**Вопрос 1** **Что произойдет в результате выполнения данного кода на C#, если он не содержит синтаксическую ошибку.**

**var fs = new System.IO.FileStream("file", System.IO.FileMode.OpenOrCreate,**

**System.IO.FileAccess.Write, System.IO.FileShare.Read);**

**var sw = new System.IO.StreamWriter(fs);**

**sw.Write("Hello world!");**

**sw.Flush();**

**sw.Close();**

**fs = new System.IO.FileStream("file", System.IO.FileMode.OpenOrCreate,**

**System.IO.FileAccess.Write, System.IO.FileShare.Read);**

**sw = new System.IO.StreamWriter(fs);**

**sw.Write("Hello c#!");**

**sw.Flush();**

**sw.Close();**

**fs = new System.IO.FileStream("file", System.IO.FileMode.Open,**

**System.IO.FileAccess.Read, System.IO.FileShare.Read);**

**var sr = new System.IO.StreamReader(fs);**

**Console.WriteLine(sr.ReadLine());**

**sr.Close();**

InvalidOperationException

на экране отобразится "Hello world!"

ни один из вышеперечисленных ответов не является верным

на экране отобразится "Hello c#!"

IOException

FileNotFoundException

**Ответ:** ни один из вышеперечисленных ответов не является верным[[1]](#footnote-1)

**Вопрос 2** **когда обьект считается достижимым**

если объект находится в том поколении, которое в данный момент очищается сборщиком мусора

если сборка мусора еще не произошла

если под объект выделена память

если на него есть хоть одна ссылка

если все ссылки указывают на этот объект

если сборщик мусора имеет ссылку на этот объект

**Ответ:** если на него есть хоть одна ссылка

**Вопрос 3** **29 В чем отличие XmlTextReader от XmlDocument**

XmlTextReader построен на SAX, а XmlDocument на DOM

XmlTextReader используется для чтения и изменения документа, а XmlDocument - только для чтения

XmlTextReader построен на DOM, а XmlDocument на SAX

XmlTextReader используется только для чтения документа, а XmlDocument - для чтения и изменения

XmlTextReader и XmlDocument построены на DOM. XmlTextReader - это обновленная версия XmlDocument

XmlTextReader и XmlDocument построены на SAX. XmlDocument - это обновленная версия XmlTextReader

**Ответ:** XmlTextReader используется только для чтения документа, а XmlDocument - для чтения и изменения

**Вопрос 4** **Чем отличается сборка мусора в NET.Framework 4.5 и выше от предъидущих поколений**

в NET.Framework 4.5 и выше добавлена фоновая сборка мусора

они одинаковы

в NET.Framework 4.5 и выше добавили дополнительный поток для сборки мусора

для NET.Framework 4.5 и выше добавили 4е поколение объектов

в версиях до NET.Framework 4.5 сборка мусора происходила в фоновом режиме

сделаны небольшие оптимизации, однако, в принципе, они одинаковы

**Ответ:** в NET.Framework 4.5 и выше добавлена фоновая сборка мусора

**Вопрос 5** **Что произойдет в результате выполнения данного кода на C#, если он не содержит синтаксическую ошибку.**

**Если он содержит ошибку, то какую?**

**var sw = new System.IO.StreamWriter("file");**

**sw.Write("123");**

**sw.Dispose();**

**sw.Write("567");**

**sw.Close();**

**Console.WriteLine("Operation completed!");**

NullReferenceException – вызван метод Dispose(), а за ним Write()

ArgumentException – имя файла не имеет расширения

после записи не вызван метод Dispose()

Operation completed!

