Основные программные конструкции C#

1. При выполнении следующего кода  
  
class Program  
{  
    static bool Method1()  
    {  
        Console.WriteLine("in Method1");  
        return false;  
    }  
    static bool Method2()  
    {  
        Console.WriteLine("in Method2");  
        return true;  
    }  
    static void Main(string[] args)  
    {  
        if (Method1() & Method2())  
        {  
            Console.WriteLine("insite statement if");  
        }  
    }  
}  
  
на экран будет выведено

Ответ: in Method1  
     in Method2

12. При выполнении следующего кода  
  
static bool Method1()  
{  
    Console.WriteLine("in Method1");  
    return false;  
}  
static bool Method2()  
{  
    Console.WriteLine("in Method2");  
    return true;  
}  
static void Main(string[] args)  
{  
    if (Method1() && Method2())  
    {  
        Console.WriteLine("insite statement if");  
    }  
}  
  
на экран будет выведено

}

Ответ: in Method1.

17. При выполнении следующего кода   
  
static bool Method1(int i)  
{  
    return (i % 2 == 0) ? true : false;  
}  
  
public static void Main(string[] args)  
{  
    int a = 2;  
    Console.WriteLine(Method1(++a\*2));  
    Console.WriteLine(Method1(++a+1));  
    Console.WriteLine(Method1(a+++2));  
}  
  
на экран будет выведено

}

Ответ: True  
     False  
     True

2. Что произойдет в результате выполнения кода

char[] s = new char[]{ 'a', 'b', 'c', 'd', 'e' };

for (int i = 1; i <= s.Length; i++)

{

    Console.Write(s[i]);

}

Ответ: На экран будет выведено bcde, а дальше произойдет ошибка времени выполнения.

3. Сколько объектов типа string будет создано во время выполнения следующего кода  
  
string s = "sdfgh";  
if (s.ToLower().Substring(1, 2).EndsWith("exe"));

}

Ответ: 3.

4. Сколько раз выполнится цикл while  
  
int i = 0;  
while(++i < 3)  
{  
    Console.WriteLine(i);  
}

}

Ответ: 2.

5. При выполнении следующего кода  
  
static void Main(string[] args)  
{  
    int a = 0;  
    decimal b = 0;  
    double c = 0.0;  
  
    Console.WriteLine(a == b);  
    Console.WriteLine(a.Equals(b));  
    Console.WriteLine(a == c);  
    Console.WriteLine(a.Equals(c));  
}  
  
на экран будет выведено

}

Ответ: True  
     False  
     True  
     False

6. Что произойдет в результате выполнения кода  
  
static void Main(string[] args)  
{  
    string s;  
    {  
        int s = 10;  
        Console.WriteLine(s);  
    }  
}

}

Ответ: Возникнет ошибка времени компиляции.

7. Как могут инициализироваться readonly поля экземпляра класса?

Ответ: 1) при объявлении; 2) в конструкторе

8. Что произойдет в результате выполнения кода Скомпилируется ли данный код  
  
static void Main(string [] args)  
{  
    int a = 10;  
    int b = 20;  
    byte c = a + b;  
    Console.WriteLine(c);  
}

}

Ответ: Возникнет ошибка времени компиляции.

9. Скомпилируется ли успешно данный фрагмент кода?  
  
private int GetID(string inputText)  
{  
    if (inputText != "")  
        return 1;  
    else if (inputText == "")  
        return 0;  
}

}

Ответ:Нет.

10. Укажите строки, которые вызовут ошибку времени компиляции  
  
1. Type t1 = typeof(System.Int32);  
2. Type t2 = Type.GetType("System.Int32");  
3. Type t3 = Type.GetType(System.Int32);  
4. Type t4 = System.Int32.GetType();

}

Ответ: 1) 3; 2) 4.

11. При выполнении следующего кода  
  
public static void Main(string[] args)  
{  
    int i, j, s = 0;  
    for (i = 0, j = 6; i < j; i++, --j)  
    {  
        s += i;  
    }  
    Console.WriteLine(s);  
}  
  
на экран будет выведено

}

Ответ: 3.

13. Укажите строки, в которых неявно типизированные переменные использованы корректно:

Ответ: 1) static int GetValue(){var t = 9; return t;}

2) var ok = true;

3) var data = (int?)null;

4) var i = 0; var j = i;

5) string s = “Hi”; var data = s;

14. Укажите все корректные способы объявления массива в C#.

Ответ: 1) int[][] x; 2) int[,] x; 3) int[] x;

15. При выполнении следующего кода  
  
public static void Main(string[] args)  
{  
    int a = 101;  
    Console.WriteLine(a>>2);  
}  
  
на экран будет выведено

}

Ответ: 25.

16. При выполнении следующего кода  
  
public static void Main(string[] args)  
{  
    int a = 2;  
    Console.WriteLine((++a)&(a--));  
}  
  
на экран будет выведено

}

Ответ: **3.**

18. Сколько раз выполнится цикл while  
  
public static void Main(string[] args)  
{  
    int count = 3;  
    do  
    {  
    Console.WriteLine("Hello!");  
    } while (count--> 0);  
}

Ответ: 4.

19. Сколько раз выполнится цикл while  
  
public static void Main(string[] args)  
{  
    const int COUNT = 3;  
    do  
    {  
    Console.WriteLine("Hello!");  
    } while (COUNT --> 0);  
}

Ответ: Возникнет ошибка времени компиляции.

20. При выполнении следующего кода   
  
public static void Main(string[] args)  
{  
    object[] s = new object[]{ "Hello", 34, true };  
    foreach (var item in s)  
    {  
        if (item is Int32)  
        {  
            Console.WriteLine(item+1);  
        }  
    }  
}  
  
на экран будет выведено

Ответ: Возникнет ошибка времени компиляции.

21. При выполнении следующего кода   
  
public static void Main(string[] args)  
{  
    dynamic s = new object[]{ "Hello", 34, true };  
    foreach (var item in s)  
    {  
        if (item is Int32)  
        {  
            Console.WriteLine(item+1);  
        }  
    }  
}  
  
на экран будет выведено

Ответ: 35.

22. При выполнении следующего кода   
  
public static void Main(string[] args)  
{  
    dynamic s = new object[]{ "Hello", 34, true };  
    foreach (var item in s)  
    {  
        if (item is Boolean)  
        {  
            Console.WriteLine(item+1);  
        }  
    }  
}  
  
на экран будет выведено

Ответ: Возникнет ошибка времени выполнения.

24. При выполнении следующего кода  
  
public static void Main(string[] args)  
{  
    dynamic s = new object[]{ "Hello", 34, true };  
    foreach (var item in s)  
    {  
        if (item is String)  
        {  
            Console.WriteLine(item+1);  
        }  
    }  
}  
  
на экран будет выведено

Ответ: Hello1.

23. При выполнении следующего кода  
  
class Test  
{  
    static void Main(string[] args)  
    {  
        int i = 1;  
        Console.WriteLine("i = {0}", i++);  
    }  
}  
  
на экран будет выведено

Ответ: i = 1.

25. При выполнении следующего кода   
  
public static void Main(string[] args)  
{  
    int? x = null;  
    int? y = x ?? -1;  
    Console.WriteLine(y??10);  
}  
  
на экран будет выведено

Ответ: -1.

26. Что будет выведено на экран в результате выполнения кода, если бы сейчас на часах было 10:00?

int h =DateTime.Now.Hour;

Console.WriteLine((h > 12) && (h < 16) ? "Добрый день" : (h > 7) && (h < 12) ? "Добрый вечер" : "Доброе утро");

Ответ: На экран будет выведено: Добрый вечер.

27. Что будет выведено на экран в результате выполнения кода:

int x = new int();

while (!(--x==0))

{

  Console.Write(x+= x == 0 ? 0 : 1);

}

Ответ: На экран будет выведено: 0000000000... и программа зациклится.

28. Укажите все способы объявления массива в результате которых он будет корректно инициализирован

Ответ: 1) int[] someArray = new int[4];

2) int [] someArray = new int[] {1,2,3,4};

3) int[] someArray = {1,2,3,4};

4) int[] someArray = new int[4] {1,2,3,4};

5) var b = new[] { "hello", null, "world" };

29. При выполнении следующего кода

public static void Main(string[] args)

{

    goto F2;

F1: Console.WriteLine("Before F2 ");

F2: Console.WriteLine("After F2");

    goto F1;

   Console.WriteLine("Done");

}

на экран будет выведено

Ответ: After F2

     Before F2

     …до бесконечности

30. Какие операторы выбора являются эквивалентом тернарной операции «?:» ?

Ответ: if...else.

31. При выполнении следующего кода  
  
bool operation1 = true;  
bool operation2 = false;  
bool operation3 = false;  
  
if(operation1 ^ operation2 ^ operation3)  
{  
    Console.WriteLine(operation1 ^ operation2);  
    if (operation1 | operation2 | operation3)  
    {  
        Console.WriteLine(operation2 | operation3);  
        if(operation1 & operation2 & operation3)  
        {  
            Console.WriteLine(operation1 & operation3);  
        }  
    }  
}  
  
на экран будет выведено

Ответ: True  
     False

32. При выполнении следующего кода

public static void Main()

{

    Int32 x = 4;

    Object o = x;

    x = 123;

    Console.WriteLine(x + ", " + (Int32)o);

}

на экран будет выведено

Ответ: 123, 4.

33. В какой кодировке хранятся символьные (char) переменные в C#?

Ответ: UTF-16.

Объявление и вызов методов в C#

1. Какое значение хранит переменная result после выполнения кода  
  
private static int Methods(int i, int j)  
{  
    return (j - i) \* i;  
}  
...  
int i = 1;  
int j = 1;  
int result = Methods(i++, i+j);//1;3

Ответ: 2.

10. Какое значение хранит переменная result после выполнения кода  
  
private static int Methods(ref int i, int j)  
{  
    i++;  
    return (j - i) \* i;  
}  
...  
int i = 1;  
int j = 1;  
int result = Methods(ref i, i+++j)+i;

Ответ:0.

2. Для следующего метода  
  
private static void Method(int i = 1, string s = "A", double d = 0.00001)  
{  
    Console.WriteLine("i = {0}, s = {1}, d = {2}", i, s, d);  
}  
  
его вызов в коде  
  
private static int n = 0;  
...  
Method(n++, n++.ToString());  
...  
равносилен следующему вызову

Ответ: Do(0, "1",0.00001);

3. Для следующего метода  
  
private static void Method(int x = 9, string s = "A", DateTime dt = default(DateTime))  
{  
    Console.WriteLine("x = {0}, s = {1}, dt = {2}", x, s, dt);  
}  
  
его вызов в коде  
  
private static int n = 0;  
...  
Method(s : (n++).ToString(), x : n++);  
...  
равносилен следующему вызову

Ответ: Do(1,"0",default(DataTime));

4. Выберите спецификации методов, в списке параметров которых есть ошибки.

Ответ: 1) private static void Method(int a, DateTime dt = DateTime.Now);

2) private static void Method(int a : 4, DateTime dt = default(DateTime));

5. Для следующего метода

private static int AddArray(int[] values)

{

    int sum = 0;

    for (int x = 0; x < values.Length; x++)

        sum += values[x];

    return sum;

}

указать операторы, которые не приведут к ошибке времени компиляции.

Ответ: 1) int[] arr = new int[] { 1, 2, 3, 4, 5 }; Console.WriteLine(AddArray(arr));

2) Console.WriteLine(AddArray(new int[] { 1, 2, 3, 4, 5 }));

6. Для следующего метода

private static int Add(params int[] values)

{

    int sum = 0;

    for (int x = 0; x < values.Length; x++)

        sum += values[x];

    return sum;

}

указать операторы, которые не приведут к ошибке времени компиляции.

Ответ: 1) int[] arr = new int[] { 1, 2, 3, 4, 5 }; Console.WriteLine(Add(arr));

2) Console.WriteLine(AddArr(new int[] { 1, 2, 3, 4, 5 }));

3) Console.WriteLine(Add(1, 2, 3, 4, 5));

7. После выполнения следующего кода значения переменных value, str1, str2 будут равны соответственно  
  
class OutReturnExample  
{  
    static void Method(out int i, out string s1, out string s2)  
    {  
        i = 44;  
        s1 = "I've been returned";  
        s2 += "Hi!";  
    }  
  
    static void Main()  
    {  
        int value = 40;  
        string str1, str2;  
        Method(out value, out str1, out str2);  
    }  
}

Ответ: Ошибка компиляции.

8. Выберите объявления методов с правильным использованием ключевого слова params:

Ответ: 1) public void Method (params double[] array) { }

2) public void Method (double volume, params double[] array) { }

9. Выберете правильные утверждения

Ответ: 1) Следующая перегрузка методов возможна:

    public void Method(int i) { }

    public void Method(out int i) { i = 5; }

2) Следующая перегрузка методов не возможна:

    private static int Add(params int[] values)

    private static int Add(int first, int second, int third)

3) Следующая перегрузка методов не возможна:

    private static int Add(params int[] values)

    private static int Add(int first, int second, int third)

11. Что будет выведено после выполнения следующего кода  
  
class OutReturnExample  
{  
    int x=20;  
    double y=3;  
    public double Method(int x, int y)  
    {  
        return this.x-y/2;  
    }  
  
    public static void Main(string[] args)  
    {  
        OutReturnExample a = new OutReturnExample();  
        Console.WriteLine(a.Method(7,9));  
    }  
}

Ответ: 16.

12. Что будет выведено после выполнения следующего кода  
  
class c1  
{  
    int x=20;  
    double y=3;  
    public static double Method(int x, int y)  
    {  
        return this.x-y/2;  
    }  
  
    public static void Main(string[] args)  
    {  
        c1 a = new c1();  
        Console.WriteLine(a.Method(7,9));  
    }  
}

Ответ: Ошибка компиляции.

13. Что будет выведено после выполнения следующего кода  
  
class c1  
{  
    public static void Method(ref int x, ref float y)  
    {  
        x\*=x;  
        y\*=2;  
    }  
  
    public static void Main(string[] args)  
    {  
        int x=2, y=3; float d=3;  
        Method(ref x, ref y);  
        Console.WriteLine(x+” “+y+d);  
    }  
}

Ответ: Ошибка компиляции

14. Что будет выведено после выполнения следующего кода  
  
class c1  
{  
    public static int Method(int x)  
    {  
        return (x>0)?x%10+Method(x/10):0;  
  
    }  
  
    public static void Main(string[] args)  
    {  
        Console.WriteLine(Method(12345));  
    }  
}

Ответ: 15.

15. При выполнении следующего кода  
  
static void Method(out int value, ref int j)  
{  
    value = 10;  
    j += 5;  
}  
static void Main(string[] args)  
{  
    int value, j = 5;  
    Method(out value, ref j);  
    Console.WriteLine(value +++ j);  
    Console.WriteLine(value + j);  
}  
  
на экран будет выведено

Ответ: 20  
     21

16. При выполнении следующего кода  
  
static void Method(out int value, ref int j)  
{  
    value = 10;  
    j = 5;  
    }   
  
static void Main(string[] args)  
{  
    int value;  
    int j;  
  
    Method(out value, ref j);  
    Console.WriteLine(value + j);  
}  
  
на экран будет выведено

Ответ: Возникнет ошибка времени компиляции.

17. При выполнении следующего кода  
  
static void Method(ref int value)  
{  
    if (value++ < 100)  
        Method(ref value);  
}   
  
static void Main(string[] args)  
{  
    int value = 1;  
    Method(ref value);  
    Console.WriteLine(value);  
}  
  
на экран будет выведено

Ответ: 101

19. При выполнении следующего кода  
  
static void Method(ref int value)  
{  
    if (++value < 100)  
        Method(ref value);  
}   
  
static void Main(string[] args)  
{  
    int value = 1;  
  
    Method(ref value);  
    Console.WriteLine(value);  
}  
  
на экран будет выведено

Ответ: 100.

18. При выполнении следующего кода  
  
static class StringAdditional  
{  
    public static bool ToInt(this String s, out int result)  
    {  
        return int.TryParse(s, out result);  
    }  
}  
class Program  
{  
    static void Main(string[] args)  
    {  
        string s = "54";  
        int result = 0;  
        s.ToInt(out result);  
        Console.WriteLine(result);  
    }  
}  
  
на экран будет выведено

Ответ: 54

20. При выполнении следующего кода, что будет выведено на экран

    static class A

    {

        internal static int ToInt(this string s)

        {

            return Convert.ToInt32(s);

        }

    }

    class Program

    {

        static void Main(string[] args)

        {

            int x = 0;

            string str=(++x).ToString();

            x += str.ToInt();

            Console.WriteLine(x++);

        }

    }

Ответ: 2.

21. При выполнении следующего кода, что будет выведено на экран

    static class A

    {

        internal static int ToInt(this string s)

        {

            return Convert.ToInt32(s);

        }

    }

    class Program

    {

        static void Main(string[] args)

        {

            int x = 1;

            string str=(++x).ToString();

            x += str.ToInt();

            Console.WriteLine(x++);

        }

    }

Ответ:4.

22. При выполнении следующего кода, что будет выведено на экран

    class Program

    {

        static class A

        {

           internal static int ToInt(this string s)

           {

               return Convert.ToInt32(s);

           }

        }

        static void Main(string[] args)

        {

            int x = 0;

            string str=(++x).ToString();

            x += str.ToInt();

            Console.WriteLine(x++);

        }

    }

Ответ: Возникнет ошибка времени компиляции.

23. При выполнении следующего кода, что будет выведено на экран

   static class A

    {

        internal static int ToInt(this string s, int i)

        {

            return i++;

        }

    }

    class Program

    {

        static void Main(string[] args)

        {

            int x = 0;

            string str=(++x).ToString();

            x += str.ToInt(x);

            Console.WriteLine(x++);

         }

    }

Ответ:2.

Создание новых типов в C#

1. Определена переменная c

Color c = Color.Blue;  
  
где   
  
enum Color{ White, Red, Green, Blue, Orange};

При выполнении какого оператора на консоль будет выведено целочисленное значение 3?

Ответ: Console.WriteLine(c.ToString("D"));

1. В перечислении Season

enum Season : sbyte {Spring = -3, Summer, Fall, Winter};

литерал Winter равен

Ответ: 0.

3. Если при определении класса не определить конструктор по умолчанию, то?

Ответ: Конструктор по умолчанию генерируется автоматически, только если класс не имеет других конструкторов.

4. Что произойдет, если при определении разделяемого метода, не обеспечить его реализацию?

Ответ: Код будет компилироваться и работать, и можно будет вызвать разделяемый метод.

5. Результатом выполнения следующего кода будет

struct Book  
{  
    public string title;  
    public string author;  
    public int copyrigth;  
  
    public Book()  
    {  
        author = "Author";  
        title = "Title";  
        copyrigth = 1000;  
  
        Console.WriteLine("This is a book.");  
    }  
  
}  
  
class Program  
{  
    static void Main(string[] args)  
    {  
        Book book = new Book();  
        Console.WriteLine("This is not a book.");  
    }  
}

Ответ: Ошибка времени компиляции.

6. Отметьте операторы, которые генерируют идентичный IL код?

Ответ: 1) int a = new int();

2) int b = 0;

3) System.Int32 d = new System.Int32();

4) Int32 e = 0;

7. Если Residence является классом

public enum ResidenceType  
{  
    House,  
    Flat,  
    Bungalow,  
    Apartment  
};  
  
public class Residence  
{  
    public ResidenceType type;  
    public int numberOfBedrooms;  
  
    public Residence(ResidenceType type, int numberOfBedrooms)  
    {  
        this.type = type;  
        this.numberOfBedrooms = numberOfBedrooms;  
    }  
}

какое сообщение будет выведено на консоль при выполнении следующего кода?

Residence myHouse = new Residence(ResidenceType.House, 2);  
Residence anotherHouse = new Residence(ResidenceType.House, 2);  
if (myHouse == anotherHouse)  
{  
    Console.WriteLine("They are the same house");  
}  
else  
{  
    Console.WriteLine("They are different houses");  
}

Ответ: They are different houses

8. Если Residence является структурой

public enum ResidenceType  
{  
    House,  
    Flat,  
    Bungalow,  
    Apartment  
};  
public struct Residence  
{  
    public ResidenceType type;  
    public int numberOfBedrooms;  
  
    public Residence(ResidenceType type, int numberOfBedrooms)  
    {  
        this.type = type;  
        this.numberOfBedrooms = numberOfBedrooms;  
    }  
}

какое сообщение будет выведено на консоль при выполнении следующего кода?

Residence myHouse = new Residence(ResidenceType.House, 4);  
Residence anotherHouse = new Residence(ResidenceType.House, 4);  
if (myHouse == anotherHouse)  
{  
    Console.WriteLine("They are the same house");  
}  
else  
{  
    Console.WriteLine("They are different houses");  
}

Ответ: Код не будет компилироваться.

9. Пусть определена структура

private struct Point : ICloneable  
{  
    public Int32 x, y;  
  
    public override String ToString()  
    {  
        return String.Format("({0}, {1})", x, y);  
    }  
  
    public Object Clone()  
    {  
        return MemberwiseClone();  
    }  
}

cоздан экземпляр структуры

Point p;  
  
и инициализированы ее поля  
  
p.x = 10;  
p.y = 20;  
  
Отметьте строки, в которых происходит упаковка.

Ответ: 1) Console.WriteLine(p.GetType());

2) ICloneable c = p;

10. Отметьте правильные утверждения

Ответ: 1) Перечисления можно объявить в классе 2) Перечисления можно объявить в пространстве имен.

11. Для какого типа используется Nullable <T>?

Ответ: Типы значения (value types).

