Code Academy

Dėstytojas

Edvinas Kesminas

Ciklai - While

Data

Šiandien išmoksite

01 While

02 While pavyzdžiai realybėje

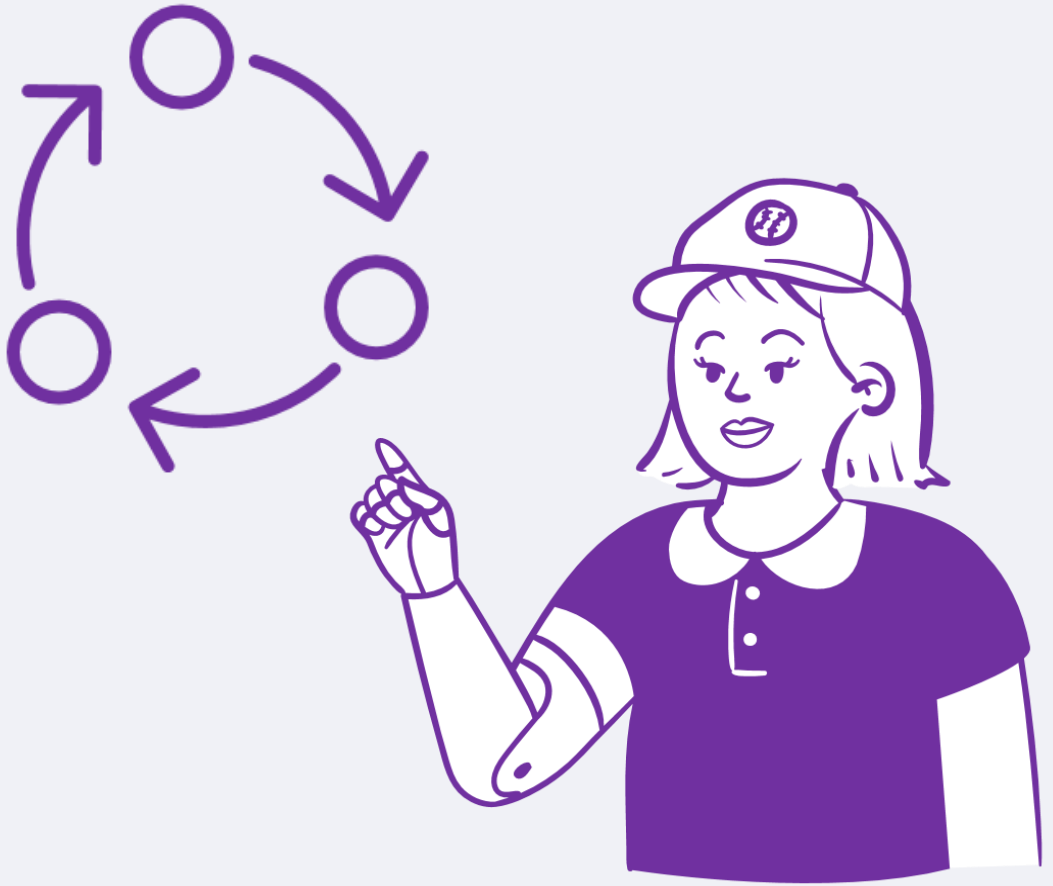
03 While in While

04 Do While

04 Do While pavyzdys realybėje

While

- "While" ciklas yra vienas iš pagrindinių ciklų, naudojamų "C#" programavimo kalboje.
- Šis ciklas leidžia vykdyti tam tikrą kodą tol, kol tam tikra sąlyga išlieka teisinga.
- Tai puiki priemonė kartoti tam tikrus veiksmus, kol įvyksta tam tikra sąlyga, arba tol, kol sąlyga tampa neteisinga.



While

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

- "While" ciklas turi paprastą struktūrą. Pirmiausia nurodome sąlygą, kurią norime patikrinti prieš kiekvieną ciklo iteraciją.
- Jei sąlyga yra teisinga, ciklo kūnas (kodas, kurį norime vykdyti) bus įvykdomas.
- Po to ciklas grįžta į sąlygos patikrinimą, ir šis procesas kartojamas, kol sąlyga tampa neteisinga.

