Учреждение образования «Полоцкий государственный университет»

Кафедра технологий программирования

Отчёт по курсу «Программирование на C#» на stepik.org

Дисциплина «Программирование на платформе .Net» (практическая часть курса)

Выполнил: 17-ИТ-4 Трубкин А.Д.

Проверил: преподаватель-стажер

Виноградова А.Д.,

старший преподаватель

Деканова М.В.

Полоцк, 2020

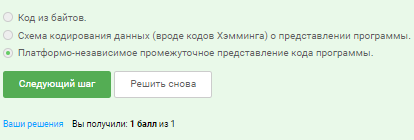
**Модуль 1 Управляемая (англ. «managed») платформа**

**Раздел 1.1Управляемый код**

**Задача 1.**

Условие.Что такое байт-код (англ. bytecode)?

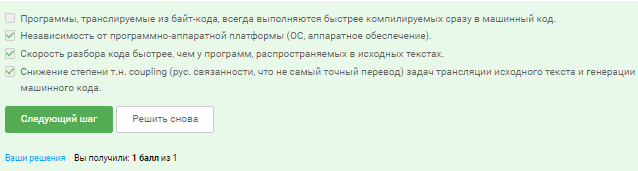
Результат:



**Задача 2.**

Условие.Какие, по мнению преподавателя, преимущества даёт байт-код представление программы? Может быть выбрано несколько пунктов.

Результат:



**Раздел 1.2Сборка мусора**

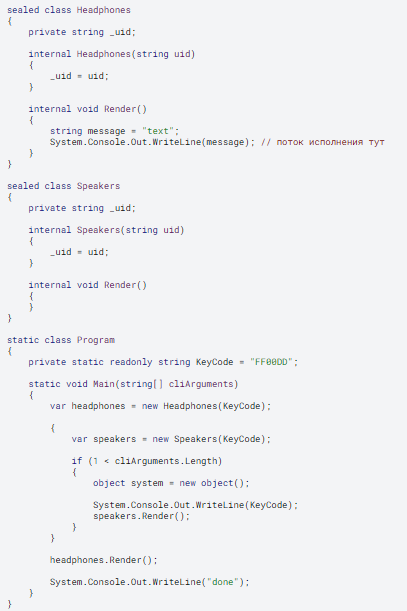
**Задача 1.**

Условие. Какие графовые структуры (в терминах объектов-вершин и ссылок-рёбер) не позволяет автоматически удалять алгоритм подсчёта ссылок? Может быть выбрано несколько пунктов.

Результат: Циклы объектов.

**Задача 2.**

Условие.Какие перечисленные ниже объекты являются корнями сборки мусора (англ. gcroots) в момент нахождения потока исполнения внутри метода Headphones.Render на строке System.Console.Out.WriteLine(message)?

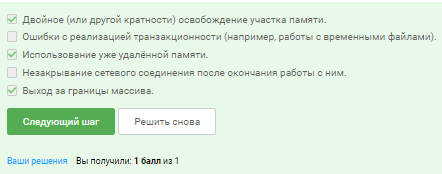


Результат:KeyCode, headphones, cliArguments, message.

**Задача 3.**

Условие.Какие из нижеперечисленных ошибок программ с ручным управлением памятью помогает не допускать механизм сборки мусора? Естественно, при условии, что программист максимально «изолирован» от понятия «указатель». Может быть выбрано несколько пунктов.

Результат:



**Раздел 1.3Класс System.Object**

**Задача 1.**

Условие. Имеют ли в C# общего предка (в терминах наследования классов) типы System.String и short?

Результат:Да.

**Задача 2.**

Условие. Как в C# сравнить на равенство адресов памяти две переменные-ссылки?

Результат:ReferenceEquals.

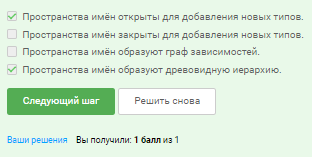
**Модуль 2 Простейшие конструкции и типы**

**Раздел 2.1** [**Файлы и модули**](https://stepik.org/lesson/209077/step/1?unit=182626)

**Задача 1.**

Условие.Выберите верные для пространств имён (англ. «namespaces») утверждения. Может быть выбрано несколько пунктов.

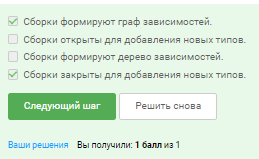
Результат:



**Задача 2.**

Условие. Выберите верные для сборок (англ. «assemblies») утверждения. Может быть выбрано несколько пунктов.

Результат:

****

**Раздел 2.2Значимые типы**

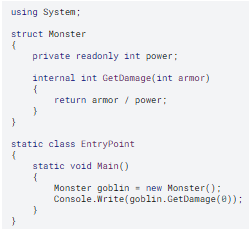
**Задача 1.**

Условие.В каких областях памяти могут размещаться объекты значимых (англ. value) типов? Может быть выбрано несколько пунктов.

Результат: Управляемая куча, Стек.

**Задача 2.**

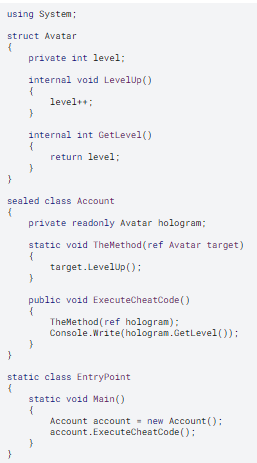
Условие. Что выведет в консоль этот код?



Результат: Будет выброшено исключение.

**Задача 3.**

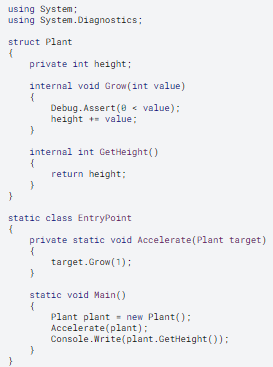
Условие.Что выведет в консоль этот код?



Результат: Код не скомпилируется.

**Задача 4.**

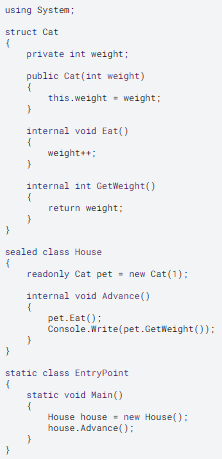
Условие.Что выведет в консоль этот код?



Результат: 0.

**Задача 5.**

Условие.Что выведет в консоль этот код?

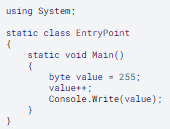


Результат: 1.

