



Программирование на языке C#.фип_дСПО_ПИ_н/с

- 1 Различие платформ .NET и .NET Framework заключается в том, что ...
- 2 ... является особенностью .Net
- 3 ... называется управляемым кодом
- 4 IL – это ...
- 5 ... - особенности, характерные для языка C#
- 6 Технологии WPF, Windows Forms и ... используются для создания десктопных приложений с графическим интерфейсом
- 7 Файлы исходного кода на языке C# имеют расширение ...
- 8 Только код ... не приводит к ошибке
- 9 Литерал ... обозначает пустую ссылку
- 10 ... – это тип данных, хранящий беззнаковые числа в диапазоне [0,255]
- 11 Приведение типов ... является небезопасным
- 12 Верно, что ...
- 13 Код ... не приводит к ошибке
- 14 Программа ... позволит вывести значения переменных a, b и с в трех разных строках
- 15 В форматированной строке `Console.WriteLine(X “фамилия`
студента: {stud_name}”); на месте символа “X” должен стоять
символ ...
- 16 Метод `Console` ... позволяет вывести на экран переданное значение
и перейти на новую строку
- 17 ... – это значение, которое будет получено в ходе выполнения
выражения `double a = 2; double b = 3; double d = (a + b)%a - b;`



- 18 ... – значение, которое будет получено в ходе выполнения выражения `int a = 2; int b = 3; double d = (a + b)/a - b;`
- 19 Оператор ... позволяет найти остаток от деления
- 20 Укажите верный порядок приоритетов математических операций
- 21 При выполнении представленного ниже кода на экран будет выведено ... `int a = 7; if(a>0) { Write("a"); } if(a>5) { Write("b"); }`
- 22 При выполнении представленного ниже кода на экран будет выведено ... `int a = 7; if(a>0) { Write("a"); } else if(a>5) { Write("b"); }`
- 23 При выполнении представленного ниже кода на экран будет выведено ... `int a = 7; if(a>0) { if(a<5) { Write("a"); } } else { Write("b"); }`
- 24 При выполнении представленного ниже кода на экран будет выведено ... `int a = 7; if(a > 0) { if(a < 5) { Write("a"); } } else { Write("b"); } else { Write("b"); }`
- 25 Операторы I и II обозначают “...”
- 26 Разница между операторами = и == заключается в том, что ...
- 27 В следующем выражении будет выполнено ... операций сравнения: `a = 5; b = 3; ((a < b) && (b == 3)) && (a <= 5)`
- 28 Логическая операция ^ истина в случае, если ...
- 29 Логическая операция & истина в случае, если ...
- 30 В языке C# имеются циклы for, while, ..., do...while
- 31 Блок кода, помещенный в цикл ...
- 32 Слово “Hello” будет выведено на экран ... раз `int a=10; do { Console.WriteLine("Hello"); a--; }while(a > 0)`
- 33 При выполнении представленного ниже кода на экран будет выведено “...” `int a=10; do { Console.Write("a"); a--; }while(a > 10) ;`
- 34 Тернарные операторы используются для ...



- 35 В ходе выполнения выражения, приведённого ниже, будет получено число ... `int a=20; int b=100; int c = a * 10 < b ? (a + b) : (a - b);`
- 36 Установите верное соотношение между методами строк и их описаниями:
- 37 Для того, чтобы значение В не вычислялось в выражении A||B, значение a должно быть ...
- 38 При использовании конструкции из нескольких else if будет выполнен только блок кода с первым ... условием
- 39 Циклическая конструкция, предназначенная для перебора элементов коллекций, называется ...
- 40 Свойство ... используется для определения длины массива
- 41 Расставьте блоки кода в правильном порядке внутри приведённого ниже цикла, чтобы программа не уходила в бесконечный цикл. Изначальные значения переменных: `a=9 b=2` Цикл: `while (true) { }`
- 42 Слово void перед именем метода означает, что ...
- 43 Основной недостаток ниже приведённого метода заключается в том, что ... `void F(double m) { double f= m * 9.8; WriteLine(f); }`
- 44 Метод с приведённым кодом должен возвращать тип данных ...
`... tax(double income) { if (income > 2400000) { return 2400000 * 0.05 + (income - 2400000) * 0.13; } else { return income * 0.05; } }`
- 45 В результате выполнения кода, приведённого ниже, значение a будет равно ... `void test_method(int a) { a++; } int a=6; test_method(a)`
- 46 В результате выполнения приведённого ниже кода, значение a будет равно ... `void test_method(ref int a) { a++; } int a=6; test_method(ref a)`
- 47 Какой из вызовов соответствует методу? `void test_method(params int[] par_arr) { }`
- 48 Небезопасность использования передачи параметров по ссылке заключается в том, что ...
- 49 ... функции, определяются и вызываются только внутри какого-либо метода