While

- "While" ciklas turi didelį lankstumą, nes sąlygą galime patikrinti bet kuriuo metu ir atlikti įvairius veiksmus ciklo kūne.
- Tai leidžia mums pritaikyti ciklą daugybei skirtingų situacijų.
- Pavyzdžiui, galime naudoti "while" ciklą skaityti ir apdoroti įvestis iš vartotojo tol, kol gausime norimą rezultatą.

```
int i = 1;
string result = "";
while (i <= 5)
{
    Console.WriteLine("Iveskite teksta:");
    string input = Console.ReadLine();
    result += " " + input;
    i++;
}
Console.WriteLine(result);
```

While

- Svarbu užtikrinti, kad sąlyga cikle galiausiai taptų neteisinga, kad nebūtų užstrigta begalinė ciklo vykdymo kilpa.
- Norint tai užtikrinti, turime kiekvienoje ciklo iteracijoje atnaujinti sąlygą.
- Tai gali būti padaryta pakeičiant reikšmes, pridėdant arba atimant, arba kitais būdais modifikuojant sąlygą, kad ji galiausiai taptų neteisinga.

While

- "While" ciklas yra labai naudingas tais at
  kai turime nežinomą arba kintantį iteracij
  skaičių.
- Pavyzdžiui, galime naudoti "while" ciklą s
  duomenis iš failo tol, kol pasieksime failo
  pabaigą.
- Tai leidžia mums veiksmingai dirbti su
  duomenimis, kurių apimtis gali skirtis kie
  kartą.

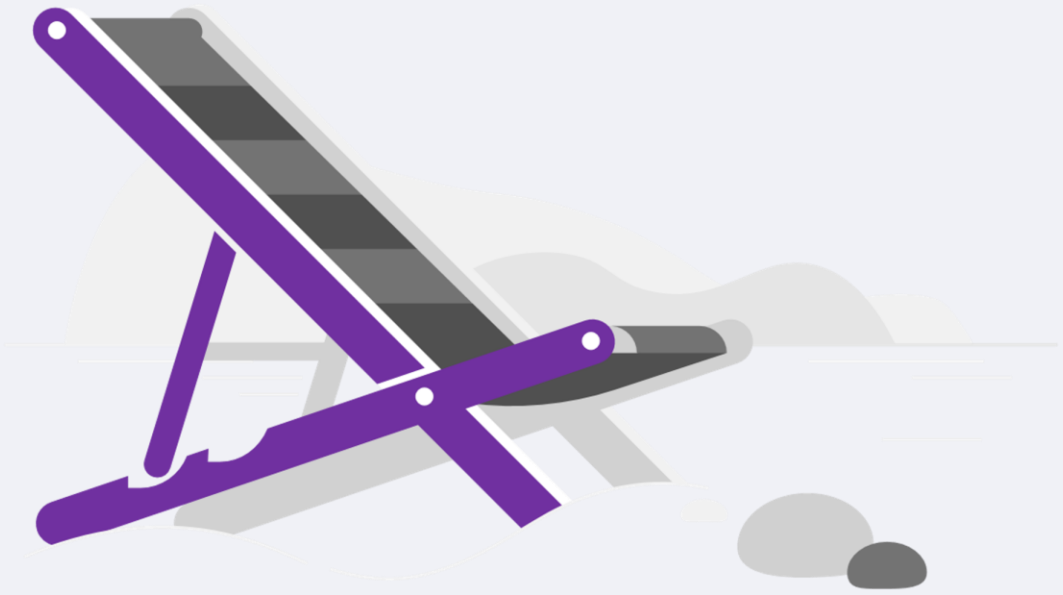
```
string filePath = "sample.txt";

try
{
    using (StreamReader sr = new StreamReader(filePath))
    {
        string line;
        while ((line = sr.ReadLine()) != null)
        {
            Console.WriteLine(line);
        }
    }
}
catch (Exception ex)
{
    Console.WriteLine("An error occurred: " + ex.Message);
}
```

While pavyzdžiai realybėje

- Galima palyginti "while" ciklą su pirkėju apsipirkimo centre. Pirkėjas gali vykdyti tam tikras prekių paieškos ir pirkimo operacijas tol, kol parduotuvėje yra prekių norimam pirkimo sąraše.