12. Что из перечисленного верно?

Ответ: 1) Переменные ссылочного типа (reference type) размещаются в стеке потока и хранят адрес объекта в управляемой куче.

2) Переменные значимого типа (value type) размещаются в стеке потока.

13. Отметьте все верные утверждения о классе SomeClass:  
  
    internal class SomeClass()  
    {  
        public void SomeClass  
        {  
        }  
    }

Ответ: 1) SomeClass наследуется от System.Object.

2) SomeClass доступен для использования только внутри сборки, в которой он определён.

14. Что будет выведено после выполнения следующего кода  
class c1  
{  
    public static void Main(string[] args)  
    {  
        Console.WriteLine(Byte.MaxValue+1);  
    }  
}

Ответ: 256.

15. Что будет выведено после выполнения следующего кода  
  
class c1  
{  
    public static void Main(string[] args)  
    {  
        byte a = Byte.MaxValue+1;  
        Console.WriteLine(a);  
    }  
}

Ответ: Ошибка компиляции.

16. Что будет выведено после выполнения следующего кода  
  
class c1  
{  
    public static void Main(string[] args)  
    {  
        object a = 2;  
        object b = a;  
        a=3;  
        Console.WriteLine(b);  
    }  
}

Ответ: 2.

17. Что будет выведено после выполнения следующего кода  
  
class c1  
{  
    public static void Main(string[] args)  
    {  
        int[] a = { 1, 2, 3, 4, 5 };  
        int[] b = a;  
        a[0] = 3;  
        foreach(int i in b) Console.Write(i+" ");  
    }  
}

Ответ: 3 2 3 4 5.

18. Что будет выведено после выполнения следующего кода  
  
class c1  
{  
    public static void Main(string[] args)  
    {  
        Type type = 12.GetType();  
        Console.WriteLine(type);  
    }  
}

Ответ: System.Int32.

19. Результатом выполнения следующего кода будет  
{  
    public static int Counter { get; private set; }  
    static A()  
    {  
        Counter = 10;  
    }  
    public A(int counter)  
    {  
        Counter = counter;  
    }  
}  
  
class Program  
{  
    static void Main(string[] args)  
    {  
        Console.WriteLine(A.Counter);  
        A a = new A(20);  
        Console.WriteLine(A.Counter);  
        Console.ReadKey();  
    }  
}

Ответ: 10  
     20

20.Результатом выполнения следующего кода будет  
  
static class A  
{  
    public static int Counter { get; private set; }   
    public A(int counter)  
    {  
        Counter = counter;  
    }  
}  
  
class Program  
{  
    static void Main(string[] args)  
    {  
        A a = new A(20);  
        Console.WriteLine(A.Counter);  
        Console.ReadKey();  
    }  
}

Ответ: Ошибка времени компиляции.

21. Результатом выполнения следующего кода будет  
  
static class A  
{  
    public static int Counter { get; private set; }   
    static A(int counter)  
    {  
        Counter = counter;  
    }  
}  
  
class Program  
{  
    static void Main(string[] args)  
    {  
        A a = new A(20);  
        Console.WriteLine(A.Counter);  
        Console.ReadKey();  
    }  
}

Ответ: Ошибка времени компиляции.

22. Результатом выполнения следующего кода будет  
  
class A  
{  
    public static int Counter { get; private set; }   
    private A()  
    {  
        Counter = 20;  
    }  
}  
  
class Program  
{  
    static void Main(string[] args)  
    {  
        Console.WriteLine(A.Counter);  
        Console.ReadKey();  
    }  
}

Ответ: 0.

23. Результатом выполнения следующего кода будет  
  
class A  
{  
    public static int Counter { get; private set; }   
    static A()  
    {  
        new A();  
    }  
    private A()  
    {  
        Counter = 20;  
    }  
}  
  
class Program  
{  
    static void Main(string[] args)  
    {  
        Console.WriteLine(A.Counter);  
        Console.ReadKey();  
    }

Ответ: 20.

24. Что будет выведено на экран в результате выполнения данного кода

    class Program

    {

      struct A

        {

            public int x;

            public string ToString()

            {

                return Convert.ToString(x++);

            }

        }

      static void Main(string[] args)

        {

            A a = new A { x = 3 };

            Console.WriteLine(a.ToString());

        }

    }

Ответ: 3.

25. Что будет выведено на экран в результате выполнения данного кода

    class Program

    {

      struct A

        {

            public int x=0;

            public string ToString()

            {

                return Convert.ToString(x++);

            }

        }

      static void Main(string[] args)

        {

            A a = new A { x = 3 };

            Console.WriteLine(a.ToString());

        }

    }

Ответ: Ошибка времени компиляции.

26. Что будет выведено на экран в результате выполнения данного кода

    class Program

    {

     class A

        {

            public int x=0;

            public A()

              {

                 x = 1;

              }

           public string ToString()

            {

                return Convert.ToString(x++);

            }

        }

      static void Main(string[] args)

        {

            A a = new A { x = 3 };

            Console.WriteLine(a.ToString());

        }

    }

Ответ: 3.

27. Что будет выведено на экран в результате выполнения данного кода

    class Program

    {

     class A

        {

            public int x=0 {get; set;}

            public A()

              {

                 x = 1;

              }

           public string ToString()

            {

                return Convert.ToString(x++);

            }

        }

      static void Main(string[] args)

        {

            A a = new A { x = 3 };

            Console.WriteLine(a.ToString());

        }

    }

Ответ: Ошибка времени компиляции.

Инкапсуляция данных и методов. Перегрузка операций

1. Отметьте оператор, который определяет индексатор:

**Ответ:** public class Item[ int index]

{

get {…}

set {…}

}

1. Статический конструктор вызывается:

**Ответ:** перед созданием первого экземпляра или ссылки на какие-либо статические члены.

1. Отметьте правильные утверждения:

**Ответ:** 1) для определения индексатора используется ключевое слово this.

2) индексаторы могут иметь более одного формального параметра.

3) в индексаторе допустимо использование опциональных параметров.

1. Отметьте правильные утверждения:

**Ответ:** 1) модификатор доступа internal – доступ к типу или члену возможен из любого кода в той же сборке, но не из другой сборки.

**2)**  если не указать модификатор доступа для типа, по умолчанию применяется уровень доступа internal.

1. При выполнении следующего блока кода

class User  
{  
    internal static int usersOnline;  
    internal User()  
    {  
        usersOnline++;  
    }  
}  
User a = new User();  
User b = new User();  
User c = new User();  
User d = new User();  
int totalUsersOnline = User.usersOnline;

значение переменой totalUsersOnline будет равно:

**Ответ:**  4

1. При выполнении следующего блока кода

class User  
{  
    internal int usersOnline;  
    internal User()  
    {  
        usersOnline++;  
    }  
}  
User a = new User();  
User b = new User();  
User c = new User();  
User d = new User();  
int totalUsersOnline = d.usersOnline;

значение переменной totalUsersOnline будет равно:

**Ответ:** 1

1. Какой метод реализуется для создания метода расширения Reverse() класса string :

**Ответ:** public static string Reverse(this string s) .

1. Отметьте операции /ключевые слова, которые не поддерживают перегрузку напрямую:

**Ответ:** 1) +=

2) ||

1. Какой метод реализуется для перегрузки операции сложения:

**Ответ:** public static ComplexNumber operator + (ComplexNumber a, ComplexNumber b)

1. Отметьте модификаторы, которые нельзя использовать для изменения метода override:

**Ответ:** 1) virtual 2) new 3) static

1. Отметьте модификаторы, которые нельзя использовать c модификатором virtual:

**Ответ:** 1) override 2) private 3) abstract 4) static

1. Переопределенный базовый метод должен иметь тип :

**Ответ:**  virtual, abstract или override .

1. Для классов по умолчанию уровень доступа в C#:

**Ответ:**  internal .

1. При выполнении следующего блока кода

public class A  
{  
    public readonly int counter = 10;  
}  
  
class Program  
{  
    static void Main(string[] args)  
    {  
        A a = new A();  
        a.counter++;  
        Console.WriteLine(a.counter);  
    }  
}

на экран будет выведено:

**Ответ:**  возникнет ошибка времени компиляции.

1. При выполнении следующего блока кода

public class A  
{  
    int counter;  
    public int Counter  
    {  
        get { return counter; }  
        set { counter = (value < 10) + [ ] value : counter; }  
    }  
}  
  
class Program  
{  
    static void Main(string[] args)  
    {  
    A a = new A();  
    a.Counter = 10;  
    a.Counter++;  
    Console.WriteLine(a.Counter);  
}

на экран будет выведено:

**Ответ:** 1

1. При выполнении следующего блока кода

public class A  
{  
    internal static int Counter { get; set; }  
}  
  
class B: A { }  
  
class Program  
{  
    static void Main(string[] args)  
    {  
        B.Counter+=10;  
        Console.WriteLine(A.Counter);  
    }  
}

на экран будет выведено:

**Ответ:** 10

1. При выполнении следующего блока кода

public class A  
{  
    internal int Counter { get; set; }  
}  
  
class B: A { }  
  
class Program  
{  
    static void Main(string[] args)  
    {  
        A a = new A();  
        B b = new B();  
        b.Counter+=10;  
        Console.WriteLine(a.Counter);  
    }  
}

на экран будет выведено:

**Ответ:** 0

1. При выполнении следующего блока кода

public class A  
{  
    private static int Counter { get; set; }  
    public class B  
    {  
        public int Counter { get; set; }//15  
        public void Go(int counter)//10  
        {  
            A.Counter = counter;//0=10  
            Console.WriteLine(this.Counter + A.Counter);//15+10  
        }  
    }  
}  
  
class Program  
{  
    static void Main(string[] args)  
    {  
        A.B b = new A.B();  
        b.Counter = 15;  
        b.Go(10);  
        Console.ReadKey();  
    }  
}

на экран будет выведено:

**Ответ:**  25

1. При выполнении следующего блока кода, что будет выведено на экран:

 class A

        {

            public static int x = 0;

            public static string s="";

            public int y = 0;

            public A()

            {

                y++;

            }

            static A()

            {

                s = Math.Cos(x).ToString();

            }

        }

        static void Main(string[] args)

        {

            A a = new A();

            Console.WriteLine("{0} | {1}",a.y, A.s);

        }

**Ответ:**  1 | 1

1. При выполнении следующего блока кода, что будет выведено на экран:

 class A

        {

            public static int x = 0;

            public static string s="";

            public int y = 0;

            public A()

            {

                y++;

            }

            static A()

            {

                s = Math.Cos(x).ToString();

            }

        }

        static void Main(string[] args)

        {

            A a = new A();

            Console.WriteLine("{0} | {1}",a.y, a.s);

        }

**Ответ:** возникнет ошибка времени компиляции .

1. При выполнении следующего блока кода, что будет выведено на экран:

  class Program

    {

        static int M1(params int[] a)

        {

            int s = a[0];

            for (int i = 1; i < a.Length; i++) s -= a[i];

            return s;

        }

        static int M1(int x, int y)

        {

            return x + y;

        }

        static void Main(string[] args)

        {

            int x = 3;

            int y = 1;

            Console.WriteLine(M1(x,y));

         }

    }

**Ответ:**  4

1. При выполнении следующего блока кода, что будет выведено на экран:

   class Program

    {

        static int M1(out int x, int y)

        {

            x=y;

            return x + y;

        }

        static int M1(params int[] a)

        {

            int s = a[0];

            for (int i = 1; i < a.Length; i++) s -= a[i];

            return s;

        }

        static void Main(string[] args)

        {

            int x = 3;

            int y = 1;

            Console.WriteLine(M1(x,y));

        }

    }

**Ответ:**  2

**Наследование, интерфейсы, абстрактные классы**

1. В следующем коде  
     
   public class C : T1, T2 { };  
   public struct S : T3, T4 { };  
     
   T1, T2, T3, T4 могут быть:

Ответ: Т1 – класс, Т1 – интерфейс, Т2 – интерфейс, Т3 – интерфейс, Т4 – интерфейс

2. Что будет выведено на экран после выполнения следующего кода

    class Program

    {

      class A

        {

            public override string ToString()

            {

                return "class A";

            }

        }

      class B : A

        {

            public override string ToString()

            {

                return "class B";

            }

        }

        static void Main(string[] args)

        {

            object[] mas = { new object(), new A(), new B() };

            foreach (var m in mas)

            {

                Console.Write(" {0} ",m.ToString());

            }

        }

    }

Ответ: System.Object  class A  class B

3.Выберете правильные утверждения:

Ответ: Невозможно создать экземпляр интерфейса напрямую

Нельзя создать экземпляры абстрактного класса

Любой неабстрактный тип, реализующий интерфейс, должен реализовать все его члены

Абстрактные классы могут содержать абстрактные методы и методы доступа

4. Что будет выведено на экран после выполнения следующего кода

    class Program

    {

      class A

        {

            public override string ToString()

            {

                return "class A";

            }

        }

      class B : A

        {

            public override string ToString()

            {

                return "class B";

            }

        }

        static void Main(string[] args)

        {

            object[] mas = { new B(), new A(), new B() };

            foreach (B m in mas)

            {

                Console.Write(" {0} ",m.ToString());

            }

        }

    }

Ответ: Ошибка времени исполнения

5. Что будет выведено на экран после выполнения следующего кода  
  
public class A  
{  
    public void A1()  
    {  
        Console.WriteLine("This is A");  
    }  
}  
  
public class B : A  
{  
    public void A1()  
    {  
        Console.WriteLine("This is B");  
        base.A1();  
    }  
}  
class Program  
{  
    static void Main(string[] args)  
    {  
        B b = new B();  
        b.A1();  
          
    }  
}

Ответ: This is B  
    This is A

6. Какой модификатор служит для запрета наследования класса:

Ответ: sealed

7. Что будет выведено на экран после выполнения следующего кода  
  
public class BaseC  
{  
    public class NestedC  
    {  
        public int x = 200;  
        public int y;  
    }  
}  
  
public class DerivedC : BaseC  
{  
    new public class NestedC  
    {  
        public int x = 100;  
        public int y;  
        public int z;  
     }  
  
    static void Main()  
    {  
        NestedC c1 = new NestedC();  
        BaseC.NestedC c2 = new BaseC.NestedC();  
        Console.WriteLine(c1.x);  
        Console.WriteLine(c2.x);  
    }  
}

Ответ:100

200

8. Выберете правильные утверждения:

Ответ: Интерфейс похож на абстрактный базовый класс. Любой класс (или структура), реализующий интерфейс, должен реализовывать все его члены.

Невозможно создать экземпляр интерфейса напрямую. Его члены реализованы всеми классами или структурами, реализующими интерфейс.

Интерфейсы могут содержать события, индексаторы, методы и свойства.

Интерфейсы не содержат реализацию методов.

Класс или структура может реализовывать несколько интерфейсов.Класс может наследовать базовому классу и также реализовывать один или несколько интерфейсов.

9. Какой уровень доступа в интерфейсах по умолчанию:

Ответ: public

10. Что будет выведено на экран после выполнения следующего кода  
  
public class A  
{  
    public A()  
    {  
        Console.WriteLine("This is A");  
    }  
}  
  
public class B : A  
{  
    static B()  
    {  
        Console.WriteLine("This is B");  
    }  
}  
class Program  
{  
    static void Main(string[] args)  
    {  
        B b = new B();  
  
    }  
}

Ответ: This is B  
    This is A

18. Что будет выведено на экран после выполнения следующего кода  
  
public class A  
{  
    public A()  
    {  
        Console.WriteLine("This is A");  
    }  
}  
  
public class B : A  
{  
    public B()  
    {  
        Console.WriteLine("This is B");  
    }  
}  
class Program  
{  
    static void Main(string[] args)  
    {  
        B b = new B();  
  
    }  
}

Ответ: This is A  
    This is B

11. Что будет выведено на экран после выполнения следующего кода  
  
public abstract class A  
{  
    public void Go()  
    {  
        Console.WriteLine("This is A");  
    }  
}  
public class B: A  
{  
    public new void Go()  
    {  
        Console.WriteLine("This is B");  
    }  
}  
public class C : A  
{  
    public new void Go()  
    {  
        base.Go();  
        Console.WriteLine("This is C");  
    }  
}  
class Program  
{  
    static void Main(string[] args)  
    {  
        B b = new B();  
        C c = new C();  
        b.Go(); //b  
        c.Go(); //a c  
    }  
}

Ответ: This is B  
    This is A  
    This is C

12. Что будет выведено на экран после выполнения следующего кода  
  
public abstract class A  
{  
    public void Go()  
    {  
        Console.WriteLine("This is A");  
    }  
}  
public class B: A  
{  
    public new void Go()  
    {  
        Console.WriteLine("This is B");  
    }  
}  
public class C : A{ }  
class Program  
{  
    static void Main(string[] args)  
    {  
        B b = new B();  
        C c = new C();  
        b.Go();//b  
        c.Go();//a  
    }  
}

Ответ: This is B  
    This is A

24. Что будет выведено на экран после выполнения следующего кода  
  
public class A  
{  
    public virtual void M()  
    {  
        Console.WriteLine("This is A");  
    }  
}  
  
public class B : A  
{  
    public new void M()  
    {  
        Console.WriteLine("This is B");  
    }  
}  
  
class Program  
{  
    static void Main(string[] args)  
    {  
        B b = new B();  
        A a = b;  
        a.M();  
    }  
}

Ответ: This is A

13. Что будет выведено на экран после выполнения следующего кода  
  
     public interface IFirst  
     {  
         void Go();  
     }  
       
     public interface ISecond  
     {  
         void Go();  
     }  
       
     public interface IGeneral: IFirst, ISecond { }  
     public class A: IGeneral  
     {  
         void IFirst.Go()  
         {  
             Console.WriteLine("This is IFirst");  
         }  
       
         void ISecond.Go()  
         {  
             Console.WriteLine("This is ISecond");  
         }  
     }  
       
     class Program  
     {  
         static void Main(string[] args)  
         {  
             IGeneral a = new A();  
             a.Go();  
         }  
     }

Ответ: Возникнет ошибка времени компиляции

14. Что будет выведено на экран после выполнения следующего кода  
  
     public interface IFirst  
     {  
         void Go();  
     }  
       
     public interface ISecond  
     {  
         void Go();  
     }  
       
     public interface IGeneral: IFirst, ISecond { }  
     public class A: IGeneral  
     {  
         void IFirst.Go()  
         {  
             Console.WriteLine("This is IFirst");  
         }  
       
         void ISecond.Go()  
         {  
             Console.WriteLine("This is ISecond");  
         }  
     }  
       
     class Program  
     {  
         static void Main(string[] args)  
         {  
             IFirst a = new A();  
             a.Go();  
         }  
     }

Ответ: This is IFirst

15. Какое (какие) ключевые слова пропущены в тексте создания абстрактного класса, чтобы можно было корректно воспользоваться полиморфизмом?

    public abstract class Figure  
    {  
        public int X { get; set; }  
        public int Y { get; set; }  
        ??? double GetPerimeter();  
    }

Ответ: public abstract

16. Что будет на экране в результате запуска приложения?

   abstract class A  
    {  
        public abstract void M();  
    }  
  
    class B : A  
    {  
        public sealed override void M()  
        {  
            Console.WriteLine("B");  
        }  
    }  
  
    class C : B  
    {  
        public override void M()  
        {  
            Console.WriteLine("C");  
        }  
    }  
  
    class Program  
    {  
        static void Main(string[] args)  
        {  
            B b = new B();  
            C c = new C();  
  
            b.M();  
            c.M();  
        }

    }

Ответ: Ошибка времени компиляции

17. Что будет на экране в результате запуска приложения?

   abstract class A  
    {  
        public abstract void M();  
    }  
  
    class B : A  
    {  
        public sealed override void M()  
        {  
            Console.WriteLine("B");  
        }  
    }  
  
    class C : B  
    {  
        public new void M()  
        {  
            Console.WriteLine("C");  
        }  
    }  
  
    class Program  
    {  
        static void Main(string[] args)  
        {  
            B b = new B();  
            C c = new C();  
  
            b.M();  
            c.M();  
        }

    }

Ответ: В

С

19. Что будет выведено на экран после выполнения следующего кода  
  
class A  
{  
    public void Test(double n)  
    {  
        Console.WriteLine("Test A");  
    }  
}  
class B : A  
{  
    public void Test(int n)  
    {  
        Console.WriteLine("Test B");  
    }  
}  
class Program  
{  
    static void Main(string[] args)  
    {  
        B b = new B();  
        b.Test(5.0);   
    }  
}

Ответ: Test A

20. Выберете правильные утверждения, описывающие следующий код  
  
interface I1  
{  
    void Do();  
}  
  
interface I2  
{  
    void Do();  
}  
  
class C : I1, I2  
{  
    void I1.Do(){. . .}  
    void I2.Do(){. . .}  
}

Ответ: Данный код содержит явную реализацию интерфейсов

Данный код позволяет выполнить разные реализации метода D+

21. Что будет выведено на экран после выполнения следующего кода  
class A  
{  
    public void Test(int n)  
    {  
        Console.WriteLine("Test A");  
    }  
}  
class B : A  
{  
    public void Test(double n)  
    {  
        Console.WriteLine("Test B");  
    }  
}  
class Program  
{  
    static void Main(string[] args)  
    {  
        B b = new B();  
        b.Test(5);  
    }  
}

Ответ: Test B

22. Что будет выведено на экран после выполнения следующего кода  
  
struct Book  
{  
    public string author;  
    public string title;  
    public int copyright;  
  
    public Book()  
    {  
        author = "aaa";  
        title = "bbb";  
        copyright = 50;  
        Console.WriteLine("This is a book.");  
    }  
}  
class Program  
{  
    static void Main(string[] args)  
    {  
        Book bk = new Book();  
        Console.WriteLine("This is not a book");  
    }  
}

Ответ: Возникнет ошибка времени компиляции

23. Что будет выведено на экран после выполнения следующего кода  
  
public interface I  
{  
    void Go();  
}  
public class A : I  
{  
    public void Go()  
    {  
        Console.WriteLine("A.Go()");  
    }  
}  
class B : A  
{  
}  
class C : B, I  
{  
    public new void Go()  
    {  
        Console.WriteLine("C.Go()");  
    }  
}  
class Program  
{  
    static void Main(string[] args)  
    {  
        B b1 = new B();  
        C c1 = new C();  
        B b2 = c1;  
        b1.Go();  
        c1.Go();  
        b2.Go();  
        ((I)b2).Go();  
    }  
}

Ответ: A.Go();  
    C.Go();  
    A.Go();  
    C.Go();

25. Что будет выведено на экран после выполнения следующего кода  
  
public class Car  
{  
    public Car()  
    {  
        System.Console.WriteLine("The Car constructor invoked.");  
    }  
}  
public class Bus : Car  
{  
    static Bus()  
    {  
        System.Console.WriteLine("The Bus constructor invoked.");  
    }  
  
    public static void Drive()  
    {  
        System.Console.WriteLine("The Drive method invoked.");  
    }  
}  
class Program  
{  
    static void Main(string[] args)  
    {  
        Bus.Drive();  
    }  
}

Ответ: The Bus constructor invoked.   
    The Drive method invoked.

