



## Программирование на языке C#.фип\_СПО(2/3)\_ИСиП\_МДК без ИМ\_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 c 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) Сопоставьте операторы и их значения:
- 22) При выполнении представленного ниже кода на экран будет выведено ... `int a = 7; if(a>0) { Console.WriteLine("a"); } if(a>5) { Console.WriteLine("b"); }`
- 23) При выполнении представленного ниже кода на экран будет выведено ... `int a = 7; if(a>0) { Console.WriteLine("a"); } else if(a>5) { Console.WriteLine("b"); }`
- 24) При выполнении представленного ниже кода на экран будет выведено ... `int a = 7; if(a>0) { if(a<5) { Console.WriteLine("a"); } } else { Console.WriteLine("b"); }`
- 25) При выполнении представленного ниже кода на экран будет выведено ... `int a = 7; if(a > 0) { if(a < 5) { Console.WriteLine("a"); } else { Console.WriteLine("b"); } } else { Console.WriteLine("b"); }`
- 26) Операторы I и II обозначают "... "
- 27) Разница между операторами = и == заключается в том, что ...
- 28) В следующем выражении будет выполнено ... операций сравнения: `a = 5; b = 3; ((a < b) && (b == 3)) && (a <= 5)`
- 29) Логическая операция ^ истина в случае, если ...
- 30) Логическая операция & истина в случае, если ...
- 31) В языке C# имеются циклы for, while, ..., do...while
- 32) Блок кода, помещенный в цикл ...
- 33) Слово "Hello" будет выведено на экран ... раз `int a=10; do { Console.WriteLine("Hello"); a--; }while(a > 0)`
- 34) При выполнении представленного ниже кода на экран будет выведено "... " `int a=10; do { Console.WriteLine("a"); a--; }while(a > 10) ;`





- 35) Тернарные операторы используются для ...
- 36) В ходе выполнения выражения, приведённого ниже, будет получено число ... `int a=20; int b=100; int c = a * 10 < b ? (a + b) : (a - b);`
- 37) Установите верное соотношение между методами строк и их описаниями:
- 38) Для того, чтобы значение В не вычислялось в выражении A||B, значение а должно быть ...
- 39) При использовании конструкции из нескольких else if будет выполнен только блок кода с первым ... условием
- 40) Циклическая конструкция, предназначенная для перебора элементов коллекций, называется ...
- 41) Свойство ... используется для определения длины массива
- 42) 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) { ..... }`
- 43) Слово void перед именем метода означает, что ...
- 44) Основной недостаток ниже приведённого метода заключается в том, что ... `void F(double m) { double f= m * 9.8; Console.WriteLine(f); }`
- 45) 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; } }`
- 46) В результате выполнения кода, приведённого ниже, значение а будет равно ... `void test_method( int a) { a++; } int a=6; test_method(a)`
- 47) В результате выполнения приведённого ниже кода, значение а будет равно ... `void test_method(ref int a) { a++; } int a=6; test_method(ref a)`
- 48) Какой из вызовов соответствует методу? `void test_method(params int[] par_arr) { ..... }`
- 49) Небезопасность использования передачи параметров по ссылке заключается в том, что ...





- 50) ... функции, определяются и вызываются только внутри какого-либо метода
- 51) ... алгоритм – это такой алгоритм, в определении которого содержится прямой или косвенный вызов этого же алгоритма.
- 52) Использование ... параметров позволяет вернуть из метода несколько значений?
- 53) Рекурсивный алгоритм ...
- 54) Верно, что ...
- 55) Верно, что ...
- 56) Верно, что ...
- 57) 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;
}
```
- 58) Сопоставьте ключевые слова и их назначение:
- 59) Результатом работы приведённой ниже программы будет ...
- ```
int x;
Inc (ref x); Console.WriteLine(x);
void Inc (ref int x) { x++; }
```
- 60) Базовый тип данных ... может возвращать метод `test()`, чтобы он мог применяться в следующей конструкции `foreach(char i in test()) { ..... }`
- 61) Для передачи параметра по ссылке используется слово ...
