**Вопрос 1** **Назовите все верные утверждения для приведения между классами и структурами**

нельзя делать приведение от объекта ссылочного типа к объекту структурного типа

нельзя делать приведение от объекта структурного типа к объекту ссылочного типа

приведение может быть определено только в одном из типов

нельзя определять приведение между пользовательским классом и классом System.Object

приведение должно быть определено в обоих типах

нельзя определять приведения между классами, если один из них является наследником другого

**Ответ:** приведение может быть определено только в одном из типов   
нельзя определять приведение между пользовательским классом и классом System.Object   
нельзя определять приведения между классами, если один из них является наследником другого

**Вопрос 2** **Какой тип исключения корректно будет вызвать на строчке 24**

Exception

InvalidOperationException

ApplicationException

ArgumentNullException

StackOverflowException

NullReferenceExcepton

**Ответ:** ArgumentNullException

**Вопрос 3** **Что отобразится на экране**

**class A {**

**public int i=0;**

**public static A operator ++ (A a){**

**++a.i;**

**return a;**

**}**

**}**

**class Program**

**{**

**static void Main(string[] args)**

**{**

**A obj1 = new A();**

**A obj2 = new A();**

**Console.WriteLine("obj1: " + (obj1.i++) +" obj2: "+ (++obj2.i));**

**}**

**}**

не объявлена перегрузка оператора постфиксного инкремента

obj1: 0 obj2: 0

obj1: 1 obj2: 1

obj1: 1 obj2: 0

не объявлена перегрузка оператора префиксного инкремента

obj1: 0 obj2: 1

**Ответ:** obj1: 0 obj2: 1

**Вопрос 4** **выберите наиболее полный перечень ключевых слов, внутри которых можно использовать "throw new Exception();"**

try и тело метода

try, catch и finally

for и try

все перечисленные

try

тело метода, for, try

**Ответ:** все перечисленные

**Вопрос 5** **В пространстве имен System.Text есть класс StringBuilder. Вы создали пространство имен MyCompany, внутри которого создали пространство имен MyProject. В пространстве имен MyProject вы создали свой класс StringBuilder. Вы хотите написать тест производительности вашего класса, по сравнению с базовым. Выберите один или несколько вариантов, чтобы ответить на вопрос, можно ли с помощью using решить проблему конфликта имен этих классов.**

писать код во вложенном пространстве имен:

using System;

using System.Text;

namespace A{

class Program{

public static void Main(){

StringBuilder s = new StringBuilder ();

//…

}

}

}

нельзя решить проблему конфликта имен

использовать оператор приведения типа

namespace A{

class Program{

public static void Main(){

StringBuilder s =

(MyCompany.MyProject.StringBuilder)new StringBuilder ();

//…

}

}

}

использовать явное указание класса для класса приложения из BCL

namespace A{

class Program{

public static void Main(){

System.Text.StringBuilder st = new System.Text.StringBuilder ();

//…

}

}

}

использовать явное указание класса через пространство имен:

using System;

using System.Text;

namespace A{

class Program{

public static void Main(){

MyCompany.MyProject.StringBuilder s =

new MyCompany.MyProject.StringBuilder ();

//…

}

}

}

использовать псевдоним для пространства имен MyCompany.MyProject

**Ответ:** использовать явное указание класса для класса приложения из BCL   
использовать явное указание класса через пространство имен   
использовать псевдоним для пространства имен MyCompany.MyProject

