



# The C# Programming Language

## Professional

Системные коллекции

# Коллекция

## ArrayList

Системная коллекция элементов `ArrayList` позволяет хранить данные любого типа приводя их к базовому типу `object`.

Для типов по значению требуется предварительная упаковка  
(не рекомендуется к использованию).

# Интерфейс

## IEqualityComparer

Методы интерфейса **IEqualityComparer**:

`bool Equals(object x, object y)` – определяет равенство объектов.

`int GetHashCode(object obj)` – возвращает хэш-код объекта.

# Интерфейс

## IComparer

Методы интерфейса **IComparer**:

`int Compare(object x, object y)` – выполняет сравнение двух объектов и возвращает значение, показывающие, является ли первый объект больше, равным или меньше второго объекта.

# Очередь

## Queue

Системная коллекция элементов **Queue**, реализует принцип очереди: FIFO – First In First Out.

Для помещения элементов в коллекцию предназначен метод `Enqueue()`.

При извлечении элементов при помощи метода `Dequeue()` происходит фактическое удаление из коллекции.

Получить значение элемента без удаления его из коллекции можно при помощи метода `Peek()`.

# Стек

## Stack

Системная коллекция элементов **Stack**, реализует принцип стека: LIFO – Last In First Out.

Для помещения элементов в коллекцию предназначен метод `Push()`.

При извлечении элементов при помощи метода `Pop()` происходит фактическое удаление из коллекции.

Получить значение элемента без удаления его из коллекции можно при помощи метода `Peek()`.

# Хэш таблица

## Hashtable

Системная коллекция **Hashtable**, хранит пары:  
«ключ - значение».

Доступ к элементам можно осуществлять по ключам.

Хранимая информация требует уникальности хэш-кодов, что означает невозможность хранения одинаковых значений.

Не рекомендуется к использованию, если размер коллекции будет менее 10 элементов.

# Словарь

## ListDictionary

Системная коллекция **ListDictionary**, хранит пары  
«ключ - значение».

Подходит для хранения небольшого количества элементов,  
поскольку организована по принципу обычного массива.



# Словарь

## HybridDictionary

Системная коллекция **HybridDictionary**, хранит пары «ключ - значение».

Ведет себя как **ListDictionary** при работе с маленькими наборами или как **Hashtable** при работе с большими наборами элементов.

Рекомендуется к использованию в тех случаях, когда невозможно определить размер коллекции заранее.

# Сортированный список

## SortedList

Системная коллекция **SortedList**, хранит пары  
«ключ - значение».

Все элементы коллекции упорядочены по ключу, при  
добавлении новых элементов упорядочивание  
происходит автоматически.

Доступ к элементам возможен как по ключам, так и по  
соответствующим индексам.

# Упорядоченный словарь

## OrderedDictionary

Системная коллекция **OrderedDictionary**, хранит пары «ключ - значение».

Размещение элементов соответствует порядку их добавления в коллекцию, что позволяет автоматически сохранять элементы в хронологическом порядке.

# БИТОВЫЙ МАССИВ

## BitArray

Системная коллекция **BitArray** предназначена для работы с битовыми данными.

Размер коллекции невозможно менять динамически, для изменения размера необходимо явно изменить значение свойства **Length**.

# БИТОВЫЙ массив

## BitVector32

Системная коллекция **BitVector32** предназначена для работы с единичным 32-битным числом.

Имеет фиксированный размер.

Набор очень удобно использовать для создания битовых масок, а также для упаковки битов.

# Строковая коллекция

## StringCollection

Системная коллекция `StringCollection`, имеет динамически изменяемый размер и позволяет хранить только строковые значения.

# Строковой словарь

## StringDictionary

Системная коллекция `StringDictionary`, хранит пары «ключ - значение» с тем лишь ограничением, что и ключи и значения обязательно должны иметь строковой тип.

# Коллекция Имя-Значение

## NameValueCollection

Системная коллекция **NameValueCollection**, хранит пары «ключ - значение».

Основным отличием этой коллекции от всех предыдущих является то, что она позволяет хранить несколько значений под одним ключом.