Обработка исключений

1. При выполнении следующего кода  
  
unchecked  
{  
    int i = -1;  
    Console.WriteLine((byte)i);  
}  
  
на экран будет выведено

Ответ: 255.

2. При выполнении следующего кода  
  
unchecked  
{  
    int i = 256;  
    byte b = (byte)i;  
    Console.WriteLine((int) b + " ");  
}  
  
на экран будет выведено

Ответ: 0.

3. При выполнении следующего кода  
  
...  
public const int END = int.MaxValue;  
public const int START = END - 100;  
...  
private static void Main(string[] args)  
{  
    int count = 0;  
    unchecked  
    {  
        for (int i = START; i <= END; i++)  
        {  
            count++;  
        }  
    }  
    Console.WriteLine(count);  
}

Ответ: Не выполнится ничего из перечисленного.

4. При выполнении следующего кода  
  
unchecked  
{  
    int i = int.MaxValue;  
    i++;  
    Console.WriteLine(i);  
}  
  
на экран будет выведено

Ответ: -2147483648.

5. При выполнении следующего кода  
  
checked  
{  
    Console.WriteLine(8 << 32);  
    Console.WriteLine(8 << 33);  
}  
  
на экран будет выведено

Ответ: 8  
     16

6. При выполнении следующего кода  
  
    int i = 0;  
    unchecked  
    {  
        while (-1 << i != 0)  
        {  
            i++;  
        }  
    }  
    Console.WriteLine(i);

Ответ: Ничего из перечисленного.

7. При выполнении следующего кода  
  
checked  
{  
    double i = 0.0/0.0;  
  
    if (i != i)  
        Console.WriteLine("not equal");  
    else  
        Console.WriteLine("equal");  
}

Ответ: На экран будет выведено not equal.

8. При выполнении следующего кода  
int i = int.MinValue;  
unchecked  
{  
    if (i != 0 && i == -i)  
    {  
        Console.WriteLine("equal");  
    }  
    else  
    {  
        Console.WriteLine("not equal");  
    }  
}

Ответ: На экран будет выведено equal.

9. При выполнении следующего кода  
  
checked  
{  
    const int START = 2000000000;  
  
    float f = START + 50;  
  
    if (f == START)  
        Console.WriteLine("true");  
    else  
        Console.WriteLine("false");  
}

Ответ: На экран будет выведено true.

10. При выполнении следующего кода  
  
private static int i = -(2147483648);  
  
...  
checked  
{  
    Console.WriteLine(-i);  
}

Ответ: Возникнет ошибка времени компиляции.

11. При выполнении следующего кода  
  
unchecked  
{  
    short s = short.MinValue;  
    char c = (char)s;  
    Console.WriteLine((int)c);  
}  
  
на экран будет выведено

Ответ: 32768.

12. При выполнении следующего кода  
  
try  
{  
    Console.WriteLine("Hello ");  
    return;  
}  
finally  
{  
    Console.WriteLine("Goodbye ");  
}  
Console.WriteLine("world!");  
  
на экран будет выведено

Ответ: Hello Goodbye.

13. При выполнении следующего кода  
  
try  
{  
    Console.Write("Hello ");  
    Thread.CurrentThread.Abort();  
}  
finally  
{  
    Console.Write("Goodbye ");  
}  
Console.Write("world!");  
  
на экран будет выведено

Ответ: Hello Goodbye.

14. При выполнении следующего кода  
  
try  
{  
    Console.WriteLine("Hello ");  
    System.Environment.Exit(0);  
}  
finally  
{  
    Console.WriteLine("Goodbye ");  
}  
Console.WriteLine("world!");  
  
на экран будет выведено

Ответ: Hello.

15. Является ли следующая функция компилируемой?  
  
public int Foo(bool b)  
{  
    throw new Exception("Some Exception");  
}

Ответ: Да.

16. При выполнении следующего кода  
  
public static int[] getArray() { return null; }  
  
private static void Main(string[] args)  
{  
    int index = 1;  
    try  
    {  
        getArray()[index = 2]++;  
    }  
    catch (Exception)  
    {  
        //empty catch  
    }   
    Console.WriteLine("index = {0}", index);  
}  
  
на экран будет выведено

Ответ: index = 2.

17. При выполнении следующего кода   
  
private static void Main(string[] args)  
{  
    try  
    {  
        int[] a = null;  
        int i = a[m1(true)];  
    }  
    catch (Exception e)  
    {  
        Console.WriteLine(e.GetType());  
    }  
}  
  
public static int m1(bool b)  
{  
    if (b)  
        throw new Exception("Some Exception");  
    return 0;  
}  
  
на экран будет выведено

Ответ: System.Exception.

18. Отметьте все строки, которые могут быть выведены на экран после выполнения следующего кода:  
  
using System;  
using System.Text;  
  
internal class Test  
{  
    private static void Main(string[] args)  
    {  
        StringBuilder word = null;  
        int i=0;  
        if (args.Length>0)  
         try  
         {  
             i = Int32.Parse(args[0]);  
         }  
         catch  
         {  
             i = 1;   
         }  
              
        switch (i)  
        {  
            case 1:  
                word = new StringBuilder('P');  
                break;  
  
            case 2:  
                word = new StringBuilder('G');  
                break;  
  
            default:  
                word = new StringBuilder('M');  
                break;  
        }  
        word.Append('a');  
        word.Append('i');  
        word.Append('n');  
        Console.WriteLine(word);  
    }  
}

Ответ: ain.

19. Отметьте все строки, которые могут быть выведены на экран после выполнения следующего кода:  
  
using System;  
using System.Text;  
  
internal class Test  
{  
    private static void Main(string[] args)  
    {  
        StringBuilder word = null;  
        int i = 0;  
        if (args.Length > 0)  
        {  
            try  
            {  
                i = Int32.Parse(args[0]);  
            }  
            catch  
            {  
                i = 1;  
            }  
        }  
        switch (i)  
        {  
            case 1:  
                word = new StringBuilder("P");  
                break;  
  
            case 2:  
                word = new StringBuilder("G");  
                break;  
  
            default:  
                word = new StringBuilder("M");  
                break;  
        }  
        word.Append('a');  
        word.Append('i');  
        word.Append('n');  
        Console.WriteLine(word);  
    }  
Ответ: 1) Main 2) Pain 3) Gain.

20. Исключения имеют типы, в конечном счете являющиеся производными от:

Ответ:System.Exception.

21. При попытке передать в метод аргумент неверного формата генерируется исключение:

Ответ: FormatException.

22. При несовместимости типа сохраняемого значения с типом массива генерируется исключение:

Ответ: ArrayTypeMismatchException.

23. При ошибке преобразования типа генерируется исключение:

Ответ: InvalidCastException.

24. При арифметическом переполнении генерируется исключение:

Ответ: ОverFlowException.

25. При выполнении следующего кода  
  
double result = 0.0;  
string counter = "5,5";  
try  
{  
    Console.WriteLine(Int32.Parse(counter));  
}  
catch(FormatException)  
{  
    result += Double.Parse(counter) \* 2;  
}  
finally  
{  
    Console.WriteLine(result);  
}  
  
на экран будет выведено

Ответ: 11.

26. При выполнении следующего кода  
  
public class A  
{  
    private int counter;  
    public int Counter {  
        get { return counter; }  
        set  
        {  
            if (value > 10)  
            {  
                throw new Exception("Error in property");  
            }  
            else  
                counter = value;  
        }  
    }  
}  
  
class Program  
{  
    static void Main(string[] args)  
    {  
        A a = new A();  
        try  
        {  
            a.Counter = 15;  
        }  
        catch(Exception) when (a.Counter < 10)  
        {  
            Console.WriteLine("Error with when");  
        }  
        catch (Exception ex)  
        {  
            Console.WriteLine(ex.Message);  
        }  
    }  
}  
  
на экран будет выведено

Ответ: Error with when.

27. При выполнении следующего кода  
  
public class A  
{  
    private int counter;  
    public int Counter {  
        get { return counter; }  
        set  
        {  
            if (value > 10)  
            {  
                throw new Exception("Error in property");  
            }  
            else  
                counter = value;  
        }  
    }  
}  
  
class Program  
{  
    static void Main(string[] args)  
    {  
        A a = new A();  
        try  
        {  
            a.Counter = 15;  
        }  
        catch(Exception) when (a.Counter > 10)  
        {  
            Console.WriteLine("Error with when");  
        }  
        catch (Exception ex)  
        {  
            Console.WriteLine(ex.Message);  
        }  
    }  
}  
  
на экран будет выведено

Ответ: Error in property.

28. Верно ли написана конструкция try catch?  
  
int a = 9;  
try  
{  
    a++;  
}  
catch(Exception) when (a < 10)  
{  
    Console.WriteLine("Error with when");  
}

Ответ: Да.

29. При выполнении следующего кода  
  
int a = 9;  
try  
{  
    a++;  
}  
catch(Exception) when (a < 10)  
{  
    Console.WriteLine("Error with when");  
}  
finally  
{  
    Console.WriteLine(a);  
}  
  
на экран будет выведено

Ответ:10.

Управление ресурсами в .NET. Сборка мусора

1. Какой метод вызывается при реализации шаблона Dispose, чтобы предотвратить выполнение метода Finalize?

Ответ: GC.SuppressFinalize.

1. Отметьте верные утверждения, характеризующие деструктор:

Ответ: 1) Деструктор не имеет параметров

2) В структурах определение деструкторов невозможно.

1. К поколению 0 сборщика мусора относятся:

Ответ: только что размещенные в куче, которые еще ни разу не помечались как подлежащие удалению в процессе сборки мусора.

1. Для оптимизации производительности сборщика мусора на какое количество поколений делиться управляемая куча?

Ответ: 3

1. При каких условиях возникает сборка мусора?

Ответ: 1) Недостаточно физической памяти в системе

2) Память, используемая объектами, выделенными в управляемой куче, превышает допустимый порог

3) Вызывается метод GC.Collect.

1. После инициализации средой CLR сборщик мусора выделяет сегмент памяти для хранения объектов и управления ими. Эта память называется?

Ответ: управляемой кучей.

1. К эфемерным относятся поколения:

Ответ: 1 и 0.

1. Какое поколение кучи содержит долгоживущие объекты?

Ответ: 2.

1. Деструктор объявляется следующим образом?

Ответ: ~MyClass(){…}

1. В какой очередности будут вызваны деструкторы в результате выполнения следующего кода?

class A{  
       ~A()  
        {   
             System.Diagnostics.Trace.WriteLine("Destructor of class A is called.");     
        }  
}  
class B : A  
{      
       ~B()  
       {  
             System.Diagnostics.Trace.WriteLine("Destructor of class B is called.");      
       }  
}  
class C : B  
{  
      ~C()  
      {  
             System.Diagnostics.Trace.WriteLine("Destructor of class C is called.");  
       }  
}  
class Program  
{  
       static void Main(string[] args)  
       {  
             C t = new C();  
       }  
}

Ответ: Destructor of class C is called.    Destructor of class B is called.    Destructor of class A is called.

1. В какой очередности будет выполнен следующий код?

class A  
{  
    public A()  
    {  
        System.Diagnostics.Trace.WriteLine("Class A");  
    }  
    ~A()  
    {  
        System.Diagnostics.Trace.WriteLine("Destructor of class A is called.");  
    }  
}  
  
class B : A  
{  
    public B()  
    {  
        System.Diagnostics.Trace.WriteLine("Class B");  
    }  
  
    ~B()  
    {  
        System.Diagnostics.Trace.WriteLine("Destructor of class B is called.");  
    }  
}  
  
class C : B  
{  
    public C()  
    {  
        System.Diagnostics.Trace.WriteLine("Class C");  
    }  
    ~C()  
    {  
        System.Diagnostics.Trace.WriteLine("Destructor of class C is called.");  
    }  
}  
  
class Program  
{  
    static void Main(string[] args)  
    {  
        C t = new C();  
    }  
}

Ответ: Class A  
     Class B  
    Class C  
     Destructor of class C is called.  
     Destructor of class B is called.  
     Destructor of class A is called.

1. Как определить максимальное количество поколений кучи, которые в настоящее время поддерживает система?

Ответ: GC.MaxGeneration.

1. Как узнать текущее поколение экземпляра класса?

Ответ: GC.GetGeneration(myClass).

Работа с файловой системой

1. Какую информацию будет хранить текстовый файл TextFile.txt в результате выполнения следующего кода  
     
   string filePath = "TextFile.txt";  
     
   string[] fileLines = { "One ", "Two", "Three " };  
   File.AppendAllLines(filePath, fileLines);  
     
   при условии, что до выполнения кода такой файл не существовал на диске

Ответ: One  
    Two  
     Three

1. Выберете правильные утверждения. Следующий конструктор класса FileStream  
     
   FileStream fs = new FileStream(name, FileMode.Create, FileAccess.Read);

Ответ: Указывает, что операционная система должна создать новый файл с именем name в режиме доступа только для чтения. Если файл уже существует, он будет переписан.

1. Выберете правильные утверждения. Следующий конструктор класса FileStream  
     
   FileStream fs = new FileStream(name, FileMode.OpenOrCreate, FileAccess.Read);

Ответ: Указывает, что операционная система должна создать новый файл с именем name в режиме доступа только для чтения. Если файл уже существует, будет сгенерировано исключение IOException.

4. Выберете правильные утверждения. Следующий конструктор класса FileStream  
  
FileStream fs = new FileStream(name, FileMode.Append, FileAccess.Write);

Ответ: Указывает, что операционная система должна открыть файл с именем name в режиме доступа только для записи и добавить данные в конец файла. Если файл не существует, то он создается.

1. Перечисление FileAccess определяет следующие константы:

Ответ: Read  
     Write  
     ReadWrite

5. Что произойдет в результате выполнения следующего кода (папка d:\\ds существует и она не пуста)?  
  
static void Main()  
{  
    Directory.Delete("d:\\ds", true);  
}

Ответ: удалится папка ds и все её содержимое.

6. Что произойдет в результате выполнения следующего кода (папка d:\\ds существует и она не пуста)?  
  
static void Main()  
{  
    Directory.Delete("d:\\ds");  
}

Ответ: возникнет ошибка выполнения.

7. Класс Directory из пространства имен

Ответ: System.IO.

8. Для использования кодировок подключается пространство имен.

Ответ: System.Text.

9. Что будет храниться в файле in.txt после выполнения следующего кода  
  
static void Main()  
{  
    FileStream file1 = new FileStream("d:\\in.txt", FileMode.Create);  
    StreamWriter writer = new StreamWriter(file1);  
    writer.Write("текст");  
}

Ответ: файл будет пустым.

10. Можно ли узнать время последнего изменения файла?

Ответ: ДА.

11. При помощи каких методов класса File можно узнать время последнего изменения файла?

Ответ: 1) File.GetLastWriteTime();

2) File.GetLastWriteTimeUTC();

12. При помощи каких методов класса File можно узнать время создания файла?

Ответ: 1) File.GetCreationTime();

2) File.GetCreationTimeUTC();

13. Что будет храниться в файле in.txt после выполнения следующего кода  
  
static void Main()  
{  
    using (StreamWriter file = new StreamWriter("in.txt"))  
    {  
        file.WriteLine("Hello world");  
    }  
}

Ответ: Hello world.

14. При помощи каких методов класса Directory можно получить все файлы указанной папки

Ответ: 1) Directory.EnumerateFiles("e:\\");

2) Directory.GetFiles("e:\\");

1. Как называется член класса задача которого хранить данные?

Ответ: Поле.

1. Какое из ниже перечисленных выражений является литералом?

Ответ: ‘\x78’

1. Как называется модификатор, указывающий на то, что экзепляр помеченного класса не может быть создан?

Ответ: abstract

1. Чему равна строка s, если следующий код возвращает значение True:

new Regex(@”[012]?\d{1}:[0,5][0-9]”).IsMatch(s)

Ответ: 21:05

1. Чему равна строка s, если слеующий код возвращает True:

new Regex(@”[012]?\d{1}:[0-5][0,9]”).IsMatch(s)

Ответ: 00:30 15

1. Какое из ниже перечисленных выражений соответствует выводу на консоль символьного литерала?

Ответ: Console.WriteLine(‘A’)

1. Какой из следующих ответов описывает правильный способ создания массива типа int с тремя элементами?

Ответ: int[] myArray = new int[3]

1. Какое контекстное ключевое слово означает, что метод, оператор или метод доступа get (в котором оно присутствует) является итератором?

Ответ: yield

1. Что будет содержать переменная val, после выполнения следующего кода:

string s = “5 +3 – 2 + 1 - 7”;

string[] n =s.Split(‘+’, ‘-‘, ‘\*’, ‘/’);

int val = 0;

foreach(string v in n) val += Int32.Parse(v);

Ответ: 18

1. Переменная которая будет доступна в одной сборке и в классах наследниках

Ответ: protected internal

1. Что будет выведено на экран при выполнении следующего кода?

int a = 7;

try

{

a++;

}

catch(Exception) when (a < 10)

{

Console.WriteLine(“Error with when”);

}

Ответ: Ничего не будет выведено

1. Как называется элемент языка C#, который описывает объект хранящийся в стэке потока?

Ответ: Структура.

1. Как называетя тип данных доступный под сокращенным названием long?

Ответ: System.Int64

1. Какое из ниже перечисленных выражений является литералом?

Ответ: true

1. Какая строка кода использует логическое отрицание?

Ответ: Console.WriteLine(!value);

1. Как называется модификатор, указывающий на то, что доступ к типу или члену возможен только из кода в том же объекте class или struct?

Ответ: private

1. Класс позволяющий использовать регулярные выражения в .NET?

Ответ: Regex

1. Необходимо найти слеющий индекс любого символа -, = или \_ типа char. Какой метод следует использовать?

Ответ: IndexOfAny()

1. Как называется установщик, необходимый для создания проектов, написания кода, вызова методов в библиотеках, компиляции кода, сборки, отладки и запуска приложений?

Ответ: Среда выполнения .NET

1. Какое из приведенных утверждений верно?

Ответ: Оператор new служит для создания экземпляра ссылочного типа и возвращения его адреса в переменную.

1. Какие из следующих строк кода отражают допустимое использование условного оператора?

Ответ: int value = amount >= 10 ? 10 : 20

1. Какое из ниже перечисленных выражений соответствует выводу на консоль логического литерала?

Ответ: Console.WriteLine(true)

1. Назовите оператор, который отвечает за возвращение значение из метода или свойства

Ответ: return

1. Какой модификатор в заголовке метода позволяет вызвать этот метод через имя класса, в котором этот метод реализован?

Ответ: static

1. Как называетя тип данных доступный под сокращенным названием float?

Ответ: System.Single

1. Как называетя тип данных доступный под сокращенным названием int?

Ответ: System.Int32

1. Как называетя тип данных доступный под сокращенным названием double?

Ответ: System.Double

1. Как называетя тип данных доступный под сокращенным названием short?

Ответ: System.Int16

## ADO.NET

1. Какие классы входят в состав поставщика ADO.NET:

DataSet

+ DataAdapter DataRow

+ Connection

+ DataReader DataRelation

+ Command

1. Каким методом можно выполнить вызов хранимой процедуры:

+ ExecuteNonQuery()

+ ExecuteScalar()

+ ExecuteReader() TableDirect StoretProcedure

1. Тип команды для оператора Insert:

+ Text TableDirect StoretProcedure SqlTransaction

ExecuteProcedure

1. Метод DataReader.Read():

Устанавливает курсор чтения на первую запись из набора

Устанавливает курсор чтения в начало считанных данных, возвращает false если данных для чтения нет

+ Устанавливает курсор чтения на следующую запись из набора, возвращает false если данных для чтения нет

Возвращает количество строк Возвращает коллекцию строк

1. Какой способ защиты от SQL-инъекций существует в ADO.NET:

Замена «опасных» символов их безопасными аналогами при помощи HttpServerUtility.HtmlEndoce()

Встроенного механизма нет, необходимо проверять введенные пользователем данные вручную

+ Использование параметризированных команд позволяет решить эту проблему Шифрование символов

Отложенное выполнение кода

1. Какой класс предоставляет набор команд SQL и подключение базы данных, которые используются для заполнения DataSet и обновления источника данных?

