Министерство образования Республики Беларусь
Учреждение образования
«Белорусский государственный университет
информатики и радиоэлектроники»
Филиал
«Минский радиотехнический колледж»

Учебный предмет
«Конструирование программ и языки программирования»

# Инструкция

по выполнению лабораторной работы №25
«Разработка, отладка и испытание программ с коллекциями»

Минск 2024 г.

# Лабораторная работа № 25

**Тема работы:** «Разработка, отладка и испытание программ с коллекциями»

## 1 Цель работы

Сформировать умение разрабатывать программы с использованием коллекций.

## 2 Задание

Номер варианта соответствует вашему номеру по списку.

1. Используя стек, напечатать символы некоторой величины строкового типа в обратном порядке.
2. Пусть имеется файл действительных чисел и некоторое число C. Используя очередь, напечатать сначала все элементы, меньшие числа C, а затем все остальные элементы.
3. Составить программу, в которой все делители заданного числа n будут выводиться в порядке возрастания (при проверке возможных делителей, не превышающих $\sqrt{n}$)
4. Расположить элементы целочисленного массива размером n в обратном порядке с использованием стека.
5. Дана величина a строкового типа из четного количества символов. Получить и напечатать величину b, состоящую из символов первой половины величины a, записанных в обратном порядке, после которых идут символы второй половины величины a, также записанные в обратном порядке. Например, при a = “привет” b должно быть равно “ипртев”.
6. Даны две непустые очереди. Элементы каждой из очередей упорядочены по возрастанию. Объединить очереди в одну с сохранением упорядоченности элементов.
7. Элементы целочисленного массива записать в очередь. Написать функцию извлечения элементов из очереди до тех пор, пока первый элемент очереди не станет четным.
8. Написать функцию, которая по произвольному указателю на один из элементов двусвязного списка подсчитывает количество элементов в этом списке.
9. Требуется создать список из элементов последовательности целых чисел, вводимых пользователем. Также требуется написать функцию удаления всех элементов списка, имеющих некоторое значение x.
10. Сформировать список целых чисел, вводимых пользователем, в порядке возрастания и без повторений элементов. Вывести элементы списка на экран и найти сумму всех элементов списка.
11. Написать функцию, которая по двум данным линейным спискам формирует новый список, состоящий из элементов, одновременно входящих в оба данных списка.

12. Написать функцию, которая удаляет из списка элементы, входящие в него только один раз.

13. Определить, образуют ли элементы списка действительных чисел геометрическую прогрессию.

14. Удалить из списка все элементы, встречающиеся более одного раза.

15. Дан список целых чисел. Продублировать в нем все простые числа.

## 3 Оснащение работы

ПК, среда Visual Studio 2019, MSword.

## 4 Основные теоретические сведения

Большая часть классов коллекций содержится в пространствах имен System.Collections (простые необобщенные классы коллекций), System.Collections.Generic (обобщенные или типизированные классы коллекций) и System.Collections.Specialized (специальные классы коллекций). Также для обеспечения параллельного выполнения задач и многопоточного доступа применяются классы коллекций из пространства имен System.Collections.Concurrent

Основой для создания всех коллекций является реализация интерфейсов IEnumerator и IEnumerable (и их обобщенных двойников IEnumerator<T> и IEnumerable<T>). Интерфейс IEnumerator представляет перечислитель, с помощью которого становится возможен последовательный перебор коллекции, например, в цикле foreach. А интерфейс IEnumerable через свой метод GetEnumerator предоставляет перечислитель всем классам, реализующим данный интерфейс. Поэтому интерфейс IEnumerable (IEnumerable<T>) является базовым для всех коллекций.

Класс ArrayList представляет коллекцию объектов. Если необходимо сохранить вместе разнотипные объекты - строки, числа и т.д., то данный класс как раз для этого подходит.

**Основные методы класса:**

– int Add(object value): добавляет в список объект value

– void AddRange(ICollection col): добавляет в список объекты коллекции col, которая представляет интерфейс ICollection - интерфейс, реализуемый коллекциями.

– void Clear(): удаляет из списка все элементы

– bool Contains(object value): проверяет, содержится ли в списке объект value. Если содержится, возвращает true, иначе возвращает false

– void CopyTo(Array array): копирует текущий список в массив array.

– ArrayList GetRange(int index, int count): возвращает новый список ArrayList, который содержит count элементов текущего списка, начиная с индекса index

– int IndexOf(object value): возвращает индекс элемента value

– void Insert(int index, object value): вставляет в список по индексу index объект value

– void InsertRange(int index, ICollection col): вставляет в список начиная с индекса index коллекцию ICollection

– int LastIndexOf(object value): возвращает индекс последнего вхождения в списке объекта value

– void Remove(object value): удаляет из списка объект value

– void RemoveAt(int index): удаляет из списка элемент по индексу index

– void RemoveRange(int index, int count): удаляет из списка count элементов, начиная с индекса index

– void Reverse(): переворачивает список

– void SetRange(int index, ICollection col): копирует в список элементы коллекции col, начиная с индекса index

– void Sort(): сортирует коллекцию

Класс List<T> из пространства имен System.Collections.Generic представляет простейший список однотипных объектов.