- 50 ... алгоритм – это такой алгоритм, в определении которого содержится прямой или косвенный вызов этого же алгоритма.
- 51 Использование ... параметров позволяет вернуть из метода несколько значений?
- 52 Рекурсивный алгоритм ...
- 53 Верно, что ...
- 54 Верно, что ...
- 55 Верно, что ...
- 56 Из ниже перечисленных методов ... являются корректными способами вернуть два значения из метода 1 метод: void quadric(double a, double b, double c, out double x1, out double x2)
{ double D = b * b - 4 * a * c; x1 = (- b + Math.Sqrt(D)) / (2 * a); x2 = (- b - Math.Sqrt(D)) / (2 * a); } 2 метод: double quadric(double a, double b, double c)
{ double D = b * b - 4 * a * c; x1 = (- b + Math.Sqrt(D)) / (2 * a); x2 = (- b - Math.Sqrt(D)) / (2 * a); return x1, x2; } 3 метод: double[] quadric(double a, double b, double c)
{ double D = b * b - 4 * a * c;
x1 = (- b + Math.Sqrt(D)) / (2 * a); x2 = (- b - Math.Sqrt(D)) / (2 * a); double[] x = new double[2]{ x1, x2 }; return x; }
- 57 Сопоставьте ключевые слова и их назначение:
- 58 Результатом работы приведённой ниже программы будет ... int x;
Inc (ref x); WriteLine(x); void Inc (ref int x) { x++; }
- 59 Какой базовый тип данных может возвращать метод test(), чтобы он мог применяться в следующей конструкции foreach(char i in test()) { }
- 60 Для передачи параметра по ссылке используется слово ...
- 61 Расположите в правильном порядке блоки кода, чтобы получить метод, позволяющий обменять местами значения двух переменных:



- 62 Чтобы приведённый ниже метод соответствовал данной функции, требуется на место пропуска подставить ... $Fm,n = \{ n+1 \}$
 $m=0 \ Fm-1,1 \quad m>0, n=0 \ Fm-1, Fm, n-1 \quad m>0, n>0 \ \text{double}$
 $F(\text{double } m, \text{double } n) \{ \quad \text{if}(m==0) \quad \{ \quad \text{return } n+1 \quad \} \}$
 $\quad \text{if}(.....) \quad \{ \quad \text{return } Fm-1,1 \quad \} \quad \text{if}(m>0 \ \&\ n>0)$
 $\quad \{ \ F(m-1, F(m,n-1)) \quad \} \ \}$
- 63 Верно, что ...
- 64 Свойства нужны для ...
- 65 Конструктор, принимающий в качестве параметра экземпляр того же класса – это конструктор ...
- 66 Верно, что ...
- 67 Особый тип данных, представляющий собой набор однотипных именованных констант, называется ...
- 68 Метод доступа ... позволяет установить значение поля.
- 69 Поле group получит значение ... если в приведённом ниже коде будет вызван первый конструктор `Student("Иванов")`
- ```
class Student
{ public string name; public string group; public int course;
 public Student(string name) : this(name, "не определена") //
```
- первый конструктор { course=1; } public Student(string name, string group) // второй конструктор { this.name = name; this.group = group; }
- 70 Конструктор по умолчанию не может включать инициализацию полей ...
- 71 Специальный метод, предназначенный для создания экземпляров класса и инициализации их полей – это ...
- 72 Верная реализация свойства представлена как ...
- 73 ... компоненты относятся ко всему классу, и для обращения к ним не требуют создания экземпляра
- 74 Модификатор ... должен иметь закрытый компонент класса
- 75 В классе, приведённом ниже, поля ... доступны извне class `Student { public string a; public string b; string c; private string d; }`



- 76 ... - это принцип ООП, предусматривающий скрытие внутреннего устройства класса и осуществление доступа только через методы и свойства
- 77 Перегруженные методы должны иметь ...
- 78 ... является верной реализацией перегрузки оператора сложения для обыкновенных дробей
- 79 Сопоставьте реализации конструкторов и их типы:
- 80 Приведенный ниже метод называется ... `public void Deconstruct(out string a, out string b, out string c) { a = this.a; b = this.b; c = this.c; }`
- 81 Ключевое слово ... используется для создания экземпляров классов
- 82 Расположите модификаторы доступа в порядке уменьшения области видимости?
- 83 Общедоступные поля должны иметь модификатор доступа ...
- 84 Закрытые поля, доступные в наследуемых классах, должны иметь модификатор ...
- 85 Верно, что ...
- 86 Скрытие метода реализуется посредством модификатора ...
- 87 Модификатор `override` используется для ...
- 88 ... - это концепция, согласно которой сущности могут совершать одинаковые действия над объектами разных типов
- 89 ... - это концепция, согласно которой одни классы, называемые родительскими, могут лежать в основе других — дочерних. При этом, дочерние классы переименуют свойства и поведение своего родителя.
- 90 Достоинствами использования наследования являются ...
- 91 ... - максимальное количество классов-родителей у производного класса
- 92 Результатом `boxing` является ...
- 93 Базовым классом для всех классов C# является ...



