

Тест 3

Ответы не мои)

Обязательно ли событие (event) должно быть членом класса?

1 балл

- ☒ да
- ☐ нет

Отменить выбор

Что выведет на экран данный код?

1 балл

```
4 public class Program
5 {
6     public static void Main()
7     {
8         IDictionary<int, string> dict = new Dictionary<int, string>();
9
10        dict.Add(new KeyValuePair<int, string>(1, "One"));
11        dict.Add(new KeyValuePair<int, string>(2, "Two"));
12
13
14        Console.WriteLine(dict.Keys.GetType().IsValueType);
15
16    }
17 }
```

- ☐ true
- ☒ false

Отменить выбор

Обобщения и исключения, коллекции

Выберите верные утверждения об исключениях

1 балл

- ☒ для каждого блока try может быть несколько блоков catch
- ☐ для каждого блока try может быть несколько блоков finally
- ☐ блоки try могут быть вложенные
- ☒ блок try может не иметь блока finally
- ☐ поиск подходящего блока catch осуществляется сверху вниз
- ☒ можно создавать пользовательские исключения наследуя их от System.Exception

Пусть задана коллекция. Можно ли хранить в коллекции list объекты разных типов?

1 балл

```
ArrayList list = new ArrayList();
```

- ☒ Да
- ☐ Нет

Отменить выбор

Какой интерфейс предназначен для применения в тех случаях, когда требуется определить относительный порядок следования двух объектов?

1 балл

- ☒ IEnumerable
- ☐ ICloneable
- ☐ IComparable
- ☐ IObject
- ☐ IGettable

Отменить выбор

Выберите верные утверждения для обобщений

1 балл

- ☐ универсальный (generic) тип не перегружается
- ☒ универсальный (generic) тип может содержать другой универсальный тип
- ☒ generic могут быть методы, классы, исключения
- ☒ generic поддерживает механизм ограничений

Что выведет на экран данный код?

1 балл

```
class Program
{
    static void Main(string[] args)
    {
        Console.Write(method("String"));
    }
    public static string method(string str)
    {
        try
        {
            if (str[91] == 'r') return "Yes";
            else return "No";
        }
        catch (StackOverflowException)
        {
            Console.Write("Exception ");
        }
        finally
        {
            Console.Write("Finally ");
            str = "working";
        }
        return str;
    }
}
```

- ☒ Exception Finally working
- ☐ Finally working
- ☐ Exception Finally String
- ☐ String Exception
- ☐ Необработанное исключение: System.IndexOutOfRangeException

Отменить выбор

Какие из перечисленных ниже ограничений наложены на тип U?

1 балл

```
public class Figure { }  
public class Rectangle : Figure { }  
  
public class LinkedSet<U, T> where U : class, T, new() { }  
  
static void Main()  
{  
  
    LinkedSet<Rectangle, Figure> чел = new LinkedSet<Rectangle, Figure>();  
    LinkedSet<Figure, String> рис = new LinkedSet<Figure, String>();  
  
}
```

- ☒ ограничение на базовый класс
- ☐ ограничение на интерфейс
- ☐ ограничение на конструктор
- ☐ ограничение ссылочного типа
- ☐ ограничение значимого типа

Как можно повторно вызывать то же самое исключение в блоке обработчика catch?

1 балл

- ☐ throw new Exception();
- ☐ finally {..}
- ☒ throw;
- ☐ catch {..}
- ☐ нет правильного ответа

Отменить выбор

Выберите верную конструкцию фильтра исключений

1 балл

- ☒ catch (Exception ex) when
- ☐ catch (Exception ex) if
- ☐ catch
- ☐ catch (Exception ex) finally

Отменить выбор

Можно ли в generic использовать ограничение new() одновременно с ограничением типа значения?

можно создавать пользовательские исключения наследуя их от System.SystemException?