Экзамен ООП

Бурлаченко Егор

ІПЗ-20к-2

Билет 21.

Вопрос 61.

Массив представляет собой совокупность переменных одного типа с общим для обращения к ним именем. В C# массивы могут быть как одномерными, так и многомерными. Массивы служат самым разным целям, поскольку они предоставляют удобные средства для объединения связанных вместе переменных.

Структура ( struct ) в C# — это пользовательский тип данных, который используется наряду с классами и может содержать какие-либо данные и методы. Структурами также являются такие типы данных как int , double и т. д.

В C# существуют две разновидности типов: ссылочные типы и типы значений. В переменных ссылочных типов хранятся ссылки на их данные (объекты), а переменные типа значений содержат свои данные непосредственно. Две переменные ссылочного типа могут ссылаться на один и тот же объект, поэтому операции над одной переменной могут затрагивать объект, на который ссылается другая переменная. При использовании типов значений каждая переменная имеет собственную копию данных, и операции с одной переменной не могут повлиять на другую (за исключением inпеременных параметров и refout переменных параметров; см. модификатор параметровin, ref и out).

Для объявления ссылочных типов используются следующие ключевые слова:

* class
* interface
* delegate
* record

В C# также предусмотрены следующие встроенные ссылочные типы:

* dynamic
* object
* строка

Строка — это объект типа String, значением которого является текст. Внутри программы текст хранится в виде упорядоченной коллекции объектов Char только для чтения. В конце строки C# нет символа, завершающего значение NULL; поэтому строка C# может содержать любое количество внедренных символов NULL ('\0').

Операция упаковки boxing характеризуется выделением памяти в управляемой куче (managed heap) под объект value type и дальнейшее присваивание указателя на этот участок памяти переменной в стеке.

Распаковка unboxing, напротив, выделяет память в стеке выполнения под объект, полученный из управляемой кучи с помощью указателя.

Казалось бы, в обоих случаях выделяется память и особой разницы быть не должно, если бы не одно но- крайне важной деталью является область памяти.

В случае же с распаковкой, память выделяется в стеке выполнения, который содержит указатель на свой конец, по совместительству являющийся началом участка памяти под новый объект.

Вывод из этого я делаю такой, что процесс упаковки должен занимать значительно больше времени, чем распаковки, ввиду возможных side effects связанных с GC и медленной скоростью выделения памяти/копирования значения в managed heap.

Приведение (превращение) типов (англ. type conversion, typecasting, coercion) — в программировании это изменение типа сущности одного типа данных в другой, что может происходить разными способами, явно или неявно. Это используется с целью воспользоваться некоторыми особенностями иерархий типов или представления типа.

Вопрос 62.

using System;

using System.IO;

namespace FileHandlingArticleApp

{

class Program

{

static void Main(string[] args)

{

if (File.Exists("TextFile1.txt"))

{

string content = File.ReadAllText("TextFile1.txt");

Console.WriteLine("Current content of file:");

Console.WriteLine(content);

}

Console.WriteLine("Please enter new content for the file:");

string newContent = Console.ReadLine();

File.AppendAllText("TextFile1.txt", newContent + "\n");

}

}

} Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание