

ЛАБОРАТОРНАЯ РАБОТА №9

Курс «Основы программирования»

Тема: Абстрактные типы данных (АТД). Коллекции в .NET.

Цель: Научиться программировать ключевые АТД с помощью массивов, связанных списков и деревьев, научиться работать с коллекциями .NET.

Темы для предварительной проработки ^[устно]:

- Фундаментальные структуры данных и абстрактные типы данных.
- Связные списки и деревья.
- Коллекции в .NET.

Общие задания ^[код] :

1. Модифицировать код проекта из лабораторной работы №5 дважды: в первом проекте данные предметной области хранить и обрабатывать в виде двусвязного списка (структуру реализовать самостоятельно), во втором – использовать коллекцию .NET List.
2. *С помощью стека* проверить, что математическое выражение в скобках записано корректно. Примеры корректных выражений:

(2+3) (1+6) ((2-3) (5+1))
2 (3) (1+6 (7+2)) ((2-3) (5+1))
2 (3+5 ((6)))

Примеры некорректных выражений:

((2+3) (4-1))
2 (3+5 ((6))

Примечание. Основной признак корректности выражения – наличие для каждой открывающей скобки соответствующей закрывающей.

3. Написать игру «Считалка», в коде которой будет эмулироваться сам процесс игры с помощью циклического связанного списка. В игре участвуют от 5 до 10 человек (имена участников ввести из текстового файла либо «попросить» в коде). Пользователь вводит строку считалки и указывает человека, с которого необходимо начать. Программа должна вывести человека, на которого придется последнее слово строки. Как можно данную задачу решить без дополнительных структур данных?
4. Модифицировать код задания 5 из лабораторной работы №2: перебрать все числа N от 1 до 50000 и вывести только те числа, для которых существует более двух (три и более) комбинаций суммы кубов. Использовать ассоциативный массив .NET (Dictionary) для решения задачи.
5. Дан текстовый файл со словами, разделенными пробелами. Написать программу, которая выводит 10 наиболее встречаемых слов в файле, отсортированных в порядке убывания частоты встречаемости. При равном числе вхождений слова упорядочивать по алфавиту.
6. Представить турнирную сетку раунда плей-офф чемпионата мира в виде дерева. Названия команд считать из файла либо «попросить» в коде. Сначала корень дерева и все узлы, кроме листьев, заполнить пустыми строками и неизвестными результатами. Листья соответствуют матчам этапа 1/16 финала. Результаты матчей генерировать произвольным образом, продвигая

команд-победителей от листьев к корню. Вывести результаты розыгрыша в таком виде:

```
BRA - ARG : 0 - 2
  BRA - FRA : 2 - 1
    BRA - COL : 4 - 3
      BRA - CHI : 2 - 1
        COL - URU : 3 - 2
          FRA - GER : 1 - 0
            FRA - NIG : 1 - 0
              GER - ALG : 3 - 1
                CRC - ARG : 0 - 2
                  MEX - CRC : 0 - 1
                    NED - MEX : 1 - 2
                      CRC - GRE : 2 - 1
                        ARG - BEL : 1 - 0
                          ARG - SWI : 3 - 2
                            BEL - USA : 3 - 2
```

7. Написать программу, в соответствии с вариантом. Решить задачу как на основе самостоятельно разработанного списка, так и на основе коллекции .NET.

Дополнительные задания ^[КОД] :

1. Написать код для обработки разреженных матриц (состоящих, в основном, из нулей) в двух проектах. Для представления матриц использовать в одном проекте двумерный массив, в другом – связные списки. Реализовать функции: заполнения и вывода матрицы, транспонирования матрицы, суммирования и перемножения матриц.

Контрольные вопросы ^[ОТЧЕТ] :

1. Кратко опишите фундаментальные структуры данных (ФСД).
2. Чем отличаются абстрактные типы данных (АТД) от ФСД?
3. Как программно реализовать односвязные и двусвязные списки?
4. Стек, дек и очередь.
5. Деревья, основные термины.
6. Деревья бинарного поиска.
7. Укажите возможности .NET по работе с коллекциями (связными списками, векторами, стеком, очередью, словарями, множествами).

Рекомендуемые источники:

- [1] Шилдт Г. С# 4.0. Полное руководство. – М.: Издательский дом «Вильямс», 2011. – 1056с.
- [2] Кормен Т., Лейзерсон Ч., Ривест Р. Алгоритмы: Построение и анализ. – М.: МЦНМО, 1999. – 960с.
- [3] Ахо А., Хопкрофт Д., Ульман Д. Структуры данных и алгоритмы. – М.: Издательский дом «Вильямс», 2001. – 384с.
- [4] Кнут Д. Искусство программирования, том 1. Основные алгоритмы. – М.: Издательский дом «Вильямс», 2000. – 720с.
- [5] Златопольский Д.М. Сборник задач по программированию. – СПб.: БХВ-Петербург, 2007. – 240с.

Варианты индивидуальных заданий.

Вариант 1.

Написать программу, которая оставляет в списке только уникальные элементы.

Вариант 2.

Написать программу, которая проверяет равенство двух списков. Списки считать равными, если они содержат одинаковые элементы в одинаковом количестве (порядок не важен).

Вариант 3.

Написать программу, которая проверяет, что один список является реверсией второго.

Вариант 4.

Написать программу, которая проверяет, что один список является под-списком второго.

Вариант 5.

Написать программу, которая проверяет, что в одном списке содержатся *все* элементы второго.

Вариант 6.

Написать программу, которая проверяет, что в одном списке не содержится *ни одного* элемента второго.

Вариант 7.

Написать программу, которая добавляет в один список элементы другого списка в обратном порядке.

Вариант 8.

Написать программу, которая оставляет в первом списке только те элементы, которые содержатся и в первом, и во втором списках.

Вариант 9.

Написать программу, которая оставляет в первом списке только те элементы, которых нет во втором списке.

Вариант 10.

Написать программу, которая проверяет, что элементы списка в обратном порядке образуют ту же цепочку, что и в прямом порядке.