FOREACH CYCLE OPERATOR

Mamarajabov Jahongir Sherzod o’g’li

Student of the Department of Computer Science and Programming Technologies,
Faculty of Information Technology, Termez State University

jahongirmamarajabov780@gmail.com

Abstract

The foreach loop is primarily designed to make working with arrays and collection values easier, allowing you to refer to each element of an array without knowing the number of elements in it. The foreach statement: enumerates the elements of a collection and executes its body for each element of the collection. This article discusses the foreach operator and how to write programs using this operator, test and execute different parts of the code based on test results.

Keywords: the foreach operator, program, network, variables, branching program.

Аннотация

Цикл foreach в первую очередь предназначен для упрощения работы с массивами и значениями коллекций, позволяя обращаться к каждому элементу массива, не зная количества элементов в нем. Оператор foreach: перечисляет элементы коллекции и выполняет свое тело для каждого элемента коллекции. В этой статье обсуждается оператор foreach и как писать программы с использованием этого оператора, тестировать и выполнять различные части кода на основе результатов тестирования.

Ключевые слова: оператор foreach, программа, сеть, переменные, программа ветвления.

In computer programming, foreach loop (or for each loop) is a control flow statement for traversing items in a collection. Foreach is usually used in place of a standard for loop statement. Unlike other for loop constructs, however, foreach loops usually maintain no explicit counter: they essentially say "do this to everything in this set", rather than "do this x times". This avoids potential off-by-one errors and makes code simpler to read. In object-oriented languages, an iterator, even if implicit, is often used as the means of traversal.

The foreach statement in some languages has some defined order, processing each item in the collection from the first to the last. The foreach statement in many other languages, especially array programming languages, does not have any particular order. This simplifies loop optimization in general and in particular allows vector processing of items in the collection concurrently.

Programming languages which support foreach loops include ABC, ActionScript, Ada, C++11, C#, ColdFusion Markup Language (CFML), Cobra, D, Daplex (query language), Delphi, ECMAScript, Erlang, Java (since 1.5), JavaScript, Lua, Objective-C (since 2.0), ParaSail, Perl, PHP, Prolog, Python, R, REALbasic, Rebol, Red, Ruby, Scala, Smalltalk, Swift, Tcl, tcsh, Unix shells, Visual Basic .NET, and Windows PowerShell. Notable languages without foreach are C, and C++ pre-C++11.

The C language does not have collections or a foreach construct. However, it has several standard data structures that can be used as collections, and foreach can be made easily with a macro.

However, two obvious problems occur:

The macro is unhygienic: it declares a new variable in the existing scope which remains after the loop.

One foreach macro cannot be defined that works with different collection types (e.g., array and linked list) or that is extensible to user types.

```
var fibNumbers = new List<int> { 0, 1, 1, 2, 3, 5, 8, 13 };
foreach (int element in fibNumbers)
{
   Console.Write($"{element} ");
}
// Output:
        // 0 1 1 2 3 5 8 13
```

The foreach statement isn't limited to those types. You can use it with an instance of any type that satisfies the following conditions:

A type has the public parameterless GetEnumerator method. Beginning with C# 9.0, the GetEnumerator method can be a type's extension method.

The return type of the GetEnumerator method has the public Current property and the public parameterless MoveNext method whose return type is bool.

The following example uses the foreach statement with an instance of the System.Span<T> type, which doesn't implement any interfaces:

```
Span<int> numbers = new int[] { 3, 14, 15, 92, 6 };
foreach (int number in numbers)
{
    Console.Write($"{number} ");
}
// Output:
        // 3 14 15 92 6
```

If the enumerator's Current property returns a reference return value (ref T where T is the type of a collection element), you can declare an iteration variable with the ref or ref readonly modifier, as the following example shows:

```
Span<int> storage = stackalloc int[10];
int num = 0;
foreach (ref int item in storage)
{
    item = num++;
}
foreach (ref readonly var item in storage)
{
    Console.Write($"{item} ");
}
// Output:
        // 0 1 2 3 4 5 6 7 8 9
```

If the foreach statement is applied to null, a NullReferenceException is thrown. If the source collection of the foreach statement is empty, the body of the foreach statement isn't executed and skipped.

Foydalanilgan adabiyotlar

1. "D Programming Language `foreach` Statement Documentation". Digital Mars. Retrieved 2008-08-04.
2. "SWI-Prolog -- foreach/2". www.swi-prolog.org. Retrieved 2020-02-10.
3. http://www.rebol.com. `{{`cite web`}}`: Missing or empty `|title=` (help)
4. http://www.red-lang.org. `{{`cite web`}}`: Missing or empty `|title=` (help)
5. "Proposed ECMAScript 4th Edition – Language Overview" (PDF). Retrieved 2020-02-21.
6. "Enhanced for Loop - This new language construct[...]" "Java Programming Language, Section: Enhancements in JDK 5". Sun Microsystems, Inc. *2004*. Retrieved 2009-05-26.