Среди его методов можно выделить следующие:

– void Add(T item): добавление нового элемента в список

– void AddRange(ICollection collection): добавление в список коллекции или массива

– int BinarySearch(T item): бинарный поиск элемента в списке. Если элемент найден, то метод возвращает индекс этого элемента в коллекции. При этом список должен быть отсортирован.

– int IndexOf(T item): возвращает индекс первого вхождения элемента в списке

– void Insert(int index, T item): вставляет элемент item в списке на позицию index

– bool Remove(T item): удаляет элемент item из списка, и если удаление прошло успешно, то возвращает true

– void RemoveAt(int index): удаление элемента по указанному индексу index

– void Sort(): сортировка списка

Класс LinkedList<T> представляет двухсвязный список, в котором каждый элемент хранит ссылку одновременно на следующий и на предыдущий элемент.

Если в простом списке List<T> каждый элемент представляет объект типа T, то в LinkedList<T> каждый узел представляет объект класса LinkedListNode<T>.

Этот класс имеет следующие свойства:

– Value: само значение узла, представленное типом T

– Next: ссылка на следующий элемент типа LinkedListNode<T> в списке. Если следующий элемент отсутствует, то имеет значение null

– Previous: ссылка на предыдущий элемент типа LinkedListNode<T> в списке. Если предыдущий элемент отсутствует, то имеет значение null

Класс Queue<T> представляет обычную очередь, работающую по алгоритму FIFO ("первый вошел - первый вышел").

У класса Queue<T> можно отметить следующие методы:

– Dequeue: извлекает и возвращает первый элемент очереди
– Enqueue: добавляет элемент в конец очереди
– Peek: просто возвращает первый элемент из начала очереди без его удаления.

Класс Stack<T> представляет коллекцию, которая использует алгоритм LIFO ("последний вошел - первый вышел"). При такой организации каждый следующий добавленный элемент помещается поверх предыдущего. Извлечение из коллекции происходит в обратном порядке - извлекается тот элемент, который находится выше всех в стеке.

В классе Stack можно выделить два основных метода, которые позволяют управлять элементами:

– Push: добавляет элемент в стек на первое место
– Pop: извлекает и возвращает первый элемент из стека
– Peek: просто возвращает первый элемент из стека без его удаления

Еще один распространенный тип коллекции представляют словари. Словарь хранит объекты, которые представляют пару ключ-значение.

Каждый такой объект является объектом структуры KeyValuePair <TKey, TValue>. Благодаря свойствам Key и Value, которые есть у данной структуры, можно получить ключ и значение элемента в словаре.

Кроме стандартных классов коллекций типа списков, очередей, словарей, стеков .NET также предоставляет специальный класс ObservableCollection. Он по функциональности похож на список List за тем исключением, что позволяет известить внешние объекты о том, что коллекция была изменена.

## 5. Порядок выполнения работы

1. Выделить ключевые моменты задачи.
2. Построить алгоритм и теоретическую объектную модель решения задачи.
3. Запрограммировать полученные алгоритмы и объектную модель.

## 6. Форма отчета о работе

*Лабораторная работа № ___*

*Номер учебной группы* _____

*Фамилия, инициалы учащегося* _____

*Дата выполнения работы* _____

*Тема работы:* _____

*Цель работы:* _____

*Оснащение работы:* _____

*Результат выполнения работы:* _____

**7. Контрольные вопросы и задания**

1. Дайте определение понятию «Стек». Поясните его функциональное назначение.
2. Дайте определение понятию «Очередь». Поясните ее функциональное назначение.
3. Дайте определение понятию «Список». Поясните его функциональное назначение. Приведите классификацию видов списков.

**8. Рекомендуемая литература**

**1. Рихтер, Дж**. CLR via C#. Программирование на платформе Microsoft .NET Framework 4.5 на языке C# / Дж. Рихтер. СПб. : Изд-во Питер, 2021. 896 с.

**2. Прайс, М. Дж**. C# 10 и .NET 6. Современная кросс-платформенная разработка / М. Дж. Прайс. СПб : Изд-во Питер, 2023. 848 с.

**3. Васильев, А.Н.** Программирование на C# для начинающих. Особенности языка / А.Н. Васильев. М. : Эксмо, 2022. 528 с.

**4. Фримен, А**. ASP.NET Core 3 с примерами на C# для профессионалов / А. Фримен. СПб. : Изд-во Вильямс, 2021. 1184 с.