- 62) Расположите в правильном порядке блоки кода, чтобы получить метод, позволяющий обменять местами значения двух переменных:





- 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  
    {  
        this.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 Общедоступные поля должны иметь модификатор доступа ...
- 81 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; }`
- 82 Ключевое слово ... используется для создания экземпляров классов
- 83 Расположите модификаторы доступа в порядке уменьшения области видимости?
- 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 перенимают 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) { Console.WriteLine(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) { Console.WriteLine(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) Есть базовый класс `class BaseTest { protected int a; protected int b; public BaseTest(int a, int b): this(a) { this.b = b; Console.WriteLine("2"); } public BaseTest(int a) { this.a = a; Console.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) { this.c = c; this.d = d; Console.WriteLine("4"); } }` Укажите верный порядок запуска конструкторов при создании экземпляра производного класса `Test`  
`t = new Test(1, 2, 3, 4);`
- 101) ... - 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) { this.b = b; } public BaseTest(int a) { this.a = a; } }` `class Test: BaseTest { private int c; private int d; public Test(int c, int d, int a, int b) { this.a = a; this.b = b; this.c = c; this.d = d; } }` { a = this.a; b = this.b; c = this.c; }
- 102) Верный вариант создания экземпляра шаблонного класса - это ...  
`class Class1<T> { public T value { get; set; } public Class1(T val) { value = val; } }`
- 103) Укажите соответствие между понятиями и их описаниями?
- 104) IL - это ...
- 105) Язык C# характерен такими особенностями, как ...
- 106) Корректно работает вариант кода ...





- 107) Тип данных, хранящий беззнаковые числа в диапазоне [0,255] – это ...
- 108) ... - это значение, которое будет получено в ходе выполнения выражения `double a = 2; double b = 3; double d = (a + b)/a - b;`
- 109) ... - это корректно работающий код
- 110) Метод `Console ...` позволяет вывести на экран переданное значение и перейти на новую строку
- 111) Верно, что ...
- 112) Значение, которое будет получено в ходе выполнения выражения - ... `int a = 2; int b = 3; double d = (a + b) / (a + 1) - 1;`
- 113) Укажите верный порядок приоритетов математических операций:
- 114) Неверно, что ...
- 115) Значение ... будет получено в переменной `c` при вычислении выражения `int a = 9; int b = 4; int c = a/b;`
- 116) Значение ... получится в результате выполнения выражения в переменной `a` `int a=1; a++;`
- 117) Функцию ... следует использовать для преобразования введенной с клавиатуры строки в целое число
- 118) Верно, что ...
- 119) Записывая выражение в состав которой входит дробь, выражение в ... дроби следует взять в скобки
- 120) Знаком ... обозначается логическое сложение
- 121) Знаком ... обозначается целочисленное деление
- 122) Тип данных ... используется для обозначения вещественных чисел
- 123) Инициализация целого числа в виде строки ... недопустима
- 124) Инициализация вещественного числа в виде строки ... недопустима
- 125) При выполнении кода, показанного ниже, ... будет выведено на экран `int a = 7; if(a > 0) { if(a < 5) { Console.WriteLine("a"); } else { Console.WriteLine("b"); } } else { Console.WriteLine("b"); }`







- 126) Логическая операция  $\wedge$  истинна в случае, если...
- 127) Выберите истинное утверждение
- 128) Установите верное соотношение между методами строк и их описаниями
- 129) 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) { ..... }`
- 130) Тернарный оператор используется ...
- 131) В приведённом ниже выражении будет выполнено ... операций сравнения `a = 5; b = 3; ((a < b) && (b == 3)) && (a <= 5)`
- 132) В ходе выполнения выражения будет получено значение ... `int a=20; int b=100; int c = a * 10 < b ? (a + b) : (a - b);`
- 133) В языке C# есть такие циклы, как ..., `while`, `do-while`, `foreach`
- 134) Логическая операция  $\&$  истина в случае ...
- 135) При выполнении показанного ниже кода на экран будет выведено ... `int a = 7; if(a>0) { Console.WriteLine("a"); } if(a>5) { Console.WriteLine("b"); }`
- 136) Метод ... переводит строку в верхний регистр
- 137) Метод ... позволяет убрать лишние пробелы в начале строки
- 138) Метод ... позволяет разделить строку на массив подстрок
- 139) ... - нерабочий вариант перебора строки.