**Раздел 2.3Конвертация, частичные классы**

**Задача 1.**

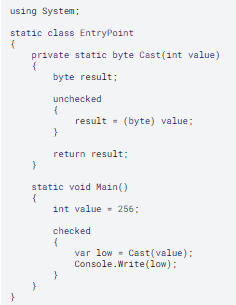
Условие.Что выведет в консоль этот код?



Результат:В зависимости от настроек компилятора: либо 0, либо будет выброшено исключение.

**Задача 2.**

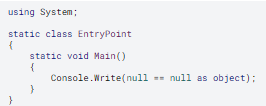
Условие.Что выведет в консоль этот код?



Результат:В зависимости от настроек компилятора: либо 0, либо будет выброшено исключение.

**Задача 3.**

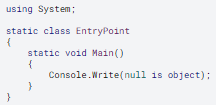
Условие.Хоть мы явно этот момент не проговаривали, попробуйте догадаться, что выведет в консоль этот код?



Результат:True.

**Задача 4.**

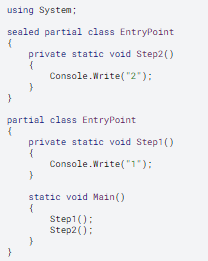
Условие.Хоть мы явно этот момент не проговаривали, попробуйте догадаться, что выведет в консоль этот код?



Результат:False.

**Задача 5.**

Условие.Что выведет в консоль этот код?

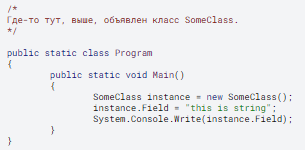


Результат:12.

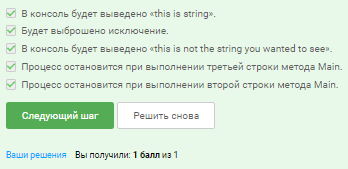
**Раздел 2.4Свойства**

**Задача 1.**

Условие.К чему из нижеперечисленного может привести выполнение следующего кода? Может быть выбрано несколько пунктов.

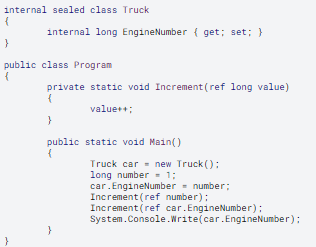


Результат:

****

**Задача 2.**

Условие.Что выведет в консоль этот код?



Результат:Код не скомпилируется.

**Задача 3.**

Условие.Реализуйте свойство HugeNumber типа long так, что б помимо функционала сохранения и возврата значения, на запись в него в консоль выводилось «set! » (без кавычек, с пробелом на конце), а на чтение — «get! » (тоже без кавычек, с пробелом на конце). Для вывода в консоль используйте [System.Console.Write](http://docs.microsoft.com/en-us/dotnet/api/system.console.write" \t "_blank).

Результат:

private long \_hugeNumber;

public long HugeNumber {

get {

Console.Write("get! ");

return \_hugeNumber;

}

set {

Console.Write("set! ");

\_hugeNumber = value;

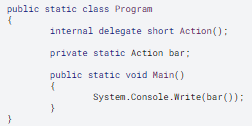
}

}

**Раздел 2.5Делегаты**

**Задача 1.**

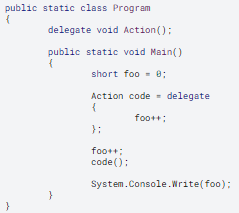
Условие.Что выведет в консоль этот код?



Результат: Будет выброшено исключение.

**Задача 2.**

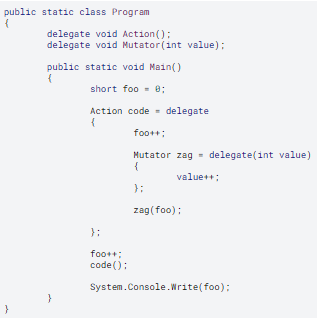
Условие.Что выведет в консоль этот код?



Результат: 2.

**Задача 3.**

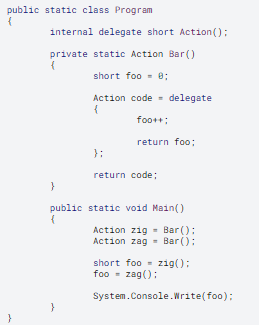
Условие.Что выведет в консоль этот код?



Результат:2.

**Задача 4.**

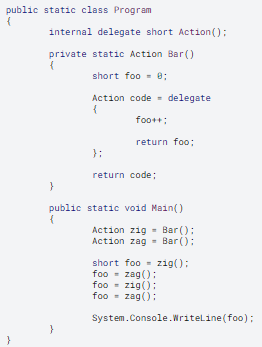
Условие.Что выведет в консоль этот код?



Результат:1.

**Задача 5.**

Условие.Что выведет в консоль этот код?

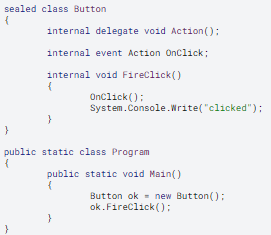


Результат:2.

**Раздел 2.6События**

**Задача 1.**

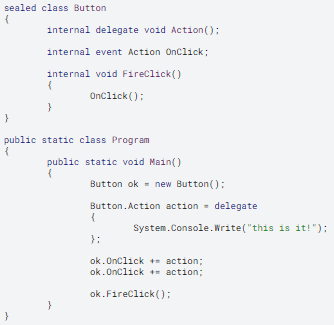
Условие.Что выведет в консоль этот код?



Результат: Будет выброшено исключение.

**Задача 2.**

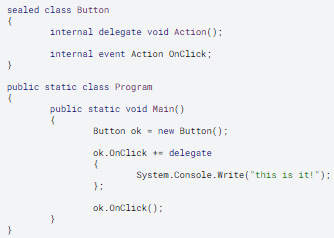
Условие.Что выведет в консоль этот код?



Результат:this is it! this is it!

**Задача 3.**

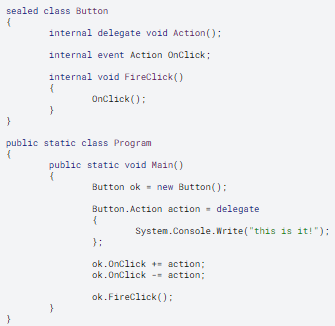
Условие.Что выведет в консоль этот код?



Результат: Код не скомпилируется.

**Задача 4.**

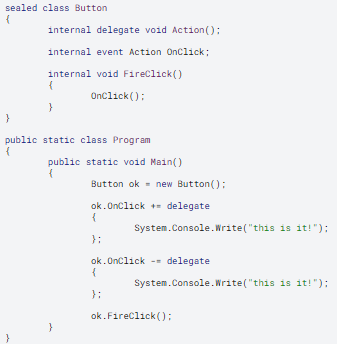
Условие.Что выведет в консоль этот код?