ObjectDisposedException – вызван метод Dispose(), а за ним Write()

неверный синтаксис создания объекта потока

**Ответ:** ObjectDisposedException – вызван метод Dispose(), а за ним Write()

**Вопрос 6** **Выберите все верные утверждения**

после сборки мусора все оставшиеся обьекты переходят в следующее поколение, если текущее не последнее

после сборки мусора все оставшиеся обьекты переходят в следующее поколение, если текущее не последнее

метод KeepAlive используется для предотвращения освобождения обьектов, на которые нет ссылок в управляемом коде

обьект уничтожается сборщиком мусора сразу, когда на него перестают ссылаться все обьекты

для уничтожения объекта, сборщик мусора прямо вызывает метод Dispose()

процесс сборки мусора разбит на поколения для увеличения скорости сборки

**Ответ:** после сборки мусора все оставшиеся обьекты переходят в следующее поколение, если текущее не последнее[[2]](#footnote-2)   
метод KeepAlive используется для предотвращения освобождения обьектов, на которые нет ссылок в управляемом коде   
процесс сборки мусора разбит на поколения для увеличения скорости сборки

**Вопрос 7** **Что произойдет в результате выполнения данного кода на C#, если он не содержит синтаксическую ошибку.**

**Если он содержит ошибку, то какую?**

**class Program {**

**static void Main(){**

**var f1 = new System.IO.FileInfo("file");**

**if (!f1.Exists) f1.Create();**

**f1.Attributes = System.IO.FileAttributes.ReadOnly |**

**System.IO.FileAttributes.Encrypted | System.IO.FileAttributes.Archive;**

**f1.Refresh();**

**var f2 = new System.IO.FileInfo("file");**

**var attr = f2.Attributes;**

**Console.Write((attr == System.IO.FileAttributes.Archive) + " ");**

**//...**

**Console.WriteLine((attr & System.IO.FileAttributes.Archive)**

**== System.IO.FileAttributes.Archive);**

**//...**

**}**

**}**

False True

True True

FormatException – неверный синтаксис получения атрибутов

неверный синтаксис установки атрибутов файла на c#

False False

InvalidOperationException – невозможно установить атрибуты

**Ответ:** False True

**Вопрос 8** **Что произойдет в результате выполнения данного кода на C#, если он не содержит синтаксическую ошибку.**

**Если он содержит ошибку, то какую?**

**static void Main(){**

**Console.WriteLine(System.IO.Path.GetPathRoot("file"));**

**}**

file

пустая строка

null

NullReferenceException

ArgumentException

System.String

**Ответ:** пустая строка

**Вопрос 9** **Что произойдет в результате выполнения данного кода на C#, если он не содержит синтаксическую ошибку.**

**Если он содержит ошибку, то какую? (код для Windows).**

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

**String s = System.IO.Path.Combine("folder", "file.txt");**

**Console.Write(s);**

**}**

c:\folder\file.txt

C:\\folder\\file.txt

C:\folder\file.txt

ArgumentException

folder\file.txt

c:\\folder\\file.txt

**Ответ:** folder\file.txt

**Вопрос 10** **Что произойдет в результате выполнения данного кода на C#, если он не содержит синтаксическую ошибку.**

**Если он содержит ошибку, то какую?**

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

**Console.WriteLine(System.IO.Path.ChangeExtension("file", "txt"));**

**}**

file.txt

C:\\file.txt

KeyNotFoundException

c:\file.txt

IOException

ArgumentException

**Ответ:** file.txt

**Вопрос 11** **Что произойдет в результате выполнения данного кода на C#, если он не содержит синтаксическую ошибку.**

**Если он содержит ошибку, то какую?**

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

**Console.WriteLine(System.IO.Path.GetExtension("file"));**

**}**

file

NullReferenceException

null

System.String

ArgumentException

пустая строка

**Ответ:** пустая строка

**Вопрос 12** **Выберите все верные утверждения**

метод Finalize() вызывает только сборщик мусора

метод Finalize() вызывает только сборщик мусора

сборщик мусора не умеет правильно разрушать типы, содержащие неуправляемые ресурсы. Код освобождения необходимо прописывать вручную

при освобождении IDisposable-обьектов, метод Dispose() рекомендуется вызывать в блоке finally

для очистки и повторного использования объекта класса из BCL, необходимо вызвать метод Dispose()

освобождение ресурсов в C# можно производить только путем вызова метода Dispose()

**Ответ:** метод Finalize() вызывает только сборщик мусора[[3]](#footnote-3)   
сборщик мусора не умеет правильно разрушать типы, содержащие неуправляемые ресурсы. Код освобождения необходимо прописывать вручную   
при освобождении IDisposable-обьектов, метод Dispose() рекомендуется вызывать в блоке finally

