



LATVIJAS  
UNIVERSITĀTE  
ANNO 1919

NACIONĀLAIS  
ATTĪSTĪBAS  
PLĀNS 2020



EIROPAS SAVIENĪBA

Eiropas Sociālais  
fonds

IEGULDĪJUMS TAVĀ NĀKOTNĒ

# Java programmēšanas pamati



VADĪBAS UN  
UZŅĒMĒJDARBĪBAS  
MĀCĪBU CENTRS

6. nodarbība

# CIKLA OPERATORI



# CIKLA OPERATORI

# PĀRSKATS

- ▶ Bieži ir sastopamas situācijas, kad ir nepieciešams izpildīt koda bloku **vairākas reizes** pēc kārtas
- ▶ Iterāciju operators **ļauj** izpildīt izteiksmi vai izteiksmju grupu **vairākas reizes** pēc kārtas
- ▶ Pieejamie iterāciju operatori:
  1. while
  2. for
  3. do...while

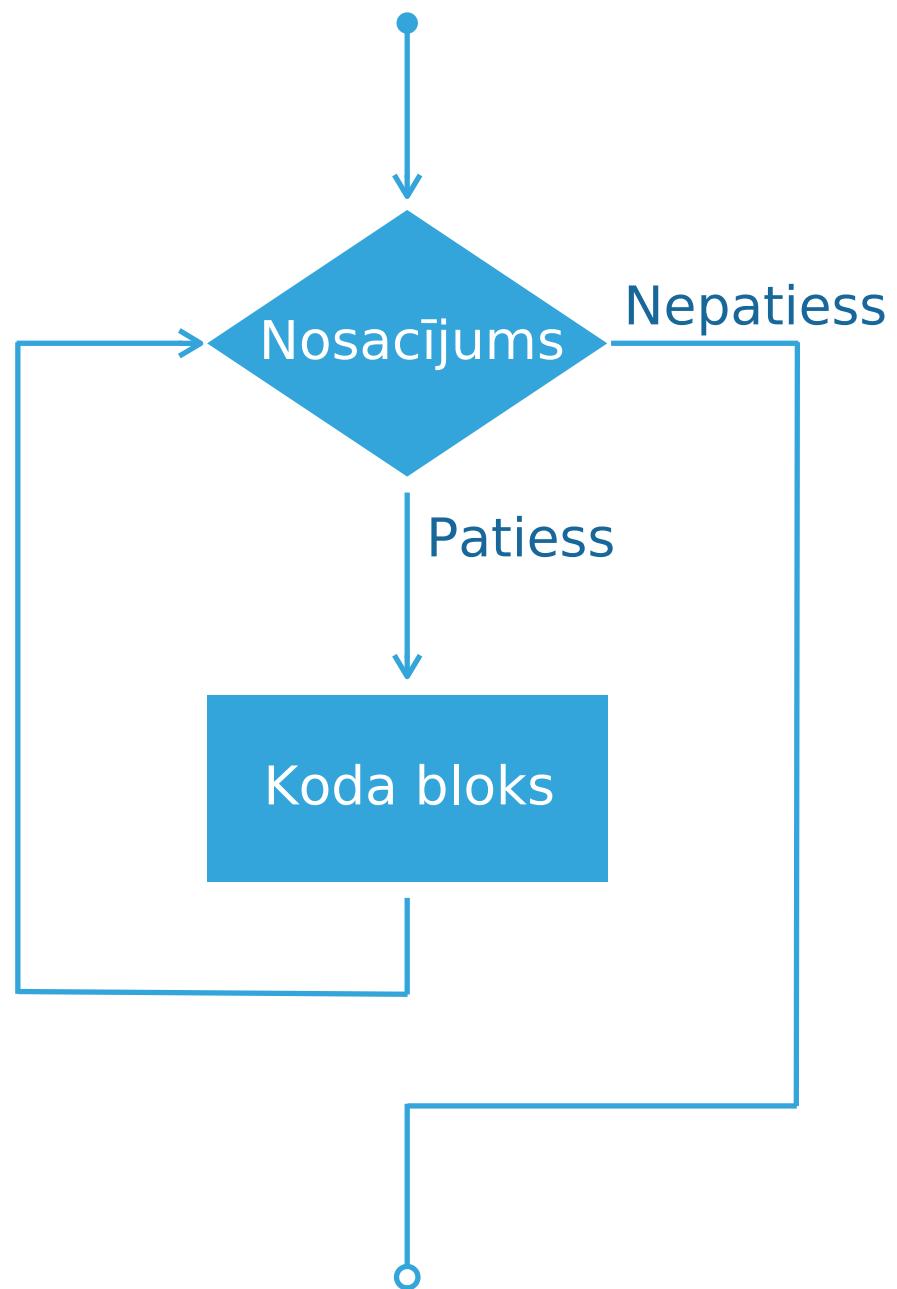
# CIKLA STRUKTŪRA

- ▶ Ar **cikla skaitītāju** var **kontrolēt** izpildes ciklu skaitu
- ▶ Cikla mainīgajam ir jābūt **inicializētam**
- ▶ Katru reizi, kad cikls ir izpildīts, tiek **palielināts vai samazināts** cikla skaitītājs