DataColumn

+ SqlDataAdapter DataRow SqlDataReader SqlConnection SqlParameter

1. Что будет выведено на экран в результате выполнения следующего блока кода

. . .

using (var connection = new SqlConnection(\_connectionString))

{

using (var cmd = new SqlCommand("SELECT GETDATE()", connection))

{

Console.WriteLine( cmd.ExecuteScalar());

}

}

. . .

+ Ошибка времени выполнения Ошибка компиляции

Результат выполнения SQL функции GETDATE() Пустая строка

1. Что представляет собой набор бинарных библиотек?

Соединение

+ Драйвер

+ Провайдер Объединение Запрос

1. В данном коде осуществляется транзакция к некоторой базе данных. Вместо пронумерованных комментариев вставьте команды управления СУБД в таком порядке, чтобы код компилировался и успешно выполнялся:

. . .

SqlConnection sqlCn = new SqlConnection(); SqlTransaction tx = null;

try

{

//1

#region Здесь указаны инструкции по формированию

...

#endregion SQL-запросов и выполнению cоотв. им команд

//2

}

catch (Exception)

{

//3

}

. . .

tx.Rollback(); tx = sqlCn.BeginTransaction(); tx.Commit();

+ tx = sqlCn.BeginTransaction(); tx.Commit(); tx.Rollback(); tx = sqlCn.BeginTransaction(); tx.Rollback(); tx.Commit(); tx.Commit(); tx = sqlCn.BeginTransaction(); tx.Rollback(); tx.Commit(); tx.Rollback(); tx.Commit();

1. Какие две модели для работы с данными существуют в ADO.NET?

+ Присоединенная Смешанная Модель OLE DB

+ Отсоединенная Модель SQL Модель LINQ

1. Какой метод следует применить для выполнения следующего запроса?

. . .

command.CommandText = String.Format(@"INSERT INTO tbPhones (Name, Surname, EMail, Phone) VALUES ('{0}', '{1}', '{2}', '{3}')", Name, Surname, EMail, Phone);

. . .

command.ExecuteReader(); command.ExecuteQuery();

+ command.ExecuteNonQuery(); command.ExecuteScalar(); command.Execute();

1. Что произойдет в результате выполнения следующего блока кода string connectionString = @"Data “

+”Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=|DataDirectory|\AppData\TestDB.mdf;"

+

"Integrated Security=True";

string param1 = "user";

string param2 = "');delete from [User];--"; int id = 4;

string queryString ="insert into [Role](Id,Name,Definition) "

+"values("+id+", '"+param1+"', '"+param2+"');";

using (SqlConnection connection = new SqlConnection(connectionString))

{

SqlCommand command = new SqlCommand(queryString, connection); connection.Open();

int result = command.ExecuteNonQuery();

}

Произойдет добавление новой записи в таблицу Role Возникнет ошибка времени компиляции

Возникнет ошибка времени выполнения

+ Произойдет добавление новой записи в таблицу Role, а затем будут удалены все записи из таблицы User

Произойдет добавление новой записи в таблицу User, а затем будут удалены все записи из таблицы Role

1. В приведенном коде есть SQL-инъекции? string connectionString = @"Data “

+”Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=|DataDirectory|\AppData\TestDB.mdf;"

+

"Integrated Security=True";

string param1 = "user";

string param2 = "');delete from [User];--"; int id = 4;

string queryString = "insert into [Role](Id,Name,Definition) "

+"values("+id+", '"+param1+"', '"+param2+"');";

using (SqlConnection connection = new SqlConnection(connectionString))

{

SqlCommand command = new SqlCommand(queryString, connection); connection.Open();

int result = command.ExecuteNonQuery();

}

+ Да Нет

1. Что произойдет после выполнения следующего кода?

+”Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=|DataDirectory|\AppData\TestDB.mdf;"

+

"Integrated Security=True";

string param1 = "user";

string param2 = "');delete from [User];--"; int id = 4;

string queryString = "insert into [Role](Id,Name,Definition) "

+"values("+id+", '"+param1+"', '"+param2+"');";

using (SqlConnection connection = new SqlConnection(connectionString))

{

SqlCommand command = new SqlCommand(queryString, connection); connection.Open();

int result = command.ExecuteNonQuery();

}

+Будут удалены записи из таблицы User Будет добавлена запись в таблицу Role Ничего не будет выполнено

Ничего не будет выполнено

Возникнет ошибка времени выполнения

1. Что произойдет в результате выполнения следующего блока кода string connectionString = @"Data “

+”Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=|DataDirectory|\AppData\TestDB.mdf;"

+"Integrated Security=True";

string param1 = "user";

string param2 = "');delete from [Role];--"; int id = 4;

string queryString =

"insert into [Role](Id,Name,Definition) "

+"values(@id, @param1, @param2);";

using (SqlConnection connection = new SqlConnection(connectionString))

{

SqlCommand command = new SqlCommand(queryString, connection); command.Parameters.AddWithValue("@id", id); command.Parameters.AddWithValue("@param1", param1); command.Parameters.AddWithValue("@param2", param2); connection.Open();

int on1 = command.ExecuteNonQuery();

}

+ Произойдет добавление новой записи в таблицу Role Возникнет ошибка времени компиляции

Возникнет ошибка времени выполнения

Произойдет добавление новой записи в таблицу Role, а затем будут удалены все записи из таблицы User

Произойдет добавление новой записи в таблицу User из таблицы Role

1. Что произойдет в результате выполнения следующего блока кода string connectionString =@"Data “

+”Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=|DataDirectory|\AppData\TestDB.mdf;"

+ "Integrated Security=True"; string param = "mu";

string queryString = "select \* from [User] where Login like @param;";

using (SqlConnection connection = new SqlConnection(connectionString))

{

SqlCommand command = new SqlCommand(queryString, connection); command.Parameters.AddWithValue("@param", param + "%");

try

{

connection.Open();

SqlDataReader reader = command.ExecuteReader(); while (reader.Read())

{

Console.WriteLine("{0, 23}{1, 23}{2, 23}{3, 20}{4, 23}",

reader[0], reader[1], reader[2], reader[3], reader[4]);

}

reader.Close();

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

}

+ Произойдет выборка всех записей логины, которых начинаются с «mu», из таблицы User и ее результат будет выведен на консоль

Возникнет ошибка времени компиляции Возникнет ошибка времени выполнения

Произойдет добавление новой записи в таблицу User

Произойдет удаление записей с логинами, которые начинаются с «mu», в таблицу User Возникнет ошибка при работе с БД

1. Что произойдет в результате выполнения следующего блока кода string connectionString =@"Data “

+”Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=|DataDirectory|\AppData\TestDB.mdf;"

+ "Integrated Security=True"; string param = "mu";

string queryString = "delete from [User] where Login like @param;";

using (SqlConnection connection = new SqlConnection(connectionString))

{

SqlCommand command = new SqlCommand(queryString, connection); command.Parameters.AddWithValue("@param", param + "%");

try

{

connection.Open();

int result = command.ExecuteNonQuery ();

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

}

Произойдет выборка всех записей логины, которых начинаются с «mu», из таблицы User и ее результат будет выведен на консоль

Возникнет ошибка времени компиляции Возникнет ошибка времени выполнения

Произойдет добавление новой записи в таблицу User

+ Произойдет удаление записей с логинами, которые начинаются с «mu», в таблицу User Возникнет ошибка при работе с БД

## XML парсинг

1. Документ XDocument может включать в себя следующие элементы:

+ Один объект XDeclaration.

+ Один объект XElement.

+ Любое количество объектов XProcessingInstruction.

+ Любое количество объектов XComment.

+ Один тип документа XDocumentType для DTD.

1. Выберите корректный вариант написания элемента с именем element, который содержит текст "value"

+ <element>value</element>

1. Какие из следующих являются языками определения схемы XML документа?

+ XML Schema XSLT

SGML

+ RELAX NG

+ DTD

1. Какие из приведенных утверждений о DTD являются верными?

Нельзя проверить корректность XML документа без DTD

+ DTD может размещаться в отдельном файле

DTD может быть помещено в начало файла XML документа DTD является наследником DDT

Ни один из вариантов не является правильным

1. Что в XML-документе означают символы &amp;lt; и &amp;gt; ?

Двойные и одинарные кавычки " ' Круглые скобки ()

Начало и окончание комментария Фигурные скобки {}

+ Угловые скобки < >

1. Является ли приведенный XML-документ корректным:

<?xml version="1.0" encoding="utf-8"?>

<mail>

<from>admin@quizful.net/from>

<[to>use](mailto:user@quizful.net)r@[quizful.ne](mailto:user@quizful.net)t</to>

<subject>xml test passed successfuly</subject>

<text>xml test passed successfuly</text>

</mail>

Нет

+ Да

269. Есть ли ошибки в приведенном XML-документе:

<?xml version="1.0" encoding="utf-8"?>

<mail>

<from>admin@quizful.net/from>

<[to>use](mailto:user@quizful.net)r@[quizful.ne](mailto:user@quizful.net)t</to>

<subject>xml test passed successfuly</subject>

<text>xml test passed successfuly</text>

</mail>

Нет

Ошибка в строке с тегом to Ошибка в строке с тегом subject Ошибка в строке с тегом text

+ Ошибка в строке с тегом from

1. Какие варианты подходят для обозначения элемента без содержимого?

+ <name/>

+ <name></name>

1. Какие из перечисленных языков являются прикладными расширениями XML?

HTML

+ XHTML SGML CSS PHP

1. XML является подмножеством HTML

GML

+ SGML XSL

Ни один из вариантов не является верным

1. Что означает сокращение DTD?

Dark Tranquil Daemon Document Type Delivery

+ Document Type Definition Dummy Translated Document Direct Type Definition

1. Перечислите виды валидации xml документов?

+ XSD

+ DTD DDT DDP

Ни один из вариантов не является верным

1. Какие классы используются для работы с xml-документами

+ XDocument

+ XmlDocument Stream StreamReader

Ни один из вариантов не является верным

1. Что произойдет в результате выполнения следующего блока кода

XmlNode firstName = document.CreateElement("FirstName"); XmlAttribute attrName = document.CreateAttribute("Name"); attrName.Value = "Popov"; firstName.Attributes.Append(attrName);

firstName.InnerText = "Popov";

+ Будет создан элемент &lt;FirstName Name="Popov"&gt;Popov&lt;/FirstName&gt; Будет создан элемент &lt;FirstName&gt;Popov&lt;/FirstName&gt;

Будет удален элемент &lt;FirstName&gt;Popov&lt;/FirstName&gt;

Будет удален элемент &lt;FirstName Name="Popov"&gt;Popov&lt;/FirstName&gt; Будет изменен элемент &lt;FirstName Name="Popov"&gt;Popov&lt;/FirstName&gt;

1. Что произойдет в результате выполнения следующего блока кода

XmlNode phoneNumber = document.CreateElement("PhoneNumber"); XmlAttribute attrNumber = document.CreateAttribute("Number"); XmlAttribute attrType = document.CreateAttribute("Type"); attrNumber.Value = ="+375(44)167-85-12";

attrType.Value = "work"; phoneNumber.Attributes.Append(attrType); phoneNumber.Attributes.Append(attrNumber); phoneNumber.InnerText = ="+375(44)167-85-12";

+ Будет создан элемент &lt;PhoneNumber Type="work" Number="+375(44)167-85- 12"&gt;+375(44)167-85-12&lt;/PhoneNumber&gt;

Будет создан элемент &lt;PhoneNumber Type="work" &gt;+375(44)167-85- 12&lt;/PhoneNumber&gt;

Будет создан элемент &lt;PhoneNumber Type="work" Number="+375(44)167-85- 12"&gt;&lt;/PhoneNumber&gt;

Будет создан элемент &lt;PhoneNumber&gt;+375(44)167-85-12&lt;/PhoneNumber&gt; Ошибка выполнения

1. Что произойдет в результате выполнения следующего блока кода

XmlNode phoneNumber = document.CreateElement("PhoneNumber"); XmlAttribute attrNumber = document.CreateAttribute("Number"); XmlAttribute attrType = document.CreateAttribute("Type"); attrNumber.Value = ="+375(44)167-85-12";

attrType.Value = "work"; phoneNumber.Attributes.Append(attrType); phoneNumber.Attributes.Append(attrNumber); phoneNumber.InnerText = ="+375(44)167-85-12";

+ Будет создан элемент &lt;PhoneNumber Type="work" Number="+375(44)167-85- 12"&gt;+375(44)167-85-12&lt;/PhoneNumber&gt;

Будет создан элемент &lt;PhoneNumber Type="work" &gt;+375(44)167-85- 12&lt;/PhoneNumber&gt;

Будет создан элемент &lt;PhoneNumber Type="work" Number="+375(44)167-85- 12"&gt;&lt;/PhoneNumber&gt;

Будет создан элемент &lt;PhoneNumber&gt;+375(44)167-85-12&lt;/PhoneNumber&gt; Ошибка выполнения

1. Какой тип валидации описывает следующий блок кода

<?xml version="1.0" encoding="utf-8"?>

<!ELEMENT AddressBook (Record\*)>

<!ELEMENT Record (IdRecord,FirstName,SecondName,PhoneNumber+>

<!ELEMENT IdRecord (#PCDATA)>

<!ELEMENT FirstName (#PCDATA)>

<!ELEMENT SecondName (#PCDATA)>

<!ELEMENT PhoneNumber (#PCDATA)>

<!ELEMENT Note (String,Tag+)>

<!ELEMENT String (#PCDATA)>

<!ELEMENT Tag (#PCDATA)>

<!ELEMENT LastChangeDate (#PCDATA)>

<!ATTLIST IdRecord Id CDATA #IMPLIED>

<!ATTLIST FirstName Name CDATA #IMPLIED>

<!ATTLIST SecondName Name CDATA #IMPLIED>

<!ATTLIST PhoneNumber Number CDATA #IMPLIED>

<!ATTLIST PhoneNumber Type (home|work|mobile) #REQUIRED>

XSD

+ DTD DDT XAD

Ни один из вариантов не является верным

1. Какой тип валидации описывает следующий блок кода

<?xml version="1.0" encoding="utf-8"?>

<xs:schema xmlns:xs="<http://www.w3.org/2001/XMLSchema>">

<xs:element name="AddressBook">

<xs:complexType>

<xs:sequence>

<xs:element name="Record" minOccurs="0" maxOccurs="unbounded">

<xs:complexType>

<xs:sequence>

<xs:element name="IdRecord" maxOccurs="1" minOccurs="1">

<xs:complexType>

<xs:attribute name="Id" type="xs:string"></xs:attribute>

</xs:complexType>

</xs:element>

<xs:element name="FirstName" maxOccurs="1" minOccurs="1">

<xs:complexType>

<xs:attribute name="Name" type="xs:string"></xs:attribute>

</xs:complexType>

</xs:element>

<xs:element name="SecondName" maxOccurs="1" minOccurs="1">

<xs:complexType>

<xs:attribute name="Name" type="xs:string"></xs:attribute>

</xs:complexType>

</xs:element>

<xs:element name="PhoneNumber" minOccurs="1" maxOccurs="unbounded">

<xs:complexType>

<xs:attribute name="Number">

<xs:simpleType>

<xs:restriction base="xs:string">

<xs:pattern value="\S?\S\d{1,5}\S\d{3}-\d{2}-\d{2}"></xs:pattern>

</xs:restriction>

</xs:simpleType>

</xs:attribute>

<xs:attribute name="Type">

<xs:simpleType>

<xs:restriction base="xs:string">

<xs:enumeration value="home"/>

<xs:enumeration value="work"/>

<xs:enumeration value="mobile"/>

</xs:restriction>

</xs:simpleType>

</xs:attribute>

</xs:complexType>

</xs:element>

</xs:sequence>

</xs:complexType>

</xs:element>

</xs:sequence>

</xs:complexType>

</xs:element>

</xs:schema>

+ XSD DTD DDT XAD

Ни один из вариантов не является верным

1. Что произойдет в результате выполнения следующего блока кода, где document является экземпляром XmlDocument, а nodes – XmlNodeList

foreach (XmlElement nodt in nodes.ChildNodes)

{

if (node.InnerText.Equals(record.InnerText)) nodes.RemoveChild(nod);

}

document.Save(xmlFile);

+ Произойдет удаление элементов, соответствующих условию, и последующее сохранение изменений в xml-файл

Произойдет удаление элементов, соответствующих условию

Произойдет изменение элементов, соответствующих условию, и последующее сохранение изменений в xml-файл

Произойдет изменение элементов, соответствующих условию

Произойдет изменение элементов, соответствующих условию и вывод xml-файла

1. Что произойдет в результате выполнения следующего блока кода, где document является экземпляром XmlDocument, а node и old – XmlNode

document.ReplaceChild(node, old); document.Save(xmlFile);

Произойдет удаление элемента old и последующее сохранение изменений в xml-файл Произойдет удаление элемента node и последующее сохранение изменений в xml-файл

+ Произойдет изменение элемента old и последующее сохранение изменений в xml-файл Произойдет изменение элемента old

Ошибка выполнения

1. Что произойдет в результате выполнения следующего блока кода, где document является экземпляром XmlDocument, а node – XmlNode

document.AppendChild(node); document.Save(xmlFile);

Произойдет удаление элемента node и последующее сохранение изменений в xml-файл Произойдет изменение элемента node и последующее сохранение изменений в xml-файл

+ Произойдет добавление элемента node и последующее сохранение изменений в xml-файл Произойдет добавление элемента

Произойдет добавление элемента node без сохранения в xml-файл

## Введение в LINQ

1. Что будет выведено на экран в результате выполнения следующего блока кода

. . .

var values = new List&lt;int&gt;(); values.Add(1);

values.Add(5); values.Add(15); values.Add(11);

var result = from i in values where i > 10

select i; values[1] = 20;

foreach(var value in result)

{

Console.WriteLine(value);

}

. . .

11

15

20

+ 20

15

11

1

20

15

11

11

15

15

11

1. Что будет выведено на экран в результате выполнения следующего блока кода

. . .

List&lt;int&gt; list = new List&lt;int&gt;() { 1, 2, 3 }; var x = list.GroupBy(i => { Console.Write(i); return i; });

var y = list.ToLookup(i => { Console.Write(i); return i; });

. . .

+ 123

123123

Ничего

List&lt;int&gt; list = new List&lt;int&gt;() { 1, 2, 3, 4, 5 }; List&lt;int&gt; all = list.FindAll( i => { Console.Write(i); return i < 3; });

. . .

12

123

Ничего

+ 12345

40. Что будет выведено на экран в результате выполнения следующего блока кода

. . .

var people = new List&lt;Person&gt;()

{

new Person() {Name = "Jon", Age = 18}, new Person() {Name = "Jane", Age = 21}, new Person() {Name = "Alice", Age = 25}, new Person() {Name = "Bob", Age = 31}

};

var results = from p in people where p.Age > 20 group p by p.Age%10 int+ [] grouped orderby grouped.Key select grouped;

foreach (var g in results)

{

Console.WriteLine("Key {0} - Count {1}",g.Key, g.Count());

}

. . .

Key 2 - Count 2

Key 3 - Count 1

Key 2 - Count 2

Key 1 - Count 3

+ Key 1 - Count 2

Key 5 - Count 1

Key 1 - Count 2

Key 5 - Count 1

Key 8 - Count 1

Key 1 - Count 1

Key 2 - Count 2

Key 8 - Count 1

var values = new object[] { "1", "2", "3", "abc", 5 }; var myList = values.OfType&lt;string&gt;().ToList(); Console.WriteLine(myList.Count());

. . .

3

5

Ошибка времени выполнения

+ 4

Ошибка компиляции

1. Что будет выведено на экран в результате выполнения следующего блока кода

. . .

var words = new List&lt;string&gt; { "abc", "abcde", "acbdef", "abcdefg" }; bool tmpValue = words.All(w => w.Contains("bc")); Console.WriteLine(tmpValue);

. . .

Ошибка времени выполнения true

+ false

Ошибка компиляции

1. С помощью какого из ниже перечисленных запросов LINQ можно вычислить n-ый член ряда Фибоначи?

int Fibonacci = Enumerable.Range(1, n).Skip(2) .Join(new { Current = 1, Prev = 0 }, (x, index) => new { Current = x.Prev + x.Current, Prev = x.Current }) .Current;

int Fibonacci = Enumerable.Range(1, n).Skip(2) .Aggregate(new { Current = 1, Prev = 0 }, (x, index) => new { Current = x.Prev + x.Current, Prev = x.Prev }) .Prev;

int Fibonacci = Enumerable.Range(1, n).Take(2) .Aggregate(new { Current = 1, Prev = 0 }, (x, index) => new { Current = x.Prev + x.Current, Prev = x.Current }) .Current;

+ int Fibonacci = Enumerable.Range(1, n).Skip(2) .Aggregate(new { Current = 1, Prev = 0 }, (x, index) => new { Current = x.Prev + x.Current, Prev = x.Current }) .Current;

. . .

var namesList = new List&lt;string&gt;()

{"Василий", "Пётр", "Фёдор", "Екатерина", "Мария", "Василиса", "Петруха"}; var a = namesList.First(e => e.Contains("Вас"));

var b = namesList.Single(e => e.Contains("Вас")); Console.Write(a == b);

. . .