**Вопрос 6** **Что отобразится на экране**

**namespace ConsoleApplication1**

**{**

**class A**

**{**

**public int i;**

**public A(int i)**

**{**

**this.i = i;**

**}**

**public static A operator \*(A a, int m)**

**{**

**return new A(a.i \* m);**

**}**

**}**

**class Program**

**{**

**static void Main(string[] args)**

**{**

**A rez = new A(14);**

**rez \*= 2;**

**Console.WriteLine(rez);**

**}**

**}**

**}**

ConsoleApplication1.A

оператор = не перегружен

неверный синтаксис перегрузки оператора \*

28

14

оператор \*= не перегружен

**Ответ:** ConsoleApplication1.A

**Вопрос 7**: **В каком случае срабатывает блок кода finally**

срабатывает, если в блоке catch используются освобождаемые ресурсы

если исключение не произошло

если произошло исключение

срабатывает, если в блоке try используются освобождаемые ресурсы

если исключение не было обработано

всегда

**Ответ:** всегда

**Вопрос 8**: **Что отобразится на экране**

**static void ThrowsMethod(){**

**try{**

**throw InvalidOperationException();**

**}finally{**

**Console.WriteLine("finally: ThrowsMethod");**

**}**

**}**

**//Main...**

**try{**

**ThrowsMethod();**

**}**

**catch(NullReferenceException){**

**Console.WriteLine("NullReference");**

**}**

**catch(FormatException f){**

**Console.WriteLine(f.ToString());**

**}**

**finally{**

**Console.WriteLine("finally: OuterMethod");**

**}**

InvalidOperationException

нельзя писать try-finally. Необходимо try-catch-finally

finally: ThrowsMethod

finally: OuterMethod

InvalidOperationException

finally: ThrowsMethod

finally: OuterMethod

неверный синтаксис около catch – нет имени переменной NullReferenceException

неверный синтаксис для оператора throw

**Ответ:** неверный синтаксис для оператора throw

**Вопрос 9**: **Сколько строк будет показано на экране**

**try{**

**int i=0;**

**for(; i<5; ++i)**

**try{**

**Console.WriteLine("I’m working!");**

**if (i==2 || i==3) throw new Exception();**

**}catch(Exception e){**

**Console.WriteLine("Exception 1");**

**}**

**//...**

**}catch(Exception ex){**

**Console.WriteLine("Exception 2");**

**}finally{**

**Console.WriteLine("Bye!");**

**}**

7

3

8

5

0

10

**Ответ:** 8

**Вопрос 10** **Какая информация содержится в свойстве InnerException объекта класса Exception**

класс Exception не содержит свойство InnerException

ничего не возвращает

возвращает объект исключения, если оно вызвано внутри объекта класса Exception, в противном случае null

возвращает объект класса InnerException

возвращает экземпляр класса Exception, вызвавший текущее исключение

на данный момент, генерирует InvalidOperationException

**Ответ:** возвращает экземпляр класса Exception, вызвавший текущее исключение

**Вопрос 11** **Каких правил необходимо придерживаться при перегрузке операторов**

Перегрузка операторов должна выполняться открытыми статическими методами класса

Пункты 1 и 2

Пункты 2 и 4

У метода-оператора тип возвращаемого значения или одного из параметров должен совпадать с типом, в котором выполняется перегрузка оператора

Пункты 1, 2 и 4

Параметры метода-оператора не должны включать ref и out

**Ответ:** Пункты 1, 2 и 4

**Вопрос 12** **Какая информация содержится в свойстве StackTrace объекта класса Exception**

строковое представление вызова элементов (методов или переменных), что привело к исключению

список методов, вызов которых привел к исключению

возвращает объект класса StackOverflowException

нет такого элемента в классе Exception

строковое представление объекта исключения

ничего не возвращает

**Ответ:** строковое представление вызова элементов (методов или переменных), что привело к исключению

**Вопрос 13** **Что отобразится на экране**

**class A {**

**public int i=0;**

**public static A operator ++ (A a){**

**++a;**

**return a;**

**}**

**}**

**class Program**

**{**

**static void Main(string[] args)**

**{**

**A obj = new A();**

**obj++;**

**Console.WriteLine(obj.i);**

**}**

**}**

0

неверный синтаксис в методе перегрузки оператора ++

StackOverflowException

неверный синтаксис инкремента obj - префиксная форма инкремента не перегружена

неверный синтаксис инкремента obj - постфиксная форма инкремента не перегружена

1

**Ответ:** StackOverflowException

**Вопрос 14** **Какой оператор следует использовать для предотвращения возникновения исключения типа OverflowException**

skipoverflow

checked

try-catch

нет такого оператора

nooverflow

unchecked

**Ответ:** unchecked

**Вопрос 15** **для чего используется оператор "throw;" (throw точка с запятой)**

для выхода из приложения в случае необрабатываемой ошибки

для генерации того же исключения, что было перехвачено catch

для вызова исключения типа Exception в catch

для выхода из блока try, если произошло необрабатываемое исключение

для перехода вверх по стеку вызовов

такой синтаксис оператора throw неверный

**Ответ:** для генерации того же исключения, что было перехвачено catch

**Вопрос 16** **Что отобразится на экране**

**namespace Namespace2 {**

**class A {**

**internal void Show() { Console.WriteLine("Hello from class A"); }**

**}**

**}**

**namespace Namespace1{**

**class Program{**

**static public void Main(){**

**Namespace2.A obj = new Namespace2.A();**

**obj.Show();**

**}**

**}**

**}**

Hello from class A

ошибка синтаксиса при объявлении переменной obj

класс A должен быть статическим, т.к. Main статический

класс А является private

неверный синтаксис для Main – пропущены принимаемые параметры

нельзя вызвать метод Show, т.к. он internal

**Ответ:** Hello from class A

**Вопрос 17** **Что произойдет, если метод ReferenceEquals, что предназначен для сравнения ссылочных типов, применить для сравнения значимых типов**