- ▶ **Cikla nosacījums**, kas nosaka vai ir jāturpina izpildīt cikls vai tā darbība ir jāpārtrauc

## CIKLS “WHILE”

- ▶ Izpilda izteiksmi vai izteiksmju bloku **kamēr (while)** tā kontrolējošā būla izteiksme ir **patiesa**
- ▶ Būla izteiksme tiek izvērtēta pirms pirmās cikla iterācijas, tātad **izpildās nevienu vai vairākas reizes**
- ▶ Parasti izmanto, kad ciklu izpildes skaits **nav precīzi zināms**

# CIKLS “WHILE”: BLOKSHĒMA



# CIKLS “WHILE”: SINTAKSE

Cikla “while”  
deklarēšanas  
atslēgvārds

Cikla nosacījums

```
while (nosacījums) {  
    izteiksmes...;  
}
```

Izteiksmes, kas tiks izpildītas  
cikla ķermenī (body)



# CIKLS “WHILE”: PIEMĒRS

## Izejas kods

```
int i = 0;
while (i < 5) {
    System.out.print("i = " + i + "; ");
    i++;
}
```

## Konsoles izvade

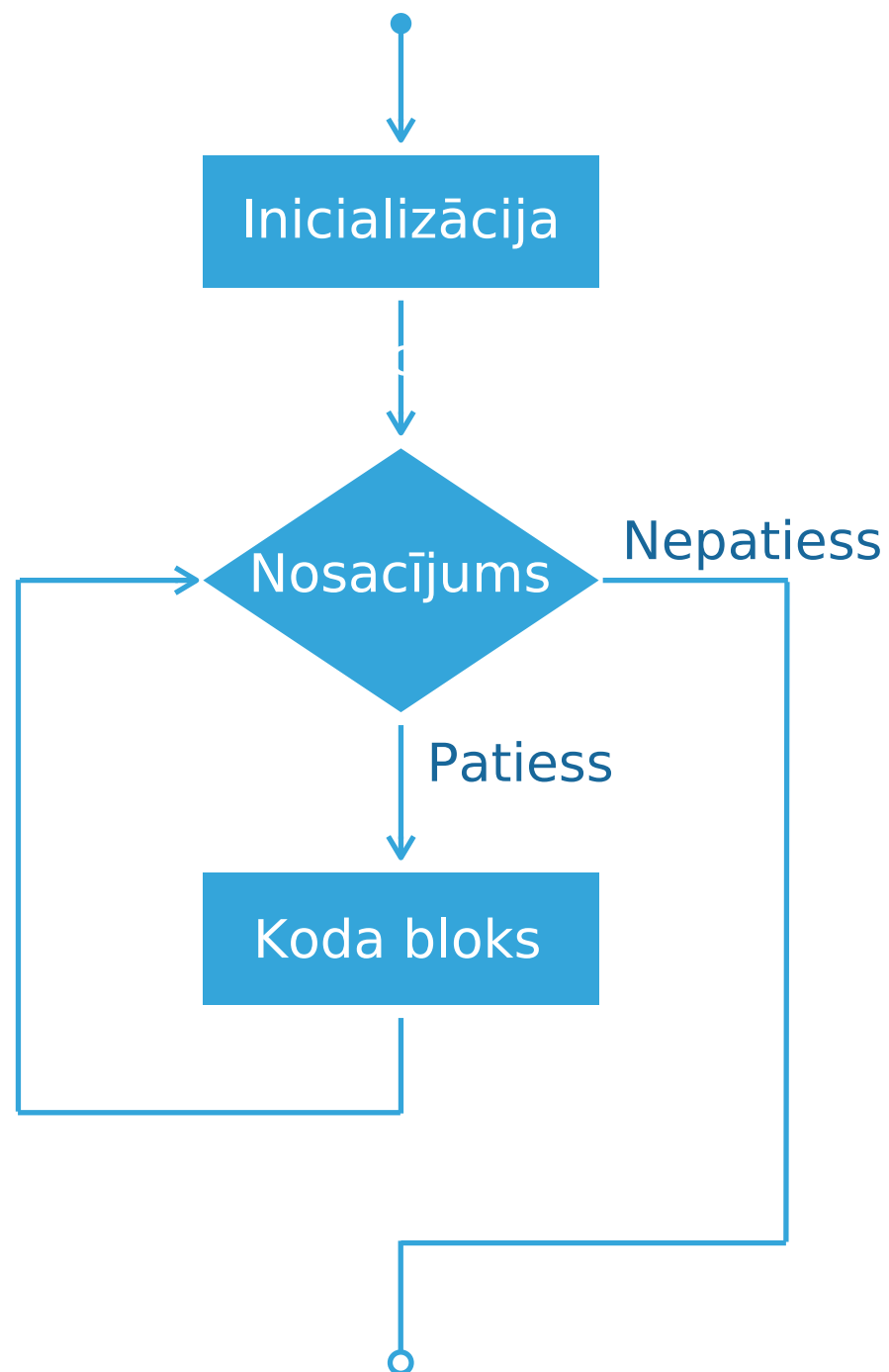
i = 0; i = 1; i = 2; i = 3; i = 4;