+ Ошибка времени выполнения Ошибка компиляции

false true

1. Выберите категорию, для которой не существует стандартных методов в LINQ. Конкатенация

Агрегация

+ Дублирование Эквивалентность

1. Что будет выведено на экран в результате выполнения следующего блока кода

List&lt;string&gt; list = new List&lt;string&gt;() { "banana", "lime", "apple" }; List&lt;string&gt; list2 = new List&lt;string&gt;() { "yellow", "green", "red" }; List&lt;string&gt; result = list.Zip(list2, (first, second)=> first + " is " + second).ToList(); result.ForEach(x => { Console.WriteLine(x); });

+ banana is yellow

lime is green

apple is red

apple is yellow lime is green banana is red

Возникнет ошибка времени компиляции Возникнет ошибка времени выполнения

List&lt;Employee&gt; employees = new List&lt;Employee&gt;()

{

new Employee(1, "Popov", "Victor", 3), new Employee(2, "Ivanov", "Dmitriy", 2),

new Employee(3, "Sidorova", "Ekaterina", 1)

};

List&lt;Payment&gt; payments = new List&lt;Payment&gt;()

{

new Payment(1, 2468.40m), new Payment(2, 4698.70m), new Payment(3, 4508.50m)

};

var result = employees.Join(payments, em => em.IdPayment, p => p.Id, (employee, payment) => new { employee.Id, employee.FirstName, employee.SecondName, payment.Value}).ToList();

result.ForEach(x => { Console.WriteLine("{0, 5}{1, 10}{2,10}{3, 10}",x.Id, x.FirstName, x.SecondName, x.Value); });

* 1. Popov Victor 3
  2. Ivanov Dmitriy 2
  3. Sidorova Ekaterina 1

+ 1 Popov Victor 4508,50

1. Ivanov Dmitriy 4698,70
2. Sidorova Ekaterina 2468,40 Возникнет ошибка времени компиляции Возникнет ошибка времени выполнения
3. Что будет выведено на экран в результате выполнения следующего блока кода class EmployeeComparer : IEqualityComparer&lt;Employee&gt;

{

public bool Equals(Employee x, Employee y)

{

if (Object.ReferenceEquals(x, y)) return true;

if (Object.ReferenceEquals(x, null) || Object.ReferenceEquals(y, null)) return false;

return x.FirstName == y.FirstName && x.SecondName == y.SecondName && x.IdPayment

== y.IdPayment;

}

public int GetHashCode(Employee employee)

{

if (Object.ReferenceEquals(employee, null)) return 0;

int hashFirstName = (employee.FirstName == null) ? 0 : employee.FirstName.GetHashCode();

int hashSecondName = (employee.SecondName == null) ? 0 : employee.SecondName.GetHashCode();

int hashIdPayment = employee.IdPayment.GetHashCode();

return hashFirstName ^ hashSecondName ^ hashIdPayment;

}

}

. . .

List&lt;Employee&gt; list1 = new List&lt;Employee&gt;()

{

new Employee(1, "Popov", "Victor", 3), new Employee(2, "Ivanov", "Dmitriy", 2),

new Employee(3, "Sidorova", "Ekaterina", 1)

};

List&lt;Employee&gt; list2 = new List&lt;Employee&gt;()

{

new Employee(1, "Popov", "Victor", 3), new Employee(2, "Grigorov", "Sergey", 2),

new Employee(3, "Sidorova", "Ekaterina", 1)

};

var result = list1.Union(list2, new EmployeeComparer()).ToList(); result.ForEach(x => {

Console.WriteLine("{0, 10}{1,10}", x.FirstName, x.SecondName);

});

+ Popov Victor Ivanov Dmitriy Sidorova Ekaterina Grigorov Sergey

Popov Victor Ivanov Dmitriy Sidorova Ekaterina Popov Victor Grigorov Sergey Sidorova Ekaterina

Возникнет ошибка времени компиляции Возникнет ошибка времени выполнения Ничего не будет выведено

1. Что будет выведено на экран в результате выполнения следующего блока кода

class EmployeeComparer : IEqualityComparer&lt;Employee&gt;

{

public bool Equals(Employee x, Employee y)

{

if (Object.ReferenceEquals(x, y)) return true;

if (Object.ReferenceEquals(x, null) || Object.ReferenceEquals(y, null)) return false;

return x.FirstName == y.FirstName && x.SecondName == y.SecondName && x.IdPayment

== y.IdPayment;

}

public int GetHashCode(Employee employee)

{

if (Object.ReferenceEquals(employee, null)) return 0;

int hashFirstName = (employee.FirstName == null) ? 0 : employee.FirstName.GetHashCode();

int hashSecondName = (employee.SecondName == null) ? 0 : employee.SecondName.GetHashCode();

int hashIdPayment = employee.IdPayment.GetHashCode();

return hashFirstName ^ hashSecondName ^ hashIdPayment;

}

}

. . .

List&lt;Employee&gt; list1 = new List&lt;Employee&gt;()

{

new Employee(1, "Popov", "Victor", 3), new Employee(2, "Ivanov", "Dmitriy", 2),

new Employee(3, "Sidorova", "Ekaterina", 1)

};

List&lt;Employee&gt; list2 = new List&lt;Employee&gt;()

{

new Employee(1, "Popov", "Victor", 3), new Employee(2, "Grigorov", "Sergey", 2),

new Employee(3, "Sidorova", "Ekaterina", 1)

};

var result = list1.Except(list2, new EmployeeComparer()).ToList(); result.ForEach(x => {

Console.WriteLine("{0, 10}{1,10}", x.FirstName, x.SecondName);

});

Popov Victor Ivanov Dmitriy Sidorova Ekaterina Grigorov Sergey

+ Ivanov Dmitriy

Возникнет ошибка времени компиляции Возникнет ошибка времени выполнения Ничего не будет выведено

1. Что будет храниться в переменной result в результате выполнения следующего блока кода List&lt;Employee&gt; list = new List&lt;Employee&gt;()

{

new Employee(1, "Popov", "Victor", 3), new Employee(2, "Ivanov", "Dmitriy", 2),

new Employee(3, "Sidorova", "Ekaterina", 1)

};

Employee res = list.FirstOrDefault(x => x.FirstName.CompareTo("Ivanof") == 0);

Возникнет ошибка InvalidOperationException

+ null

Объект Employee(2, "Ivanov", "Dmitriy", 2) Объект Employee(1, "Popov", "Victor", 3) Объект Employee(3, "Sidorova", "Ekaterina", 1)

1. Что будет храниться в переменной result в результате выполнения следующего блока кода

var result = ("this is a test phrase").Split(' ').Aggregate((cur, next) => next + " " + cur);

Возникнет ошибка InvalidOperationException

+ phrase test a is this is this a test phrase this is a test phrase a is phrase test this

1. Что будет храниться в переменной result в результате выполнения следующего блока кода

("this is a test phrase").Split(' ').OrderBy(x => x).Aggregate((cur, next) => cur + " " + next);

Возникнет ошибка InvalidOperationException phrase test a is this

is this a test phrase this is a test phrase

+ a is phrase test this

1. Что будет храниться в переменной result в результате выполнения следующего блока кода

("this is a test phrase").Split(' ').OrderByDescending(x => x).Aggregate((cur, next) => cur + " " + next);

Возникнет ошибка InvalidOperationException phrase test a is this

is this a test phrase

+this test phrase is a a is phrase test this

1. Что будет храниться в переменной result в результате выполнения следующего блока кода

("this is a test phrase").Split(' ').OrderBy(x => x).Aggregate((cur, next) => next + " " + cur);

Возникнет ошибка InvalidOperationException phrase test a is this

is this a test phrase

+this test phrase is a a is phrase test this

1. Что будет храниться в переменной result в результате выполнения следующего блока кода

("this is a test phrase").Split(' ').OrderByDescending(x => x).Aggregate((cur, next) => next + " " + cur);

Возникнет ошибка InvalidOperationException phrase test a is this

is this a test phrase this test phrase is a

+a is phrase test this

1. Что будет храниться в переменной result в результате выполнения следующего блока кода

List &lt; Employee&gt;

list = new List&lt; Employee

{

new Employee(1, "Popov", "Victor", 3); new Employee(2, "Ivanov", "Dmitriy", 2); new Employee(3, "Sidorova", "Ekaterina", 1

};

Employee res = list.First(x => x.FirstName.CompareTo("Ivanof") == 0);

+Возникнет ошибка InvalidOperationException null

Объект Employee(2, "Ivanov", "Dmitriy", 2) Объект Employee(1, "Popov", "Victor", 3)

Объект Employee(3, "Sidorova", "Ekaterina", 1)

1. Что будет храниться в переменной result в результате выполнения следующего блока кода

List&lt; Employee&gt;

list = new List&lt; Employee()

{

new Employee(1, "Popov", "Victor", 3); new Employee(2, "Ivanov", "Dmitriy", 2);

new Employee(3, "Sidorova", "Ekaterina", 1)

};

Employee res = list.Find(x => x.FirstName.CompareTo("Ivanof") == 0);

Возникнет ошибка InvalidOperationException

+null

Объект Employee(2, "Ivanov", "Dmitriy", 2) Объект Employee(1, "Popov", "Victor", 3) Объект Employee(3, "Sidorova", "Ekaterina", 1)

1. Что будет храниться в переменной result в результате выполнения следующего блока кода

("apple orange banana apricot").Split(' ').ToList().Find(x => x.StartsWith("ap")); Выберите один ответ:

+ apple apricot

Возникнет ошибка InvalidOperationException apple apricot

null

1. Что будет храниться в переменной result в результате выполнения следующего блока кода

List list = new List() { "apple", "orange", "banana", "strawberry", "apricot" }; var result = list.Skip(2).Take(1);

Выберите один ответ:

null

Возникнет ошибка InvalidOperationException banana strawberry

apple

+ banana

("apple orange apricot banana").Split(' ').ToList().Last(x => x.StartsWith("ap")); Выберите один ответ:

Возникнет ошибка InvalidOperationException

+ apricot null apple banana

1. Что будет храниться в переменной result в результате выполнения следующего блока кода

List list = new List() { "apple", "orange", "banana", "strawberry", "apricot" }; var result = list.Skip(2).Take(2);

Выберите один ответ:

+ banana strawberry

Возникнет ошибка InvalidOperationException apple

banana null

1. Что будет храниться в переменной result в результате выполнения следующего блока кода

List list = new List() { "apple", "orange", "banana", "strawberry", "apricot" }; var result = list.SkipWhile(x => x.CompareTo("banana") != 0);

Выберите один ответ:

strawberry apricot

Возникнет ошибка InvalidOperationException banana

+ banana strawberry apricot null

list.Single(x => x.StartsWith("ap")); Выберите один ответ:

+ Возникнет ошибка InvalidOperationException null

banana apricot apple

1. Что будет храниться в переменной result в результате выполнения следующего блока кода

List list = new List() { "apple", "orange", "banana", "strawberry", "apricot" }; var result = list.TakeWhile(x => x.CompareTo("banana") != 0);

Выберите один ответ:

null

apple orange banana

Возникнет ошибка InvalidOperationException banana strawberry apricot

+ apple orange

1. Что будет храниться в переменной result в результате выполнения следующего блока кода

("apple orange banana apricot").Split(' ').ToList().FindAll(x => x.StartsWith("ap")); Выберите один ответ:

apricot apple

Возникнет ошибка InvalidOperationException

+ apple apricot null

var result = list.Skip(5).Take(1); Выберите один ответ:

+ null

banana strawberry apple

Возникнет ошибка InvalidOperationException banana

## Делегаты и события

1. Для следующей иерархии классов public class Person

{

}

public class Employee : Person

{

}

и методов

static Employee FindByTitle(String title)

{

. . .

return new Employee();

}

static void AddToContacts(Person person)

{

. . .

}

какие из перечисленных операторов присваивания являются законными

+ Func&lt;String, Employee> findEmployee = FindByTitle;

+ Func&lt;String, Person> findPerson = FindByTitle;

+ Action&lt;Person> addPersonToContacts = AddToContacts;

+ Action&lt;Employee> addEmployeeToContacts = AddToContacts; Нет верных ответов

1. Для следующей иерархии классов public class Person

{

}

public class Employee : Person

{

}

и методов

static Employee FindByTitle(String title)

{

. . .

return new Employee();

}

static void AddToContacts(Employee employee)

{

. . .

}

какие из перечисленных операторов присваивания являются законными

+ Func&lt;String, Employee> findEmployee = FindByTitle;

+ Func&lt;String, Person> findPerson = FindByTitle; Action&lt;Person> addPersonToContacts = AddToContacts;

Action&lt;Person> addPersonToContacts = new Delegate(AddToContacts);

+ Action&lt;Employee> addEmployeeToContacts = AddToContacts;

1. Для следующей иерархии классов public class Person

{

}

public class Employee : Person

{

}

и методов

static Person FindByTitle(String title)

{

. . .

return new Employee();

}

static void AddToContacts(Person person)

{

. . .

}

какие из перечисленных операторов присваивания являются законными

Func&lt;String, Employee> findEmployee = FindByTitle; Func&lt;String, Employee> findEmployee = new Delegate(FindByTitle);

+ Func&lt;String, Person> findPerson = FindByTitle;

+ Action&lt;Person> addPersonToContacts = AddToContacts;

+ Action&lt;Employee> addEmployeeToContacts = AddToContacts;

1. Отметьте правильные утверждения

+ CLR предоставляет каждому типу делегата методы BeginInvoke и EndInvoke, чтобы обеспечить возможность асинхронного вызова делегата

+ Делегат – это экземпляр типа делегата, который может содержать указатель на статический метод

Можно создавать новые типы, производные от типа делегата Делегат не может быть равен null

Делегат не может хранить цепочку методов

1. Что будет выведено на экран при выполнении следующего блока кода

. . .

public delegate void Del(string message);

public class MethodClass

{

public void Method1(string message)

{

Console.WriteLine(message + "!");

}

public void Method2(string message)

{

Console.WriteLine(message + "?");

}

}

class Program

{

public static void Method3(string message)

{

Console.WriteLine(message + ".");

}

static void Main(string[] args)

{

MethodClass obj = new MethodClass(); Del d1 = obj.Method1;

Del d2 = obj.Method2;

Del d3 = Method3;

Del allMethodsDelegate = d1 + d2; allMethodsDelegate += d3; allMethodsDelegate("Run");

}

}

+ Run!

Run?

Run.

1. Что будет выведено на экран в результате выполнения следующего блока кода

int count = 0; int total = 0;

Func&lt;int> countUp = () => count++; for (int i = 0; i &lt; 5; i++)

{

total += countUp();

}

Console.WriteLine("Total = " + total);

Total = 5

Total = 0

+ Total = 10

Total = 15

Total = 20

1. Можно ли делегату Func(T, TResult) присвоить следующее лямбда-выражение Func&lt;double, double> fact = null; fact = (x) => (x > 1) ? x \* fact(x - 1) : 1;

+ Да Нет

277. Какой результат вернет делегат fact(5) если он задан следующим лямбда-выражением Func&lt;double, double&gt; fact = null; fact = (x) =&gt; (x &gt; 1) ? x \* fact(x - 1) : 1;

+ 120

125

5

4

1

1. Метод GetMethodImpl()

+ возвращает статический метод, представленный текущим делегатом возвращает последний метод, представленный текущим делегатом возвращает первый метод, представленный текущим делегатом имплементирует вызов всех методов текущего делегата

нет верного ответа

1. Что будет выведено на экран после выполнения следующего кода

delegate void Del(int i, double j); class MathClass

{

static void Main() {

MathClass m = new MathClass(); Del d = m.MultiplyNumbers;

for (int i = 1; i &lt;= 3; i++) d(i, 2);

}

void MultiplyNumbers(int m, double n)

{

System.Console.Write(m \* n + " ");

}

}

пустой экран

+ 2 4 6

1 3 5

ошибка компиляции ошибка выполнения

1. Следующие утверждения не верны

События, не имеющие подписчиков, никогда не возникают

+ Если событие имеет несколько подписчиков, то при его возникновении происходит асинхронный вызов обработчиков событий.

События, не имеющие подписчиков, никогда не возникают

В библиотеке классов .NET Framework в основе событий лежит делегат EventHandler и базовый класс EventArgs

+ Для одного события может быть не более 128 подписчиков

1. Над делегатами могут производиться следующие операции

+ +=

--

+ ==

===

+ !=

1. Метод Finalize()

+ позволяет объекту попытаться освободить ресурсы и выполнить другие операции по очистке перед тем, как объект будет утилизирован сборщиком мусора

возвращает последний метод, представленный текущим делегатом удаляет последний метод, представленный текущим делегатом выполняет все методы делегата

вызывает сборщик мусора

1. Что будет выведено на экран в результате выполнения следующего блока кода

delegate double function(ref double x); static event function onFunc;

static double Method(ref double x) { return x += 5; } static double Method2(ref double x) { return x \* x; } static double Method3(ref double x) { return x + x; }

static void Main(string[] args)

{

double x = 9; onFunc += Method; onFunc += Method2; onFunc += Method3;

Console.WriteLine(onFunc(ref x)); Console.WriteLine(x);

}

28

28

+ 28

14

14

14

392

392

64. Что будет выведено на экран в результате выполнения следующего блока кода static double Method(ref double x) { return x += 5; }

static double Method2(ref double x) { return x \*= x; } static double Method3(ref double x) { return x += x; }

static void Main(string[] args)

{

double x = 3; onFunc += Method; onFunc += Method2; onFunc += Method3;

Console.WriteLine(onFunc(ref x)); Console.WriteLine(x);

}

16

+ 128

128

8

8

128

8

1. Для следующего блока кода

static double Method(double x) { return x += x; } static Func&lt;double, double> function;

какие из перечисленных операторов присваивания являются законными

function = (double x) { return x; }; function = (int x) { return (double)x; };

+ function = Method;

+ function = delegate (double x) { return x; };

+ function = (x) => { return x; };

1. Что будет выведено на экран в результате выполнения следующего блока кода delegate void function(ref double x);

static event function onFunc;

static void Method(ref double x) { Console.WriteLine(x += 5); } static void Method2(ref double x) { Console.WriteLine(x \* x); } static void Method3(ref double x) { Console.WriteLine(x + x); }

static void Main(string[] args)

{

double x = 3; onFunc += Method; onFunc += Method2; onFunc += Method3;

onFunc(ref x);

}

9

6

+ 8

64

16

8

64

128

Возникнет ошибка времени компиляции

1. Что будет выведено на экран в результате выполнения следующего блока кода delegate void function(ref double x);

static event function onFunc;

static void Method(ref double x) { Console.WriteLine(x += 5); } static void Method2(ref double x) { Console.WriteLine(x \*= x); } static void Method3(ref double x) { Console.WriteLine(x += x); }

static void Main(string[] args)

{

double x = 3; onFunc += Method; onFunc += Method2; onFunc += Method3;

onFunc(ref x);

}

8

9

6

8

64

16

+ 8

64

128

Возникнет ошибка времени компиляции

## Инкапсуляция данных и методов. Перегрузка операций

1. Отметьте оператор, который определяет индексатор public object Item[int index]

{

get {...}

set {...}

}

public Item[int index]

{

get {...}

set {...}

}

public object this[int index]

{

get { }

set { }

}

+ public class Item[int index]

{

get {...}

set {...}

}

1. Статический конструктор вызывается

после создания первого экземпляра класса или ссылки на какие-либо статические члены сразу после начала выполнения программы

+ перед созданием первого экземпляра или ссылки на какие-либо статические члены после создания первого экземпляра класса, но перед ссылкой на любой статический член только когда осуществляется доступ к члену класса, который обращается к статическому ресурсу

1. Отметьте правильные утверждения

+ Для определения индексатора используется ключевое слово this

Индексаторы должны использовать в качестве индекса целочисленное значение Индексаторы нельзя перегружать

+ Индексаторы могут иметь более одного формального параметра

+ В индексаторе допустимо использование опциональных параметров Индексаторы не могут иметь более одного формального параметра

Параметры индексатора не могут быть описаны с использованием ключевого слова params

1. Отметьте правильные утверждения

+ Модификатор доступа internal – доступ к типу или члену возможен из любого кода в той же сборке, но не из другой сборки

Модификатор доступа protected – доступ к типу или члену возможен из любого кода в той же сборке или другой сборке, ссылающейся на него

Если не указать модификатор доступа для члена типа, уровень доступа для него по умолчанию будет internal

+ Если не указать модификатор доступа для типа, по умолчанию применяется уровень доступа internal

Если не указать модификатор доступа при объявлении глобальной переменной, уровень доступа для нее по умолчанию будет protected

1. При выполнении следующего блока кода class User

{

internal static int usersOnline; internal User()

{

usersOnline++;

}

}

User a = new User(); User b = new User(); User c = new User(); User d = new User();

int totalUsersOnline = User.usersOnline;

значение переменной totalUsersOnline будет равно 1

0

+ 4

Возникнет ошибка времени выполнения Возникнет ошибка времени компиляции

1. При выполнении следующего блока кода

class User

{

internal int usersOnline; internal User()

{

usersOnline++;

}

}

User a = new User(); User b = new User(); User c = new User(); User d = new User();

int totalUsersOnline = d.usersOnline;

значение переменной totalUsersOnline будет равно

+ 1

4

0

Возникнет ошибка времени выполнения Возникнет ошибка времени компиляции

1. Какой метод реализуется для создания метода расширения Reverse() класса string? public void Reverse(this string s);

public static string Reverse();

+ public static string Reverse(this string s); public static void Reverse(this string s); public string Reverse(string s);

1. Отметьте операции/ключевые слова, которые не поддерживают перегрузку напрямую true

+ +=

\*

+ ||

+

1. Какой метод реализуется для перегрузки операции сложения

public static ComplexNumber operation + (ComplexNumber a, ComplexNumber b)

+ public static ComplexNumber operator + (ComplexNumber a, ComplexNumber b) public static ComplexNumber this + (ComplexNumber a, ComplexNumber b) public static ComplexNumber + (ComplexNumber a, ComplexNumber b)