**Вопрос 13** **Какие наиболее известные API используются для реализации XML-анализаторов**

XML.API

SAX

SSX

DOM

MMX

DOC

**Ответ:** SAX   
DOM

**Вопрос 14** **Для чего используется перечисление System.IO.FileShare**

cодержит константы, позволяющие управлять типом доступа, который другие объекты FileStream могут осуществлять к тому же файлу

описывает какие права есть у других потоков, которые могут читать или записывать в файл позже

описывает каким образом файл должен быть открыт операционной системой

описывает, какой тип файла создать, бинарный или текстовый

определяет константы чтения, записи или чтения и записи файла

описывает, что делать с файлом после закрытия потока

**Ответ:** cодержит константы, позволяющие управлять типом доступа, который другие объекты FileStream могут осуществлять к тому же файлу

**Вопрос 15** **Выберите все верные утверждения**

сборка мусора в поколении 1 уничтожает объекты только этого поколения

кучу можно рассматривать как совокупность двух куч: куча больших объектов и куча маленьких объектов

сборка мусора в основном сводится к уничтожению короткоживущих объектов, что обычно занимают только небольшую часть кучи

большинство объектов уничтожаются при сборке мусора в последнем поколении

первое поколение называется поколение 1

размер выделенной памяти под поколения объектов строго фиксирован

**Ответ:** кучу можно рассматривать как совокупность двух куч: куча больших объектов и куча маленьких объектов   
сборка мусора в основном сводится к уничтожению короткоживущих объектов, что обычно занимают только небольшую часть кучи

**Вопрос 16** **для каких типов необходимо применять механизм финализации**

типов, что используют неуправляемый код

пользовательских типов данных

типов, что используют код из других языков программирования

типов, объекты которых создаются через оператор new

типов, что содержат ссылки на объекты классов

типов, что занимают много оперативной памяти

**Ответ:** типов, что используют неуправляемый код

**Вопрос 17** **Для чего используется тег remarks в xml-документации**

для описания какой базовый класс используется

для указания ссылок на комментарии в другом файле

для указания ссылки, что будет отображаться в разделе «смотри также»

для описания реализуемых интерфейсов

для описания дополнительной информации типе

для описания исключений, что могут произойти

**Ответ:** для описания дополнительной информации типе

**Вопрос 18** **Что произойдет в результате выполнения данного кода на C#, если он не содержит синтаксическую ошибку.**

**Если он содержит ошибку, то какую?**

**class Program {**

**static void Main()**

**{**

**string newStr =**

**System.Text.RegularExpressions.Regex.Replace("Обычная 12 строка с 75 числами",**

**@"[^[\D\s]]", string.Empty);**

**Console.WriteLine(newStr);**

**}}**

Обычная строка с числами (один пробел между словами)

FormatException

12 75

ArgumentException

Обычная строка с числами (может быть один или 2 пробела между словами)

1275

**Ответ:** Обычная строка с числами (один пробел между словами)[[4]](#footnote-4)

**Вопрос 19** **Выберите все верные утверждения**

при необходимости, метод Finalize() можно явно перегрузить

оператор using может использоваться для освобождения неуправляемых ресурсов

оператор using преобразуется в блок try-finally

try-finally используется только для освобождения ресурсов

как принимаемый параметр оператора using, можно использовать любой объект

оператор using преобразуется в блок try-catch-finally

**Ответ:** оператор using может использоваться для освобождения неуправляемых ресурсов   
оператор using преобразуется в блок try-finally

**Вопрос 20** **Имеет ли значение, использовать как разделитель уровней папок символ \ (слеш) или значение, возвращаемое свойством System.IO.Path.DirectorySeparatorChar**

имеет, если писать приложение для различных операционных систем, т.к. разделитель путей у них может отличаться

не имеет, т.к. везде используется разделитель / или \

имеет, т.к. везде используется разделитель :

имеет, т.к. везде используется разделитель / или :

не имеет, т.к. везде используется разделитель \

имеет, т.к. везде используется разделитель \ или :