While pavyzdžiai realybėje



- Kelionė namo: Galima lyginti "while" ciklą su kelione namo. Pavyzdžiui, kai keliaujate atostogų metu, gali būti keletas dalykų, kuriuos turite nuveikti, kol grįšite namo.
- Galite turėti sąrašą vietų, kurias norite aplankyti, ir jas pereiti vieną po kito, kol pasieksite savo tikslą.
- Kiekvieną kartą, kai aplankote vietą, patikrinate, ar pasiekėte savo tikslą. Jei ne, tęskite kelionę tol, kol pasieksite norimą pabaigą.

While pavyzdžiai realybėje

- Galima lyginti "while" ciklą su mokytojo konsultacijomis.
- Pavyzdžiui, jei esate studentas ir turite klausimų dėl tam tikros temos, galite eiti pas savo mokytoją konsultacijai.
- Įeinate į kabinetą, pateikiate klausimus ir girdite atsakymą.
- Galbūt turite daugiau klausimų, todėl pakartojate konsultaciją, kol visi jūsų klausimai bus išsprendžiami ir jūsų supratimas bus užtikrintas.
- Šioje analogijoje konsultacija atlieka "while" ciklo funkciją, o jūsų klausimai ir atsakymai yra kiekviena iteracija, kol pasieksite supratimo tikslą.

While in While

```
int i = 1;
int j = 1;

while (i <= 3)
{
    Console.WriteLine("Outer loop: " + i);

    while (j <= 3)
    {
        Console.WriteLine("Inner loop: " + j);
        j++;
    }

    j = 1; // Nustatome vidinio ciklo skaitliuką į pradinę reikšmę
    i++;
}
```

- "While" ciklas gali būti naudingas ir galingas įrankis programavime, kuris leidžia mums kurti sudėtingesnes programos loginės struktūros.
- Be to, vienas iš šios struktūros variantų yra vadinamas "while" ciklas viduje kito "while" ciklo. Tai reiškia, kad vienas "while" ciklas gali būti įterptas į kito "while" ciklo kūną, leidžiant mums dar labiau išplėsti ir kontroluoti vykdymo eigą.
- Ši ciklų kaskadavimo technika gali būti pritaikoma sprendžiant įvairias užduotis ir problemas.
- Svarbu tikrinti sąlygas ir tinkamai valdyti kiekvieno ciklo vykdymą, kad būtų išvengta begalinio kilpavimo.
- Tinkamai naudojant "while" ciklą viduje kito "while" ciklo, galima sukurti efektyvias ir lanksčias programų struktūras, kurios atitinka konkrečius užduočių reikalavimus.

Užduotis nr. 1

- Parašykite "while" ciklą, kuris spausdina skaičius nuo 1 iki 5 į konsolę.
- Modifikuokite pirmą užduotį, kad po pirmojo ciklo eitų antrasis ciklas, kuris spausdintų skaičius nuo 5 iki 1.

- Parašykite "while" ciklą, kuris spausdina visus lyginius skaičius nuo 2 iki 10 į konsolę.
- Modifikuokite antrą užduotį, kad po pirmojo ciklo eitų antrasis ciklas, kuris spausdintų nelyginius skaičius nuo 1 iki 9.

- Parašykite "while" ciklą, kuris prašo įvesti skaičius iš vartotojo tol, kol įvestas skaičius yra didesnis nei 100.
- Modifikuokite trečią užduotį, kad po pirmojo ciklo eitų antrasis ciklas, kuris prašytų įvesti skaičius tol, kol įvestas skaičius yra teigiamas.

Užduotis nr. 2

- Parašykite "while" ciklą, kuris skaičiuoja ir spausdina faktorialą iš įvesto skaičiaus.
- Modifikuokite ketvirtą užduotį, kad po pirmojo ciklo eitų antrasis ciklas, kuris apskaičiuotų ir spausdintų visų įvestų skaičių faktorialus tol, kol nebūna įvestas neigiamas skaičius.

