Interfaces

|  |  |  |
| --- | --- | --- |
| Назва | Загальний вигляд | Приклад |
| ICloneable  Клонування об’єктів | public interface ICloneable  {      object Clone();  } | public object Clone()      {          return new Person { Name = this.Name, Age = this.Age };      } |
| IComparable  Сортування об’єктів | public interface IComparable  {      int CompareTo(object o);  } | public int CompareTo(object o)      {          Person p = o as Person;          if (p != null)              return this.Name.CompareTo(p.Name);          else              throw new Exception("Невозможно сравнить два объекта");      } |
| IComparer  Порівняння двох об’єктів | interface IComparer  {      int Compare(object o1, object o2);  } | public int Compare(Person p1, Person p2)      {          if (p1.Name.Length > p2.Name.Length)              return 1;          else if (p1.Name.Length < p2.Name.Length)              return -1;          else              return 0;      } |
| ICustomFormatter  Строкове предсталення значення об’єкта | public interface ICustomFormatter  {  string Format(string format, object arg, IFormatProvider formatProvider);  } | public string Format(string format, object arg, IFormatProvider formatProvider)  {  if ("ue" != format)  return \_FormatOther(format, arg);  var s = Convert.ChangeType(arg ,typeof(string)) as string;  if (null == s) // sanity  s = "";  return Uri.EscapeDataString(s);  } |
| IConvertible  преобразуют значение реализующей ссылки или типа значения в тип среды CLR с эквивалентным значением. |  | bool IConvertible.ToBoolean(IFormatProvider provider)  {  if( (x != 0.0) || (y != 0.0) )  return true;  else  return false;  } |
| IDisposable  механизм для освобождения неуправляемых ресурсов. | public interface IDisposable  {  void Dispose();  } | public void Dispose()  {  Dispose(true);  GC.SuppressFinalize(this);  } |
| IDictionary  Представляет небазовую коллекцию пар "ключ-значение". |  |  |

|  |  |  |
| --- | --- | --- |
| IEnumerable  Предоставляет перечислитель, который поддерживает простой перебор элементов неуниверсальной коллекции. | public interface IEnumerable  {  IEnumerator GetEnumerator();  } | IEnumerator IEnumerable.GetEnumerator()  {  return (IEnumerator) GetEnumerator();  }  public PeopleEnum GetEnumerator()  {  return new PeopleEnum(\_people);  } |
| IEnumerator  Поддерживает простой перебор по неуниверсальной коллекции | public interface IEnumerator  {  object Current { get; }  bool MoveNext();  void Reset();  } | public bool MoveNext()  {  position++;  return (position < \_people.Length);  }  public void Reset()  {  position = -1;  } |
| IEqualityComparer  Определяет методы, поддерживающие сравнение объектов на предмет равенства. | public interface IEqualityComparer  {  bool Equals(object x, object y);  int GetHashCode(object obj);  } | public new bool Equals(object x, object y)  {  return myComparer.Compare(x, y) == 0;  }  public int GetHashCode(object obj)  {  return obj.ToString().ToLower().GetHashCode();  } |
| IEquatable  Определяет обобщенный метод, который реализуется типом значения или классом для создания метода с целью определения экземпляров. | public interface IEquatable<T>  {  bool Equals(T other);  } | public bool Equals(Person other)  {  if (other == null)  return false;  if (this.uniqueSsn == other.uniqueSsn)  return true;  else  return false;  } |
| IFormattable  Предоставляет функциональные возможности для форматирования значения объекта в представление строки | public interface IFormattable  {  string ToString(string format, IFormatProvider formatProvider);  } | public string ToString(string format, IFormatProvider provider)  {  if (String.IsNullOrEmpty(format)) format = "G";  if (provider == null) provider = CultureInfo.CurrentCulture;  switch (format.ToUpperInvariant())  {  case "G":  case "C":  return temp.ToString("F2", provider) + " °C";  case "F":  return Fahrenheit.ToString("F2", provider) + " °F";  case "K":  return Kelvin.ToString("F2", provider) + " K";  default:  throw new FormatException(String.Format("The {0} format string is not supported.", format));  }  } |
|  |  |  |
|  |  |  |
|  |  |  |