Результат: Будет выброшено исключение.

**Задача 5.**

Условие.Что выведет в консоль этот код?



Результат:this is it!.

**Задача 6.**

Условие.Реализуйте событие без аргументов c типом System.Action и именем OnEvent так, что б на добавление в него нового делегата в консоль выводилось «add!» (без кавычек), а на удаление «remove!» (тоже без кавычек). Для вывода в консоль используйте [System.Console.Write](http://docs.microsoft.com/en-us/dotnet/api/system.console.write" \t "_blank).Подсказка: вам могут пригодиться функции [Delegate.Combine](https://docs.microsoft.com/en-us/dotnet/api/system.delegate.combine?view=netframework-4.7.2" \t "_blank) и [Delegate.Remove](https://docs.microsoft.com/en-us/dotnet/api/system.delegate.remove?view=netframework-4.7.2" \t "_blank).

Результат:

private event Action \_onEvent;

internal event Action OnEvent

{

add

{

System.Console.Write("add!");

\_onEvent += value;

}

remove {

System.Console.Write("remove!");

\_onEvent -= value;

}

}

internal void FireEvent()

{

var tmp = \_onEvent;

if (tmp != null) tmp();

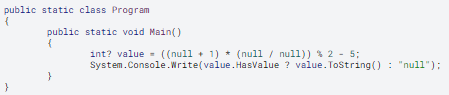
}

**Модуль 3 Продвинутые конструкции и типы**

**Раздел 3.1** [**Значи**](https://stepik.org/lesson/209077/step/1?unit=182626)**мые типы, допускающие присвоение null**

**Задача 1.**

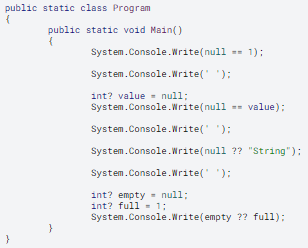
Условие.Что выведет в консоль этот код?



Результат:null.

**Задача 2.**

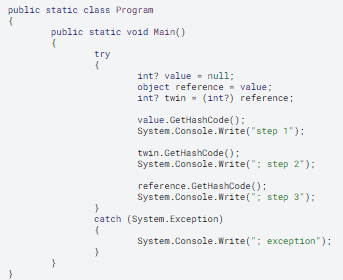
Условие.Что выведет в консоль этот код?



Результат:False True String 1!

**Задача 3.**

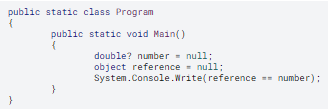
Условие.Что выведет в консоль этот код?



Результат:step 1; step 2; exeption.

**Задача 4.**

Условие.Хоть это и не обсуждалось в уроке, попробуйте догадаться, что выведет в консоль этот код?

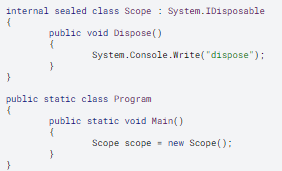


Результат:Код не скомпилируется.

**Раздел 3.2Интерфейс IDisposable**

**Задача 1.**

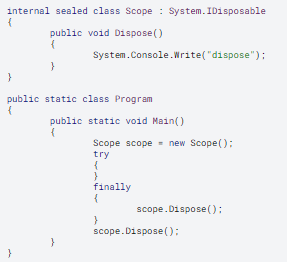
Условие.Что выведет в консоль этот код?



Результат:Ничего не выведет.

**Задача 2.**

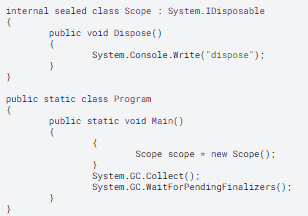
Условие.Что выведет в консоль этот код?



Результат:disposedispose

**Задача 3.**

Условие.Что выведет в консоль этот код?



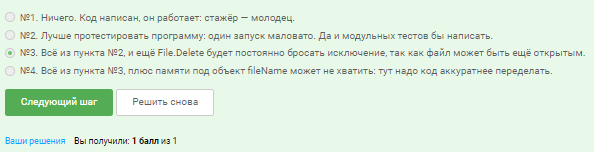
Результат: Ничего не выведет.

**Задача 4.**

Условие.Представим, что мы работаем над крупным проектом: программирует много людей, приложение исполняется в несколько нитей. Нам надо отрецензировать код стажёра:



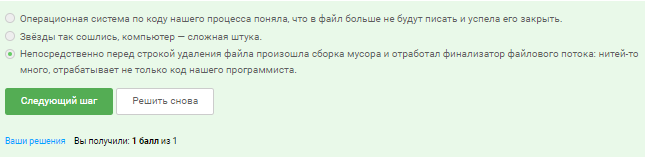
Результат:



**Задача 5.**

Условие.А как так вышло, что у стажёра из предыдущего шага код нормально отработал?

Результат:



**Задача 6.**

Условие.А как поправить код, что б файл удалялся вне зависимости от того, когда будет запущен сборщик мусора?

Напоминалка, код был таким:



Результат:using().

**Раздел 3.3**[**Интерфейс IEnumerable**](https://stepik.org/lesson/209077/step/1?unit=182626)

**Задача 1.**

Условие.Напишите статическую функцию Display, перебирающую и печатающую в консоль ([System.Console.Write](http://docs.microsoft.com/en-us/dotnet/api/system.console.write" \t "_blank)) не более десяти элементов, переданной ей последовательности ([Object.ToString](https://docs.microsoft.com/en-us/dotnet/api/system.object.tostring" \t "_blank)). Оператор foreach и функции LINQ, хоть мы до них ещё не дошли, использовать нельзя.

Результат:

private static void Display<T>(System.Collections.Generic.IEnumerable<T> values){

var ie = values.GetEnumerator();

int i = 0;

while(ie.MoveNext())

{

var item = ie.Current;

System.Console.Write(item);

if(i == 9) break;

i++;

}

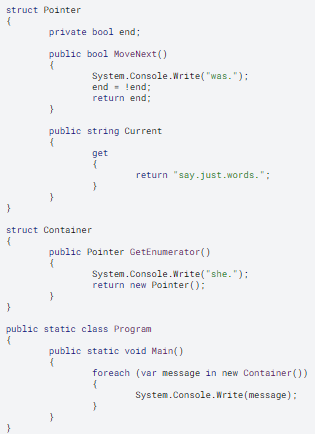
ie.Dispose();

}

**Раздел 3.4** [**Операторforeach**](https://stepik.org/lesson/209077/step/1?unit=182626)

**Задача 1.**

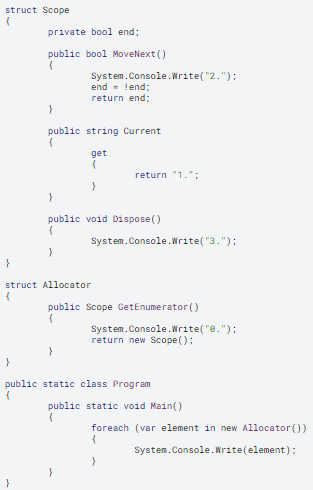
Условие.Что выведет в консоль этот код?