- Parašykite programą, kuri prašytų vartotojo įvesti teigiamą skaičių ir tuomet išvestų skaitmenis atskirai. Pavyzdžiui, įvedus skaičių 12345, programa turėtų išvesti šiuos skaitmenis vieną po kito: 1, 2, 3, 4, 5.

- Parašykite programą, kuri prašytų vartotojo įvesti skaičių n ir tuomet išvestų trikampį iš žvaigždučių naudojant "while" ciklus. Pavyzdžiui, įvedus skaičių 5, programa turėtų išvesti tokį trikampį:

Užduotis nr. 3

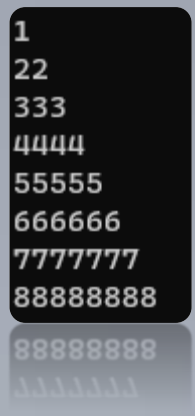
- Parašykite programą, kuri paprašo vartotojo įvesti sveikąjį skaičių ir tą skaičių gražina. Jeigu vartotojas įveda blogą skaičių, tai programa turi informuoti, kad įvestas blogas skaičius ir prašyti įvesti vėl. Kol vartotojas neįveda tinkamo skaičiaus metodas turi vis prašyti įvesti simbolius.

- Parašykite programą, kuri paprašytų vartotojo įvesti skaičių, kurį nori pakelti laipsniu ir tada paprašytų įvesti laipsnio skaičių, kuriuo nori pakelti. Skaičių pakeltą duotu laipsniu turėtų išvesti į ekraną.

- Parašykite programą, kuri išvestu vienoje eilutėje skaičių grupes tokiu principu: -> 1 -> 11 -> 111 -> 1111 -> 11111 -> ..... Programa turi paprašyti nurodyti skaičių ir grupių kiekį. Tikrinkite ar buvo įvestas skaitmuo, o ne tekstas.

Užduotis nr. 4

- Parašykite programą, kuri paprašo vartotojo įvesti sveikąjį skaičių ir pagal įvestą skaičių turėtų atitinkamai nupiešti trikampį kaip duota pavyzdyje:



- Parašykite programą, kuri prašytų vartotojo įvesti sumą pinigų, kurią jis nori išgryninti bankomate. Programa turi išvesti į ekraną naudodama While loop kada ir kokią kupiurą atiduoda vartotojui. Pvz: 50 eurų, 50 eurų, 20 eurų, 10 eurų (Vartotojas įvedė 130 eurų).

Do While

- Do While ciklas yra kita iteracijos kontrolės struktūra, kuri naudojama programavime.
- Šis ciklas labai panašus į "while" ciklą, tačiau turi svarbų skirtumą: sąlyga tikrinama po kiekvienos ciklo iteracijos.
- Tai reiškia, kad ciklas visada vykdomas bent vieną kartą, net jei sąlyga nuo pradžių yra neteisinga.
- Šis ciklas yra naudingas, kai norime užtikrinti, kad tam tikri veiksmai būtų įvykdyti bent vieną kartą, nepriklausomai nuo sąlygos teisingumo. Taip pat jis gali būti naudojamas, kai norime leisti vartotojui atlikti tam tikrus veiksmus ir tikrinti sąlygą tik po to, kai veiksmai yra vykdomi bent vieną kartą.

```
int number;
do
{
    Console.WriteLine("Įveskite skaičių: ");
} while (!int.TryParse(Console.ReadLine(), out number));
Console.WriteLine("Įvestas teisingas skaičius: " + number);
```

Do While

```
string choice;
do
{
    Console.WriteLine("Pasirinkite veiksmą (įveskite 'exit' norėdami išeiti): ");
    choice = Console.ReadLine();
    Console.WriteLine("Jūsų pasirinkimas: " + choice);
} while (choice != "exit");
```

- Galime turėti situaciją, kai norime parodyti pranešimą ir paklausti vartotojo, ar jis nori tęsti arba baigti programą.
- Nors vartotojo pirmasis atsakymas gali būti "ne", mes vis tiek norime parodyti pranešimą bent vieną kartą.
- Tuo atveju, "DoWhile" ciklas būtų tinkamas pasirinkimas, nes jis užtikrintų, kad pranešimas būtų parodytas ir vartotojui būtų suteikta galimybė atlikti veiksmą, net jei pirmasis atsakymas yra "ne".