- 94) Через ключевое слово ... можно вызвать метод базового класса
- 95) ... - это корректный вариант переопределения виртуального метода func1
- 96) Класс, у которого объявленные методы не содержат реализации, называется ... класс
- 97) Имеется базовый класс `class BaseTest { public override string ToString() { return "A"; } }` И производный класс: `class Test: BaseTest { public string ToString() { return "B"; } }` В результате выполнения приведённого ниже кода будет выведено ... `BaseTest[] d= { new Test(), new Test() }; foreach(BaseTest test in d) { Write(test.ToString()); }`
- 98) Имеется базовый класс `class BaseTest { public override string ToString() { return "A"; } }` И производный класс: `class Test: BaseTest { public override string ToString() { return "B"; } }` В результате выполнения приведённого ниже кода будет выведено ... `BaseTest[] d= { new Test(), new Test() }; foreach(BaseTest test in d) { Write(test.ToString()); }`
- 99) Укажите ошибку, допущенную в переопределении метода: `class BaseTest { public string method1() { return "A"; } } class Test: BaseTest { public override string method1() { return "B"; } }`
- 100) Неверно, что ...
- 101) Есть базовый класс `class BaseTest { protected int a; protected int b; public BaseTest(int a, int b): this(a) { b = b; WriteLine("2"); } public BaseTest(int a) { a = a; WriteLine("3"); } }` И производный класс `class Test: BaseTest { private int c; private int d; public Test(int c, int d, int a, int b): base(a,b) { c = c; d = d; WriteLine("4"); } }` Укажите верный порядок запуска конструкторов при создании экземпляра производного класса `Test t = new Test(1, 2, 3, 4);`
- 102) ... - это верные способы исправления ошибок в приведённом ниже коде `class BaseTest { protected int a; protected int b; public BaseTest(int a, int b): this(a) { b = b; } public BaseTest(int a) { a = a; } }` class `Test: BaseTest { private int c; private int d; public Test(int c, int d, int a, int b) { a = a; b = b; c = c; d = d; } }` { `a = this.a; b = this.b; c = this.c;` }
- 103) Укажите соответствие между понятиями и их описаниями?



- (104) ... список — список, в котором каждый элемент хранит ссылки на предыдущий и следующий элементы. Представлен классом `LinkedList`
- (105) В C# коллекции представлены набором шаблонных классов из пространства имен `System.Collections`. ...
- (106) Результатом применения метода `Dequeue()` к пустой очереди будет то, что ...
- (107) Результатом применения метода `Peek` к непустому стеку будет то, что метод ...
- (108) Результатом применения метода `Peek` к пустому стеку будет то, что ...
- (109) Верны утверждения, что ...
- (110) ... - это способ организации данных в коллекции, при котором элементы могут покинуть коллекцию в том же порядке, в котором в были помещены в нее.
- (111) Метод ... позволяет добавить новый элемент в конец списка `List`
- (112) Метод ... позволяет извлечь элемент из стека и избежать ошибки при пустом стеке
- (113) Метод ... позволяет проверить наличие в словаре определенного ключа
- (114) Выберите верные способы избежать возникновения ошибки при пустом стеке:
- (115) Свойство ... позволяет получить количество элементов в списке
- (116) Каждый элемент словаря имеет тип данных ...
- (117) Свойство ... должно использоваться для получения значений ключей элементов: `foreach(var phone in PhoneList) { Console.WriteLine($"phone: {phone...} name: {phone.Value}"); }`
- (118) Результатом попытки обратиться к элементу словаря по несуществующему ключу будет ...
- (119) Метод `dict.TryGetValue("meow", out V value)` в случае отсутствия в словаре элемента с ключом "meow" вернёт значение ...
- (120) Расставьте в правильном порядке изображения наполнения списка, так, чтобы они соответствовали строкам программы. Написать в ответе 1 любой из верных вариантов.



- (121) Сопоставьте типы коллекций с их описаниями:
- (122) Использование в словаре ..., имеющих разные типы недопустимо.
- (123) Все ... из пространства имен System.Collections.Generic являются шаблонными типами данных
- (124) Верно, что ...
- (125) При возникновении исключения в блоке try на экран будет выведено ... `int a =1; int b = 0; try { a = a / b; } catch when (a>0) { WriteLine(2); }`
- (126) При возникновении исключения в блоке try на экран будет выведено ... `int a =-1; int b = 0; try { a = a / b; } catch when (a==0) { WriteLine(1); } catch when (a>0) { WriteLine(2); }`
- (127) При возникновении исключения в блоке try на экран будет выведено ... `int a =0; int b = 0; try { a = a / b; } catch when (a==0) { WriteLine(1); } catch(DivideByZeroException) { WriteLine(2); }`
- (128) При возникновении исключения в блоке try на экран будет выведено ... `int a =2; int b = 0; try { a = a / b; } catch when (a==0) { WriteLine(1); } catch(DivideByZeroException) { WriteLine(2); }`
- (129) При наличии ...
- (130) Переменная `ex` может применяться для... `catch (DivideByZeroException ex) { // выполняемые инструкции }`
- (131) Неверно, что ...
- (132) Ключевое слово ... используется для принудительного вызова исключения
- (133) Принудительно вызвать можно не только производные типы исключений, вызов исключения базового класса ... также возможен
- (134) Создание собственных классов ... позволяет отслеживать ситуации, не являющиеся ошибками с точки зрения языка, но ошибочные с точки зрения логики программы
- (135) Свойство ... позволяет получить строковое значение с текстом ошибки?



