10а ТЕМА - 6

Цикъл с предусловие while в C#

цикълът while се използва за многократно изпълнение на блок от код, **докато** дадено логическо условие е истина (true). Нарича се цикъл с **предусловие**, защото проверката се прави *преди* всяко изпълнение на тялото на цикъла.

Синтаксис

```
while (условие)
{
    // Код, който се изпълнява
}
```

- Ако условието е false още в началото, кодът в кръдравите скоби **няма да се изпълни нито веднъж**.

Примерен код: Брояч от 1 до 5

Най-простата употреба – повтаряне на действие фиксиран брой пъти.

```
int counter = 1;
while (counter <= 5)
{
    Console.WriteLine("Число: " + counter);
    counter++; // Увеличаваме, за да не стане безкраен цикъл
}
```

Примерен код: Четене на данни до команда "Stop"

Това е класически пример, при който не знаем колко пъти ще се завърти цикълът (зависи от потребителя).

```
Console.WriteLine("Въвеждай думи (напиши 'Stop' за край):");
string input = Console.ReadLine();
while (input != "Stop")
{
    Console.WriteLine("Ти въведе: " + input);
    input = Console.ReadLine(); // Четем следващата стойност
}
Console.WriteLine("Програмата приключи.");
```

Пример: **Симулация на банкомат (ATM) с проверка на наличност и лимит на опитите.**
Кодът съчетава цикъла **while** с логически проверки (**if-else**), броячи и работа с числови данни.
Пример: Теглене на пари с валидация

```
double balance = 500.0; // Начална сума в картата
int attempts = 3;       // Лимит на опитите за въвеждане на сума
Console.WriteLine("Добре дошли! Вашата наличност е: " + balance + " лв.");
while (attempts > 0)
{
    Console.Write("Въведете сума за теглене (Оставащи опити: " + attempts + "): ");
    double amount = double.Parse(Console.ReadLine());
    if (amount <= 0)
    {
        Console.WriteLine("Грешка: Сумата трябва да е положително число.");
    }
    else if (amount <= balance)
    {
        balance -= amount; // Намаляваме баланса
        Console.WriteLine("Успешно изтеглихте " + amount + " лв. Остатък: " + balance + " лв.");
        break; // Прекъсваме цикъла, защото операцията е успешна
    }
    else
    {
        Console.WriteLine("Нямате достатъчно наличност!");
    }
    attempts--; // Намаляваме опитите при всяка неуспешна или невалидна стъпка
}
if (attempts == 0)
{
    Console.WriteLine("Превишихте лимита от опити. Картата е блокирана.");
}
Console.WriteLine("Благодарим, че използвахте нашите услуги!");
```

Разлика: **do-while винаги изпълнява кода поне веднъж**, дори условието в началото да е грешно. Проверката се случва накрая.
Практически пример, където **do-while** е по-добрият избор от обикновения **while**.
Пример: Валидация на вход (Положително число)
Потребителят трябва да въведе число и ако то е невалидно (например отрицателно), трябва да опитва отново и отново:_____

```
int number;
do
{
    Console.Write("Моля, въведете положително число (по-голямо от 0): ");
    number = int.Parse(Console.ReadLine());
    if (number <= 0)
    {
        Console.WriteLine("Грешка! Числото не е положително.");
    }
} while (number <= 0); // Цикълът се върти, докато числото е грешно
Console.WriteLine($"Браво! Избрахте числото: {number}");
```

Пример: Текстово меню с избор
В конзолните приложения менюто, което трябва да се появи поне веднъж, и да стои на екрана, докато потребителят не избере "Изход":

```
string choice;
do
{
    Console.WriteLine("\n--- ГЛАВНО МЕНЮ ---");
    Console.WriteLine("1. Нова игра");
    Console.WriteLine("2. Настройки");
    Console.WriteLine("3. Изход");
    Console.Write("Изберете опция (1-3): ");
    choice = Console.ReadLine();
    switch (choice)
    {
        case "1":
            Console.WriteLine("Стартиране на играта...");
            break;
```

```
        case "2":
            Console.WriteLine("Отваряне на настройки...");
            break;
        case "3":
            Console.WriteLine("Излизане от програмата. Довиждане!");
            break;
        default:
            Console.WriteLine("Невалиден избор, опитайте пак.");
            break;
    }
} while (choice != "3"); // Цикълът продължава, докато не изберем "3"
```

Упражнение с (while): Сумиране на цифрите на число
Използваме **while**, ако числото е 0, цикълът изобщо не започва.
Логика: Взимаме последната цифра с % 10 и я премахваме с /10 _____

```
Console.Write("Въведете цяло число: ");
int number = int.Abs(int.Parse(Console.ReadLine())); // Взимаме модул, за да работи и с
отрицателни
int sum = 0;
while (number > 0)
{
    sum += number % 10; // Добавяме последната цифра към сумата
    number /= 10;       // Премахваме последната цифра
}
Console.WriteLine("Сумата на цифрите е: " + sum);
```

Упражнение (do-while): Валидация на парола (ограничен достъп)
Използваме **do-while**, потребителят въвежда парола **поне веднъж**, преди да проверим дали е вярна._____

```
string password;
string correctPassword = "C#isCool123";
do
{
    Console.Write("Въведете парола за достъп: ");
```

```
    password = Console.ReadLine();

    if (password != correctPassword)
    {
        Console.WriteLine("Грешна парола! Опитайте отново.");
    }
} while (password != correctPassword);
Console.WriteLine("Достъпът е разрешен!");
```

Упражнение (Комбинирано): Математическа игра "Докато не кажеш край"
Програма генерира две случайни числа и иска отговор. Използва се **do-while** за цялата игра и **while** за проверка на конкретен отговор:

```
Random rnd = new Random();
string command;
do
{
    int a = rnd.Next(1, 11);
    int b = rnd.Next(1, 11);
    int result;
    Console.Write($"Колко е {a} + {b}? ");
    result = int.Parse(Console.ReadLine());
    // Докато отговорът е грешен, питай пак
    while (result != (a + b))
    {
        Console.Write("Грешно! Опитай пак: ");
        result = int.Parse(Console.ReadLine());
    }
    Console.WriteLine("Точно така!");
    Console.Write("Искаш ли друга задача? (да/край): ");
    command = Console.ReadLine();
} while (command != "край");
```

Един цикъл за да е безкраен, когато **условието му винаги е истина (true)** и вътре в тялото му липсва механизъм, който да го промени или да го прекъсне (чрез break). Пример: Логическа грешка (Забравен брояч)
Променливата се проверява, но стойността ѝ никога не се променя вътре в цикъла.

```
int i = 1;
while (i <= 5)
{
    Console.WriteLine("Това ще се печата вечно, защото 'i' винаги е 1...");
    // Тук липсва i++; или break
}
```

Нарочен безкраен цикъл (**Internal Exit**)
Изход от безкраен цикъл чрез ключова дума **break**. При липса на **break**, цикълът няма спре.

```
while (true)
{
    Console.Write("Въведи 'exit', за да спреш: ");
    string input = Console.ReadLine();
    if (input == "exit")
    {
        break; // Ако изтриете този ред, цикълът става безкраен
    }
    Console.WriteLine("Продължаваме...");
}
```

За да спрем безкраен цикъл в конзолата?
При стартиран код в конзолата започне да печата безкрайно, за стоп се натиска клавишната комбинация:
Ctrl + C
Това ще прекрати изпълнението на програмата принудително.