Темы урока

Циклы

- Циклы являются такой же важной частью структурного программирования, как условные операторы.
- С помощью циклов можно организовать повторение выполнения участков кода.
- Обычно разделяют следующие виды циклов:
    - Циклы **с постусловием**
        - Сначала выполняем блок действий, потом смотрим, не надо ли повторить выполнение этого блока ещё раз?
    - Циклы **с предусловием**
        - Сначала смотрим, должны ли мы выполнить определенный блок действий.
            - В случае если да, выполняем блок действий снова задаемся вопросом о необходимости повторения.
            - Если нет, пропускаем блок действий тела цикла, переходим к следующим действиям этого уровня вложенности.
    - Циклы **со счетчиком**
    - Так называемые, **совместные циклы**. Они задают выполнение некоторой операции для объектов из заданного множества, без явного указания порядка перечисления этих объектов.

Цикл с постусловием: do...while

- Оператор do выполняет определенный блок кода, пока условное логическое выражение истинно.
- Выполняется как минимум один раз.

Самостоятельная работа

Написать программу, которая будет запрашивать строку у пользователя (никуда её не сохраняя) до тех, пока пользователь не введет “exit”.

```
string str;
Console.WriteLine("Enter text string (or \"exit\" to finish):");
do
{
	str = Console.ReadLine();
} while (str != "exit");
```

Досрочный выход из цикла: break

- Команда досрочного выхода break применяется, когда необходимо прервать выполнение цикла, в котором условие выхода ещё не достигнуто (в середине зацикленного кода.
- **Например**, когда при выполнении тела цикла обнаруживается ошибка, после которой дальнейшая работа цикла не имеет смысла.
- **Другой пример**: можно сделать вечный цикл *do {} while (true)*, у которого единственной точкой выхода и будет break.

Самостоятельная работа

Модифицировать программу таким образом, чтобы цикл был вечным, а выход осуществлялся бы с использованием оператора break.

```
string str;
Console.WriteLine("Enter text string (or \"exit\" to finish):");
do
{
	str = Console.ReadLine();
	if (str == "exit")
		break;
} while (true);
```

Пропуск итерации: continue

- Данный оператор применяется, когда:
  - В текущей итерации цикла необходимо пропустить все команды до конца тела цикла.
  - При этом сам цикл прерываться не должен, условия продолжения или выхода должны вычисляться обычным образом.

Самостоятельная работа

Модифицировать программу таким образом, чтобы на экран выводилась длина введенной строки в формате “Entered string length is X” для всех строк, короче или равным 15 символам.
Для строк больше 15 символов должна выводиться надпись “Too long string. Try another:”, и после вывода этой надписи цикл должен возвращался к началу.

```
string str;
Console.WriteLine("Enter text string (or \"exit\" to finish):");
do
{
	str = Console.ReadLine();
	if (str.Length > 15)
	{
		Console.WriteLine($"Too long string. Try another:");
		continue;
	}

	Console.WriteLine($"Entered string length is {str.Length}.");
	if (str == "exit")
		break;

} while (true);
```

Цикл с предусловием: while

- Оператор while выполняет определенный блок кода, если условное логическое выражение равно значению true.
- Выполняется 0 или более раз.

Самостоятельная работа

Написать программу, которая бы считала сумму чисел, расположенную в заданном массиве используя цикл while. В каждой итерации цикла необходимо выводить промежуточный результат подсчета суммы.

```
var arr = new[] { 7, 43, 23, 32, 34 };

int i = 0;
int sum = 0;
while (i < arr.Length)
{
	sum += arr[i];
	i++;
	Console.WriteLine($"Intermediate sum is {sum}.");
}

Console.WriteLine($"The sum is {sum}");
```

Цикл со счетчиком: for

- Оператор for выполняет определенный блок кода, каждую итерацию изменяя значение счётчика, пока условное логическое выражение истинно.
- Выполняется 0 и более раз.
- Удобен для перебора элементов массива.