- 140) После выполнения кода, показанного ниже, на экран будет выведено значение ... `string a = "aBc"; string b = "abc"; Console.WriteLine(a == b);`
- 141) ... - верные варианты инициализации массива
- 142) 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]);`





- 143) После выполнения кода, показанного ниже, на экран будет выведено значение ... `string a = "ab"; string b = "c"; Console.WriteLine(a + b);`
- 144) Основной недостаток такого метода заключается в том, что ... `void F(double m) { double f= m * 9.8; Console.WriteLine(f); }`
- 145) Вызов ... соответствует приведённому ниже методу `void test_method(params int[] par_arr) { ..... }`
- 146) Верно, что ...
- 147) Корректные способы вернуть два значения из метода, это ...
- 148) Для передачи параметра по ссылке используется слово ...
- 149) Сопоставить ключевые слова и их назначение
- 150) Расположите в правильном порядке блоки кода, чтобы получить метод, позволяющий обменивать местами значения двух переменных:
- 151) Слово `void` перед именем метода означает, что метод ...
- 152) После выполнения кода, показанного ниже, на экран будет выведено значение ... `void test_method(ref int a) { a++; } int a=6; test_method(a)`
- 153) ... функции, определяются и вызываются только внутри какого-то метода?
- 154) Верно, что ...
- 155) 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.Write(a); Console.Write(b);`
- 156) 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.Write(a); Console.Write(b);`
- 157) 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.Write(a); Console.Write(b);`
- 158) Метод, возвращающий строковое значение, должен иметь тип ...





- 159) При использовании массива параметров все параметры должны быть ... типа
- 160) При использовании внутри метода условной конструкции каждый блок кода должен возвращать ...
- 161) Если переменная передается в метод в качестве ..., то она должна называться так же, как и ...
- 162) Слово ... перед именем метода означает, что метод не возвращает значение
- 163) Внутри ... можно вызывать другие ...
- 164) Значение ... нельзя передать в параметр типа double
- 165) 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); }`
- 166) Сопоставьте реализации конструкторов и их типы:
- 167) Расположите модификаторы доступа в порядке уменьшения области видимости:
- 168) Верную реализация перегрузки оператора сложения для обыкновенных дробей – это ...
- 169) Конструктор по умолчанию не может включать инициализацию полей ...
- 170) Поля класса ... доступны извне  
`class Student { public string a; public string b; string c; private string d; }`
- 171) Свойства нужны для ...
- 172) Верно, что ...
- 173) 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 { this.course=1; } public Student(string name, string group) // второй конструктор { this.name = name; this.group = group; } }`
- 174) Перегруженные методы должны иметь ...





- 175) ... - это метод, позволяющий разложить экземпляр класса на его составляющие значения.
- 176) Свойство может иметь метод доступа `get` для чтения и/или ... для записи.
- 177) Закрытый компонент класса может иметь модификатор доступа ...
- 178) Конструктор - это специальный метод, предназначенный для создания экземпляров ... и инициализации их полей.
- 179) При помощи ... класса можно получить значения закрытых полей
- 180) Методы классов могут быть вызваны через оператор ... от экземпляра класса
- 181) В C# можно перегружать как бинарные, так и ... операторы, а также операторы сравнения
- 182) Для одного набора типов операндов можно сделать только одну перегрузку ...
- 183) Укажите соответствие между понятиями и их описаниями:
- 184) Верный вариант создания экземпляра шаблонного класса это - ...  
`class Class1<T> { public T value { get; set; } public Class1(T val) { value = val; } }`
- 185) 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"; } }`
- 186) 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) { Console.Write(test.ToString()); }`
- 187) 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) { Console.Write(test.ToString()); }`
- 188) ... не является одним из достоинств использования наследования





- 189) Модификатор `override` используется для ...
- 190) Через ключевое слово ... можно вызвать метод базового класса
- 191) Базовым классом для всех классов C# является ...
- 192) Сопоставьте операторы и их значения