- 136 В коде, приведённом ниже отсутствует часть ... `class FileLoaderException : Exception { public FileLoaderException(string message)... { } }`
- 137 ... – это тип исключения, который срабатывает при передаче некорректного значения в параметры метода
- 138 При выполнении приведённой ниже строки, если ввести с клавиатуры число 300, возникает исключение ... `byte a = byte.Parse(Console.ReadLine());` если ввести с клавиатуры число 300?
- 139 Конструктор класса `Exception` принимает в качестве параметра текст сообщения об ...
- 140 Чтобы на экран вывелоось число 1, необходимо указать вместо многоточия производный тип исключения ... `try { int[] a = {1,2,3,4,5}; a[10]=15; } catch (...) { WriteLine(1); } catch { WriteLine(2); }`
- 141 Сопоставьте типы исключений и ситуации, при которых они возникают:
- 142 Блок ... используется для размещения части кода, которая должна выполняться независимо от того, произошло ли исключение в блоке `try`
- 143 Основной целью ... исключений является предотвращения внезапного завершения работы программы вследствие ошибки?
- 144 Расставьте блоки кода в корректном порядке
- 145 Неименованные методы, предназначенные для инициализации делегатов, называются ... методами
- 146 Количество обработчиков у события может быть равным ...
- 147 Делегат позволяет вызвать метод ...
- 148 Есть два метода и делегат: `int f1() { return 2; } int f2() { return 3; } delegate int Del();` В результате выполнения приведённого ниже кода будет получено значение ... `Del d; d = f1; d += f2; Console.WriteLine(d());`
- 149 ... может содержать только методы с одним типом возвращаемого значения.
- 150 Есть два метода и делегат: `void f1() { Write("A"); } void f2() { Write("B"); } delegate void Del();` Какое значение будет выведено на экран в результате выполнения кода? `Del d; d = f1; d += f2; d();`



- (151) При вызове делегата, содержащего несколько методов возвращающих значение, в качестве результата будет получено значение ... добавленного метода.
- (152) Лямбда выражения ...
- (153) Лямбда выражения ...
- (154) Верно, что ...
- (155) Соотнесите делегаты и соответствующие им методы:
- (156) Обработчик события является ...
- (157) ... позволяет записать сложный элемент в xml-файл в автоматическом режиме
- (158) Делегат может применяться для ...
- (159) Создан делегат delegate void Message(); И два метода void Hello() { Console.WriteLine("Hello"); } void Hi() { Console.WriteLine("Hi"); } В результате выполнения следующей ниже программы, в консоль будет выведено ... Message mes = new Message(Hello); mes += Hi; mes -= Hello; mes -= Hi; mes();
- (160) Создан делегат delegate void Message(); И два метода void Hello() { Console.WriteLine("Hello"); } void Hi() { Console.WriteLine("Hi"); } В результате выполнения следующей ниже программы, в консоль будет выведено ... (перечислите через запятую без пробелов) Message mes1 = Hello; mes1 += Hi; mes1 += Hello; mes1 += Hello; mes1 -= Hello; mes1();
- (161) Создан делегат delegate int Operation(int x, int y); и методы: int Add(int x, int y) { return x + y; } int Multiply(int x, int y) { return x \* y; } В результате выполнения следующей ниже программы, в консоль будет выведено ... Operation del = Add; del += Multiply; int result = del(6, 5); Console.WriteLine(result);
- (162) Создан делегат delegate int Operation(int x, int y); Укажите верный порядок добавления в делегат указанных методов, если известно что вызов делегата в такой программе: Operation op; ..... Console.WriteLine(op(4,5)); приводит к следующему выводу данных: В С А В 9 Каждый метод может быть добавлен не более одного раза
- (163) Делегату delegate int Operation (int val); соответствует метод ...



- 164) При возникновении события имеющего несколько обработчиков ...
- 165) Оператор ... используется для добавления обработчика события
- 166) Интерфейсы это ... типы данных
- 167) В коллекции с типом данных интерфейса, могут храниться ... любых классов реализующих этот интерфейс
- 168) Создать экземпляр интерфейса ...
- 169) Интерфейс может включать такие компоненты, как ...
- 170) Класс может реализовать ... интерфейсов
- 171) 

```
interface Interface1 { void Execute(); } interface Interface2 { void Execute(); } class Test : Interface1, Interface2 { void Interface1.Execute() { Console.WriteLine("A"); } void Interface2.Execute() { Console.WriteLine("B"); } }
```

 При исполнении следующей ниже программы ...  
`Test t = new Test(); t.Execute();`
- 172) Чтобы исправить ошибку в приведённом ниже коде, требуется ...  

```
interface IAction { void Move(); } class RunAction : IAction { private void Move() { WriteLine("I am running"); } }
```
- 173) В приведённом ниже примере ошибка заключается в том, что ...  
`public abstract interface IAction { void Move(); }`
- 174) Интерфейсы могут наследоваться от ...
- 175) Метод ... реализует метод Execute в классе Test  

```
interface Interface1 { void Execute(); } interface Interface2 { void Execute(); } class Test : Interface1, Interface2 { void Execute() { Console.WriteLine("A"); } }
```
- 176) Соотнесите интерфейсы и соответствующие им классы:
- 177) ... интерфейсы – интерфейсы, позволяющие работать с различными типами определяемыми в качестве параметров.
- 178) Указать модификатор доступа для компонентов интерфейса ...
- 179) Отличия между интерфейсом и абстрактным классом заключаются в том, что ...