Результат:she.was.say.just.words.was.

**Задача 2.**

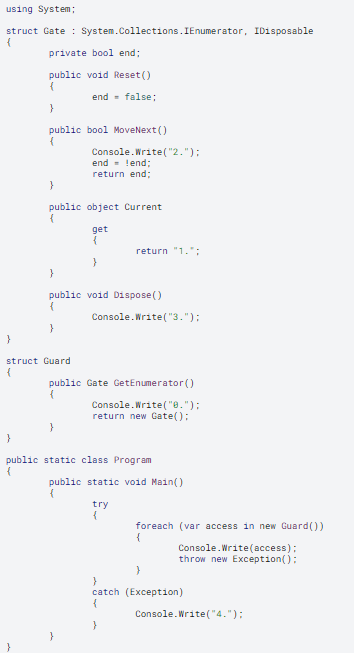
Условие.Что выведет в консоль этот код?



Результат:0.2.1.2.

**Задача 3.**

Условие.Что выведет в консоль этот код?

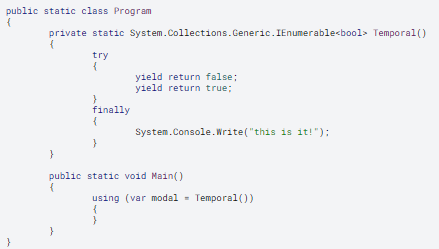


Результат:0.2.1.3.4.

**Раздел 3.5**[**Оператор yield**](https://stepik.org/lesson/209077/step/1?unit=182626)

**Задача 1.**

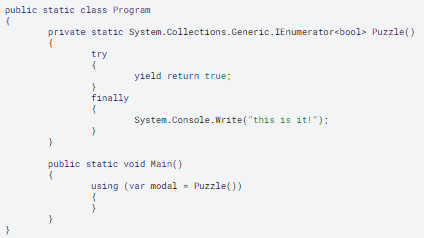
Условие.Что выведет в консоль этот код?



Результат:Код не скомпилируется.

**Задача 2.**

Условие.Что выведет в консоль этот код?



Результат:Ничего не скомпилируется.

**Задача 3.**

Условие.Напишите функцию Even, возвращающую последовательность чётных неотрицательных чисел типа [long](https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/keywords/long" \t "_blank). Давайте считать ноль чётным. Переполнение тоже можно не обрабатывать.

Результат:

private static System.Collections.Generic.IEnumerable<long> Even(){

long i = -2;

while (true)

yield return i += 2;

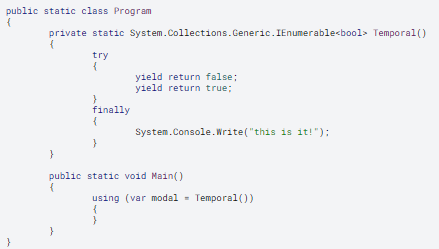
}

**Модуль 4Асинхронный ввод-вывод**

**Раздел 4.1 Формулировка задачи масштабируемости**

**Задача 1.**

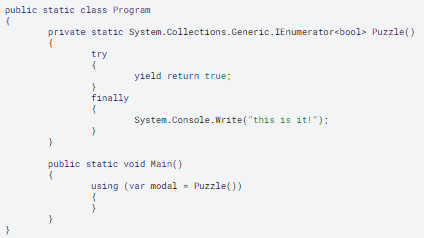
Условие.Что выведет в консоль этот код?



Результат:Код не скомпилируется.

**Задача 2.**

Условие.Что выведет в консоль этот код?



Результат:Ничего не скомпилируется.

**Раздел 4.2Асинхронный ввод-вывод в .Net**

**Задача 1.**

Условие.В предыдущем шаге преподаватель допустил описку в коде: устно сказал всё верно, а на доске нарисовал не то, что надо. Какая строка должна была быть на самом деле?

Результат:var data = new byte[fs.Length].

**Задача 2.**

Условие.Какой минимальный набор объектов нужно явно запросить у ОС Windows для организации асинхронного ввода-вывода? Может быть выбрано несколько пунктов.

Результат:Непосредственно объект ввода-вывода (файл или именованный канал или пр.).

**Раздел 4.3Интерфейс класса AsyncEnumerator**

**Задача 1.**

Условие.Какой механизм ОС Windows позволяет организовать прерывание (с последующим возобновлением) исполнения тела функции?

Результат:Fibers(волокна).

**Задача 2.**

Условие.Какой механизм языка C#, помимо оператора await, позволяет организовать прерывание (с последующим возобновлением) исполнения тела функции?

Результат: Оператор yield.

**Задача 3.**

Условие.Как итераторы, порождаемые оператором yieldreturn, так и волокна, являются объектами пользовательского адресного пространства. То есть, ответственность за их (фактически кооперативное) планирование лежит на плечах программиста, ОС о них «ничего не знает». И раз мы ограничены в количестве создаваемых нитей (объектов пространства ядра), то можно попробовать решить задачу масштабирования по количеству клиентов на неблокирующем I/O в пространстве пользователя.

Получится ли у нас волокнами (или итераторами) организовать переключение между задачами (клиентами) ввода-вывода при использовании классических, блокирующих поток исполнения, функций?

Результат:Нет.

**Раздел 4.4Реализация класса AsyncEnumerator**

**Задача 1.**

Условие.Хотим завести очередь. Какой тип стандартной библиотеки лучше всего для этого взять, если нет каких-то особенных требований?

Результат:System.Collections.Generic.Queue<t>.

**Задача 2.**

Условие.Как организовать в коде на C# критическую секцию (akamutex)?

Результат:Использовать оператор lock.

**Задача 3.**

Условие.Какими функциями можно организовать ожидание и пробуждение нити по событию в .NET (аналоги методов wait/notify из Java или conditionalvariables из библиотеки POSIX Threads)?

Результат:Monitor.Wait / Monitor.Pulse (Monitor.PulseAll).

