Nested Loops

SoftUni Team

Technical Trainers

Software University

SoftUni

Software University

https://softuni.bg

Have a Question?

sli.do

#prgm-for-qa

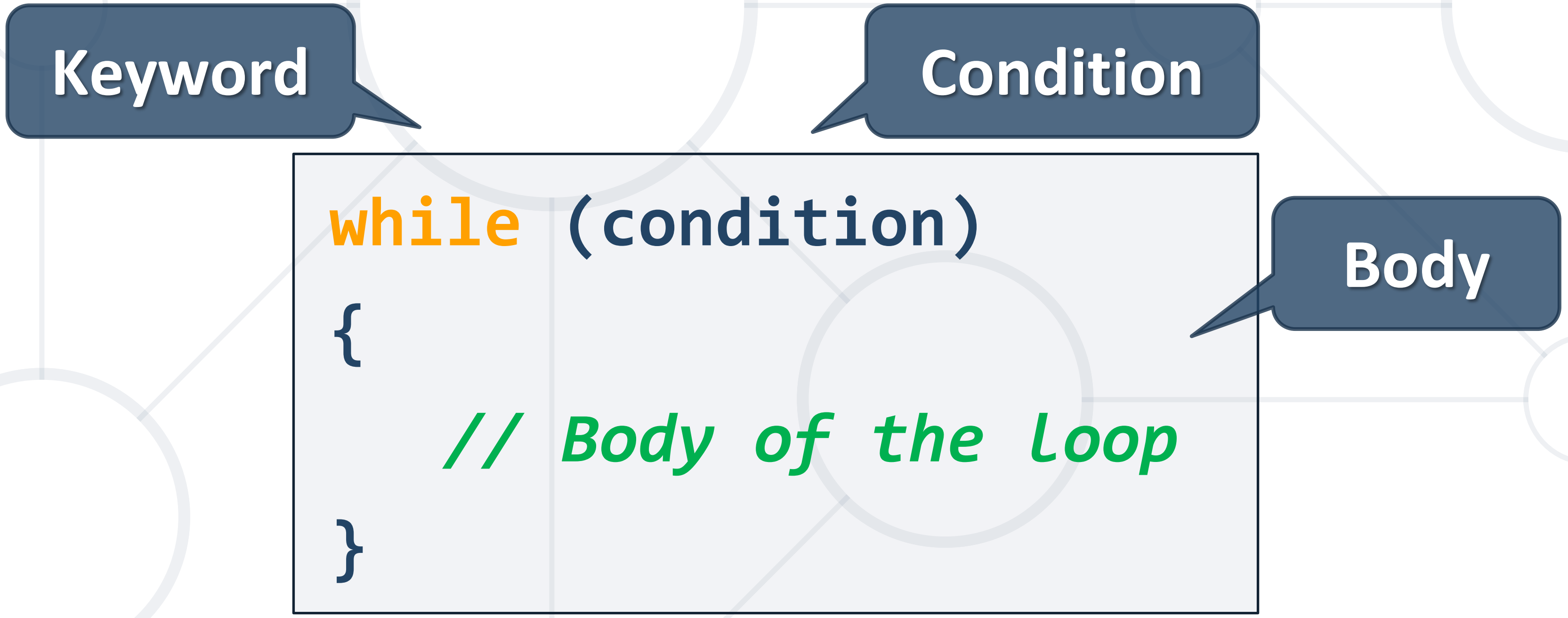
Table of Contents

Review

While Loops

While Loop

- Control flow **statement**
  - Executes code repeatedly while a condition is **true**



```
while (condition)
{
    // Body of the loop
}
```

Example: While Loop

- Print the numbers from **1 to 5**

```
int i = 1;
while (i <= 5)
{
    Console.WriteLine(i);
    i++;
}
```

1 … 5

While or For?