- 180 Класс может не содержать реализации всех компонентов интерфейса, в таком случае он должен быть объявлен как ..., а реализация должна быть сделана в производных классах
- 181 Неабстрактный класс, реализующий несколько интерфейсов должен содержать реализацию ... всех интерфейсов
- 182 Ошибка, допущенная в реализации интерфейса ...  
interface ILoader { public bool Load(string link); } class data\_adapter:  
ILoader { public bool Load() {  
реализация метода } }
- 183 Компонент данного интерфейса – это ... interface ILoader {  
public const string defaultPatch = "./"; }
- 184 Компонент данного интерфейса – это ... interface ILoader {  
public string FileName { get; set; } }
- 185 Компонент данного интерфейса – это ... interface ILoader {  
public event LoadHandler LoadEvent; }
- 186 Расположите типы в порядке их наследования
- 187 ... – это последовательность байтов, которую можно использовать для записи или чтения из вспомогательного запоминающего устройства, являющегося одним из устройств хранения информации
- 188 Конструкция using используется для ...
- 189 Неверно, что в случае попытки записи данных в несуществующий файл ...
- 190 Верно, что ...
- 191 Класс ... позволяет считывать данные из файлов
- 192 Относительно приведённого кода верно, что ... using  
(StreamWriter writer = new StreamWriter("test.txt", false)) {  
WriteLine(text); }
- 193 Метод ... следует использовать для построчного чтения файла
- 194 Верно, что ...
- 195 Соотнесите компоненты данных на C# с элементами XML-документа:



- 196 Названия элементов в XML являются ... зависимыми
- 197 Данный элемент имеет свойство ...
- 198 ... - ошибка, допущенная в XML файле.
- 199 ... позволяет записать сложный элемент в xml-файл в автоматическом режиме
- 200 При создании экземпляра класса `xmlSerializer` в конструктор необходимо передать ...
- 201 Для записи сериализуемого объекта в XML-файл следует использовать
- 202 Верно, что ...
- 203 ... XML – элемент, внутрь которого помещаются все остальные элементы
- 204 Объекты в JSON представлены в виде набора пар ...-значение
- 205 Для десериализации JSON применяется шаблонный метод ..., принимающий в качестве параметра файловый поток
- 206 При использовании `System.Xml` для открытия и хранения xml-файла используется класс ...
- 207 Расположите блоки кода в правильном порядке для сериализации объекта в JSON:
- 208 Элемент WPF ... используется в качестве корневого элемента окна
- 209 Контейнер ... позволяет располагать вложенные элементы аналогично таблице в строках и столбцах
- 210 Контейнер ... лучше использовать для предоставления пользователю возможности выбрать несколько необязательных опций
- 211 Свойство ... позволяет задать набор строк для контейнера `Grid`
- 212 Если для компоновки элементов в окне приложения используется `Grid`, то для объединения двух ячеек, находящихся в соседних столбцах и размещения в получившихся объединенных ячейках, элемента управления следует ...





- (213) Укажите верный порядок тегов, в соответствии с уровнями вложенности:
- (214) Чтобы изменить ориентацию дочерних элементов в панели StackPanel, требуется ...
- (215) Создаваемый по умолчанию файл ... включает хамл разметку элементов окна
- (216) Контейнер компоновки ... пристыковывает дочерние элементы к различным сторонам панели
- (217) Работа WPF-приложений основана на использовании ...
- (218) Метод ... отвечает за применение разметки описанной в файле XAML
- (219) Событие ... возникает при нажатии кнопки пользователем
- (220) Верно, что ...
- (221) Внутри одного окна можно использовать ... контейнеров для компоновки
- (222) Неверно, что одним из преимуществ WPF является ...
- (223) Имеется класс:
- (224) В WPF отсутствует контейнер для компоновки ...
- (225) Одна из задач элемента ... - размещение в окне надписи, поясняющей текстовое поле
- (226) Сопоставьте элементы управления и их назначение
- (227) XAML представляет собой ...
- (228) Событие ... вызывается при изменении выбранного элемента в ComboBox
- (229) IL – это ...
- (230) Язык C# характерен такими особенностями, как ...
- (231) Корректно работает вариант кода ...