**Раздел 4.5Оператор await**

**Задача 1.**

Условие.Напишите функцию,  которая асинхронно читает байты из одного потока, интерпретируя их, как ASCII-символы, и асинхронно записывает во второй, сохраняя порядок, лишь те, что находятся в интервале от 'a' до 'z'. То есть в выходной поток должны писаться лишь байты, являющиеся кодами строчных букв английского алфавита в кодировке [ASCII](https://docs.microsoft.com/en-us/dotnet/api/system.text.encoding.ascii?view=netframework-4.7.2#System_Text_Encoding_ASCII).

Результат:

private static async Task TransferSymbols(Stream source, Stream destination) {

var block = new byte[0x1000];

int read = 0;

while ((read = await source.ReadAsync(block, 0, block.Length)) != 0) {

string res = "";

int i = 0;

for (int runs = 0; runs <block.Length; runs++) {

byte c = block[runs];

if (c >= 97 & c <= 122) {

res += Convert.ToChar(c);}}

block = Encoding.ASCII.GetBytes(res);

await destination.WriteAsync (block, 0, block.Length);}}

**Раздел 4.6 Контекстсинхронизации**

**Задача 1.**

Условие.В какой нити исполнится код, следующий в функции после оператора await, если известно, что её вызвали из нити с умолчательным (англ. default) контекстом синхронизации. Ни в самой функции, ни в вызвавшем её коде, каких-либо действий по смене контекста синхронизации или переадресации задач не происходит. Оператор await был применён к «честной» асинхронной функции, прерывающей своё исполнение.

Результат:В одной из нитей пула по умолчанию (англ. Default thread pool).

**Задача 2.**

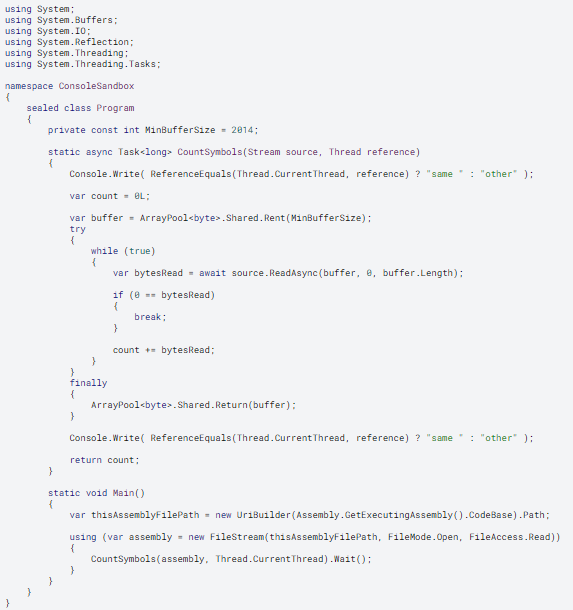
Условие.Выясните, какой контекст синхронизации у нити, исполняющей метод Main в консольном .NET-приложении при вашем окружении (ОС, виртуальная машина, версия стандартной библиотеки и пр.).

Верным ответом будет любая строка, так как сложно гарантировать, что на всём зоопарке реализаций .NET (.netcore, .netframework, mono и пр.) поведение будет одинаковым.

Результат:10.

**Задача 3.**

Условие.Что выведет в консоль этот код?



Результат:same other.

**Задача 4.**

Условие.Выясните (уже не «в уме», а используя среду разработки), что будет выведено в консоль, если в коде из предыдущего шага чтение файла сборки заменить на чтение объекта класса [System.IO.MemoryStream](https://docs.microsoft.com/en-us/dotnet/api/system.io.memorystream?view=netframework-4.7.2" \t "_blank), создаваемого вот так:varassembly = newMemoryStream(newbyte[31])

Результат:same same.

**Раздел 4.7Подводные камни**

**Задача 1.**

Условие.Можно ли воспользоваться первым советом урока и удалить оператор await из этого кода:



Результат:Нет.

**Задача 2.**

Условие.Что выведет в консоль этот код?

Результат: Будет выброшено исключение.

**Задача 3.**

Условие.Выясните, возникнет ли мёртвая блокировка, если в функции Main консольного приложения позвать метод Wait какой-либо задачи, полученной из асинхронной функции. Таким образом, проверим больше ли одной нити в контексте синхронизации.  
  
Верным ответом будет любая строка, так как сложно гарантировать, что на всём зоопарке реализаций .NET (.netcore, .netframework, mono и пр.) поведение будет одинаковым.

Результат: Нет.

**Модуль 5 LINQ (Language Integrated Query**

**Раздел 5.1 Введение**

**Задача1.**

Условие.Попробуйте вспомнить пример DSL (domain-specificlanguage), отличного от SQL, из своей практики или опыта. Ответ мы проверить не сможем: подойдёт любая строка.

Результат:Mongoquery.

**Раздел 5.2Запчасти**

**Задача1.**

Условие.Какой тип у переменной «android»?



Результат:Droid.

**Задача2.**

Условие.Какой тип у переменной «callback»?

var callback = (Func<int>) delegate{ return0; };

Результат:Func<int>.

**Задача3.**

Условие.Объявите переменную «cSharpPeople» — массив объектов анонимного типа, содержащего одно свойство «Name». Количество элементов в массиве — четыре. Значения имён для них: «AndersHejlsberg», «EricLippert», «LucaCardelli», «ErikMeijer».

Результат:

var cSharpPeople = new []{

new {Name = "Anders Hejlsberg"},

new {Name = "Eric Lippert"},

new {Name = "Luca Cardelli"},

new {Name = "Erik Meijer"}

};

**Задача4.**

Условие.Вам где-то свыше дан массив объектов анонимного типа: «monsters». Известно, что у его элементов есть два свойства: «Name» и «Level». Заведите переменную анонимного типа, взяв имя нулевого монстра, уровень первого и добавьте ему свойство «Strength» равное 0. Имя переменной — «frankenstein».

Для имени и уровня попробуйте использовать т.н. projectioninitialization: синтаксис без явного указания имени свойства анонимного типа.

Результат:

var frankenstein = new {

Name = monsters[0].Name,

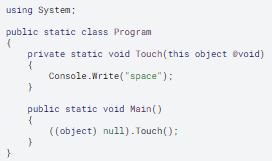
Level = monsters[1].Level,

Strength = 0

};

**Задача5.**

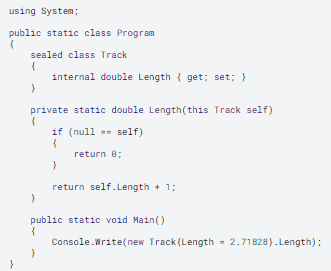
Условие.Что выведет в консоль этот код?