Do While pavyzdys realybėje

- Galime įsivaizduoti, kad einame į šokio pamokas ir mokomės naujo šokio. Įeidami į pamokas, nežinome, ar mums gerai seksis. Tačiau mokytojas mus paskatina pradėti šokti ir nuolat mums sako, kad pratimai bus kartojami, kol pasieksime tam tikrą šokio lygį.
- Taigi, pradedame šokti ir mokytis šokio žingsnių. Nepriklausomai nuo to, ar mums pavyksta šokį atlikti teisingai, ar ne, mes tęsiame mokymąsi ir kartojame šokio žingsnius. Tik po kiekvienos šokio iteracijos mokytojas tikrina mūsų pažangą ir sako, ką reikėtų tobulinti ar koreguoti.

Užduotis nr. 5

- Parašykite programą, kuri leistų vartotojui įvesti skaičius, kol jis pasirenka baigti. Programa turi skaičiuoti visų įvestų skaičių sumą ir ją atvaizduoti. Vartotojui turi būti leidžiama įvesti bet kokį skaičių, ir jei jis nori tęsti įvedimą, programa turi leisti tai padaryti. Jei vartotojas įveda „Baigti“ (Nesvarbu ar upper ar lower case) į ekraną turėtų atspausdinti galutinį rezultatą.

- Slaptažodžio patikrinimas. Parašykite programą, kuri leidžia vartotojui įvesti slaptažodį, kol jis įveda teisingą slaptažodį. Pradinis teisingas slaptažodis gali būti nustatytas programos koduose. Vartotojui turi būti leidžiama įvesti slaptažodį daug kartų, kol jis įveda teisingą slaptažodį

- Parašykite programą, kuri sugeneruoja atsitiktinį skaičių nuo 1 iki 100 (Math.Random). Vartotojui turi būti leidžiama spėti šį skaičių. Programa turi parodyti, ar vartotojo spėjimas yra didesnis arba mažesnis už sugeneruotą skaičių. Vartotojui bus leidžiama spėti tol, kol jis atspės teisingą skaičių

Projektas 1

- Pratęskite savo 3 pamokos projektą parašydami žaidimui meniu ir scenarijus, kuriuose naudotų while arba doWhile ciklus ir Math.Random(). Žaidimo pradžioje turėtumėte paprašyti vartotojo vardo, taip pat žaidimas turėtų kaupti virtualius taškus už gerus sprendimus. Pvz jei žaidėjas ateina į pelkę ir ten sutinka žmogėdrą tai žaidėjas turėtų turėti galimybę kovoti su žmogėdra kol žaidėjas turi daugiau nei 100 gyvybės taškų. Žaidėjo kovos mechanizmas galėtų remtis į „popierius, žirklės, akmuo“ principą. Žaidėjas turi rinktis vieną iš 3 atakų, o žmogėdrai atsitiktiniu būdu būna generuojama ataka, ciklo metu patikrinama ar žaidėjas prarado gyvybės taškų ar atėmė žmogėdrai. Pvz aukšta ataka įveikia žemą ataką, žema ataka įveikia vidutinę ataką, vidutinė ataka įveikia aukštą ir pnš.

Nuorodos

https://learn.microsoft.com/en-us/dotnet/csharp/language-reference/statements/iteration-statements

https://learn.microsoft.com/en-us/dotnet/csharp/language-reference/operators/null-coalescing-operator

https://learn.microsoft.com/en-us/dotnet/csharp/language-reference/builtin-types/nullable-value-types

https://learn.microsoft.com/en-us/dotnet/csharp/language-reference/operators/null-forgiving