**Ответ:** имеет, если писать приложение для различных операционных систем, т.к. разделитель путей у них может отличаться

**Вопрос 21** **Что произойдет в результате выполнения данного кода на C#, если он не содержит синтаксическую ошибку.**

**Если он содержит ошибку, то какую?**

**static void Main(){**

**string pattern = @"не";**

**string text = "Не может быть совсем не может быть";**

**var option = System.Text.RegularExpressions.RegexOptions.IgnoreCase;**

**var newReg = new System.Text.RegularExpressions.Regex(pattern, option);**

**var matches = newReg.Matches(text);**

**foreach (System.Text.RegularExpressions.Match mat in matches)**

**Console.WriteLine(**

**"Значение обьекта " +**

**mat.Value);**

**Console.WriteLine(**

**"Число найденных совпадений " + matches.Count);**

**}**

Значение обьекта Не  
 Значение обьекта не  
 Число найденных совпадений 2

Значение обьекта не  
 Значение обьекта не  
 Число найденных совпадений 2

Значение обьекта Не  
 Число найденных совпадений 1

Значение обьекта не  
 Число найденных совпадений 1

Значение обьекта Не  
 Значение обьекта Не  
 Число найденных совпадений 2

ArgumentException

**Ответ:** Значение обьекта Не   
 Значение обьекта не  
 Число найденных совпадений 2

**Вопрос 22** **Что произойдет в результате выполнения данного кода на C#, если он не содержит синтаксическую ошибку.**

**Если он содержит ошибку, то какую?**

**class Program {**

**static void Main(){**

**var fs = new System.IO.FileStream("file", System.IO.FileMode.OpenOrCreate,**

**System.IO.FileAccess.Write, System.IO.FileShare.Read);**

**var sw = new System.IO.BinaryWriter(fs);**

**sw.Write(12);**

**sw.Write("16");**

**sw.Flush();**

**sw.Close();**

**fs = new System.IO.FileStream("file", System.IO.FileMode.Open,**

**System.IO.FileAccess.Read, System.IO.FileShare.Read);**

**var s = new System.IO.BinaryReader(fs);**

**string str = s.ReadString();**

**int z = s.ReadInt32();**

**s.Close();**

**Console.WriteLine(str + " " + z);**

**}**

**}**

FormatException

FileNotFoundException

12 16

InvalidOperationException

16 12

EndOfStreamException

**Ответ:** EndOfStreamException

**Вопрос 23** **Для чего используется перечисление System.IO.FileAccess**

определяет константы чтения, записи или чтения и записи файла

описывает, какой тип файла создать, бинарный или текстовый

описывает какие права есть у других потоков, которые могут читать или записывать в файл позже

описывает необходимо ли вызывать метод Flush() автоматически

описывает, что делать с файлом после закрытия потока

описывает каким образом файл должен быть открыт операционной системой

**Ответ:** определяет константы чтения, записи или чтения и записи файла

**Вопрос 24** **Выберите все верные утверждения**

финализатор можно переопределять в классах и структурах

чтобы перегрузить метод Finalize(), необходимо реализовать интерфейс IDisposable

если управляемые объекты ссылаются на неуправляемые объекты, используя свои собственные дескрипторы файлов, разработчику необходимо явно освобождать неуправляемые объекты вручную

фоновая сборка мусора может удалить неиспользуемые объекты в эфемерных поколениях, а также при необходимости может расширить кучу во время сборки мусора поколения 1

третье поколение называется эфемерным

параллельная сборка мусора отключена в NET.Framework 4.5 и выше

**Ответ:** если управляемые объекты ссылаются на неуправляемые объекты, используя свои собственные дескрипторы файлов, разработчику необходимо явно освобождать неуправляемые объекты вручную   
фоновая сборка мусора может удалить неиспользуемые объекты в эфемерных поколениях, а также при необходимости может расширить кучу во время сборки мусора поколения 1