- (232) Тип данных, хранящий беззнаковые числа в диапазоне [0,255] – это ...
- (233) ... – это значение, которое будет получено в ходе выполнения выражения `double a = 2; double b = 3; double d = (a + b)/a - b;`
- (234) ... – это корректно работающий код
- (235) Метод `Console ...` позволяет вывести на экран переданное значение и перейти на новую строку
- (236) Верно, что ...
- (237) Значение, которое будет получено в ходе выполнения выражения – ... `int a = 2; int b = 3; double d = (a + b) / (a + 1) - 1;`
- (238) Укажите верный порядок приоритетов математических операций:
- (239) Неверно, что ...
- (240) Значение ... будет получено в переменной с при вычислении выражения `int a = 9; int b = 4; int c = a/b;`
- (241) Значение ... получится в результате выполнения выражения в переменной `a` `int a=1; a++;`
- (242) Функцию ... следует использовать для преобразования введенной с клавиатуры строки в целое число
- (243) Верно, что ...
- (244) Записывая выражение в состав которой входит дробь, выражение в ... дроби следует взять в скобки
- (245) Знаком ... обозначается целочисленное деление
- (246) Тип данных ... используется для обозначения вещественных чисел
- (247) Инициализация целого числа в виде строки ... недопустима
- (248) Инициализация вещественного числа в виде строки ... недопустима
- (249) При выполнении кода, показанного ниже, ... будет выведено на экран `int a = 7; if(a > 0) { if(a < 5) { Write("a"); } else { Write("b"); } } else { Write("b"); }`
- (250) Логическая операция `^` истинна в случае, если...



- (251) Выберите истинное утверждение
- (252) Установите верное соотношение между методами строк и их описаниями
- (253) Даны начальные значения переменных `a=9 b=2` Расставьте блоки кода в правильном порядке внутри цикла, чтобы программа не уходила в бесконечный цикл `while (true) { ..... }`
- (254) Тернарный оператор используется ...
- (255) В приведённом ниже выражении будет выполнено ... операций сравнения `a = 5; b = 3; ((a < b) && (b == 3)) && (a <= 5)`
- (256) В ходе выполнения выражения будет получено значение ... `int a=20; int b=100; int c = a * 10 < b ? (a + b) : (a - b);`
- (257) В языке C# есть такие циклы, как ..., while, do-while, foreach
- (258) Логическая операция & истина в случае ...
- (259) При выполнении показанного ниже кода на экран будет выведено ...  
`int a = 7; if(a>0) { Console.WriteLine("a"); } if(a>5) { Console.WriteLine("b"); }`
- (260) Метод ... переводит строку в верхний регистр
- (261) Метод ... позволяет убрать лишние пробелы в начале строки
- (262) Метод ... позволяет разделить строку на массив подстрок
- (263) ... - нерабочий вариант перебора строки.
- (264) После выполнения кода, показанного ниже, на экран будет выведено значение ...  
`string a = "aBc"; string b = "abc"; WriteLine(a == b);`
- (265) ... - верные варианты инициализации массива
- (266) После выполнения кода, показанного ниже, на экран будет выведено значение ...  
`int[] a = {1,2,3} int[] b = a; b[0] = 123; Console.WriteLine(a[0]);`
- (267) После выполнения кода, показанного ниже, на экран будет выведено значение ...  
`string a = "ab"; string b = "c"; WriteLine(a + b);`



- (268) Основной недостаток такого метода заключается в том, что ... void F(double m) { double f= m \* 9.8; Console.WriteLine(f); }
- (269) Вызов ... соответствует приведённому ниже методу void test\_method(params int[] par\_arr) { ..... }
- (270) Верно, что ...
- (271) Корректные способы вернуть два значения из метода, это ...
- (272) Для передачи параметра по ссылке используется слово ...
- (273) Сопоставить ключевые слова и их назначение
- (274) Расположите в правильном порядке блоки кода, чтобы получить метод, позволяющий обменять местами значения двух переменных:
- (275) Слово void перед именем метода означает, что метод ...
- (276) После выполнения кода, показанного ниже, на экран будет выведено значение ... void test\_method(ref int a) { a++; } int a=6; test\_method(a)
- (277) ... функции, определяются и вызываются только внутри какого-то метода?
- (278) Верно, что ...
- (279) В результате выполнения приведённого ниже метода будет получено значение ... void swap(ref int a, ref int b) { int c = a; a = b; b = c; } int a=1; int b=2; swap(ref a, ref b); Console.WriteLine(a); Console.WriteLine(b);
- (280) В результате выполнения приведённого ниже метода будет получено значение ... void swap(int a, int b) { int c = a; a = b; b = c; } int a=1; int b=2; swap(a,b); Console.WriteLine(a); Console.WriteLine(b);
- (281) В результате выполнения приведённого ниже метода будет получено значение ... void swap(int a, int b) { a = b; b = a; } int a=1; int b=2; swap(a,b); Console.WriteLine(a); Console.WriteLine(b);
- (282) Метод, возвращающий строковое значение, должен иметь тип ...
- (283) При использовании массива параметров все параметры должны быть ... типа