Software University

- `while` and `for` loops repeat blocks of code
- Use `for` when you know in advance the number of repetitions
  - For example, repeat exactly 10 times
- Use `while` when you don't know when the exit condition will be met
  - For example, repeat until `0` is reached

The "break" Operator

- Used for **prematurely exiting** the loop
- Can only be executed from the **body**, during **an iteration** of the loop
- **break** immediately exits from the loop
  - The rest of the loop body **is skipped**

```
int i = 1;
while (true)
{
  if (i > 10)
    break;
  i++;
}
```

Complex Loops

Loops with a Special Step

Complex Loops

- For-loops may have different **steps**

```
for (int i = n; i >= 1; i--) …
```

Step -1

```
for (int j = 1; j <= n; j += 2) …
```

Step +2

```
for (int k = 1; k <= n; k *= 2) …
```

Step *2

```
for (int d = n; d > 0; d /= 2) …
```

Step /2

Do...While Loops

- The **do … while (…)** loop repeats a block of code until an **exit condition** is met
  - The loop body is always executed at least **once**
- **while (…)** loop uses an exit condition at the **start**
- **do … while (…)** loop uses an exit condition at the **end**

```
int i = 1;
do {
  Console.WriteLine(i);
  i++;
} while (i <= 10);
```

Nested Loops

Introduction

Real Life Example: Clock

- Imagine **how the clock works**
  - A sequence of **iterations**
  - At each iteration, **the rightmost digit is increased**
  - When a digit **overflows** (reaches 10), it starts from **0** and the digit on its left is increased

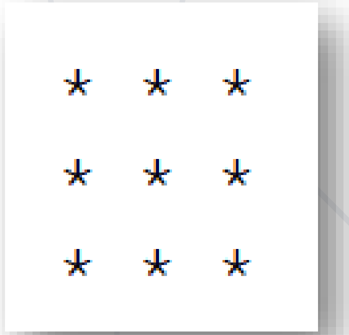
0 0 1 0

Loops Inside Other Loops

Nested Loops

- We can nest a **loop inside another loop**:

```
int n = 3;
for (int row = 1; row <= n; row++)
{
  for (int col = 1; col <= n; col++)
  {
    Console.Write(" *");
  }
  Console.WriteLine();
}
```



```
 * * *
 * * *
 * * *
```

Nested Loops

- Nested loops == several **loops** placed **inside each other**
- **Nested loops** are used:
  - To execute multiple times an **action**, which **executes** multiple **actions**
  - To implement more **complex** calculations and program logic

Multiple Levels of Nested Loops

```
for (int floor = 1; floor <= n; floor++)
{
  for (int row = 1; row <= n; row++)
   {
    for (int col = 1; col <= n; col++)
    {
      // …
    }
  }
}
```

The loop variable names must be different

Nested For-Loops

Nested For Loops

- The syntax for a **nested for loop** in C# is as follows:

```
// Outer Loop
for (init; condition; increment)
{
  // Inner Loop
  for (init; condition; increment)
  {
      // Commands
  }
}
```

Example: Nested For Loops

```
int rows = 3;
int columns = 2;
for (int r = 1; r <= rows; r++)
{
  Console.WriteLine("row = " + r);
  for (int c = 1; c <= columns; c++)
   {
     Console.WriteLine("  column = " + c);
   }
}
```

```
// Output
row = 1
   column = 1
   column = 2
row = 2
   column = 1
   column = 2
row = 3
   column = 1
   column = 2
```

Problem: Triangle of Stars

- Write a program to print a **triangle of stars** like shown below:
  - Read the **size** of a triangle from the console
  - Print a **triangle of stars**

```
5
```

```
*
**
***
****
*****
```

```
7
```

```
*
**
***
****
*****
******
*******
```

Solution: Triangle of Stars

```
int size = int.Parse(Console.ReadLine());
for (int row = 1; row <= size; row++)
{
  for (int col = 1; col <= row; col++)
  {
    Console.Write("*");
  }
 Console.WriteLine();
}
```

