|  |  |  |
| --- | --- | --- |
| Дата | Содержание выполняемой работы | Подпись  руководителя |
| 22.04.2020 | **Практическая работа №15**  Тема: Структуры данных, коллекции и классы-прототипы. Библиотечные классы коллекций.  Наименование работы: Коллекции. Пространство имен System.Collections.  Цель: Сформировать навыки создания программ с использованием класса-коллекции.  Задание 1**.**  Изучил предложенную теорию. Кратко законспектировал.  Обзор коллекций.  В C# коллекция представляет собой совокупность объектов. Коллекции упрощают решение многих задач программирования благодаря тому, что предлагают готовые решения для создания целого ряда типичных, но порой трудоемких для разработки структур данных.  Необобщенные коллекции.  Необобщенные коллекции определяются в пространстве имен System.Collections. Необобщенные коллекции представляют собой структуры данных общего назначения, оперирующие ссылками на объекты. Таким образом, они позволяют манипулировать объектом любого типа, хотя и не типизированным способом. В этом состоит их преимущество и в то же время недостаток.  Классы необобщенных коллекций.  Ниже приведены классы необобщенных коллекций:  ArrayList Определяет динамический массив, т.е. такой массив, который может при необходимости увеличивать свой размер.  Hashtable Определяет хеш-таблицу для пар "ключ-значение".  Queue Определяет очередь, или список, действующий по принципу "первым пришел — первым обслужен".  SortedList Определяет отсортированный список пар "ключ-значение".  Stack Определяет стек, или список, действующий по принципу "первым пришел — последним обслужен".  Обобщенные коллекции.  Благодаря внедрению обобщений прикладной интерфейс Collections API значительно расширился, в результате чего количество классов коллекций и интерфейсов удвоилось. Обобщенные коллекции объявляются в пространстве имен System.Collections.Generic.  Классы обобщенных коллекций.  Классы обобщенных коллекций по большей части соответствуют своим необобщенным аналогам, хотя в некоторых случаях они носят другие имена. Отличаются они также своей организацией и функциональными возможностями.  Класс ArrayList.  В классе ArrayList поддерживаются динамические массивы, расширяющиеся и сокращающиеся по мере необходимости. В языке C# стандартные массивы имеют фиксированную длину, которая не может изменяться во время выполнения программы.  Класс Hashtable.  Класс Hashtable предназначен для создания коллекции, в которой для хранения ее элементов служит хеш-таблица. Информация сохраняется в хеш-таблице с помощью механизма, называемого хешированием. Преимущество хеширования заключается в том, что оно обеспечивает постоянство времени выполнения операций поиска, извлечения и установки значений независимо от величины массивов данных.  Стек: классы Stack и Stack<T>.  Стек (stack) — это контейнер, работающий по принципу "последний вошел, первый вышел" (last in, first out — LIFO). Класс коллекции, поддерживающий стек, носит название Stack. В нем реализуются интерфейсы ICollection, IEnumerable и ICloneable.  Очередь: классы Queue и Queue<T>.  Очередь (queue) — это коллекция, в которой элементы обрабатываются по схеме "первый вошел, первый вышел" (first in, first out — FIFO). Элемент, вставленный в очередь первым, первым же и читается.  Связный список: класс LinkedList<T>.  Класс LinkedList<T> представляет собой двухсвязный список, в котором каждый элемент ссылается на следующий и предыдущий.  Сортированный список: класс SortedList<TKey, TValue>.  Если нужна коллекция, отсортированная по ключу, можно воспользоваться SortedList<TKey, TValue> Этот класс сортирует элементы на основе значения ключа. Можно использовать не только любой тип значения, но также и любой тип ключа.  Словарь: класс Dictionary<TKey, TValue>.  Словарь (dictionary) представляет собой сложную структуру данных, позволяющую обеспечить доступ к элементам по ключу. Главное свойство словарей — быстрый поиск на основе ключей. Можно также свободно добавлять и удалять элементы, подобно тому, как это делается в List<T>, но без накладных расходов производительности, связанных с необходимостью смещения последующих элементов в памяти.  Сортированный словарь: класс SortedDictionary<TKey, TValue>.  Класс SortedDictionary<TKey, Tvalue> представляет дерево бинарного поиска, в котором все элементы отсортированы на основе ключа. Тип ключа должен реализовать интерфейс IComparable<TKey>.  Множества: классы HashSet<T> и SortedSet<T>.  Коллекция, содержащая только отличающиеся элементы, называется множеством (set). В составе .NET 4 имеются два множества — HashSet<T> и SortedSet<T>. Оба они реализуют интерфейс ISet<T>. Класс HashSet<T> содержит неупорядоченный список различающихся элементов, а в SortedSet<T> элементы упорядочены.  Битовые коллекции.  Если требуется иметь дело с множеством битов, можно применить класс BitArray и структуру BitVector32. Класс BitArray расположен в пространстве имен System.Collections, a BitVector32 — в пространстве System.Collections.Specialized.  Класс BitArray.  Класс BitArray служит для хранения отдельных битов в коллекции. Коллекции типа BitArray подлежат индексированию. По каждому индексу указывается отдельный бит в коллекции, причем нулевой индекс обозначает младший бит.  Структура BitVector32.  Если необходимое количество бит известно заранее, то вместо BitArray можно использовать структуру BitVector32. Класс BitArray при необходимости может расти, а структура BitVector32 лишена такой возможности.  Специальные и наблюдаемые коллекции.  Специальные коллекции.  В среде .NET Framework предусмотрен ряд специальных коллекций, оптимизированных для работы с данными конкретного типа или для их обработки особым образом.  Наблюдаемые коллекции.  В случае если нужна информация о том, когда элементы коллекции удаляются или добавляются, можно использовать класс ObservableCollection<T>. Этот класс был определен для WPF и предназначен для того, чтобы пользовательский интерфейс мог получать информацию об изменениях коллекции.  Задание 2.  Создал программы по образцу. Листинг кода переписал в Приложение 14.  Задание 2.1  Создание приложения с использованием библиотечного класса-коллекции. Листинг кода задания 2.1 переписал в Приложение 14. Результат выполнения работы представлен на рисунке 31    Рисунок 31 – результат выполнения работы  **Контрольные вопросы:**   1. Укажите, для чего предназначена коллекция.   Ответ: Коллекции упрощают решение многих задач программирования благодаря тому, что предлагают готовые решения для создания целого ряда типичных, но порой трудоемких для разработки структур данных.   1. Опишите необобщенные коллекции.   Ответ: Необобщенные коллекции определяются в пространстве имен System.Collections. Необобщенные коллекции представляют собой структуры данных общего назначения, оперирующие ссылками на объекты.   1. Опишите специальные коллекции.   Ответ: В среде .NET Framework предусмотрен ряд специальных коллекций, оптимизированных для работы с данными конкретного типа или для их обработки особым образом.   1. Опишите битовые коллекции.   Ответ: Если требуется иметь дело с множеством битов, можно применить класс BitArray и структуру BitVector32. Класс BitArray расположен в пространстве имен System.Collections, a BitVector32 — в пространстве System.Collections.Specialized.   1. Опишите обобщенные коллекции.   Ответ: Как правило, обобщенные коллекции действуют по тому же принципу, что и необобщенные, за исключением того, что обобщенные коллекции типизированы. Это означает, что в обобщенной коллекции можно хранить только те элементы, которые совместимы по типу с ее аргументом.   1. Опишите параллельные коллекции.   Ответ: Они могут безопасно использоваться в многопоточной программе, где возможен одновременный доступ к коллекции со стороны двух или больше параллельно исполняемых потоков. |  |