1. Отметьте модификаторы, которые нельзя использовать для изменения метода override

+ virtual

+ new public internal

+ static

1. Отметьте модификаторы, которые нельзя использовать c модификатором virtual

+ override

+ private

+ abstract

+ static public

1. Переопределенный базовый метод должен иметь тип override или abstract

virtual abstract

protected internal

+ virtual, abstract или override

1. Для классов по умолчанию уровень доступа в c# public

protected internal private

protected

+ internal

public readonly int counter = 10;

}

class Program

{

static void Main(string[] args)

{

A a = new A(); a.counter++;

Console.WriteLine(a.counter);

}

}

на экран будет выведено

10

0

11

Возникнет ошибка времени выполнения

+ Возникнет ошибка времени компиляции

int counter; public int Counter

{

get { return counter; }

set { counter = (value < 10) + [ ] value : counter; }

}

}

class Program

{

static void Main(string[] args)

{

A a = new A(); a.Counter = 10; a.Counter++;

Console.WriteLine(a.Counter);

}

на экран будет выведено

10

11

+ 1

Возникнет ошибка времени выполнения Возникнет ошибка времени компиляции

internal static int Counter { get; set; }

}

class B: A { }

class Program

{

static void Main(string[] args)

{

B.Counter+=10; Console.WriteLine(A.Counter);

}

}

на экран будет выведено

0

-1

+ 10

Возникнет ошибка времени выполнения Возникнет ошибка времени компиляции

class Program

{

static void Main(string[] args)

{

A a = new A(); B b = new B(); b.Counter+=10;

Console.WriteLine(a.Counter);

}

}

на экран будет выведено

+ 0

10

-1

Возникнет ошибка времени выполнения Возникнет ошибка времени компиляции

{

public int Counter { get; set; } public void Go(int counter)

{

A.Counter = counter; Console.WriteLine(this.Counter + A.Counter);

}

}

}

class Program

{

static void Main(string[] args)

{

A.B b = new A.B(); b.Counter = 15; b.Go(10); Console.ReadKey();

}

}

на экран будет выведено

+ 25

15

0

Возникнет ошибка времени выполнения Возникнет ошибка времени компиляции

1. При выполнении следующего блока кода, что будет выведено на экран: class A

{

public static int x = 0; public static string s=""; public int y = 0;

public A() { y++; }

static A() { s = Math.Cos(x).ToString(); }

}

static void Main(string[] args)

{

A a = new A();

Console.WriteLine("{0} | {1}",a.y, A.s);

}

Возникнет ошибка времени компиляции

+ 1 | 1

1 |

|

| 1

0 | 1

1. При выполнении следующего блока кода, что будет выведено на экран: class A

{

public static int x = 0; public static string s=""; public int y = 0;

public A() { y++; }

static A() { s = Math.Cos(x).ToString(); }

}

static void Main(string[] args)

{

A a = new A(); Console.WriteLine("{0} | {1}",a.y, a.s);

}

+ Возникнет ошибка времени компиляции 1 | 1

1 |

|

| 1

0 | 1

1. class Program

{

static int M1(params int[] a)

{

int s = a[0];

for (int i = 1; i &lt; a.Length; i++) s -= a[i]; return s;

}

static int M1(int x, int y)

{ return x + y; } static void Main(string[] args)

{

int x = 3; int y = 1;

Console.WriteLine(M1(x,y));

}

}

0

Возникнет ошибка времени компиляции 1

2

3

+ 4

5

1. class Program

{

static int M1(out int x, int y)

{ x=y; return x + y; } static int M1(params int[] a)

{

int s = a[0];

for (int i = 1; i &lt; a.Length; i++) s -= a[i]; return s;

}

static void Main(string[] args)

{

int x = 3; int y = 1;

Console.WriteLine(M1(x,y));

}

}

0

Возникнет ошибка времени компиляции 1

+ 2

3

4

5

## Коллекции и обобщенные типы

1. Что будет выведено на консоль при выполнении следующего кода static class Program

{

static IEnumerable Squares(this int from, int to)

{

for(int i = from; i &lt;= to; i++) yield return (int)i \* i;

}

static void Main(string[] args)

{

var min = 1;

foreach (int i in min.Squares(4))

{

Console.WriteLine(i);

}

}

}

1

1

2

4

8

2

4

6

8

+ 1

4

9

16

1

2

3

4

1. При компиляции следующего кода

class Hash&lt;TKey, TValue> : Dictionary&lt;TKey, TValue> where TKey : struct, new()

where TValue : class, new()

{. . .}

возникнет ошибка компиляции потому что

нельзя наследовать от обобщенного класса Dictionary нельзя иметь два обобщенных одинаковых типа

+ ограничения new() не может быть использовано с ограничением struct нельзя использовать два ограничения для одного класса

ошибки не возникнет

1. При определении анонимного типа компилятор преобразует его в: Новую строку с методом get

Новую структуру с методами get и set.

+ Новый класс со свойствами и методами get и set. Новый класс со свойствами и методом get.

Хэш-таблицу со значениями, определенными в инициализаторе.

1. При выполнении следующего кода class Program

{

static IEnumerable&lt;string> Get()

{

yield return "Hello"; Console.WriteLine("Hello"); yield return "World";

}

static void Main(string[] args)

{

foreach (string s in Get())

{

Console.WriteLine(s); Console.WriteLine("C#");

}

}

}

на экран будет выведено

Hello C#

+ Hello

C#

Hello

World

C#

Hello Hello World C#

Hello C#

World C#

1. При выполнении следующего кода class Program

{

static void Main(string[] args)

{

Hashtable something = new Hashtable(); sandwich.Add("a", "bacon");

sandwich.Add("c","audi");

sandwich.Add("b", "java");

foreach (DictionaryEntry de in something) Console.WriteLine(de.Value);

}

на экран будет выведено

bacon audi java

audi bacon java

java

bacon audi

+ bacon

java

audi

Выдаст ошибку

class Program

{

static void Main(string[] args)

{

SortedList something = new SortedList(); sandwich.Add("a", "bacon");

sandwich.Add("c","audi");

sandwich.Add("b", "java");

foreach (DictionaryEntry de in something) Console.WriteLine(de.Value);

}

на экран будет выведено

bacon audi java

audi bacon java

java

bacon audi

+ bacon

java

audi

Выдаст ошибку

96. Неуниверсальные коллекции хранят

+ элементы типа Object и требуют приведения элементы типа Object и не требуют приведения элементы значимых типов и не требуют приведения элементы ссылочных типов и не требуют приведения

public class Example

{

public static void Main()

{

Stack&lt;string> dinosaurs = new Stack&lt;string>(); dinosaurs.Push("1");

dinosaurs.Push("2"); dinosaurs.Push("3"); dinosaurs.Push("4"); dinosaurs.Pop(); dinosaurs.Push("5"); dinosaurs.Peek();

foreach (string dinosaur in dinosaurs) Console.Write(dinosaur+” “);

}

}

на экран будет выведено

+ 5 3 2 1

5 4 3 2 1

5 5 3 2 1

1 2 3 4 5

4 4 3 2 1

1. При выполнении следующего кода public class Example

{

public static void Main()

{

bool[] mybool = new bool[3] { false, true, true}; BitArray newbool = new BitArray(mybool); BitArray dinosaurs = new BitArray(3,true); dinosaurs.Set(1 , false); dinosaurs.Xor(newbool);

foreach (bool dinosaur in dinosaurs) Console.Write(dinosaur+” “);

}

}

на экран будет выведено

false false false

+ true true false true true true

false true true true false false false false true

Ошибка компиляции

1. При выполнении следующего кода List&lt;int> list = new List&lt;int>()

{

4,6,2,5,7,5,3,5,7,8,5,3,3,1,0,6,9,7,4

};

ISet&lt;int> items = new SortedSet&lt;int>(); list.ForEach(x => items.Add(x)); items.ToList().ForEach(x => Console.Write(x + " "));

на экран будет выведено

4 6 2 5 7 5 3 5 7 8 5 3 3 1 0 6 9 7 4

+ 0 1 2 3 4 5 6 7 8 9

4 6 2 5 7 3 8 1 0 9

0 1 2 3 3 3 4 4 5 5 5 5 6 6 7 7 7 8 9

9 8 7 6 5 4 3 2 1 0

1. При выполнении следующего кода List&lt;int> list = new List&lt;int>()

{

4,6,2,5,7,5,3,5

};

Stack&lt;int> items = new Stack&lt;int>(); list.ForEach(x => items.Push(x)); items.ToList().ForEach(x => Console.Write(x + " "));

на экран будет выведено 4 6 2 5 7 5 3 5

+ 5 3 5 7 5 2 6 4

2 3 4 5 5 5 6 7

7 6 5 5 5 4 3 2

Будет бесконечно выводиться 5

1. При выполнении следующего кода

List&lt;int> list = new List&lt;int>()

{

4,6,2,5,7,5,3,5

};

Stack&lt;int> items = new Stack&lt;int>(); list.ForEach(x => items.Push(x));

while (items.Count > 0)

Console.Write(items.Pop() + " "); на экран будет выведено

4 6 2 5 7 5 3 5

+ 5 3 5 7 5 2 6 4

2 3 4 5 5 5 6 7

7 6 5 5 5 4 3 2

Будет бесконечно выводиться 5

1. При выполнении следующего кода List&lt;int> list = new List&lt;int>()

{

4,6,2,5,7,5,3,5

};

Stack&lt;int> items = new Stack&lt;int>(); list.ForEach(x => items.Push(x));

while (items.Count > 0)

Console.Write(items.Peek() + " "); на экран будет выведено

4 6 2 5 7 5 3 5

5 3 5 7 5 2 6 4

2 3 4 5 5 5 6 7

7 6 5 5 5 4 3 2

+ Будет бесконечно выводиться 5

1. Скомпилируется ли следующий код?

ArrayList items = new ArrayList(); items.Add(new List&lt;Employee>()

{

new Employee(1, "Popov", "Victor", 3), new Employee(2, "Ivanov", "Dmitriy", 2),

new Employee { Id =3, FirstName = "Sidorova", SecondName = "Ekaterina", IdPayment = 1 }

});

items.Add(new Dictionary&lt;int, Payment>()

{

{1, new Payment(2457.64m)},

{2, new Payment {Value = 3453.45m } },

{3, new Payment(3467.87m) }

});

items.Add(new int[] { 1, 3, 6, 3, 9, 2 }); items.Add("Hell+ [] world");

+ Да Нет

1. Что произойдет после выполнения следующего кода?

ArrayList items = new ArrayList(); items.Add(new List&lt;Employee>()

{

new Employee(1, "Popov", "Victor", 3), new Employee(2, "Ivanov", "Dmitriy", 2),

new Employee { Id =3, FirstName = "Sidorova", SecondName = "Ekaterina", IdPayment = 1 }

});

items.Add(new Dictionary&lt;int, Payment>()

{

{1, new Payment(2457.64m)},

{2, new Payment {Value = 3453.45m } },

{3, new Payment(3467.87m) }

});

items.Add(new int[] { 1, 3, 6, 3, 9, 2 }); items.Add("Hell+ [] world");

+Код скомпилируется без ошибок. Так как ArrayList принимает Object Код скомпилируется без ошибок. Так как ArrayList принимает String

Код скомпилируется без ошибок и будет выведен список обхектов из ArrayList Возникнет ошибка компиляции

Возникнет ошибка времени выполнения

## Наследование. Интерфейсы и абстрактные классы

1. В следующем коде

public class C : T1, T2 { }; public struct S : T3, T4 { };

T1, T2, T3, T4 могут быть

+ Т1 – класс

+ Т1 – интерфейс Т2 – класс

+ Т2 – интерфейс Т3 – класс

+ Т3 – интерфейс Т4 – класс

+ Т4 – интерфейс

1. Что будет выведено на экран после выполнения следующего кода class Program

{

class A

{

}

public override string ToString()

{ return "class A"; }

class B : A

{

public override string ToString()

{ return "class B"; }

}

static void Main(string[] args)

{

object[] mas = { new object(), new A(), new B() }; foreach (var m in mas)

{ Console.Write(" {0} ",m.ToString()); }

}

}

Ошибка времени исполнения class A class B

+ System.Object class A class B System.Object System.Object System.Object null class A class B

1. Выберете правильные утверждения

+ Невозможно создать экземпляр интерфейса напрямую

Интерфейсы не могут содержать индексаторы и свойства в качестве членов Интерфейс не может быть унаследован от нескольких интерфейсов

+ Нельзя создать экземпляры абстрактного класса Интерфейсы могут содержать реализации методов

+ Любой неабстрактный тип, реализующий интерфейс, должен реализовать все его члены

+ Абстрактные классы могут содержать абстрактные методы и методы доступа

1. Что будет выведено на экран после выполнения следующего кода class Program

{

class A

{

}

public override string ToString()

{ return "class A"; }

class B : A

{

public override string ToString()

{ return "class B"; }

}

static void Main(string[] args)

{

object[] mas = { new B(), new A(), new B() }; foreach (B m in mas)

{ Console.Write(" {0} ",m.ToString()); }

}

}

+ Ошибка времени исполнения class B System.Object class B class B class A class B

class B class B class B

System.Object System.Object System.Object

public class A

{

public void A1()

{

Console.WriteLine("This is A");

}

}

public class B : A

{

public void A1()

{

Console.WriteLine("This is B"); base.A1();

}

}

class Program

{

static void Main(string[] args)

{

B b = new B(); b.A1();

}

}

This is A This is B

+ This is B

This is A

This is A This is B

выдаст ошибку

109. Какой модификатор служит для запрета наследования класса

+ sealed seled abstract protected private

public class BaseC

{

public class NestedC

{

public int x = 200; public int y;

}

}

public class DerivedC : BaseC

{

new public class NestedC

{

public int x = 100; public int y; public int z;

}

static void Main()

{

NestedC c1 = new NestedC(); BaseC.NestedC c2 = new BaseC.NestedC(); Console.WriteLine(c1.x); Console.WriteLine(c2.x);

}

}

200

200

100

100

200

100

+ 100

200

Ошибка компиляции

1. Выберете правильные утверждения

+Интерфейс похож на абстрактный базовый класс. Любой класс (или структура), реализующий интерфейс, должен реализовывать все его члены.

+Невозможно создать экземпляр интерфейса напрямую. Его члены реализованы всеми классами или структурами, реализующими интерфейс.

+Интерфейсы могут содержать события, индексаторы, методы и свойства.

+Интерфейсы не содержат реализацию методов.

+Класс или структура может реализовывать несколько интерфейсов. Класс может наследовать базовому классу и также реализовывать один или несколько интерфейсов.

Интерфейсы могут содержать поля, индексаторы, методы и свойства.

1. Какой уровень доступа в интерфейсах по умолчанию

+ public private internal protected sealed

public class B : A

{

static B()

{

Console.WriteLine("This is B");

}

}

class Program

{

static void Main(string[] args)

{

B b = new B();

}

}

This is A This is B

This is A This is B

+ This is B

This is A

компилятор выдаст ошибку

public class B: A

{

public new void Go()

{

Console.WriteLine("This is B");

}

}

public class C : A

{

public new void Go()

{

base.Go(); Console.WriteLine("This is C");

}

}

class Program

{

static void Main(string[] args)

{

B b = new B(); C c = new C(); b.Go();

c.Go();

}

}

This is A This is B

This is B This is A

This is A This is B This is C

+ This is B

This is A

This is C

Компилятор выдаст ошибку

1. Что будет выведено на экран после выполнения следующего кода

public abstract class A

{

public void Go()

{

Console.WriteLine("This is A");

}

}

public class B: A

{

public new void Go()

{

Console.WriteLine("This is B");

}

}

public class C : A{ } class Program

{

static void Main(string[] args)

{

B b = new B(); C c = new C(); b.Go();

c.Go();

}

}

This is A This is B

+ This is B

This is A

This is A This is B This is C

This is B This is A This is C

Компилятор выдаст ошибку

1. Что будет выведено на экран после выполнения следующего кода

public interface IFirst

{

void Go();

}

public interface ISecond

{

void Go();

}

public interface IGeneral: IFirst, ISecond { } public class A: IGeneral

{

void IFirst.Go()

{

Console.WriteLine("This is IFirst");

}

void ISecond.Go()

{

Console.WriteLine("This is ISecond");

}

}

class Program

{

static void Main(string[] args)

{

IGeneral a = new A(); a.Go();

}

}

This is IFirst This is ISecond

Возникнет ошибка времени выполнения Компилятор выдаст ошибку

+ Возникнет ошибка времени компиляции

1. Что будет выведено на экран после выполнения следующего кода

public interface IFirst

{

void Go();

}

public interface ISecond

{

void Go();

}

public interface IGeneral: IFirst, ISecond { } public class A: IGeneral

{

void IFirst.Go()

{

Console.WriteLine("This is IFirst");

}

void ISecond.Go()

{

Console.WriteLine("This is ISecond");

}

}

class Program

{

static void Main(string[] args)

{

IFirst a = new A(); a.Go();

}

}

This is IFirst This is ISecond This is ISecond This is IFirst

+ This is IFirst

Возникнет ошибка времени выполнения Возникнет ошибка времени компиляции

1. Какое (какие) ключевые слова пропущены в тексте создания абстрактного класса, чтобы можно было корректно воспользоваться полиморфизмом?

public abstract class Figure

{

public int X { get; set; } public int Y { get; set; }

??? double GetPerimeter();

}

abstract

private abstract

+ public abstract protected abstract public override

1. Что будет на экране в результате запуска приложения? abstract class A

{ public abstract void M(); }

class B : A

{

public sealed override void M()

{ Console.WriteLine("B"); }

}

class C : B

{

public override void M()

{ Console.WriteLine("C"); }

}

class Program

{

static void Main(string[] args)

{

B b = new B(); C c = new C(); b.M();

c.M();

}

}

BC CB

Ошибка времени исполнения BB

+ Ошибка времени компиляции

1. Что будет на экране в результате запуска приложения?

abstract class A

{ public abstract void M(); }

class B : A

{

public sealed override void M()

{ Console.WriteLine("B"); }

}

class C : B

{

public new void M()

{ Console.WriteLine("C"); }

}

class Program

{

static void Main(string[] args)

{

B b = new B(); C c = new C(); b.M();

c.M();

}

}

+ BC CB

Ошибка времени исполнения BB

Ошибка времени компиляции

public class A

{

public A()

{

Console.WriteLine("This is A");

}

}

public class B : A

{

public B()

{

Console.WriteLine("This is B");

}

}

class Program

{

static void Main(string[] args)

{

B b = new B();

}

}

This is A This is B

Компилятор выдаст ошибку

+ This is A

This is B

This is B This is A

class A

{

public void Test(double n)

{

Console.WriteLine("Test A");

}

}

class B : A

{

public void Test(int n)

{

Console.WriteLine("Test B");

}

}

class Program

{

static void Main(string[] args)

{

B b = new B(); b.Test(5.0);

}

}

+ Test A Test B

Возникнет ошибка времени выполнения Возникнет ошибка времени компиляции Компилятор выдаст ошибку

123. Выберете правильные утверждения, описывающие следующий код

interface I1

{

void Do();

}

interface I2

{

void Do();

}

class C : I1, I2

{

void I1.Do(){. . .}

void I2.Do(){. . .}

}

Данный код не скомпилируется по причине конфликта имен Данный код содержит неявную реализацию интерфейсов

+ Данный код содержит явную реализацию интерфейсов

+ Данный код позволяет выполнить разные реализации метода D+

Данный код не скомпилируется, поскольку в C# невозможна реализация нескольких интерфейсов.

class A

{

public void Test(int n)

{

Console.WriteLine("Test A");

}

}

class B : A

{

public void Test(double n)

{

Console.WriteLine("Test B");

}

}

class Program

{

static void Main(string[] args)

{

B b = new B(); b.Test(5);

}

}

Test A

+ Test B

Возникнет ошибка времени выполнения Возникнет ошибка времени компиляции Компилятор выдаст ошибку

struct Book

{

public string author; public string title; public int copyright;

public Book()

{

author = "aaa"; title = "bbb"; copyright = 50;

Console.WriteLine("This is a book.");

}

}

class Program

{

static void Main(string[] args)

{

Book bk = new Book(); Console.WriteLine("This is not a book");

}

}

This is not a book This is a book

Возникнет ошибка времени выполнения

+ Возникнет ошибка времени компиляции Компилятор выдаст ошибку

public interface I

{

void Go();

}

public class A : I

{

public void Go()

{

Console.WriteLine("A.Go()");

}

}

class B : A

{

}

class C : B, I

{

public new void Go()

{

Console.WriteLine("C.Go()");

}

}

class Program

{

static void Main(string[] args)

{

B b1 = new B(); C c1 = new C(); B b2 = c1; b1.Go();

c1.Go();

b2.Go(); ((I)b2).Go();

}

}

A.Go();

C.Go();

A.Go();

A.Go();

A.Go();

C.Go();

C.Go();

A.Go();

+ A.Go();

C.Go();

A.Go();

C.Go();

A.Go();

C.Go();

C.Go();

C.Go();

public class A

{

public virtual void M()

{

Console.WriteLine("This is A");

}

}

public class B : A

{

public new void M()

{

Console.WriteLine("This is B");

}

}

class Program

{

static void Main(string[] args)

{

B b = new B(); A a = b; a.M();

}

}

+ This is A This is B

This is A This is B

Возникнет ошибка времени выполнения Возникнет ошибка времени компиляции

public class Car

{

public Car()

{

System.Console.WriteLine("The Car constructor invoked.");

}

}

public class Bus : Car

{

static Bus()

{

System.Console.WriteLine("The Bus constructor invoked.");

}

public static void Drive()

{

System.Console.WriteLine("The Drive method invoked.");

}

}

class Program

{

static void Main(string[] args)

{

Bus.Drive();

}

}

The Drive method invoked.