Результат:space.

**Задача6.**

Условие.Хоть мы явно этот момент не проговаривали, попробуйте догадаться, что выведет в консоль этот код?



Результат:2.71828.

**Задача7.**

Условие.Напишите метод-расширение CompareWithEpsilon для типа double с тремя параметрами: this, double-число с которым сравниваем, epsilon-погрешность типа double. Функция должна возвращать true, если для параметров (назовём их self, reference и epsilon) выполняется |self - reference| \leqepsilon∣*self*−*reference*∣≤*epsilon*, и false в противном случае.

Результат:

public static bool CompareWithEpsilon(this double self,

double reference,

double epsilon) =>Math.Abs(self - reference) <= epsilon;

**Задача8.**

Условие.Объявите переменную-делегат «aboveTheLimit» с помощью лямбда-выражения, возвращающую true в случае, если значение его единственного double-параметра больше чем 300000.0 и false в противном случае.

Подсказка-напоминание: свой тип для каждого делегата придумывать не обязательно, ведь есть [System.Func](https://docs.microsoft.com/en-us/dotnet/api/system.func-1?view=netframework-4.7.2" \t "_blank), [System.Action](https://docs.microsoft.com/en-us/dotnet/api/system.action?view=netframework-4.7.2" \t "_blank) и [System.Predicate](https://docs.microsoft.com/en-us/dotnet/api/system.predicate-1?view=netframework-4.7.2" \t "_blank).

Результат: Predicate <double>aboveTheLimit = (double num) =>num >= 300000.0;

**Раздел 5.3 СтроимLINQtoObjects**

**Задача1.**

Условие.Написать функцию «SelectMany», порождающую на каждый элемент входной последовательности новую и объединяющую их в одну на выходе.

Например, возьмём последовательность чисел: ***7, 13, 2***. И пусть у нас есть функция, для данного числа возвращающая тройку: само число, его квадрат и произведение на два. Тогда SelectMany из последовательности выше и означенной функции породит новую последовательность: ***7, 49, 14, 13, 169, 26, 2, 4, 4***.

Результат:

public static IEnumerable<TP>SelectMany<TS, TP>(this IEnumerable<TS> source, Func<TS, IEnumerable<TP>> produce)

{foreach (TS element in source) {

foreach (TP subElement in produce(element))

{yield return subElement; }}}

**Задача2.**

Условие.Написать функцию «Concat», конкатенирующую две последовательности в одну новую, где сначала идут элементы первой, а потом — второй последовательности.

Например, конкатенация двух последовательностей символов ***'s', 't', 'e', 'p', 'i', 'k'*** и ***'s', 'i', 't', 'e'*** должна давать на выходе одну: ***'s', 't', 'e', 'p', 'i', 'k', 's', 'i', 't', 'e'***.

Результат:

public static ICollection<T>Concat<T>

(

IEnumerable<T> first,

IEnumerable<T> second

)

{

List<T> res = new List<T>();

res.AddRange (first);

res.AddRange (second);

returnres;

}

**Задача3.**

Условие.Написать функцию «Zip», порождающую последовательность «пар» элементов из двух входных последовательностей («пара» склеивается из элементов, имеющих одинаковый порядковый номер в своих последовательностях).

Например, если в качестве «пары» брать произведение чисел, то из двух последовательностей натуральных чисел Zip выдаст последовательность их квадратов.

Результат:

public static IEnumerable<TResult> Zip<TL, TR, TResult>

(this IEnumerable<TL> left, IEnumerable<TR> right, Func<TL, TR, TResult>glue ){

using ( varleft\_enum = left.GetEnumerator())

using ( varright\_enum = right.GetEnumerator())

while (left\_enum.MoveNext() &&right\_enum.MoveNext()){

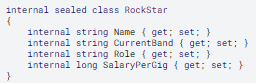
yield return glue (left\_enum.Current, right\_enum.Current);

}}

**Раздел 5.4 Query expressions syntax**

**Задача1.**

Условие.Поупражняемся в написании LINQ expressions. Вам сверху снизошёл массив stars с объектами типа RockStar:



Напишите выражение, выбирающее из массива имена рок-звёзд в порядке убывания их гонорара за концерт.

Подсказки:

* За сортировку в LINQ отвечает конструкция [orderby](https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/keywords/orderby-clause" \t "_blank).
* Содержимое массива stars:



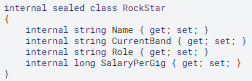
Результат:

from star in stars orderbystar.SalaryPerGig descending

selectstar.Name

**Задача 2.**

Условие.Продолжаем упражняться в написании LINQ expressions. Вам сверху снизошёл всё тот же массив stars с объектами типа RockStar:



Напишите выражение, выбирающее из массива членов группы DeepPurple и возвращающее последовательность объектов со свойствами Name (из такого же свойства объекта типа RockStar) и Instrument (из свойства Role объекта типа RockStar).

Подсказки:

* Содержимое массива stars:



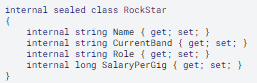
Результат:

from star in stars where star.CurrentBand == "Deep Purple"

select new { Name = star.Name, Instrument = star.Role }

**Задача 3.**

Условие.Продолжаем упражняться в написании LINQ expressions. Вам сверху снизошёл всё тот же массив stars с объектами типа RockStar:



Напишите LINQ выражение, которое группирует объекты из массива по полю CurrentBand и возвращает последовательность новых объектов с двумя полями: Name (общее значение свойства CurrentBand для объектов группы) и Size (сколько человек оказалось в группе).

У вас, конечно, получится написать код исключительно на extension-методах, но постарайтесь всё-таки использовать конструкцию from ... instars ... select ...

Подсказки:

* За группировку в LINQ отвечает конструкция [groupby](https://docs.microsoft.com/en-us/dotnet/csharp/linq/group-query-results" \t "_blank).
* Содержимое массива stars:



Результат:

from star in stars group star by star.CurrentBand into band

select new { Name = band.Key, Size = band.Count() }

**Раздел 5.5 LINQtoSQL**

**Задача1.**

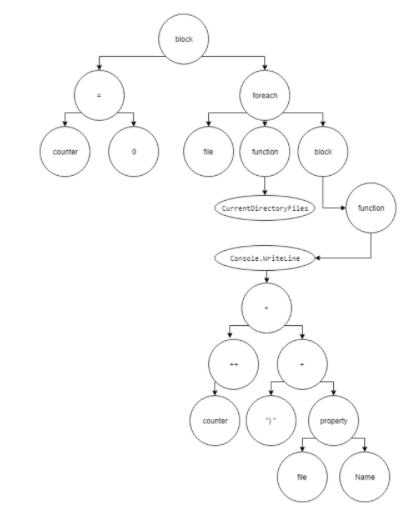
Условие.Что выведет в консоль этот код?



