Liepājas valsts tehnikums

**Cikls ar priekšnosacījumu (while) programmēšanas valodā Python**

Mācību materiāls

Darba autors: Maksims Telepnevs

Liepāja 2025

**Ievads mācību materiālam: "Cikls ar priekšnosacījumu programmēšanas valodā Python"**

Ciklā ar nosacījumu **while**, cikls tiek izpildīts, kamēr ir patiess tā noteicošais nosacījums. Tāpēc šo ciklu dažreiz sauc arī par **"kamēr" ciklu**. Bieži vien **while** cikls tiek izmantots situācijās, kad nav iespējams iepriekš paredzēt, cik reizes būs jāizpilda cikla pamatteksts. Ikdienas dzīvē **while** ciklu var sastapt algoritmos, piemēram: "Kamēr redzes laukā ir mašīnas, palikt uz vietas" vai "Kamēr kastē ir detaļas, izņemt detaļu no kastes".

**Kāpēc tas ir svarīgi?**

* **Automātizācija** – atkārto darbības bez manuālas ievades.
* **Dinamiska izpilde** – darbojas, kamēr nosacījums ir izpildīts.
* **Efektivitāte** – samazina koda dublēšanu.

***Cikls while piemers:***Programma, kas izvada skaitļus no 1 līdz 10, izmantojot while ciklu:

# Inicializē mainīgo i ar sākuma vērtību 1

i = 1

# Kamēr i ir mazāks vai vienāds ar 10, izpildīt ciklu

while i <= 10:

# Izvada pašreizējo i vērtību

print(i)

# Palielina i par 1 (sagatavo nākamo vērtību)

i = i + 1

**Detalizēts skaidrojums**

1. Inicializācija

i = 1 – Mainīgais i tiek uzstādīts uz sākuma vērtību 1.

1. Cikla nosacījuma pārbaude

while i <= 10: – Pārbauda, vai i vērtība nepārsniedz 10.

Ja nosacījums patiess (≤10), izpilda cikla darbības.

Ja nepatiess (>10), cikls pārtrauc darbību.

1. Cikla darbības

print(i) – Izvada i pašreizējo vērtību.

i = i + 1 – Palielina i par 1 (piemēram, no 1 uz 2).

1. Cikla atkārtošanās

Pēc katras iterācijas programma atgriežas pie nosacījuma pārbaudes:

i = 1 → izpilda (izvada 1, i kļūst 2)

i = 2 → izpilda (izvada 2, i kļūst 3)

...

i = 10 → izpilda (izvada 10, i kļūst 11).

1. Cikla pārtraukšana

Kad i = 11:

Nosacījums 11 <= 10 kļūst nepatiess.

Cikls beidz darboties.

Programma turpina ar nākamajām instrukcijām (ja tādas ir).

1. Rezultāts pēc cikla

Beigās mainīgais i saglabā vērtību 11, jo tā bija pēdējā piešķirtā vērtība pirms nosacījuma pārbaudes.

**Izvads konsole:**

1

2

3

4

5

6

7

8

9

10

**Vispārīga sintakse:**

while **nosacījums**:

darbiba\_1

darbiba\_2

...

darbiba\_n

**Obligātās cikla sastāvdaļas:**

1. **Nosacījums**

* Rakstās aiz atslēgvārda **while**
* Jābūt loģiskai izteiksmei (izteiksme, kas atgriež **True/False**)
* Obligāti jāietver kaut viens no:
  + **Salīdzināšanas operatori**: ==, !=, <, >, <=, >=
  + **Loģiskie operatori**: and, or, not kombinācijās
* Pēc nosacījuma vienmēr liek **kolu :**

***Piemērs:***

**while** (x < 10) **and** (y != 0): # Kombinēts nosacījums

1. **Cikla pamatteksts**

* Viena vai vairākas darbības
* Obligāta atkāpe (tradicionāli 4 atstarpes)
* Darbības izpildās secīgi

***Piemērs:***

**while** x **>** 0:

print("Vērtība:", x)

x **-=** 1 # Atkāpe nozīmē, ka šī ir cikla daļa

1. **Mainīgo atjaunināšana**
   * Obligāti jāietver darbības, kas **maina nosacījumā lietotos mainīgos**
   * Bez šīm darbībām risks: **bezgalīgs cikls**
   * Tipiskas formas:
     + Skaitītāja palielināšana: skaitītājs = skaitītājs + 1
     + Vērtības samazināšana: x -= 0.5
     + Statuss maiņa: aktivs = False

**Praktiskie uzdevumi: Cikls ar priekšnosacījumu programmēšanas valodā Python**

**1. Uzdevums**

**2. Uzdevums**

**3. Uzdevums**

**Secinājumi**

**Izmantotajie informācijas avoti:**

1. <https://informatics.msk.ru/mod/book/view.php?id=13732>
2. <https://skillbox.ru/media/code/tsikly-v-python-kak-rabotayut-i-kakie-byvayut/#stk-2>