- (284) При использовании внутри метода условной конструкции каждый блок кода должен возвращать ...
- (285) Если переменная передается в метод в качестве ..., то она должна называться так же, как и ...
- (286) Слово ... перед именем метода означает, что метод не возвращает значение
- (287) Внутри ... можно вызывать другие ...
- (288) Значение ... нельзя передать в параметр типа double
- (289) Правильный способ вызова показанного ниже метода выглядит как ...

```
void Display(int[] numbers) { int result = 0; for (int i = 0; i < numbers.Length; i++) { result += numbers[i]; } Console.WriteLine(result); }
```
- (290) Сопоставьте реализации конструкторов и их типы:
- (291) Расположите модификаторы доступа в порядке уменьшения области видимости:
- (292) Верную реализацию перегрузки оператора сложения для обыкновенных дробей – это ...
- (293) Конструктор по умолчанию не может включать инициализацию полей ...
- (294) Поля класса ... доступны извне 

```
class Student { public string a; public string b; string c; private string d; }
```
- (295) Свойства нужны для ...
- (296) Верно, что ...
- (297) Если в приведенном ниже коде будет вызван первый конструктор 

```
Student("Иванов"), то поле group получит значение ...
```

```
class Student { public string name; public string group; public int course; public Student(string name) : this(name, "не определена") // первый конструктор { course=1; } public Student(string name, string group) // второй конструктор { name = name; group = group; } }
```
- (298) Перегруженные методы должны иметь ...
- (299) ... – это метод, позволяющий разложить экземпляр класса на его составляющие значения.





- (300) Свойство может иметь метод доступа `get` для чтения и/или ... для записи.
- (301) Закрытый компонент класса может иметь модификатор доступа ...
- (302) Конструктор - это специальный метод, предназначенный для создания экземпляров ... и инициализации их полей.
- (303) При помощи ... класса можно получить значения закрытых полей
- (304) Методы классов могут быть вызваны через оператор ... от экземпляра класса
- (305) В C# можно перегружать как бинарные, так и ... операторы, а также операторы сравнения
- (306) Для одного набора типов операндов можно сделать только одну перегрузку ...
- (307) Укажите соответствие между понятиями и их описаниями:
- (308) Ошибка, допущенная в переопределении приведённого ниже метода, заключается в том, что ...  
`class BaseTest { public string method1() { return "A"; } } class Test: BaseTest { public override string method1() { return "B"; } }`
- (309) Существует базовый класс: `class BaseTest { public override string ToString() { return "A"; } }` А также производный класс: `class Test: BaseTest { public string ToString() { return "B"; } }` В результате выполнения приведённого ниже кода будет выведено ...  
`BaseTest[] d= { new Test(), new Test() }; foreach(BaseTest test in d) { Write(test.ToString()); }`
- (310) Существует базовый класс: `class BaseTest { public override string ToString() { return "A"; } }` А также производный класс: `class Test: BaseTest { public override string ToString() { return "B"; } }` В результате выполнения приведённого ниже кода будет выведено ...  
`BaseTest[] d= { new Test(), new Test() }; foreach(BaseTest test in d) { Write(test.ToString()); }`
- (311) ... не является одним из достоинств использования наследования
- (312) Модификатор `override` используется для ...
- (313) Через ключевое слово ... можно вызвать метод базового класса
- (314) Базовым классом для всех классов C# является ...



- (315) ... класс содержит методы, не содержащие реализации
- (316) Результатом упаковки является тип ...
- (317) Виртуальные методы должны иметь реализацию в ... классе
- (318) Для скрытия ... применяется override
- (319) Один ... класс не может иметь несколько родительских классов
- (320) Для инициализации ... полей можно вызвать конструктор базового класса.
- (321) Производные классы не имеют доступа к ... полям базового класса
- (322) ... тип данных, позволяет создавать классы имеющие общий функционал, но работающие с переменными различных типов
- (323) Сопоставьте типы коллекций с их описаниями
- (324) Результатом применения метода Dequeue() к пустой очереди будет то, что ...
- (325) Результатом применения метода Peek к непустому стеку будет то, что метод ...
- (326) Верно, что ...
- (327) Верные способы избежать возникновения ошибки при пустом стеке это ...
- (328) Каждый элемент словаря имеет тип данных ...
- (329) Результатом попытки обратиться к элементу словаря по несуществующему ключу будет ...
- (330) Результатом применения метода Peek к пустому стеку будет то, что ...
- (331) Метод ... позволяет добавить элементы в стек?
- (332) Метод ... позволяет извлечь элемент из стека?
- (333) Метод ... позволяет безопасно извлечь элемент из очереди?



