

**1. Перечислите возможности типа «структура» (value-типов).**

1.	Определение индекатора	
2	Переопределение конструктора по умолчанию	
3	Наследование от классов	
4	Использование событий	
5	Определение статического конструктора	
6	Множественное наследование	
7	Реализация интерфейсов	
8	Boxing и Unboxing	

**2. Укажите верные особенности, присущие исключениям в языке C#**

1	исключения представляются операторами	
2	исключения представляются классами	
3	все классы исключений являются потомками класса исключений Exception	
4	запрещено создавать пользовательские исключения	
5	можно создать обработчик исключения, который будет отлавливать все исключения	

**3. Укажите верные особенности, присущие пространствам имен в языке C#**

1	пространства имен можно вкладывать друг в друга	
2	запрещено создавать пользовательские пространства имен	
3	все пространства имен являются потомками класса NameSpace	
4	благодаря пространствам имен избегаются конфликты классами	
5	пространства имен поддерживают глобальные события	

**4. Укажите верные особенности, присущие перегрузке операторов в языке C#**

1	в бинарных операторах оба операнда должны быть одинакового типа.	
2	тип операнда унарных операторов должен быть таким же, как и у класса, для которого перегружается оператор	
3	в бинарных операторах хотя бы один из операндов должен быть такого же типа, как и у его класса.	
4	в параметрах оператора нельзя использовать модификатор ref или out	
5	перегрузка операторов возможна только в структурах	

**5. Укажите верные особенности, присущие свойствам в языке C#**

1	свойство не определяет место в памяти	
2	свойство может быть передано методу в качестве параметра ref или out	
3	свойство не подлежит перегрузке.	
4	свойство всегда должно быть public	
5	свойство не должно изменять состояние базовой переменной при вызове аксессуара get	

**6. Укажите верные особенности, применения модификаторов к свойствам в языке C#**

1	действию модификатора доступа подлежит только один аксессуар: set или get, но не оба сразу	
2	модификатор должен обеспечивать более общий доступ к аксессуару, чем доступ на уровне свойства	
3	к свойствам нельзя применять модификатор internal	
4	модификатор должен обеспечивать более ограниченный доступ к аксессуару, чем доступ на уровне свойства	
5	к свойствам нельзя применять модификатор private	

**7. В каком месте в программе может быть определен делегат (тип)?**

1	только как член класса или структуры.	
2	только внутри абстрактного класса.	
3	только как член интерфейса.	
4	в любом месте, где может быть определен класс.	
5	внутри методов и аксессуаров.	

**8. Возможно ли из тела лямбда-выражения обращаться к переменным (локальным, полям, свойствам и т.п.), определенным за пределами выражения?**

1	Только к переменным значимого типа.	
2	Можно обратиться как к локальным переменным и аргументам метода, в котором определено выражение, так и к полям и свойствам класса, в котором оно находится.	
3	Только к полям и свойствам класса, в котором определено лямбда-выражение.	
4	Только через аргументы, переданные в лямбда-выражения.	
5	Только к локальным переменным и аргументам метода, в котором определено выражение.	

**9. В каком месте в коде можно использовать лямбда-выражения?**

1	Можно использовать в любом месте, где можно определить метод.	
2	В арифметических операциях.	
3	Только в операциях LINQ (язык интегрированных запросов).	
4	В местах, где компилятор ожидает встретить делегат (функцию обратного вызова).	
5	В аксессорах свойств и событий.	

**10. Укажите состав сборки на языке программирования C#:**

1	манифест	
2	MSIL	
3	OBJ	
4	EXE	
5	метаданные	
6	реестр	

**11. Укажите особенности JIT компилятора:**

1	В процессе выполнения программы компилируются только те ее части, которые требуется выполнить в данный момент	
2	При ПОВТОРНЫХ вызовах метода JIT компилятор осуществляет повторную быструю перекомпиляцию	
3	Откомпилированные инструкции JIT сохраняет в специальные файлы.	
4	Повторное выполнение откомпилированных инструкций выполняется быстрее.	
5	JIT компилятор учитывает особенности архитектуры CPU компьютера.	

**12. Укажите особенности класса System.IO.Stream.**

1	является байтовым потоком	
2	является базовым для всех остальных классов потоков.	
3	является виртуальным классом – создать объект класса Stream возможно.	
4	является абстрактным классом – создать объект класса Stream невозможно.	