Problem: Building

- Write a program to **print a table**, representing a **building**:
  - **Odd** floors hold **apartments** (type **A**), e.g. **A10**, **A11**, **A12**, …
  - **Even** floors hold **offices** (type **O**), e.g. **O20**, **O21**, **O22**, …
  - The **last floor** holds large apartments (type **L**), e.g. **L60**, **L61**, **L62**
  - Identifiers consist of: **{type}{floor}{number}**, e.g. **L65**, **A12**, **O24**
  - Example:

```
L60 L61 L62 L63 L64 L65
A30 A31 A32 A33 A34 A35
O20 O21 O22 O23 O24 O25
A10 A11 A12 A13 A14 A15
```

Example: Building

- **Input**: the **count of floors** and the **count of estates per floor**
- **Output**: the building plan (rectangular table of estates)

```
6
4
```

```
L60 L61 L62 L63
A50 A51 A52 A53
O40 O41 O42 O43
A30 A31 A32 A33
O20 O21 O22 O23
A10 A11 A12 A13
```

Solution: Building

```
int floors = int.Parse(Console.ReadLine());
int rooms = int.Parse(Console.ReadLine());
for (int f = floors; f >= 1; f--)
{
  for (int r = 0; r < rooms; r++)
  {
    if (f == floors) // Print last floor: L{f}{r}
    else if (f % 2 == 0) // Print office: O{f}{r}
    else // Print apartment: A{f}{r}
  }
  Console.WriteLine();
}
```

The outer loop iterates through the floors

The inner loop iterates through the rooms

Nested While Loops

Nested While Loops

```
// Outer Loop
while (condition)
{
    // Inner Loop
    while (condition)
    {
        // Statements
    }
}
```

Example: Nested While Loops

```
int row = 1;
while (row <= 2)
{
  Console.WriteLine($"Row: {row}");
  int col = 1;
  while (col <= 3)
  {
    Console.WriteLine($"  Column: {col}");
    col++;
  }
  row++;
}
```

```
// Output
Row: 1
  Column: 1
  Column: 2
  Column: 3
Row: 2
  Column: 1
  Column: 2
  Column: 3
```

Problem: Travel Savings

- Calculate the **money collection** for multiple travel destinations:
  - Read **destination** and **needed budget** for destination
  - Read many times amounts of collected money, until they are **enough** for the destination (starting from 0)
    - Print "`Collected: {sum}`" where sum is formatted to 2nd digit or "`Going to {destination}`"
  - Read another destination and budget and collect money again
  - A destination "`End`" ends the program

Example: Travel Savings

```
Bali
3500
800
1800
1000
Brazil
4600
5000
End
```

```
Collected: 800.00
Collected: 2600.00
Collected: 3600.00
Going to Bali!
Collected: 5000.00
Going to Brazil!
```

Solution: Travel Savings

```
string destination = Console.ReadLine();
while (destination != "End")
{
  double neededSum = double.Parse(Console.ReadLine());
  double collectedSum = 0;
  while (collectedSum < neededSum)
  {
    collectedSum += double.Parse(Console.ReadLine());
    Console.WriteLine($"Collected: {collectedSum:F2}");
  }
 Console.WriteLine($"Going to {destination}!");

  destination = Console.ReadLine();
}
```

Summary

- For-loops can use different **steps**
- **Nested loops** are loops **within another loop**
  - Nested `for` loops, e.g. process data by rows and columns
  - Nested `while` loops, e.g. nested repeating logic with exit conditions

Software University

Questions?

SoftUni

Software University

SoftUni Creative

SoftUni Digital

SoftUni Foundation

SoftUni Kids

СофтУни БУДИТЕЛ

Finance Academy

SoftUni Diamond Partners

Software University

SmartIT

Coca-Cola HBC Bulgaria

SUPER HOSTING .BG

createX

Postbank

Решения за твоето утре

DRAFT KINGS

VIVACOM

AMBITIONED

INDEAVR

Serving the high achievers

SOFTWARE GROUP

PHAR VISION

Trainings @ Software University (SoftUni)

- Software University – High-Quality Education, Profession and Job for Software Developers
  - softuni.bg, about.softuni.bg
- Software University Foundation
  - softuni.foundation
- Software University @ Facebook
  - facebook.com/SoftwareUniversity

License