- (334) Метод ... позволяет позволять добавить элемент в список?
- (335) Метод ... позволяет безопасно получить первый элемент очереди без его извлечения?
- (336) Метод очереди TryPeek возвращает значение типа
- (337) Метод ... позволяет проверить, присутствует ли элемент в списке?
- (338) ... - это пространство имен, содержащее основные классы коллекций
- (339) Сопоставьте типы исключений и ситуации при которых они возникают:
- (340) Тип исключения ... возникнет при выполнении строки `byte a = byte.Parse(Console.ReadLine());` если ввести с клавиатуры число 300
- (341) Переменная `ex` в приведённом ниже блоке кода может применяться для ... `catch (DivideByZeroException ex) { // выполняемые инструкции }`
- (342) Верно, что ...
- (343) Сопоставьте свойства класса `Exception` и их назначение:
- (344) При возникновении исключения в блоке `try` приведённый ниже код выведет ... `int a = 1; int b = 0; try { a = a / b; } catch when (a==0) { WriteLine(1); } catch when (a>0) { WriteLine(2); }`
- (345) При возникновении исключения в блоке `try` приведённый ниже код выведет ... `int a = -1; int b = 0; try { a = a / b; } catch when (a==0) { WriteLine(1); } catch when (a>0) { WriteLine(2); }`
- (346) Верно, что ...
- (347) В приведённом ниже коде необходимо указать производный тип исключения ..., чтобы вместо многоточия на экран вывелоась число 1 `try { int[] a = {1,2,3,4,5}; a[10]=15; } catch (...) { WriteLine(1); } catch { WriteLine(2); }`
- (348) Свойство ... позволяет получить строковое значение с текстом ошибки?
- (349) Оператор ... используется для генерации исключений
- (350) Код, который потенциально может привести к исключительной ситуации, необходимо помещать в блок ...



- (351) В блоке ... задается обработчик исключения
- (352) Оператор ... используется для задания условий в фильтрах исключений
- (353) Соотнесите делегаты и соответствующие им методы:
- (354) Создан делегат: `delegate int Operation (int val);` ... соответствует данному делегату. Какие из указанных методов соответствуют данному делегату?
- (355) При возникновении события имеющего несколько обработчиков ...
- (356) Создан делегат `delegate int Operation(int x, int y);` и методы: `int Add(int x, int y) { return x + y; }` `int Multiply(int x, int y) { return x * y; }` ... - это число, которое будет получено в результате такой программы `Operation del = Add; del += Multiply; int result = del(6, 5); Console.WriteLine(result);`
- (357) Создан делегат `delegate void Message();` И два метода `void Hello() { Console.WriteLine("Hello"); }` `void Hi() { Console.WriteLine("Hi"); }` В результате выполнения следующей программы ... `Message mes = new Message(Hello); mes += Hi; mes -= Hello; mes -= Hi; mes();`
- (358) Верно, что ...
- (359) Верно, что ...
- (360) Есть два метода и делегат: `void f1(string s) { Write(s+A+s); }` `string f2() { return "B"; }` `delegate void Del(string s);` В результате выполнения кода будет выведено значение ... `Del d; d = f1; Console.WriteLine(d(f2()));`
- (361) Верно, что ...
- (362) Верны утверждения, что ...
- (363) Верно ли утверждение: Если ...-выражение содержит несколько действий, то они помещаются в фигурные скобки
- (364) Механизм “...” в C# позволяет отделить внутреннюю логику класса от пользовательского интерфейса
- (365) Соотнесите интерфейсы и соответствующие им классы:



- (366) Расположите типы в порядке их наследования:
- (367) Верно, что ...
- (368) Метод ... реализует метод Execute в классе Test      void Execute();  
} interface Interface2 {    void Execute(); } class Test : Interface1,  
Interface2 {    void Execute()    {        Console.WriteLine("A");    } }
- (369) В определении интерфейса неверно то, что ... public abstract  
interface IAction {    void Move(); } Выберите верное утверждение:
- (370) Интерфейсы это ... типы
- (371) Интерфейс может включать такие компоненты, как ...
- (372) Интерфейсы могут наследоваться от ...
- (373) Соотнесите компоненты данных на C# с элементами  
XML-документа:
- (374) Ошибки в XML-файле заключаются в том, что ...
- (375) В приведённом блоке кода ... using (StreamWriter writer = new  
StreamWriter("test.txt", false)) { WriteLine(text); }
- (376) Верно, что ...
- (377) Верно, что ...
- (378) Конструкция using применяется для ...
- (379) В случае попытки записи данных в несуществующий файл ...
- (380) Верно, что ...
- (381) Объекты в JSON представлены в виде набора пар ключ-...
- (382) Для открытия и хранения xml-файла при использовании  
System.Xml используется класс ...
- (383) Файловый поток – это ...



- (384) Если для компоновки элементов в окне приложения используется Grid, то для объединения двух ячеек, находящихся в соседних столбцах и размещения в получившихся объединенных ячейках, элемента управления следует ...
- (385) Верно, что ...
- (386) Сопоставьте элементы управления и их назначение:
- (387) XAML представляет собой ...
- (388) При нажатии кнопки пользователем возникает событие ...
- (389) Создаваемый по умолчанию файл ... включает xaml разметку элементов окна
- (390) Укажите верный порядок тегов, в соответствии с уровнями вложенности:
- (391) Чтобы изменить ориентацию дочерних элементов в панели StackPanel, требуется использовать свойство ...
- (392) ... - это элемент WPF, который используется в качестве корневого элемента окна
- (393) ... - это контейнер, позволяющий компоновать элементы произвольным образом путем указания их координат
- (394) Контейнер ... позволяет располагать вложенные элементы аналогично таблице в строках и столбцах
- (395) В основе технологии wpf лежит использование ...
- (396) Важным достоинством WPF является возможность декларативного определения графического интерфейса с помощью специального языка разметки ...
- (397) Событие ... генерируется при установке флагка CheckBox в отмеченное состояние