Лекция 9. Цикл foreach с многомерными массивами

Цель лекции: Формирование у студентов знаний, связанных с использованием оператора цикла foreach с многомерными массивами.
В результате изучения данной лекции студенты будут способны:

- Продемонстрировать знание синтаксиса использования оператора foreach
- Продемонстрировать понимание особенности работы оператора foreach с многомерными массивами

Оператор цикла foreach можно также использовать для циклического обращения к элементам многомерного массива. В этом случае элементы многомерного массива возвращаются по порядку следования строк от первой до последней, как демонстрирует приведенный ниже пример программы.

```
// Использовать оператор цикла foreach для обращения к двумерному массиву.
using System;
class ForeachDemo2 {
static void Main() {
int sum = 0;
int[,] nums = new int[3,5];
// Задать первоначальные значения элементов массива nums.
for (int i = 0; i < 3; i++)
for (int j=0; j < 5; j++)
nums[i,j] = (i+l)*(j+l);
// Использовать цикл foreach для вывода значений
// элементов массива и подсчета их суммы,
foreach(int x in nums) {
Console.WriteLine("Значение элемента равно: " +
x);
sum += x;
}
Console.WriteLine("Сумма равна: " + sum);
}
}
```



Выполнение этой программы дает следующий результат.

```
Значение элемента равно: 1
Значение элемента равно: 2
Значение элемента равно: 3
Значение элемента равно: 4
Значение элемента равно: 5
Значение элемента равно: 2
Значение элемента равно: 4
Значение элемента равно: 6
Значение элемента равно: 8
Значение элемента равно: 10
Значение элемента равно: 3
Значение элемента равно: 6
Значение элемента равно: 9
Значение элемента равно: 12
Значение элемента равно: 15
Сумма равна: 90
```



Оператор foreach допускает циклическое обращение к массиву только в определенном порядке: от начала и до конца массива, поэтому его применение кажется, на первый взгляд, ограниченным. Но на самом деле это не так. В большом числе алгоритмов, самым распространенным из которых является алгоритм поиска, требуется именно такой механизм. В качестве примера ниже приведена программа, в которой цикл foreach используется для поиска в массиве определенного значения. Как только это значение будет найдено, цикл прервется.

```
// Поиск в массиве с помощью оператора цикла
foreach.
using System;
class Search {
static void Main() {
int[] nums = new int [10]; int val;
```

```
bool found = false;
// Задать первоначальные значения элементов массива nums.
for (int i = 0; i < 10; i++) nums[i] = i;
val = 5;
// Использовать цикл foreach для поиска заданного
// значения в массиве nums.
foreach(int x in nums) {
if(x == val)    {
found = true; break;
}
}
```



```
if(found)
Console.WriteLine("Значение найдено!");
}
}
```

При выполнении этой программы получается следующий результат.

Значение найдено!

Оператор цикла foreach отлично подходит для такого применения, поскольку при поиске в массиве приходится анализировать каждый его элемент. К другим примерам применения оператора цикла foreach относится вычисление среднего, поиск минимального или максимального значения среди ряда заданных значений, обнаружение дубликатов и т.д. Как будет показано далее в этой книге, оператор цикла foreach оказывается особенно полезным для работы с разными типами коллекций.

Заключение

В рамках данной лекции были изучены особенности применения оператора цикла `foreach` для обработки многомерных массивов и реализации типовых алгоритмов анализа данных в языке C#. Были рассмотрены порядок обхода элементов многомерного массива (построчно от начала до конца), использование оператора `break` при поиске значений, а также круг практических задач, для которых `foreach` является наиболее эффективным решением.



Применение оператора `foreach` при работе с многомерными массивами позволяет заменить громоздкие вложенные циклы `for` единой лаконичной конструкцией в тех случаях, когда требуется последовательно обработать все элементы без привязки к их конкретным индексам. Это делает код более читаемым, снижает риск ошибок при работе с границами измерений и оптимально подходит для алгоритмов поиска, подсчета суммы, вычисления среднего арифметического, а также нахождения минимальных и максимальных значений.

Вопросы для самопроверки

1. **Порядок обхода:** В какой последовательности оператор цикла `foreach` извлекает элементы двумерного или многомерного массива?

2. **Преимущество перед `for`:** В чем заключается главное синтаксическое преимущество использования цикла `foreach` по сравнению с циклом `for` при необходимости просуммировать все элементы трехмерного массива?

3. **Алгоритм поиска:** Как с помощью цикла `foreach` и оператора `break` организовать эффективный поиск заданного значения в массиве? Зачем в этом алгоритме используется оператор `break`?

4. **Практическое применение:** Перечислите типовые алгоритмические задачи, для решения которых использование оператора `foreach` является наиболее удобным и оправданным.

5. **Ограничения оператора:** В каких задачах обработки многомерных массивов (например, при работе с матрицами) цикл `foreach` применить не удастся и потребуется вернуться к вложенным циклам `for`?

Список используемых источников

1. **Шилдт Г.** C# 4.0: полное руководство. — М.: Вильямс, 2019. — 1056 с. *(Глава 7: Массивы и строки — раздел «Применение цикла foreach с многомерными массивами»).*

2. **Троелсен Э., Джепикс Ф.** Язык программирования C# и платформа .NET. — СПб.: Диалектика, 2022. — 1312 с. *(Раздел: Итерационные конструкции и работа с массивами).*

3. **Албахари Д., Албахари Б.** C#. Справочник. Полное описание языка. — СПб.: БХВ-Петербург, 2023. — 1088 с. *(Спецификация инструкции `foreach` для многомерных массивов).*

4. **Павловская Т. А.** C#. Программирование на языке высокого уровня: Учебник для вузов. — СПб.: Питер, 2020. — 432 с. *(Глава: Обработка массивов и типовые алгоритмы).*

5. **Официальная документация Microsoft Learn:** Использование инструкции `foreach` с массивами (руководство по программированию на C#) [Электронный ресурс]. — URL: microsoft.com (дата обращения: 13.11.2026).