоритмам

* 1. Для хранения данных реализовываем в хеш-таблице реализуем 2 наследника абстрактного класса IDictionary<TKey,TElement>: выполнить реализацию, использующую перемешивание сложением (IDictionaryOpenAddressing<TKey,TElement>), и реализацию, использующую связные списки (IDictionarySeperateChanging<TKey,TElement>).
  2. Выполнить реализацию шаблонного класса (ListData<TKey,TElement>) – списковая структура для реализации перемешивания сцеплением.
  3. В перемешивании сложением программа должна выбрасывать исключение, если таблица заполнена.
  4. Все методы поиска и удаления должны выбрасывать исключение в ситуациях, когда ключ не найден.
  5. Поддержка дополнительного интерфейса SortedSequence позволяет внутри класса Tree получить индекс элемента в дереве (IndexOf) или получить подпоследовательность (поддерево) существующего дерева (GetSubsequence). IndexOf возвращает индекс первого встретившегося в дереве элемента, ключ которого совпадает с заданным.
  6. Хеш-таблица реализована как шаблонный класс. Это позволяет работать с разными типами данных ключа и информации.
  7. Программа предоставляет функцию измерения времени выполнения алгоритма на трех видах входных данных:
* Произвольная последовательность
* Последовательность, отсортированная в правильном порядке
* Последовательность, отсортированная в обратном порядке.

# Требования к интерфейсу

Программа должна обладать пользовательским интерфейсом (консольным или графическим). При этом должна быть возможность как автоматической, так и ручной проверки корректности работы алгоритмов.

# Требования к форматам входных и выходных данных

Требования к форматам входных данных

Программа должна позволять выбрать любой из реализованных алгоритмов сортировки и запустить его на (достаточно произвольных) исходных данных.

## Требования к форматам выходных данных

Должна быть возможность просмотра как исходных данных, так и результата.

# Требования к unit-тестам

* 1. Тестирование перемешивания сцеплением
     1. Вывод пустой таблицы
     2. Вставка 1
     3. Вставка 3
     4. Вставка 2
     5. Поиск 0 (ожидаем 0)
     6. Поиск 1 (ожидаем 1)
     7. Поиск 2 (ожидаем 1)
     8. Поиск 3 (ожидаем 1)
     9. Поиск 4 (ожидаем 0)
     10. Вставка 6.
     11. Вставка 7.
     12. Вставка 8. (переполнение)
     13. Удаление 8 (нет такого элемента)
     14. Удаление 1.
     15. Очистить память
  2. Тестирование перемешивания сцеплением
     1. Вывод пустой таблицы
     2. Вставка 1
     3. Вставка 3
     4. Вставка 2
     5. Поиск 0 (ожидаем 0)
     6. Поиск 1 (ожидаем 1)
     7. Поиск 2 (ожидаем 1)
     8. Поиск 3 (ожидаем 1)
     9. Поиск 4 (ожидаем 0)
     10. Вставка 6.
     11. Вставка 7.
     12. Вставка 8.
     13. Удаление 8
     14. Удаление 1.
     15. Очистить память
  3. Закончить работу.