**13. Что могут обеспечивать регулярные выражения**

1	эффективный поиск в тексте по заданному шаблону	
2	улучшенную работу класса String	
3	расширяет функциональность класса System.Text	
4	редактирование, замену и удаление подстрок	

**14. Что означает в регулярных выражениях запись {n,}**

1	Соответствует предшествующему шаблону, повторенному не менее n и не более m раз	
2	Соответствует предшествующему шаблону, повторенному n или более раз	
3	Соответствует в точности n экземплярам предшествующего шаблона	
4	Соответствует предшествующему шаблону, повторенному 0 или n раз	

**15. Выберите элементы, которые нельзя пометить атрибутом.**

1	Интерфейсы	
2	Все перечисленное можно пометить атрибутом	
3	Возвращаемые значения	
4	Классы	
5	Методы	
6	Структуры	

**16. От какого типа должны наследоваться пользовательские классы настраиваемых атрибутов?**

1	1. System.Reflection.CustomAttribute	
2	2. System.ValueType	
3	3. System.Object	
4	4. System.Reflection.MemberInfo	
5	5. System.Attribute	

**17. Укажите, какие требования обязательны при написании своего атрибута:**

1	С помощью атрибута AttributeUsage должна быть указана его область действия	
2	Класс должен быть наследован от System.Attribute	
3	Должен быть объявлен private конструктор	
4	Имя класса должно заканчиваться на Attribute	

**18. Выберите все правильные утверждения относительно интерфейсов:**

1	Интерфейсы нельзя наследовать	
2	От интерфейсов нельзя наследоваться	
3	В интерфейсах нельзя определять поля	
4	В интерфейсах можно писать определения методов	
5	В интерфейсах можно писать реализацию методов	

**19. Укажите, какие члены класса могут наследоваться?**

1	операторы	
2	свойства	
3	методы	
4	индексаторы	
5	поля	
6	конструкторы	

**20. Укажите основные особенности виртуального метода**

1	виртуальный метод должен обязательно возвращать значение	
2	виртуальный метод должен обязательно первым параметром принимать this	
3	виртуальный метод не может быть static или abstract	
4	переопределять виртуальный метод не обязательно.	
5	виртуальный метод объявляется только в виртуальном классе	
6	виртуальный метод должен быть объявлен последним в списке методов класса	
7	вариант выполняемого виртуального метода выбирается по типу объекта, а не по типу ссылки на этот объект	

**21. Для чего предназначен метод Finalize?**

1	У Finalize нет особого предназначения	
2	Finalize - аналог блока finally	
3	Позволяет объекту выполнить корректную очистку, прежде чем сборщик мусора освободит занимаемую объектом память	
4	Для детерминированного уничтожения объектов типа	

**22. Какие объекты относятся к поколению 2 сборщика мусора**

1	Поколение 2. Идентифицирует объект, которому удалось пережить более одного прохода сборщика мусора.	
2	Поколение 2. Идентифицирует объект, которому удалось пережить один проход сборщика мусора.	
3	Поколение 2. Идентифицирует объект, которому не удалось пережить проход сборщика мусора.	
4	Поколение 2. Идентифицирует объект, которому удалось пережить менее одного прохода сборщика мусора.	

**23. Когда вызываются статические конструкторы классов в C#?**

1	После каждого обращения к статическим полям, методам и свойствам	
2	Строгий порядок вызова не определен (в случайное время)	
3	Статических конструкторов в C# нет	
4	Один раз при первом создании экземпляра класса или при первом обращении к статическим членам класса	
5	По требованию сборщика мусора	

**24. Выберите правильные характеристики класса**

`sealed class SomeClass { }`

1	SomeClass наследуется от System.Object	
2	SomeClass является значимым типом (value-type)	
3	SomeClass доступен для использования только внутри сборки, в которой он определен	
4	SomeClass не может быть базовым классом	
5	SomeClass могут наследовать другие классы	
6	SomeClass может использоваться внешними сборками	

**25. Укажите верные особенности массивов в C#**

1	массивы унаследованы от класса System.Collections.Array	
2	освобождение памяти сборщиком мусора	
3	при выделении памяти под массив значения всех простых типов устанавливаются в «0»	
4	при выделении памяти под массив значения логического типа устанавливаются в true	
5	при выделении памяти под массив значения всех ссылочных типов устанавливаются в «null»,	