**Вопрос 25** **Для чего используется тег summary в xml-документации**

для описания исключений, что могут произойти

для описания, какие свойства и методы есть у типа

для описания типа или члена типа

для описания типа возвращаемого значения

для описания возможных возвращаемых значений

для указания ссылки на базовый класс

**Ответ:** для описания типа или члена типа

**Вопрос 26** **Что произойдет в результате выполнения данного кода на C#, если он не содержит синтаксическую ошибку.**

**Если он содержит ошибку, то какую?**

**static void Main(){**

**string s1 = "Не время для драконов 04:40:05\n04:40:55\n04:40:r5";**

**var reg = new System.Text.RegularExpressions.Regex(**

**@"[0-9]+:[0-9]+:[0-9]+",**

**System.Text.RegularExpressions.RegexOptions.IgnoreCase);**

**var mc = reg.Matches(s1);**

**foreach (System.Text.RegularExpressions.Match mat in mc)**

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

**Console.WriteLine(mc.Count.ToString());**

**}**

04:40:r5

04:40:r5

04:40:r5

3

InvalidOperationException

04:40:05  
 04:40:55  
 2

04:40:05  
1

04:40:05  
 04:40:55  
 04:40:r5  
 3

ArgumentException

**Ответ:** ArgumentException[[5]](#footnote-5)

**Вопрос 27** **Для чего используется перечисление System.IO.FileMode**

cодержит константы, позволяющие управлять типом доступа, который другие объекты FileStream могут осуществлять к тому же файлу

описывает, что делать с файлом после закрытия потока

описывает каким образом файл должен быть открыт операционной системой

описывает, какой тип файла создать, бинарный или текстовый

определяет константы чтения, записи или чтения и записи файла

описывает какие права есть у других потоков, которые могут читать или записывать в файл позже

**Ответ:** описывает каким образом файл должен быть открыт операционной системой

**Вопрос 28** **Что произойдет в результате выполнения данного кода на C#, если он не содержит синтаксическую ошибку. Если он содержит ошибку, то какую? Файла file.txt не существует до создания потока, другие ошибки произойти не могут.**

**class Program {**

**static void Main(){**

**using(var sw = new System.IO.StreamWriter("file.txt"))**

**sw.Write("hello");**

**}**

**}**

ArgumentException

FileNotFoundException

файл создастся

файл не создастся и сключения не произойдет

IOException

InvalidOperationException

**Ответ:** файл создастся

**Вопрос 29** **Что произойдет в результате выполнения данного кода на C#, если он не содержит синтаксическую ошибку.**

**Если он содержит ошибку, то какую?**

**StreamWriter sw = null;**

**try{ sw = new StreamWriter("file");**

**for(int i = 0; i < 10000; ++i)**

**sw.Write("Hello world!");**

**}catch(Exception ex){**

**Console.WriteLine("Output stream error");**

**}**

**finally{**

**if (sw != null) sw.Close();**

**}**

InvalidOperationException

не вызван метод Dispose()

не вызван метод Flush()

ArgumentException

запишет в файл 10000 строк "Hello world!"

отсутствует второй параметр для конструктора StreamWriter

**Ответ:** не вызван метод Flush()

**Вопрос 30** **Для чего используется тег see в xml-документации**

для описания ссылки на метод или поле, которое может вызываться из текущей среды компиляции

для описания типа или члена типа

для описания реализуемых интерфейсов

для описания, какие свойства и методы есть у типа

для указания ссылки, что будет отображаться в разделе «смотри также»

для описания дополнительной информации типе

**Ответ:** для описания ссылки на метод или поле, которое может вызываться из текущей среды компиляции

1. На самом деле - на экране отобразится "Hello c#!" [↑](#footnote-ref-1)
2. Выбрать одно из двух значений (вероятнее всего **первое**) – после сборки мусора все оставшиеся обьекты переходят в следующее поколение, если текущее не последнее [↑](#footnote-ref-2)
3. Выбрать одно из двух значений (вероятнее всего **первое**) – метод Finalize() вызывает только сборщик мусора [↑](#footnote-ref-3)
4. На самом деле – Обычная 12 строка с 75 числами [↑](#footnote-ref-4)
5. На самом деле – 04:40:05  
    04:40:55  
    2 [↑](#footnote-ref-5)