# Обобщенные Список, Очередь и Стек

List<T>, Queue<T>, Stack<T>

Обобщенные коллекции `List<T>`, `Queue<T>` и `Stack<T>`, имеют те же функциональные возможности, что и их необобщенные аналоги, но позволяют эффективно управлять контролем типов, а также исключают необходимость использования упаковки/распаковки.

# Обобщенный словарь

Dictionary<TKey, TValue>

Обобщенный словарь **Dictionary**<TKey, TValue>, позволяет выполнять контроль типов ключа и значения.

# Обобщенный сортированный список

`SortedList<TKey, TValue>`

Список `SortedList<TKey, TValue>`, упорядоченный по ключу.

Как критерий упорядочивания используется реализация интерфейса `IComparer<T>`.

# Обобщенный сортированный словарь

SortedDictionary<TKey, TValue>

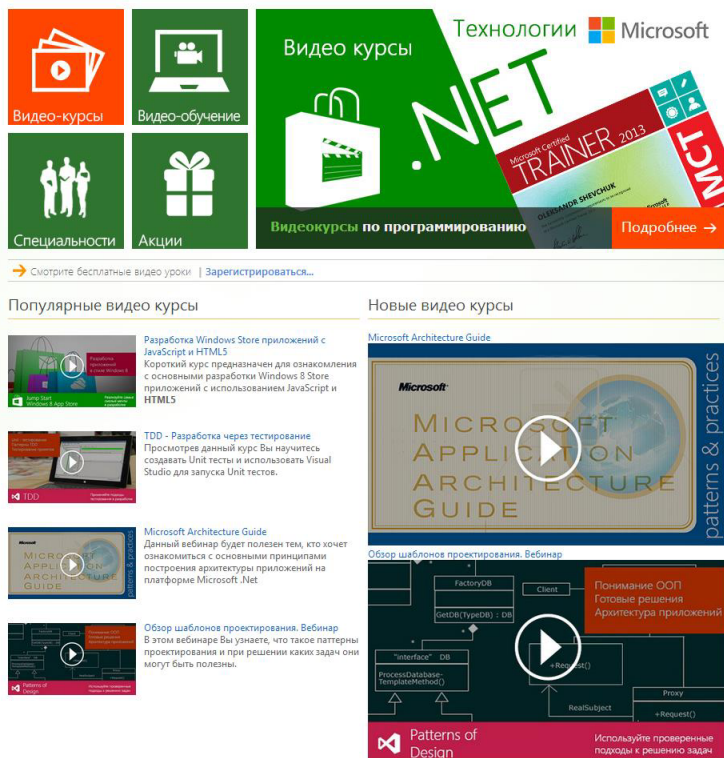
**SortedDictionary**<TKey, TValue> - представляет собой набор пар «ключ - значение», упорядоченных по ключу.

# Обобщенный связанный список

`LinkedList<T>`

`LinkedList<T>` - двунаправленный список.

# Q&A



Перейти к видеопорталу  
[video.cbsystematics.com](http://video.cbsystematics.com)

## Гарантия качества

Видео курсы Учебного центра CyberBionic Systematics - результат проверенной годами методики обучения программистов. Они разработаны сертифицированными тренерами Microsoft для учебного центра CyberBionic Systematics

## Преимущества видео обучения

- Вы можете просматривать учебный материал повторно необходимое количество раз
- Вы можете делать остановки в обучении для выполнения задания с учетом Вашей способности восприятия нового материала
- Вы обучаетесь у сертифицированных тренеров Microsoft

Видео курсы - это возможность обучаться самостоятельно, а также многократно просматривать и повторять материал учебного курса, если Вы обучаетесь очно или on-line. Мы рекомендуем видеообучение также специалистам, которым нужно систематизировать и углубить знания, полученные ранее в ВУЗе.