The Car constructor invoked. The Drive method invoked.

+ The Bus constructor invoked.

The Drive method invoked.

The Car constructor invoked. The Bus constructor invoked. The Drive method invoked.

The Bus constructor invoked. The Car constructor invoked. The Drive method invoked.

## Обработка исключений

1. При выполнении следующего кода unchecked

{

int i = -1; Console.WriteLine((byte)i);

}

на экран будет выведено

-1

1

0

+ 255

256

1. При выполнении следующего кода unchecked

{

int i = 256;

byte b = (byte)i; Console.WriteLine((int) b + " ");

}

на экран будет выведено

-1

1

+ 0

255

256

...

public const int END = int.MaxValue; public const int START = END - 100;

...

private static void Main(string[] args)

{

int count = 0; unchecked

{

for (int i = START; i <= END; i++)

{

count++;

}

}

Console.WriteLine(count);

}

На экран будет выведено 100 На экран будет выведено 101 На экран будет выведено 99

Возникнет ошибка времени компиляции Возникнет ошибка времени выполнения

+ Не выполнится ничего из перечисленного

132. При выполнении следующего кода unchecked

{

int i = int.MaxValue; i++;

Console.WriteLine(i);

}

на экран будет выведено

2147483648

2147483647

+ -2147483648

-2147483647

checked

{

Console.WriteLine(8 << 32);

Console.WriteLine(8 << 33);

}

на экран будет выведено

0

0

+ 8

16 8

8

16

24

134. При выполнении следующего кода

...

int i = 0; unchecked

{

while (-1 << i != 0)

{

i++;

}

}

Console.WriteLine(i);

...

На экран будет выведено 15 На экран будет выведено 16 На экран будет выведено 31 На экран будет выведено 32

Возникнет ошибка времени компиляции Возникнет ошибка времени выполнения

+ Ничего из перечисленного

checked

{

double i = 0.0/0.0;

if (i != i)

Console.WriteLine("not equal"); else

Console.WriteLine("equal");

}

Возникнет ошибка времени компиляции Возникнет ошибка времени выполнения

+ На экран будет выведено not equal На экран будет выведено equal

136. При выполнении следующего кода

...

int i = int.MinValue; unchecked

{

if (i != 0 && i == -i)

{

Console.WriteLine("equal");

}

else

{

Console.WriteLine("not equal");

}

}

...

Возникнет ошибка времени компиляции Возникнет ошибка времени выполнения На экран будет выведено not equal

+ На экран будет выведено equal

checked

{

const int START = 2000000000; float f = START + 50;

if (f == START)

Console.WriteLine("true"); else

Console.WriteLine("false");

}

+ На экран будет выведено true На экран будет выведено false

Возникнет ошибка времени компиляции Возникнет ошибка времени выполнения

138. При выполнении следующего кода private static int i = -(2147483648);

...

checked

{

Console.WriteLine(-i);

}

...

На экран будет выведено 2147483648 На экран будет выведено -2147483647 На экран будет выведено 0

На экран будет выведено 2147483648 На экран будет выведено 2147483647

+ Возникнет ошибка времени компиляции

unchecked

{

short s = short.MinValue; char c = (char)s; Console.WriteLine((int)c);

}

на экран будет выведено

65535

65536

0

-32767

-32768

32767

+ 32768

140. При выполнении следующего кода try

{

Console.WriteLine("Hello "); return;

}

finally

{

Console.WriteLine("Goodbye ");

}

Console.WriteLine("world!");

на экран будет выведено Hello world!

+ Hello Goodbye

Hello Goodbye world!

Goodbye world!

try

{

Console.Write("Hello "); Thread.CurrentThread.Abort();

}

finally

{

Console.Write("Goodbye ");

}

Console.Write("world!");

на экран будет выведено Hello world!

+ Hello Goodbye

Hello Goodbye world!

Goodbye world!

1. При выполнении следующего кода try

{

Console.WriteLine("Hello "); System.Environment.Exit(0);

}

finally

{

Console.WriteLine("Goodbye ");

}

Console.WriteLine("world!"); на экран будет выведено

+ Hello

Hello Goodbye world!

Hello world!

Goodbye world!

1. Является ли следующая функция компилируемой?

public int Foo(bool b)

{

throw new Exception("Some Exception");

}

+ Да

Нет, поскольку нет возвращаемого из функции значения Нет, поскольку генерируется неправильное исключение

1. При выполнении следующего кода public static int[] getArray() { return null; } private static void Main(string[] args)

{

int index = 1; try

{

getArray()[index = 2]++;

}

catch (Exception)

{

//empty catch

}

Console.WriteLine("index = {0}", index);

}

на экран будет выведено

index = 0

index = 1

+ index = 2

index = 3

1. При выполнении следующего кода

private static void Main(string[] args)

{

try

{

int[] a = null;

int i = a[m1(true)];

}

catch (Exception e)

{

Console.WriteLine(e.GetType());

}

}

public static int m1(bool b)

{

if (b)

throw new Exception("Some Exception"); return 0;

}

на экран будет выведено

Some Exception System.NullReferenceException

+ System.Exception

Возникнет ошибка времени компиляции Возникнет ошибка времени выполнения

1. Отметьте все строки, которые могут быть выведены на экран после выполнения следующего кода:

using System; using System.Text;

internal class Test

{

private static void Main(string[] args)

{

StringBuilder word = null; int i=0;

if (args.Length>0) try

{

i = Int32.Parse(args[0]);

}

catch

{

i = 1;

}

switch (i)

{

case 1:

word = new StringBuilder('P'); break;

case 2:

word = new StringBuilder('G'); break;

default:

word = new StringBuilder('M'); break;

}

word.Append('a');

word.Append('i');

word.Append('n'); Console.WriteLine(word);

}

}

Main Pain Gain

+ ain niaP niaG niaM

1. Отметьте все строки, которые могут быть выведены на экран после выполнения следующего кода:

using System; using System.Text;

internal class Test

{

private static void Main(string[] args)

{

StringBuilder word = null; int i = 0;

if (args.Length > 0)

{

try

{

i = Int32.Parse(args[0]);

}

catch

{

i = 1;

}

}

switch (i)

{

case 1:

word = new StringBuilder("P"); break;

case 2:

word = new StringBuilder("G"); break;

default:

word = new StringBuilder("M"); break;

}

word.Append('a');

word.Append('i');

word.Append('n'); Console.WriteLine(word);

}

}

+ Main

+ Pain

+ Gain niaP niaG niaM ain

1. Исключения имеют типы, в конечном счете являющиеся производными от

+ System.Exception System System.Throw System.Try

1. При попытке передать в метод аргумент неверного формата генерируется исключение ОverFlowException

+ FormatException InvalidCastException ArrayTypeMismatchException IndexOutOfRangeException

1. При несовместимости типа сохраняемого значения с типом массива генерируется исключение

ОverFlowException FormatException InvalidCastException

+ ArrayTypeMismatchException IndexOutOfRangeException

1. При ошибке преобразования типа генерируется исключение ОverFlowException

FormatException

+ InvalidCastException ArrayTypeMismatchException IndexOutOfRangeException

1. При арифметическом переполнении генерируется исключение

+ ОverFlowException FormatException InvalidCastException ArrayTypeMismatchException IndexOutOfRangeException

double result = 0.0; string counter = "5,5"; try

{

Console.WriteLine(Int32.Parse(counter));

}

catch(FormatException)

{

result += Double.Parse(counter) \* 2;

}

finally

{

Console.WriteLine(result);

}

на экран будет выведено

5.5

11

11

5.5

+ 11

Возникнет ошибка времени компиляции Возникнет ошибка времени выполнения

public class A

{

private int counter; public int Counter {

get { return counter; } set

{

if (value > 10)

{

throw new Exception("Error in property");

}

else

counter = value;

}

}

}

class Program

{

static void Main(string[] args)

{

A a = new A(); try

{

a.Counter = 15;

}

catch(Exception) when (a.Counter < 10)

{

Console.WriteLine("Error with when");

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

}

}

на экран будет выведено

Error in property Error with when

Error in property

+ Error with when

Возникнет ошибка времени компиляции Возникнет ошибка времени выполнения

public class A

{

private int counter; public int Counter {

get { return counter; } set

{

if (value > 10)

{

throw new Exception("Error in property");

}

else

counter = value;

}

}

}

class Program

{

static void Main(string[] args)

{

A a = new A(); try

{

a.Counter = 15;

}

catch(Exception) when (a.Counter > 10)

{

Console.WriteLine("Error with when");

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

}

}

на экран будет выведено

+ Error in property Error with when

Error in property Error with when

Возникнет ошибка времени компиляции Возникнет ошибка времени выполнения

+ Да Нет

156. Что будет выведено на экран при выполнении следующего кода? int a = 7;

try

{

a++;

}

catch(Exception) when (a < 10)

{

Console.WriteLine("Error with when");

}

+Ничего не будет выведено 7

8

10

Error with when

int a = 9; try

{

a++;

}

catch(Exception) when (a < 10)

{

Console.WriteLine("Error with when");

}

finally

{

Console.WriteLine(a);

}

на экран будет выведено

Error with when Error with when 10

+ 10

Возникнет ошибка времени компиляции Возникнет ошибка времени выполнения

158. Какое значение хранит переменная result после выполнения кода private static int Methods(int i, int j)

{

return (j - i) \* i;

}

...

int i = 1; int j = 1;

int result = Methods(i++, i+j);

...

0

1

+ 2

private static void Method(int i = 1, string s = "A", double d = 0.00001)

{

Console.WriteLine("i = {0}, s = {1}, d = {2}", i, s, d);

}

его вызов в коде

private static int n = 0;

...

Method(n++, n++.ToString());

...

равносилен следующему вызову

+ Do(0, "1",0.00001);

Do(0,"0",0.00001);

Do(1,"1",0.00001);

1. Для следующего метода

private static void Method(int x = 9, string s = "A", DateTime dt = default(DateTime))

{

Console.WriteLine("x = {0}, s = {1}, dt = {2}", x, s, dt);

}

его вызов в коде

private static int n = 0;

...

Method(s : (n++).ToString(), x : n++);

...

равносилен следующему вызову

+ Do(1,"0",default(DataTime));

Do(0,"1",default(DataTime));

Do(0,"0",default(DataTime));

Do(1,"1",default(DataTime));

1. Выберите спецификации методов, в списке параметров которых есть ошибки private static void Method(int a, DateTime dt = default(DateTime));

+ private static void Method(int a, DateTime dt = DateTime.Now);

+ private static void Method(int a : 4, DateTime dt = default(DateTime)); private static void Method(int a = 4, DateTime dt = new DataTime());

private static int AddArray(int[] values)

{

int sum = 0;

for (int x = 0; x &lt; values.Length; x++) sum += values[x];

return sum;

}

указать операторы, которые не приведут к ошибке времени компиляции.

+ int[] arr = new int[] { 1, 2, 3, 4, 5 }; Console.WriteLine(AddArray(arr));

+ Console.WriteLine(AddArray(new int[] { 1, 2, 3, 4, 5 }));

Console.WriteLine(AddArray(1, 2, 3, 4));

1. Для следующего метода

private static int Add(params int[] values)

{

int sum = 0;

for (int x = 0; x &lt; values.Length; x++) sum += values[x];

return sum;

}

указать операторы, которые не приведут к ошибке времени компиляции.

+int[] arr = new int[] { 1, 2, 3, 4, 5 }; Console.WriteLine(Add(arr));

+Console.WriteLine(Add(new int[] { 1, 2, 3, 4, 5 }));

+ Console.WriteLine(Add(1, 2, 3, 4, 5));

1. После выполнения следующего кода значения переменных value, str1, str2 будут равны соответственно

class OutReturnExample

{

static void Method(out int i, out string s1, out string s2)

{

i = 44;

s1 = "I've been returned"; s2 += "Hi!";

}

static void Main()

{

int value = 40; string str1, str2;

Method(out value, out str1, out str2);

}

}

44, "I've been returned", null 40, "I've been returned", null 44, "I've been returned", "Hi!"

+ Ошибка компиляции

1. Выберите объявления методов с правильным использованием ключевого слова params: public void Method (params object[] objectArray, params int[] intArray) { }

+ public void Method (params double[] array) { }

public void Method (params double[] array, double valume) { }

+ public void Method (double volume, params double[] array) { }

1. Выберете правильные утверждения

+ Следующая перегрузка методов возможна:

public void Method(int i) { }

public void Method(out int i) { i = 5; } Следующая перегрузка методов возможна:

public void Method(ref int i) { i = 6; } public void Method(out int i) { i = 5; }

Следующая перегрузка методов не возможна:

private static int Add(params int[] values) private static int Add(int[] values)

+ Следующая перегрузка методов не возможна:

private static int Add(params int[] values)

private static int Add(int first, int second, int third)

+ Следующая перегрузка методов не возможна:

private static int Add(params int[] values)

private static int Add(int first, int second, int third)

1. Какое значение хранит переменная result после выполнения кода private static int Methods(ref int i, int j)

{

}

...

i++;

return (j - i) \* i;

int i = 1; int j = 1;

int result = Methods(ref i, i+++j)+i;

...

+ 0

1

2

1. Что будет выведено после выполнения следующего кода

class OutReturnExample

{

int x=20; double y=3;

public double Method(int x, int y)

{

return this.x-y/2;

}

public static void Main(string[] args)

{

OutReturnExample a = new OutReturnExample(); Console.WriteLine(a.Method(7,9));

}

}

+ 16

15,5

3

2,5

18,5

Ошибка компиляции

1. Что будет выведено после выполнения следующего кода class c1

{

int x=20; double y=3;

public static double Method(int x, int y)

{

return this.x-y/2;

}

public static void Main(string[] args)

{

c1 a = new c1(); Console.WriteLine(a.Method(7,9));

}

}

16

15,5

3

2,5

18

+ Ошибка компиляции

1. Что будет выведено после выполнения следующего кода

class c1

{

public static void Method(ref int x, ref float y)

{

x\*=x; y\*=2;

}

public static void Main(string[] args)

{

int x=2, y=3; float d=3; Method(ref x, ref y); Console.WriteLine(x+” “+y+d);

}

}

4 63

4 9

+ Ошибка компиляции

1. Что будет выведено после выполнения следующего кода class c1

{

public static int Method(int x)

{

return (x>0)?x%10+Method(x/10):0;

}

public static void Main(string[] args)

{

Console.WriteLine(Method(12345));

}

}

+ 15

Бесконечное выполнение Ошибка компиляции

0

12345

static void Method(out int value, ref int j)

{

value = 10;

j += 5;

}

static void Main(string[] args)

{

int value, j = 5; Method(out value, ref j);

Console.WriteLine(value +++ j); Console.WriteLine(value + j);

}

на экран будет выведено

+ 20

21 21

21

Возникнет ошибка времени компиляции Возникнет ошибка времени выполнения 6

6

173. При выполнении следующего кода static void Method(out int value, ref int j)

{

value = 10;

j = 5;

}

static void Main(string[] args)

{

int value; int j;

Method(out value, ref j); Console.WriteLine(value + j);

}

на экран будет выведено

15

10

+ Возникнет ошибка времени компиляции Возникнет ошибка времени выполнения 5

static void Method(ref int value)

{

if (value++ < 100) Method(ref value);

}

static void Main(string[] args)

{

int value = 1; Method(ref value);

Console.WriteLine(value);

}

на экран будет выведено

100

+ 101

Возникнет ошибка времени компиляции Возникнет ошибка времени выполнения 99

175. При выполнении следующего кода static class StringAdditional

{

public static bool ToInt(this String s, out int result)

{

return int.TryParse(s, out result);

}

}

class Program

{

static void Main(string[] args)

{

string s = "54"; int result = 0;

s.ToInt(out result); Console.WriteLine(result);

}

}

на экран будет выведено

+ 54

0

Возникнет ошибка времени компиляции Возникнет ошибка времени выполнения

static void Method(ref int value)

{

if (++value < 100) Method(ref value);

}

static void Main(string[] args)

{

int value = 1; Method(ref value);

Console.WriteLine(value);

}

на экран будет выведено

+ 100

101

Возникнет ошибка времени компиляции Возникнет ошибка времени выполнения 99

177. При выполнении следующего кода, что будет выведено на экран static class A

{

internal static int ToInt(this string s)

{ return Convert.ToInt32(s); }

}

class Program

{

static void Main(string[] args)

{

int x = 0;

string str=(++x).ToString(); x += str.ToInt(); Console.WriteLine(x++);

}

}

0

1

+ 2

3

4

5

Возникнет ошибка времени исполнения Возникнет ошибка времени компиляции

static class A

{

internal static int ToInt(this string s)

{ return Convert.ToInt32(s); }

}

class Program

{

static void Main(string[] args)

{

int x = 1;

string str=(++x).ToString(); x += str.ToInt(); Console.WriteLine(x++);

}

}

0

1

2

3

+ 4

5

Возникнет ошибка времени исполнения Возникнет ошибка времени компиляции

class Program

{

static class A

{

internal static int ToInt(this string s)

{ return Convert.ToInt32(s); }

}

static void Main(string[] args)

{

int x = 0;

string str=(++x).ToString(); x += str.ToInt(); Console.WriteLine(x++);

}

}

0

1

2

3

4

5

Возникнет ошибка времени исполнения

+ Возникнет ошибка времени компиляции

static class A

{

internal static int ToInt(this string s, int i)

{ return i++; }

}

class Program

{

static void Main(string[] args)

{

int x = 0;

string str=(++x).ToString(); x += str.ToInt(x); Console.WriteLine(x++);

}

}

0

1

+ 2

3

4

5

Возникнет ошибка времени исполнения Возникнет ошибка времени компиляции

# Основные программные конструкции C#

1. При выполнении следующего кода class Program

{

static bool Method1()

{

Console.WriteLine("in Method1"); return false;

}

static bool Method2()

{

Console.WriteLine("in Method2"); return true;

}

static void Main(string[] args)

{

if (Method1() & Method2())

{

Console.WriteLine("insite statement if");

}

}

}

на экран будет выведено

in Method1

+ in Method1

in Method2 in Method1 in Method2

inside statement if

1. Что произойдет в результате выполнения кода char[] s = new char[]{ 'a', 'b', 'c', 'd', 'e' };

for (int i = 1; i &lt;= s.Length; i++)

{

Console.Write(s[i]);

}

На экран будет выведено abcde На экран будет выведено bcde

+ На экран будет выведено bcde, а дальше произойдет ошибка времени выполнения Возникнет ошибка компиляции

1. Сколько объектов типа string будет создано во время выполнения следующего кода

string s = "sdfgh";

if (s.ToLower().Substring(1, 2).EndsWith("exe"));

Нет правильных ответов 2

+ 3

4

1. Сколько раз выполнится цикл while int i = 0;

while(++i < 3)

{

Console.WriteLine(i);

}

+ 2

3

1

0

1. При выполнении следующего кода

static void Main(string[] args)

{

int a = 0; decimal b = 0; double c = 0.0;

Console.WriteLine(a == b); Console.WriteLine(a.Equals(b)); Console.WriteLine(a == c); Console.WriteLine(a.Equals(c));

}

False True False False True True True False

+ True

False

True

False True True True True

1. Что произойдет в результате выполнения кода static void Main(string[] args)

{

string s;

{

int s = 10; Console.WriteLine(s);

}

}

Возникнет ошибка времени выполнения. На экран будет выведено значение 10.

На экран будет выведена пустая строка.

+ Возникнет ошибка времени компиляции. На экран будет выведено значение null.

1. Как могут инициализироваться readonly поля экземпляра класса?

+ при объявлении; в instance методах;

с помощью свойств; в static методах;

+ в конструкторе

1. Что произойдет в результате выполнения кода скомпилируется ли данный код static void Main(string [] args)

{

int a = 10; int b = 20;

byte c = a + b;

Console.WriteLine(c);

}

На экран будет выведено значение 20.

+ Возникнет ошибка времени компиляции. Возникнет ошибка времени выполнения.