Результат:365.

**Задача 2.**

Условие.Какой код может описывать это AST?



Результат: var counter = 0; foreach (var file in CurrentDirectoryFiles()) { Console.WriteLine((counter++) + ") " + file.Name); }

**Задача 3.**

Условие.Какой минимальный набор интерфейсов нужно реализовать, если хотим организовать доступ к нашим данным через методы-расширения из [System.Linq](https://docs.microsoft.com/en-us/dotnet/api/system.linq?view=netframework-4.7.2" \t "_blank) с последующей генерацией кода стороннего языка запросов? (ака какие ключевые слова спрашивать у поисковика на эту тему).

Результат:IQueryable и IQueryProvider.

**Задача4.**

Условие.Как методы-расширения [IQueryable](https://docs.microsoft.com/en-us/dotnet/api/system.linq.iqueryable?view=netframework-4.7.2" \t "_blank), так и методы-расширения [IEnumerable](https://docs.microsoft.com/en-us/dotnet/api/system.collections.generic.ienumerable-1?view=netframework-4.7.2" \t "_blank) можно разделить на две группы: терминальные и нетерминальные (в англ. терминологии «deffered»: методы с отложенным исполнением). В чём между ними разница?

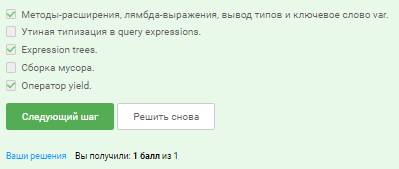
Результат:Терминальный метод производит вычисления над данными в момент его вызова, когда нетерминальный кэширует запрос для выполнения уже после окончания вызова.

**Раздел 5.6Заключение**

**Задача1.**

Условие.Без каких конструкций и механизмов, **по мнению преподавателя**, язык не может «заявлять», что в нём реализован LINQ? Т.е. отметьте запчасти из которых собран этот каркас.

Результат:



**Модуль6Интеграция в экосистему**

**Раздел6.1 Мотивация**

**Задача1.**

Условие.Что такое COM?

Результат:Бинарный протокол (стандарт) взаимодействия объектов, возможно находящихся в разных процессах.

**Задача2.**

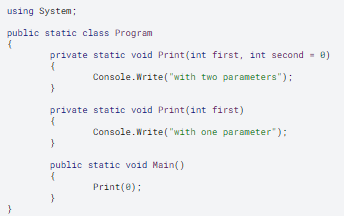
Условие.К какому классу сервисов, предоставляемых ОС, относится COM?

Результат:IPC (межпроцессное взаимодействие).

**Раздел6.2Опциональные параметры**

**Задача1.**

Условие.Хоть мы явно этот момент не проговаривали, попробуйте догадаться, что выведет в консоль этот код?



Результат:with one parameter.

**Задача 2.**

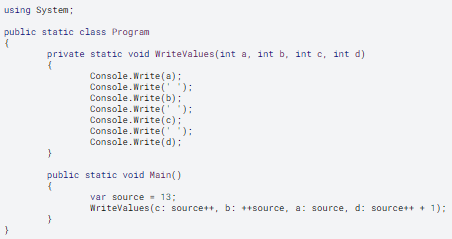
Условие.В какой момент происходит подстановка значения по умолчанию для параметра функции?

Результат:На этапе компиляции кода, вызывающего функцию.

**Раздел6.3Именованные аргументы**

**Задача1.**

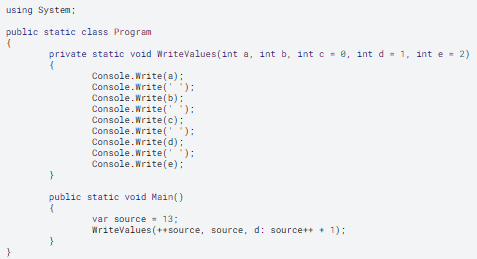
Условие.Хоть приведённый ниже код и является примером того, как не надо программировать, давайте разберёмся, что он выведет в консоль?



Результат:15 15 13 16.

**Задача 2.**

Условие.Снова пример не самого удачного дизайна (и имена переменных, кроме всего прочего, тут по-прежнему плохи), но для закрепления материала он подходит. Что выведет в консоль этот код?

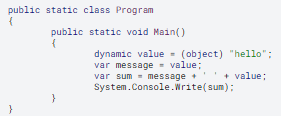


Результат:14 14 0 15 2.

**Раздел6.4Проверка типов на этапе исполнения**

**Задача1.**

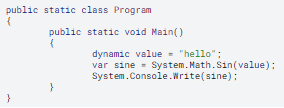
Условие.Что выведет в консоль этот код?



Результат:hellohello.

**Задача 2.**

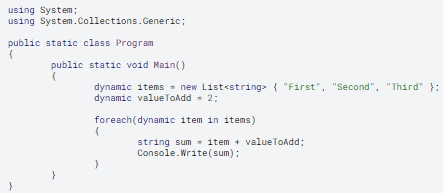
Условие.Что выведет в консоль этот код?



Результат:Будет выброшено исключение.

**Задача3.**

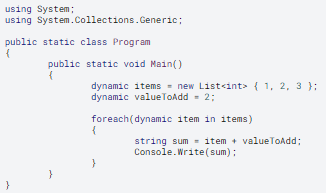
Условие.Что выведет в консоль этот код?



Результат:First2Second2Third2.

**Задача4.**

Условие.Что выведет в консоль этот код?



Результат:Будет выброшено исключение.

**Раздел6.5P/Involve**

**Задача1.**

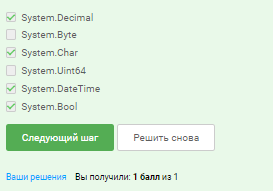
Условие.Зачем нужен механизм P/Invoke в платформе .NET?

Результат:Вызывать функции из динамически-загружаемых библиотек (DLL).

**Задача2.**

Условие.Какие из этих value-типов стандартной библиотеки **не** являются blittable? Может быть выбрано несколько пунктов.

Результат:



**Контрольная работа. Часть 1**

**Задача1.**

Условие.Какой из нижеперечисленных модификаторов доступа является самым строгим?

Результат:private.

**Задача2.**

Условие.Могут ли get и set одного property иметь отличные друг от друга модификаторы доступа?

Результат:Да.

**Задача3.**

Условие.Как без использования дополнительных структур данных вернуть несколько значений из метода?

Результат:ref и out.