метод ReferenceEquals вернет false

InvalidOperationException

метод ReferenceEquals вернет -1

StackOverflowException

метод ReferenceEquals вернет true

ClassCastException

**Ответ:** метод ReferenceEquals вернет false

**Вопрос 18** **Что отобразится на экране**

**public class Program{**

**static int M(int v){**

**if (v > 4) throw new IndexOutOfRangeException(" value > 4 ");**

**else return v;**

**}**

**static void F(){**

**try{**

**for(int i=0; i<10; ++i) Console.Write(M(i));**

**}catch(IndexOutOfRangeException ex){**

**Console.Write(ex.Message);**

**}**

**}**

**static void Main(String []args){**

**F();**

**}**

**}**

01234 IndexOutOfRangeException

в методе M не все ветви кода возвращают значение

01234 value > 4

Выход за пределы массива

IndexOutOfRangeException

123456789

**Ответ:** 01234 value > 4

**Вопрос 19** **Что отобразится на экране**

**Console.Write(String.Format("Hello, {0}! Fun in {2}? When you come {3}?", "Anny", "China", "home"));**

FormatException

Hello, Anny! Fun in China? When you come?

NullReferenceException

ArgumentNullException

Hello, {0}! Fun in {2}? When you come {3}?

Hello, Anny! Fun in Китай? When you come home?

**Ответ:** FormatException

**Вопрос 20** **Обычно объекты классов сравниваются по ссылке. Т.е. даже если значения объектов равны, при сравнении вернется false:**

**class MyString{**

**public MyString(char[]data){**

**data\_ = data;**

**}**

**char[]data\_;**

**}**

**MyString**

**s1 = new MyString(new char[]{'c','h','a','r'}),**

**s2 = new MyString(new char[]{'c','h','a','r'});**

**Console.WriteLine(s1==s2);//false**

**Объясните почему объекты string из BCL ведут себя иначе.**

объекты string сравниваются по ссылке

CLR автоматически строковые типы сравнивает по значению

для объектов string перегружен оператор =

в BCL есть 3 класса строк: string, String и StringBuilder. Первый сравнивает по значению, а остальные по ссылке

для объектов string перегружен оператор сравнения

они входят в стандартную библиотеку классов

**Ответ:** для объектов string перегружен оператор сравнения

**Вопрос 21** **Что отобразится на экране**

**class MyInteger {**

**private int value;**

**public MyInteger(int v)**

**{**

**value = v;**

**}**

**public MyInteger()**

**{**

**}**

**public static operator int(MyInteger v1){**

**return v1.value;**

**}**

**public int Get() { return value; }**

**}**

**class Program**

**{**

**static void Main(string[] args)**

**{**

**MyInteger r = new MyInteger(-4);**

**int z = -(int)r;**

**Console.WriteLine(z);**

**}**

**}**

Неверный синтаксис перегрузки оператора преобразования – не хватает второго параметра

-4

4

Неверный синтаксис перегрузки оператора преобразования – не хватает implicit или explicit

Поле value не инициализировано в конструкторе без параметров

**Ответ:** Неверный синтаксис перегрузки оператора преобразования – не хватает implicit или explicit

**Вопрос 22** **Выберите пункт, в котором все предлагаемые операторы, можно перегрузить (перечислены через запятую).**

true, ==, ||, >=, <<

!, |, (), <<=, &

>>, %, ^=, as, \*

->, --, |=, typeof, >>

&&, ?:, +, /, new

-, ~, =, ++, []

**Ответ:** true, ==, ||, >=, <<

**Вопрос 23** **В c++ унарный оператор не принимал параметры, а в С# принимает. Объясните, почему это так**

чтобы отличить его постфиксные и префиксные формы

унарный оператор в c# является статическим методом, поэтому он должен принимать два объекта для изменения

унарный оператор в c# является статическим методом, поэтому он должен принимать один объект для изменения

это уменьшает длину программного кода

c++ унарный оператор изменял значение переменной путем изменения поля класса

в с++ можно было создавать глобальные переменные, что уменьшало к-во принимаемых параметров. В c# создавать глобальные переменные нельзя, поэтому используется такой синтаксис.