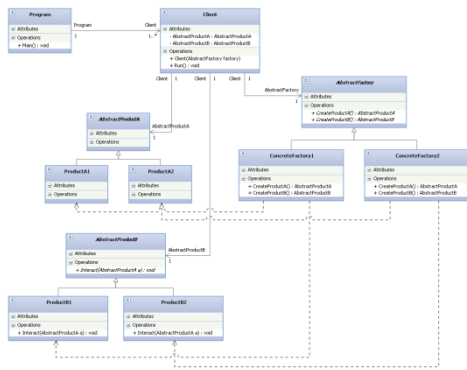


Windows Azure



CyberBionic Systematics®  
TestProvider

**CyberBionic**  
systematics



Задачи, с которыми сталкиваются разработчики программного обеспечения, как правило, довольно однотипны. Кроме того, в том или ином виде они уже были решены до нас. Шаблоны проектирования представляют собой коллекцию тщательно отобранных, наиболее общих принципов решения типовых проблем. Их высокий уровень абстракции позволяет отделить основные принципы реализации от конкретных прикладных областей, что, в свою очередь, дает прекрасную возможность не просто реализовывать шаблоны непосредственно на практике, но и использовать их как некий набор условных обозначений для четкой классификации даже самых сложных задач. В этом контексте, шаблоны проектирования являются неким общим языком, который исключает неоднозначность толкования и значительно ускоряет процесс разработки.

## Описание курса:

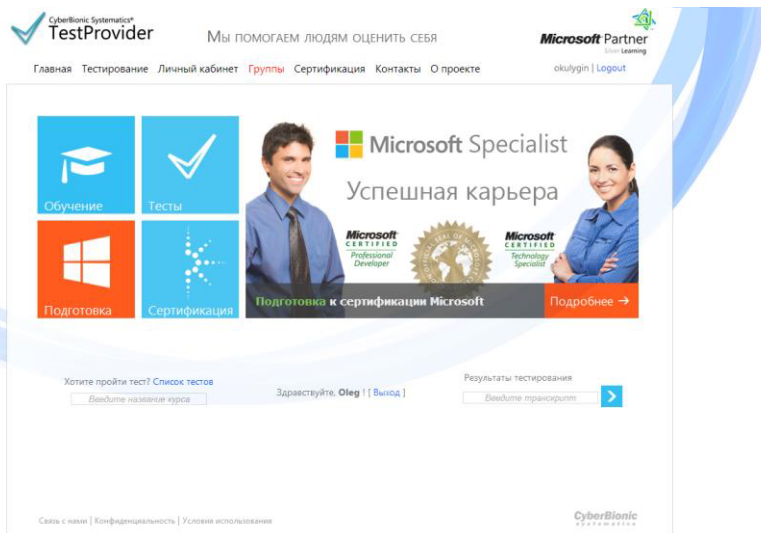
Курс "Шаблоны проектирования" поможет Вам в кратчайшие сроки освоить приемы проектирования. Вы сможете четко классифицировать задачи проектирования и однозначно описывать наиболее подходящие способы их решения. Каждый шаблон представляет собой инструмент, который Вы будете неоднократно использовать в своей практике, получая при этом все преимущества, которые дают надежные, проверенные временем решения.

Длительность:  
20 часов/10 дней.

Узнать более подробно о курсе на сайте:

[edu.cbsystematics.com](http://edu.cbsystematics.com)





Перейти к тестированию  
[www.TestProvider.com](http://www.TestProvider.com)

## Тестирование IT-специалистов

**TestProvider** обеспечивает надежную и объективную оценку технических знаний и опыта работы IT-специалиста с программными продуктами Microsoft.

## Сертификация IT-специалистов

Подтвердите ваш практический опыт работы с технологиями **Microsoft**, получив сертификацию, соответствующую той работе, которую вы выполняете сейчас или желаете получить в будущем.

Компании **Microsoft**, **CyberBionic Systematics** и **Intel** на базе портала [TestProvider](http://TestProvider) компании CyberBionic Systematics с использованием платформы Microsoft Azure совместно с Министерством науки и образования Украины проводят Всеукраинское дистанционное мониторинговое исследование уровня сформированности у выпускников учебных заведений навыков использования информационно-коммуникативных технологий в практической деятельности.

# **CyberBionic** **s y s t e m a t i c s**

*Coevolution of humans and machines.*