**Задача4.**

Условие.Представим, что вы объявили в коде свой struct. Может ли его экземпляр как-нибудь оказаться не на стеке, а в динамической памяти?

Результат:Да.

**Задача5.**

Условие.В каких случаях создастся объект в динамической памяти (heap)? (MyClass – class, MyStruct - struct).

Результат:MyClassobj = newMyClass();

**Задача6.**

Условие.Что выведет следующий код?



Результат:oh.

**Задача7.**

Условие.Поддерживается ли в языке C# множественное наследование?

Результат:Нет, не поддерживается.

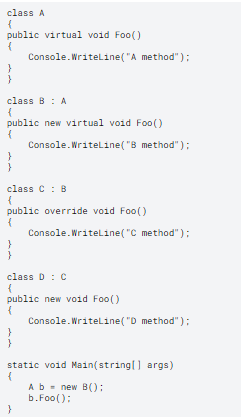
**Задача8.**

Условие.Какими могут быть методы в интерфейсах (отметить правильные варианты ответов)?

Результат:Выбор любого из других предложенных вариантов — неверное решение.

**Задача9.**

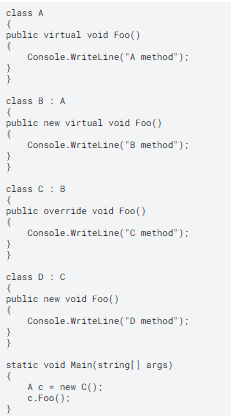
Условие.Что будет выведено после запуска функции Main?



Результат:A method.

**Задача10.**

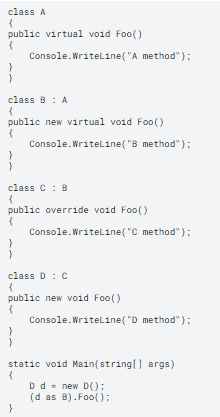
Условие.А в этом случае что будет выведено после запуска функции Main?



Результат:A method.

**Задача11.**

Условие.А в этом случае что будет выведено после запуска функции Main?

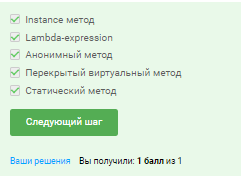


Результат:Cmethod.

**Задача12.**

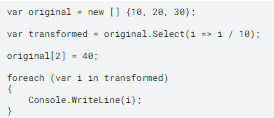
Условие.Что из нижеперечисленного можно передать в конструктор делегата?

Результат:



**Задача13.**

Условие.Что выведет следующий код?



Результат:1, 2, 4.

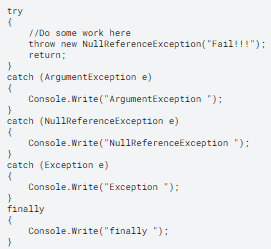
**Задача14.**

Условие.Следует ли избегать операций boxing'а и почему?

Результат:Да. Это дорогостоящий вычислительный процесс. Когда тип значения помещен в рамку, новый объект должен быть выделен и построен. В меньшей степени приведение, необходимое для распаковки, также требует больших вычислительных затрат.

**Задача15.**

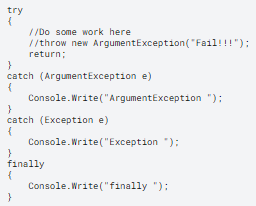
Условие.Что выведет следующий код?



Результат:NullReferenceExceptionfinally.

**Задача15.**

Условие.А этот код выведет что-нибудь?

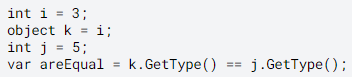


Результат:Выведетfinally.

**Контрольная работа. Часть 2**

**Задача1.**

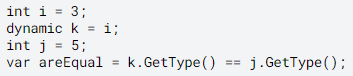
Условие.Что и почему будет лежать в переменной areEqual?



Результат:True, потому что k будет иметь тип int,и для каждого используемого типа в памяти будет находиться только один TypeObject.

**Задача2.**

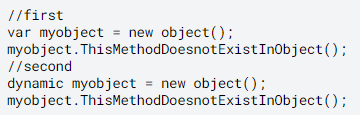
Условие.А в этом случае что и почему будет лежать в переменной areEqual?



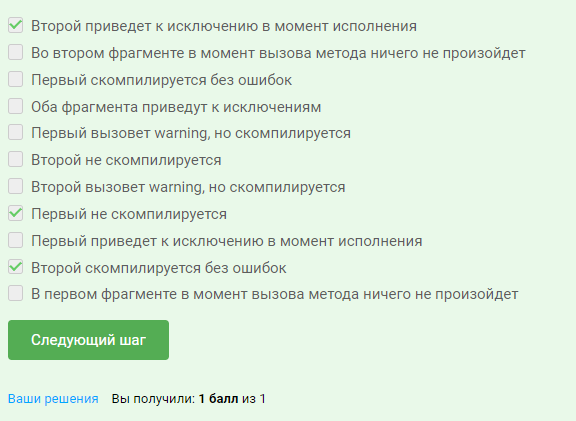
Результат:True, потому что k будет иметь тип int,и для каждого используемого типа в памяти будет находиться только один TypeObject.

**Задача3.**

Условие.Выберите верные высказывания о следующих фрагментах кода (у объекта типа object действительно нет такого метода)



Результат:



**Задача4.**

Условие.Возможны ли утечки памяти в C#?

Результат:Конечно, возможны и я готов указать (на следующем шаге), как их можно получить.

**Задача5.**

Условие.Если на предыдущем шаге вы выбрали третий ответ, то своё пояснение можете оставить тут.

Результат:Естественно, устроить утечку памяти можно и в программе на C#. Бывают эзотерические способы вроде: выделили и забыли освободить память вне управляемой кучи (посредством класса Marshal). Но это вырожденный случай: далеко не каждый проект использует такую память.

Для организации утечки исключительно «managed» средствами нужно к какому-то gcroot «подцепить» объект (прямо или опосредованно). Не исчерпывающим но часто встречающимся примером может служить схема с анонимным делегатом, который мы регистрируем, как функцию обратного вызова (англ. callback). Сам по себе объект-делегат занимает немного памяти. А вот объекты, на которые он ссылается через замыкание, могут быть объёмными. Например, наш делегат может что-то менять в пользовательском интерфейсе. Получается следующая последовательность: регистрируем делегат; при вызове он обращается к какому-то, скажем, окну; пользователь закрывает окно, мы в коде забываем убрать делегат из подписчиков на событие. Псевдо-код:

WindowbigGuiObject = … ;

lowBattaryPowerDetector.registerCallback(delegate{ bigGuiObject.ChangeTheme(); });