Process finished with exit code 0

## CIKLS “FOR”

- ▶ Kontroles mehānisms, kas ļauj atkārtot izteiksmi vai izteiksmju bloku, **palielinot** vai **samazinot** un **izvērtējot** cikla **skaitītāja vērtību**
- ▶ Būla izteiksme tiek izvērtēta pirms pirmās cikla iterācijas, tātad **izpildās nevienu vai vairākas reizes**
- ▶ Tiek izmantots, kad **iterāciju** skaits ir iepriekš zināms

# CIKLS “FOR”: BLOKSHĒMA



# CIKLS “FOR”: SINTAKSE

Cikla “for”  
Deklarešanas  
atslēgvārds

Inicializē skaitītāja(u)  
vērtību(as)

Palielina/Samazina  
skaitītāja vērtību

Cikla nosacījums

```
for (inicializācija; nosacījums; modifikācija) {  
    izteiksmes;  
}
```

Izteiksmes, kas tiek izpildītas  
cikla ķermenī (body)

# CIKLS “FOR”: PIEMĒRS

## Izejas kods

```
for (int i = 0; i < 5; i++) {  
    System.out.print("i = " + i + "; ");  
}
```

## Konsoles izvade

i = 0; i = 1; i = 2; i = 3; i = 4;

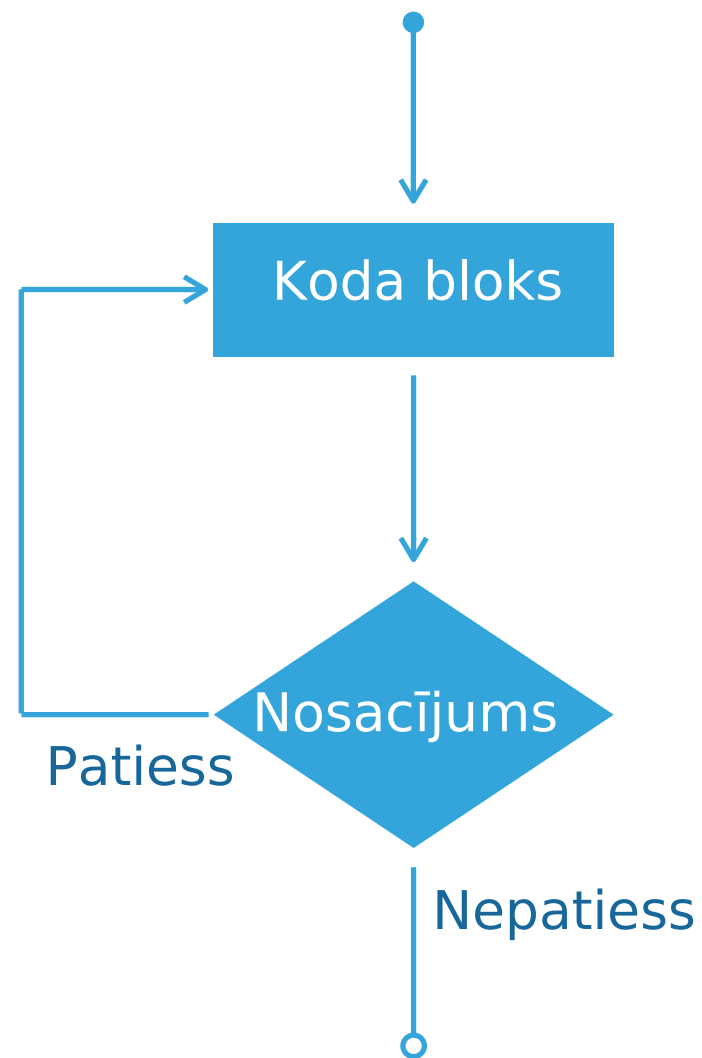
Process finished with exit code 0



## CIKLS “DO WHILE”

- ▶ Izpilda izteiksmi vai izteiksmju bloku **kamēr (while)** tā kontrolējošā būla izteiksme ir **patiesa**
- ▶ Būla izteiksme tiek izvērtēta pēc pirmās cikla iterācijas, tātad **izpildīta vienu vai vairakas reizes**
- ▶ Parasti lieto, kad ir nepieciešams izpildīt ciklu **vismaz vienu reizi** un iterāciju **skaits nav precīzi zināms**

# CIKLS “DO WHILE”: BLOKSHĒMA



# CIKLS “DO WHILE”: SINTAKSE

Cikla “do...while”  
deklarēšanas  
atslēgvārds

Izteiksmes, kas tiek izpildītas  
cikla ķermenī (body)

```
do {  
    izteiksmes;  
} while (nosacījums);
```

Cikla nosacījums

# CIKLS “DO WHILE”: PIEMĒRS

## Izejas kods

```
int i = 0;  
do {  
    System.out.print("i = " + i + "; ");  
    i++;  
} while (i < 5);