**Ответ:** унарный оператор в c# является статическим методом, поэтому он должен принимать один объект для изменения

**Вопрос 24** **Для чего используется класс ApplicationException**

служит для обработки исключений возникающих внутри CLR

для обработки в приложении исключений, возникающих в CLR

служит базовым классом для определяемых пользователем исключений

служит базовым классом для определяемых CLR исключений

нет такого класса исключения в BCL

служит базовым классом для определяемых приложением исключений

**Ответ:** служит базовым классом для определяемых приложением исключений

**Вопрос 25** **Что отобразится на экране**

**static void Main(string[] args)**

**{**

**String st = null;**

**f(st);**

**}**

**static void f(string s){**

**try{**

**Console.WriteLine(s);**

**}catch(System.IO.IOException x){**

**Console.WriteLine("Exception: IOException");**

**}catch(NullReferenceException){**

**Console.WriteLine("Exception: NullReferenceException");**

**}**

**}**

пустая строка

null

NullReferenceException

ArgumentNullException

cтрока "Exception: NullReferenceException"

строка "Exception: IOException"

**Ответ:** пустая строка

**Вопрос 26** **Выберите операторы, что должны перегружаться парами**

<= и >=

<< и >>

< и >

+= и -=

==, и !=

++ и –

**Ответ:** <= и >=   
< и >   
==, и !=

**Вопрос 27** **Что отобразится на экране (выберите все правильные варианты ответов)**

**class MyInteger {**

**private int value;**

**public MyInteger(int v) {**

**value = v;**

**}**

**public MyInteger(){ }**

**public int Get() { return value; }**

**public static MyInteger operator - (MyInteger v1, MyInteger v2) {**

**return new MyInteger(-v1.value);**

**}**

**}**

**class Program{**

**static void Main(string[] args) {**

**MyInteger r = new MyInteger(-4);**

**r = -r;**

**Console.WriteLine(r.Get());**

**}**

**}**

поле value не инициализировано для конструктора без параметров

нет нужной перегрузки оператора ‘-’ (минус)

неверный синтаксис для Console.WriteLine - передача числа. а не строки

объявлена перегрузка бинарного оператора, однако код как для перегрузки унарного

4

неверный синтаксис перегрузки оператора - (минус)

**Ответ:** нет нужной перегрузки оператора ‘-’ (минус)   
объявлена перегрузка бинарного оператора, однако код как для перегрузки унарного

**Вопрос 28** **Что отобразится на экране**

**string s = null;**

**try{**

**s = string.Format("Hello{0}! My {2} {3}",", C#", "favorite", "language!");**

**}catch(FormatException){**

**Console.Write("FormatException");**

**}catch(Exception ex){**

**Console.Write("Exception "+ex.Message);**

**}**

**finally{**

**Console.Write(" finally ");**

**}**

**Console.Write(s);**

Hello, C#! My favorite

FormatException finally

ошибка синтаксиса — нет переменной типа FormatException в catch

FormatException

NullReferenceException

Hello, C#! My favorite language!

**Ответ:** FormatException finally

**Вопрос 29** **Что отобразится на экране**

**public int Multiply (int a, int b){**

**a = a \* b;**

**if (a > 100000) throw OverflowException();**

**return a;**

**}**

**static void Main(String []args){**

**int max = int.MaxValue/4;**

**for(int i=0;i<max;++i)**

**for(int j=0;j<max;++j)**

**Console.WriteLine(Multiply(i, j)+ "");**

**}**

выведет числа на экран, однако в определенный момент вызовет OverflowException

объект OverflowException должен принимать аргументы

ошибка синтаксиса вызова исключения

нельзя вызвать метод Multiply в методе Main, т. к. он нестатический

выведет числа на экран

ошибка синтаксиса вызова исключения и нельзя вызвать метод Multiply в методе Main, т.к. он нестатический

**Ответ:** ошибка синтаксиса вызова исключения и нельзя вызвать метод Multiply в методе Main, т.к. он нестатический

**Вопрос 30** **О каких способах проверки равенства следует помнить при перегрузке операторов отношений**

равенство типов объектов в принципе

равенство типов BCL

равенство ссылок

равенство значений

равенство пользовательских типов

равенство типов классов

**Ответ:** равенство ссылок   
равенство значений