Самостоятельная работа

Дан двумерный массив целых чисел. Чтобы было нагляднее, давайте представим, что это оценки, полученные за неделю неким учеником, разбитые по дням недели: 0 - Пн, 1 - Вт, и т.д. Необходимо написать программу, которая бы посчитала и вывела на экран средний балл по каждому дню и суммарный средний бал за неделю.
Средние баллы должны быть выведены с точностью до десятых долей.
Если данных недостаточно, вывести для этого дня “N/A” (not applicable, не применимо).

Дано:

```
// Weekly school marks
var marks = new[]
{
       new [] { 2, 3, 3, 2, 3}, // Monday (it was a good weekend :)
       new [] { 2, 4, 5, 3},    // Tuesday (anyway better than Monday)
       null,                    // Wednesday (felt sick, stayed at home :( )
       new [] { 5, 5, 5, 5},    // Thursday (God Mode :)
       new [] { 4 }             // Friday (Very short day)
};
```

Ожидаемый результат:

```
The average mark for day #0 is 2.6
The average mark for day #1 is 3.5
The average mark for day #2 is N/A
The average mark for day #3 is 5.0
The average mark for day #4 is 4.0
The average mark for all the week is 3.6
Press any key to exit...
```

Решение:

```
// Weekly school marks
var marks = new[]
{
	new [] { 2, 3, 3, 2, 3}, // Monday (it was a good weekend :)
	new [] { 2, 4, 5, 3},    // Tuesday (anyway better than Monday)
	null,                    // Wednesday (felt sick, stayed at home :( )
	new [] { 5, 5, 5, 5},    // Thursday (God Mode :)
	new [] { 4 }             // Friday (Very short day)
};

// Overall average mark by the week
int totalSum = 0;

// Overall number of marks by the week
int numberOfMarks = 0;

// Iterating through the days
for (int dayIndex = 0; dayIndex < marks.Length; dayIndex++)
{
	Console.Write($"The average mark for day #{dayIndex} is ");

	if (marks[dayIndex] == null)
	{
		Console.WriteLine("N/A");
		continue;
	}

	// Iterating through the marks during that day
	int dailySum = 0;
	for (int markIndex = 0; markIndex < marks[dayIndex].Length; markIndex++)
	{
		dailySum += marks[dayIndex][markIndex];
	}

	Console.WriteLine($"{(float)dailySum / marks[dayIndex].Length:0.0}");

	totalSum += dailySum;
	numberOfMarks += marks[dayIndex].Length;
}

Console.WriteLine(
	"The average mark for all the week is " +
	$"{(float)totalSum / numberOfMarks:0.0}");
```

Совместный цикл: foreach...in

- Оператор foreach выполняет определенный блок кода для каждого элемента в множестве, например, в массиве.
- Выполняется 0 или несколько раз.

Домашнее задание

1. Написать консольное приложение, которое запрашивает натуральное число и выводит количество четных цифр в нем.

```
Пример работы программы:
> Введите положительное натуральное число не более 2 миллиардов:
> -5 /это ввод пользователя/
> Введено неверное значение! Попробуйте ещё раз:
> 300000000000 /это ввод пользователя/
> Ошибка System.OverflowException! Попробуйте ещё раз:
> ABCD /это ввод пользователя/
> Ошибка System.FormatException! Попробуйте ещё раз:
> 1234567 /это ввод пользователя/
> В числе 1234567 содержится следующее количество четных цифр: 3.
> Нажмите любую клавишу для выхода...
```

2. Написать консольное приложение, которое запрашивает 1) сумму первоначального взноса, 2) ежедневный процент дохода и 3) желаемую сумму накопления.
   Программа должна вывести номер дня, когда накопление впервые превысит желаемое.

   Пример работы программы (при корректном вводе):

```
> Введите сумму первоначального взноса в рублях:
> 100 /это ввод пользователя/
> Введите ежедневный процент дохода в виде десятичной дроби (1% = 0.01):
> 0.0003 /это ввод пользователя/
> Введите желаемую сумму накопления в рублях:
> 200 /это ввод пользователя/
> Необходимое количество дней для накопления желаемой суммы: 2311.
> Нажмите любую клавишу для выхода...
```

- Не забывать обрабатывать все предсказуемые исключения.