1. Скомпилируется ли успешно данный фрагмент кода? private int GetID(string inputText)

{

if (inputText != "") return 1;

else if (inputText == "") return 0;

}

Да

+ Нет

1. Укажите строки, которые вызовут ошибку времени компиляции
2. Type t1 = typeof(System.Int32);
3. Type t2 = Type.GetType("System.Int32");
4. Type t3 = Type.GetType(System.Int32);
5. Type t4 = System.Int32.GetType();

1

2

+ 3

+ 4

1. При выполнении следующего кода

public static void Main(string[] args)

{

int i, j, s = 0;

for (i = 0, j = 6; i < j; i++, --j)

{

s += i;

}

Console.WriteLine(s);

}

на экран будет выведено

2

+ 3

6

21

1. При выполнении следующего кода static bool Method1()

{

Console.WriteLine("in Method1"); return false;

}

static bool Method2()

{

Console.WriteLine("in Method2"); return true;

}

static void Main(string[] args)

{

if (Method1() && Method2())

{

Console.WriteLine("insite statement if");

}

}

на экран будет выведено

+ in Method1 in Method1 in Method2 in Method1 in Method2

inside statement if

1. Укажите строки, в которых неявно типизированные переменные использованы корректно

public var Add(int x, int y) {return x + y};

+ static int GetValue(){var t = 9; return t;} public void Do(var t) {Console.WriteLine(t);};

+ var ok = true; var data;

var data = null;

+ var data = (int?)null;

+ var i = 0; var j = i;

+ string s = “Hi”; var data = s; var? data = 12;

1. Укажите все корректные способы объявления массива в C#. []int x;

int x[]; int[10] x[];

+ int[][] x;

+ int[,] x;

+ int[] x;

1. При выполнении следующего кода public static void Main(string[] args)

{

int a = 101; Console.WriteLine(a>>2);

}

на экран будет выведено

ошибка времени компиляции ошибка времени исполнения

+ 25

50

1

101

1. При выполнении следующего кода

public static void Main(string[] args)

{

int a = 2; Console.WriteLine((++a)&(a--));

}

на экран будет выведено

ошибка времени компиляции 10

2

ошибка времени исполнения

+ 3

1

1. При выполнении следующего кода static bool Method1(int i)

{

return (i % 2 == 0) ? true : false;

}

public static void Main(string[] args)

{

int a = 2; Console.WriteLine(Method1(++a\*2)); Console.WriteLine(Method1(++a+1)); Console.WriteLine(Method1(a+++2));

}

на экран будет выведено

False False False True True False

+ True

False

True True True True

1. Сколько раз выполнится цикл while

public static void Main(string[] args)

{

int count = 3; do

{

Console.WriteLine("Hello!");

} while (count--> 0);

}

+ 4

3

Возникнет ошибка времени компиляции 0

1. Сколько раз выполнится цикл while public static void Main(string[] args)

{

const int COUNT = 3; do

{

Console.WriteLine("Hello!");

} while (COUNT --> 0);

}

4

3

+ Возникнет ошибка времени компиляции 0

1. При выполнении следующего кода

public static void Main(string[] args)

{

object[] s = new object[]{ "Hello", 34, true }; foreach (var item in s)

{

if (item is Int32)

{

Console.WriteLine(item+1);

}

}

}

на экран будет выведено

0

35

+ Возникнет ошибка времени компиляции Возникнет ошибка времени выполнения 34

1. При выполнении следующего кода public static void Main(string[] args)

{

dynamic s = new object[]{ "Hello", 34, true }; foreach (var item in s)

{

if (item is Int32)

{

Console.WriteLine(item+1);

}

}

}

на экран будет выведено

0

+ 35

Возникнет ошибка времени компиляции Возникнет ошибка времени выполнения 34

1. При выполнении следующего кода

public static void Main(string[] args)

{

dynamic s = new object[]{ "Hello", 34, true }; foreach (var item in s)

{

if (item is Boolean)

{

Console.WriteLine(item+1);

}

}

}

на экран будет выведено

0

35

Возникнет ошибка времени компиляции

+ Возникнет ошибка времени выполнения 34

1. При выполнении следующего кода class Test

{

static void Main(string[] args)

{

int i = 1;

Console.WriteLine("i = {0}", i++);

}

}

на экран будет выведено

+ i = 1

i = 2

i = {0}1

i = {0}2

1. При выполнении следующего кода

public static void Main(string[] args)

{

dynamic s = new object[]{ "Hello", 34, true }; foreach (var item in s)

{

if (item is String)

{

Console.WriteLine(item+1);

}

}

}

на экран будет выведено

Hell Hello ello

Возникнет ошибка времени компиляции Возникнет ошибка времени выполнения

+ Hello1

1. При выполнении следующего кода public static void Main(string[] args)

{

int? x = null; int? y = x ?? -1;

Console.WriteLine(y??10);

}

на экран будет выведено

null 10

Возникнет ошибка времени компиляции Возникнет ошибка времени выполнения

+ -1

1. Что будет выведено на экран в результате выполнения кода, если бы сейчас на часах было 10:00?

int h =DateTime.Now.Hour;Console.WriteLine((h &gt; 12) &amp;&amp; (h &lt; 16) ? "Добрый день" : (h &gt; 7) &amp;&amp; (h &lt; 12) ? "Добрый вечер" : "Доброе утро");

Будет ошибка времени компиляции

На экран будет выведено: Добрый день

+ На экран будет выведено: Добрый вечер На экран будет выведено: Доброе утро

1. Что будет выведено на экран в результате выполнения кода:

int x = new int();while (!(--x==0)){ Console.Write(x+= x == 0 ? 0 : 1);} Будет ошибка времени компиляции

На экран будет выведено: x+= x == 0 ? 0 : 1

+ На экран будет выведено: 0000000000... и программа зациклится На экран будет выведено: 0

На экран будет выведено: 11111111111... и программа зациклится На экран будет выведено: -10101010101... и программа зациклится На экран будет выведено: -10

1. Укажите все способы объявления массива в результате которых он будет корректно инициализирован

int[4] someArray;

+ int[] someArray = new int[4];

+ int[] someArray = new int[] {1,2,3,4};

+ int[] someArray = {1,2,3,4};

+ int[] someArray = new int[4] {1,2,3,4}; int someArray[] = new int[4];

+ var b = new[] { "hello", null, "world" };

public static void Main(string[] args)

{

goto F2;

F1: Console.WriteLine("Before F2 "); F2: Console.WriteLine("After F2");

goto F1;

Console.WriteLine("Done");

}

на экран будет выведено

Before F2 After F2 Before F2 After F2 Done Before F2 After F2 Done After F2 Before F2

+ After F2

Before F2

…до бесконечности Before F2

After F2

…до бесконечности

210. Какие операторы выбора являются эквивалентом тернарной операции «?:» ? foreach

if

+ if...else switch case

bool operation1 = true; bool operation2 = false; bool operation3 = false;

if(operation1 ^ operation2 ^ operation3)

{

Console.WriteLine(operation1 ^ operation2); if (operation1 | operation2 | operation3)

{

Console.WriteLine(operation2 | operation3); if(operation1 & operation2 & operation3)

{

Console.WriteLine(operation1 & operation3);

}

}

}

на экран будет выведено

True False False True True True

+ True

False True False True False False False

public static void Main()

{

Int32 x = 4; Object o = x; x = 123;

Console.WriteLine(x + ", " + (Int32)o);

}

на экран будет выведено

+ 123, 4

123, 123

4, 4

Возникнет ошибка приведения типов

1. В какой кодировке хранятся символьные (char) переменные в C#?

Win1252 UTF-4

+ UTF-16

Windows1251 KOI-8

# Работа с файловой системой

1. Какую информацию будет хранить текстовый файл TextFile.txt в результате выполнения следующего кода

string filePath = "TextFile.txt";

string[] fileLines = { "One ", "Two", "Three " }; File.AppendAllLines(filePath, fileLines);

при условии, что до выполнения кода такой файл не существовал на диске Возникнет ошибка времени выполнения

One Two Three

+ One

Two

Three

1. Выберете правильные утверждения. Следующий конструктор класса FileStream FileStream fs = new FileStream(name, FileMode.Create, FileAccess.Read);

Указывает, что операционная система должна создать новый файл с именем name в режиме доступа только для чтения. Если файл уже существует, будет сгенерировано исключение IOException.

+ Указывает, что операционная система должна создать новый файл с именем name в режиме доступа только для чтения. Если файл уже существует, он будет переписан.

Не существует.

1. Выберете правильные утверждения. Следующий конструктор класса FileStream FileStream fs = new FileStream(name, FileMode.OpenOrCreate, FileAccess.Read);

+ Указывает, что операционная система должна создать новый файл с именем name в режиме доступа только для чтения. Если файл уже существует, будет сгенерировано исключение IOException.

Указывает, что операционная система должна создать новый файл с именем name в режиме доступа только для чтения. Если файл уже существует, он будет переписан.

Не существует.

1. Выберете правильные утверждения. Следующий конструктор класса FileStream FileStream fs = new FileStream(name, FileMode.Append, FileAccess.Write);

Указывает, что операционная система должна открыть файл с именем name в режиме доступа только для записи. Если файл не существует, будет сгенерировано исключение IOException.

+ Указывает, что операционная система должна открыть файл с именем name в режиме доступа только для записи и добавить данные в конец файла. Если файл не существует, то он создается.

Не существует.

1. Перечисление FileAccess определяет следующие константы: Read

Write

+ Read

Write

ReadWrite Read

Write ReadOrWrite

1. Что произойдет в результате выполнения следующего кода (папка d:\\ds существует и она не пуста)?

static void Main()

{

Directory.Delete("d:\\ds", true);

}

удалится все содержимое папки ds удалятся все файлы в папке ds

+ удалится папка ds и все её содержимое удалятся все папки в папке ds возникнет ошибка выполнения

1. Что произойдет в результате выполнения следующего кода (папка d:\\ds существует и она не пуста)?

static void Main()

{

Directory.Delete("d:\\ds");

}

удалится все содержимое папки ds удалятся все файлы в папке ds удалится папка ds и все её содержимое удалятся все папки в папке ds

+ возникнет ошибка выполнения

1. Класс Directory из пространства имен

+ System.IO System System.Steam System.File System.Directory System.Text

1. Для использования кодировок подключается пространство имен System.IO

System System.Steam System.File System.Directory

+ System.Text

1. Что будет храниться в файле in.txt после выполнения следующего кода static void Main()

{

FileStream file1 = new FileStream("d:\\in.txt", FileMode.Create); StreamWriter writer = new StreamWriter(file1); writer.Write("текст");

}

текст

+ файл будет пустым ошибка компиляции

1. Можно ли узнать время последнего изменения файла?

+ Да Нет

1. При помощи каких методов класса File можно узнать время последнего изменения файла?

+ File.GetLastWriteTime();

+ File.GetLastWriteTimeUTC(); File.SetLastWriteTime(); File.GetLastAccessTime();

1. При помощи каких методов класса File можно узнать время создания файла?

+ File.GetCreationTime();

+ File.GetCreationTimeUTC(); File.GetLastWriteTime(); File.GetLastAccessTime();

1. Что будет храниться в файле in.txt после выполнения следующего кода static void Main()

{

using (StreamWriter file = new StreamWriter("in.txt"))

{

file.WriteLine("Hello world");

}

}

+ Hello world

файл будет пустым ошибка компиляции

1. При помощи каких методов класса Directory можно получить все файлы указанной папки

+ Directory.EnumerateFiles("e:\\");

+ Directory.GetFiles("e:\\"); Directory.EnumerateFileSystemEntries("e:\\"); Directory.GetFileSystemEntries("e:\\");

# Создание новых типов в C#

1. Определена переменная c

Color c = Color.Blue; где

enum Color{ White, Red, Green, Blue, Orange};

При выполнении какого оператора на консоль будет выведено целочисленное значение 3. Console.WriteLine(c);

Console.WriteLine(c.ToString());

+ Console.WriteLine(c.ToString("D")); Console.WriteLine(c.ToString("G"));

1. В перечислении Season

enum Season : sbyte {Spring = -3, Summer, Fall, Winter}; литерал Winter равен

-1

+ 0

Перечисление не имеет смысла

1. Если при определении класса не определить конструктор по умолчанию, то

+ Конструктор по умолчанию генерируется автоматически, только если класс не имеет других конструкторов.

Конструктор по умолчанию для класса автоматически сгенерируется в любом случае. Код не скомпилируется.

1. Что произойдет, если при определении разделяемого метода, не обеспечить его реализацию?

Код не будет компилироваться.

Код будет компилироваться, но не будет работать.

+ Код будет компилироваться и работать, и можно будет вызвать разделяемый метод.

1. Результатом выполнения следующего кода будет

struct Book

{

public string title; public string author; public int copyrigth;

public Book()

{

author = "Author"; title = "Title"; copyrigth = 1000;

Console.WriteLine("This is a book.");

}

}

class Program

{

static void Main(string[] args)

{

Book book = new Book(); Console.WriteLine("This is not a book.");

}

}

This is a book.

This is not a book.

This is a book.

+ Ошибка времени компиляции. Ошибка времени выполнения.

1. Отметьте операторы, которые генерируют идентичный IL код

+ int a = new int();

+ int b = 0; Int16 с = 0;

+ System.Int32 d = new System.Int32();

+ Int32 e = 0; Byte f = 0;

public class Residence

{

public ResidenceType type; public int numberOfBedrooms;

public Residence(ResidenceType type, int numberOfBedrooms)

{

this.type = type;

this.numberOfBedrooms = numberOfBedrooms;

}

}

какое сообщение будет выведено на консоль при выполнении следующего кода?

Residence myHouse = new Residence(ResidenceType.House, 2); Residence anotherHouse = new Residence(ResidenceType.House, 2); if (myHouse == anotherHouse)

{

Console.WriteLine("They are the same house");

}

else

{

Console.WriteLine("They are different houses");

}

They are the same house

+ They are different houses

Код не будет компилироваться

public struct Residence

{

public ResidenceType type; public int numberOfBedrooms;

public Residence(ResidenceType type, int numberOfBedrooms)

{

this.type = type;

this.numberOfBedrooms = numberOfBedrooms;

}

}

какое сообщение будет выведено на экран при выполнении следующего кода?

Residence myHouse = new Residence(ResidenceType.House, 4); Residence anotherHouse = new Residence(ResidenceType.House, 4); if (myHouse == anotherHouse)

{

Console.WriteLine("They are the same house");

}

else

{

Console.WriteLine("They are different houses");

}

They are the same house They are different houses

+ Код не будет компилироваться

public override String ToString()

{

return String.Format("({0}, {1})", x, y);

}

public Object Clone()

{

return MemberwiseClone();

}

}

создан экземпляр структуры Point p;

и инициализированы ее поля

p.x = 10;

p.y = 20;

Отметьте строки, в которых происходит упаковка. Console.WriteLine(p.ToString());

+ Console.WriteLine(p.GetType());

+ ICloneable c = p;

Point p2 = (Point)p.Clone();

1. Отметьте правильные утверждения

+ Перечисления можно объявить в классе;

+ Перечисления можно объявить в пространстве имен; Перечисления можно объявить в методе.

1. Для какого типа используется Nullable &lt;T&gt?

Ссылочные типы (reference types) Примитивные типы

+ Типы значения (value types) Кассы и структуры

Только int, bool, и DateTime

1. Что из перечисленного верно

+ Переменные ссылочного типа (reference type) размещаются в стеке потока и хранят адрес объекта в управляемой куче.

Переменные ссылочного типа (reference types) размещаются в управляемой куче. Переменные значимого типа (value type) размещаются в неуправляемой куче.

+ Переменные значимого типа (value type) размещаются в стеке потока.

Переменные ссылочного типа (reference types) размещаются в стеке потока и хранят адрес объекта в неуправляемой куче.

1. Отметьте все верные утверждения о классе SomeClass: internal class SomeClass()

{

public void SomeClass

{

}

}

SomeClass является значимым типом (value-type). SomeClass может использоваться внешними сборками.

+ SomeClass наследуется от System.Object.

экземпляр SomeClass нельзя создать вызовом new SomeClass(), поскольку в классе не определен открытый конструктор без параметров.

+ SomeClass доступен для использования только внутри сборки, в которой он определён.

1. Что будет выведено после выполнения следующего кода class c1

{

public static void Main(string[] args)

{

Console.WriteLine(Byte.MaxValue+1);

}

}

+ 256

0

Ошибка компиляции Ошибка выполнения

-127

byte a = Byte.MaxValue+1; Console.WriteLine(a);

}

}

256

0

+ Ошибка компиляции Ошибка выполнения

-127

244. Что будет выведено после выполнения следующего кода class c1

{

public static void Main(string[] args)

{

object a = 2; object b = a; a=3;

Console.WriteLine(b);

}

}

3

+ 2

Ошибка компиляции

int[] a = { 1, 2, 3, 4, 5 };

int[] b = a;

a[0] = 3;

foreach(int i in b) Console.Write(i+" ");

}

}

+ 3 2 3 4 5

1 2 3 4 5

Ошибка компиляции

246. Что будет выведено после выполнения следующего кода class c1

{

public static void Main(string[] args)

{

Type type = 12.GetType(); Console.WriteLine(type);

}

}

+ System.Int32 System.byte 12

Ошибка компиляции

public class A

{

public static int Counter { get; private set; } static A()

{

Counter = 10;

}

public A(int counter)

{

Counter = counter;

}

}

class Program

{

static void Main(string[] args)

{

Console.WriteLine(A.Counter); A a = new A(20); Console.WriteLine(A.Counter); Console.ReadKey();

}

}

+ 10

20 10

10

Ошибка времени компиляции. Ошибка времени выполнения.

static class A

{

public static int Counter { get; private set; } static A(int counter)

{

Counter = counter;

}

}

class Program

{

static void Main(string[] args)

{

A a = new A(20); Console.WriteLine(A.Counter); Console.ReadKey();

}

}

20

0

+ Ошибка времени компиляции. Ошибка времени выполнения.

class A

{

public static int Counter { get; private set; } private A()

{

Counter = 20;

}

}

class Program

{

static void Main(string[] args)

{

Console.WriteLine(A.Counter); Console.ReadKey();

}

}

20

+ 0

Ошибка времени компиляции. Ошибка времени выполнения.

class A

{

public static int Counter { get; private set; } static A()

{

new A();

}

private A()

{

Counter = 20;

}

}

class Program

{

static void Main(string[] args)

{

Console.WriteLine(A.Counter); Console.ReadKey();

}

}

+ 20

0

Ошибка времени компиляции. Ошибка времени выполнения.

struct A

{

public int x;

public string ToString()

{

return Convert.ToString(x++);

}

}

static void Main(string[] args)

{

A a = new A

{

x = 3

};

Console.WriteLine(a.ToString());

}

}

Ошибка времени исполнения Ошибка времени компиляции 0

1

2

+ 3

4

}

static void Main(string[] args)

{

A a = new A

{

x = 3

};

Console.WriteLine(a.ToString());

}

}

Ошибка времени исполнения

+ Ошибка времени компиляции 0

1

2

3

4

public string ToString()

{

return Convert.ToString(x++);

}

}

static void Main(string[] args)

{

A a = new A

{

x = 3

};

Console.WriteLine(a.ToString());

}

}

Ошибка времени исполнения Ошибка времени компиляции 0

1

2

+ 3

4

public string ToString()

{

return Convert.ToString(x++);

}

}

static void Main(string[] args)

{

A a = new A

{

x = 3

};

Console.WriteLine(a.ToString());

}

}

Ошибка времени исполнения

+ Ошибка времени компиляции 0

1

2

3

4

# Управление ресурсами в .NET. Сборка мусора

1. Какой метод вызывается при реализации шаблона Dispose, чтобы предотвратить выполнение метода Finalize?

GC.NoFinalize System.NoFinalization GC.SuppressFinalization

+ GC.SuppressFinalize GC.NoFinalization

1. Отметьте верные утверждения, характеризующие деструктор Деструкторы можно перегружать

+ Деструктор не имеет параметров. Модификатор доступа деструктора всегда public

+ В структурах определение деструкторов невозможно

1. К поколению 0 сборщика мусора относятся которые уже «пережили» один процесс сборки мусора

+ только что размещенные в куче, которые еще ни разу не помечались как подлежащие удалению в процессе сборки мусора

такого понятия не существует

1. Для оптимизации производительности сборщика мусора на какое количество поколений делиться управляемая куча?

2

5

4

+ 3

1. При каких условиях возникает сборка мусора?

+ Недостаточно физической памяти в системе.

+ Память, используемая объектами, выделенными в управляемой куче, превышает допустимый порог.

+ Вызывается метод GC.Collect. После использования объекта.

1. После инициализации средой CLR сборщик мусора выделяет сегмент памяти для хранения объектов и управления ими. Эта память называется

собственной кучей

+ управляемой кучей свободной кучей относительной кучей

1. К эфемерным относятся поколения 1 и 2

2 и 3

+ 1 и 0

1 и 3

все поколения

1. Какое поколение кучи содержит долгоживущие объекты?

-1

0

+ 2

3

4

все поколения

1. Деструктор объявляется следующим образом? public MyClass(){…}

+ ~MyClass(){…}

public void Destructor(){…}

1. В какой очередности будут вызваны деструкторы в результате выполнения следующего кода?

class A

{

}

A() { System.Diagnostics.Trace.WriteLine("Destructor of class A is called."); }

class B : A

{

~B() { System.Diagnostics.Trace.WriteLine("Destructor of class B is called."); }

}

class C : B

{

~C() { System.Diagnostics.Trace.WriteLine("Destructor of class C is called."); }

}

class Program

{

static void Main(string[] args)

{ C t = new C(); }

}

+ Destructor of class C is called. Destructor of class B is called. Destructor of class A is called. Destructor of class A is called. Destructor of class B is called. Destructor of class C is called. Destructor of class C is called. Destructor of class A is called. Destructor of class B is called.

1. В какой очередности будет выполнен следующий код?

class A

{

public A()

{

System.Diagnostics.Trace.WriteLine("Class A");

}

~A()

{

System.Diagnostics.Trace.WriteLine("Destructor of class A is called.");

}

}

class B : A

{

public B()

{

System.Diagnostics.Trace.WriteLine("Class B");

}

~B()

{

System.Diagnostics.Trace.WriteLine("Destructor of class B is called.");

}

}

class C : B

{

public C()

{

System.Diagnostics.Trace.WriteLine("Class C");

}

~C()

{

System.Diagnostics.Trace.WriteLine("Destructor of class C is called.");

}

}

class Program

{

static void Main(string[] args)

{

C t = new C();

}

}

+ Class A

Class B

Class C

Destructor of class C is called.

Destructor of class B is called.

Destructor of class A is called.

1. Как определить максимальное количество поколений кучи, которые в настоящее время поддерживает система?

+ GC.MaxGeneration GC.GetGeneration(myClass) GC.KeepAlive(myClass)

1. Как узнать текущее поколение экземпляра класса?

GC.MaxGeneration

+ GC.GetGeneration(myClass) GC.KeepAlive(myClass)