```

## Konsoles izvade

i = 0; i = 1; i = 2; i = 3; i = 4;

Process finished with exit code 0

# IZTEIKSMJU ZAROŠĀNĀS CIKLOŠ



# IZTEIKSMJU ZAROŠANĀS CIKLOS

- ▶ Zarošanās izteiksmes tiek lietotas, lai mainītu cikla izpildes secību atkarībā no kāda nosacījuma
- ▶ Pieejamās zarošanās izteiksmes:
  1. break
  2. continue

## IZTEIKSME “BREAK”

- ▶ Pārtrauc **iekšējo** for, while, do...while ciklu
- ▶ Kad tiek izsaukta break izteiksme, tad cikls tiek **pārtraukts** un programmas darbība turpinās ar nākošo izteiksmi pēc cikla

# IZTEIKSME “BREAK”: PIEMĒRS

## Izejas kods

```
for (int i = 0; i < 10; i++) {  
    if (i == 3) {  
        break;  
    }  
    System.out.print("i = " + i + "; ");  
}
```

## Konsoles izvade

i = 0; i = 1; i = 2;

Process finished with exit code 0

## IZTEIKSME “CONTINUE”

- ▶ Ciklam **for**, atslēgvārds **continue**, **pārceļ** tā izpildi pie cikla **modificēšanas izteiksmes**
- ▶ Ciklam **while** vai **do...while** atslēgvārds **continue**, **pārceļ** tā izpildi pie cikla **būla izteiksmes**

# IZTEIKSME “CONTINUE”: PIEMĒRS

## Izejas kods

```
for (int i = 0; i < 10; i++) {  
    if (i % 2 == 0) {  
        continue;  
    }  
    System.out.print("i = " + i + "; ");  
}
```

## Konsoles izvade

i = 1; i = 3; i = 5; i = 7; i = 9;

Process finished with exit code 0



# ATSAUCES

- ▶ [https://www.tutorialspoint.com/java/java\\_loop\\_control.htm](https://www.tutorialspoint.com/java/java_loop_control.htm)
- ▶ <https://www.baeldung.com/java-loops>
- ▶ <https://www.developer.com/java/data/using-different-types-of-java-loops-looping-in-java.html>
- ▶ <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/branch.html>



LATVIJAS  
UNIVERSITĀTE  
ANNO 1919