Турсунов Баходурхон

1. **В чем разница между статическим и динамическим массивом и как понятие фактора роста связано с динамическими массивами?**

Статический массив имеет фиксированный размер, заранее выделенный в памяти, и не может изменяться. Динамический массив, напротив, может изменять свой размер в процессе выполнения программы, используя динамическое выделение памяти. Фактор роста в контексте динамических массивов определяет, насколько увеличится размер массива при необходимости его расширения, что позволяет оптимизировать операции изменения размера и улучшить время выполнения операций.

1. **Опишите процесс извлечения элемента из массива и объясните, почему он может быть сложным при работе с большими массивами.**

Извлечение элемента из массива включает обращение к нему по индексу. Операция может усложниться при работе с большими массивами из-за возможных кэш-промахов, медленного доступа к оперативной памяти, сложностей в управлении памятью и алгоритмических вызовов. Объем данных также влияет на производительность, требуя оптимизированных методов доступа для уменьшения времени операции.

1. **Как объявляется инлайн-массив и какой атрибут используется для определения его длины?**

Объявление: MyInlineArray inlineArray = new();

Определение длины: [System.Runtime.CompilerServices.InlineArray(length: 10)]

1. **Сравните и противопоставьте цикл while и цикл for с точки зрения синтаксиса и типичных случаев использования.**

* **Цикл** **while** – while (условие) {}

Применяется, когда условие выполнения необходимо проверять перед каждой итерацией, и количество итераций заранее неизвестно.

* **Цикл for –** for (инициализация; условие; выражение\_итерации) {}

Цикл удобен, когда известно количество итераций и требуется более компактный синтаксис для инициализации, условию и итерации. Он часто используется при работе с коллекциями и массивами.

1. **Что такое бесконечный цикл в программировании и как из него выйти?**

Это цикл, который продолжает выполняться бесконечно, так как его условие всегда истинно. Бесконечные циклы могут быть созданы непреднамеренно или специально в случаях, когда требуется бесконечное выполнение определенных операций.

Применение:

While (true){}

Как выйти:

While (true) {

If(условие\_выхода){

Break;

}

}

Оператор break используется для прерывания выполнения цикла, а изменение условия позволяет контролировать, когда цикл завершится. Важно обеспечивать условия выхода, чтобы избежать бесконечных циклов и управлять выполнением программы.s

Practice:

1. Напишите программу, которая проверяет, содержится ли заданное число в массиве.

```

int[] arr = { 2, 3, 4, 5, 6, 5, 4, 3, 2, 3, 45, 56, 6, 45, 4 };

int isHaveNumInArr = int.Parse(Console.ReadLine());

int counter = 0;

foreach (var num in arr)

{

if (isHaveNumInArr == num)

{

counter++;

}

}

Console.WriteLine($"Заданное число в массиве встречается {counter} раза");

// 2 вариант

//for (int i = 0; i < arr.Length; i++)

//{

// if (isHaveNumInArr == arr[i]) counter++;

//}

//Console.WriteLine($"Заданное число в массиве встречается {counter} раза");

```

1. Создайте массив чисел. Используя цикл, найдите и выведите сумму всех элементов массива.

```

int[] arr = { 5,5,2,3,4,56,7,89,9,7,6,45,43,3,2,12,23,4,5,6};

int sumNumOfArr = 0;

for (int i = 0; i < arr.Length; i++)

{

sumNumOfArr += arr[i];

}

Console.WriteLine($"Сумма чисел в массиве равна {sumNumOfArr}");

```

1. Создайте строку и используйте цикл для вывода каждого символа строки на новой строке.

```

string str = "Hello, world!";

for (int i = 0; i < str.Length; i++)

{

Console.WriteLine(str[i]);

}

```