

PROGRAMMĒŠANAS PAMATI

MAINĪGIE

ARTIMĒTISKIE APRĒĶINI

OPERATORI



Prezentācija veidota, lai atkārtotu tēmu par mainīgajiem un to tipiem, deklarēšanu (norādītā saite aizved uz Learning Apps interaktīvu uzdevumu). Pēc atkārtojuma sākas tēma par datu ievades un izvades veidiem, kurā sīkāk apskatīta ar Scanner klase (ļauj ievadīt datus no tastatūras. Lai veiksmīgāk apgūtu lietotāja datu ievadi programmā, prezentācijā iekļauta informācija par vienkāršu izteiksmju vērtību aprēķinu (tajā skaitā par viena mainīgā, saliktajiem, piešķiršanas un loģiskajiem operatoriem) un operāciju veikšanu ar veseliem skaitļiem.

Šodien:

- Atkārtojums par mainīgajiem - <https://learningapps.org/display?v=phguadvej18>
- Ievads par operatoriem
- Klase *Scanner*
- Vienkāršu izteiksmju vērtību aprēķins. Veselo skaitļu operācijas

*Rezultāts: Prot veidot izteiksmes, izmantojot aritmētiskos, salīdzināšanas un mainīgo operatorus. Prot uzrakstīt vienkāršu programmu, izmantojot klasi *Scanner*. Izprot atšķirības starp operatoriem.*

Atkārtojums par mainīgajiem

Mainīgo deklarējot norāda:

- mainīgā datu tipu,
- mainīgā identifikatoru;
- var norādīt mainīgā vērtību (inicializēt mainīgo).

Mainīgā deklarācijas vispārēja forma ir: Datu tips identifikators = vērtība;

`double pi = 3.14159;`

Uzdevums: <https://learningapps.org/display?v=phguadvej18>

Datu ievade un izvade

Datu izvadīšanai uz ekrāna lieto metodes:

`System.out.println()`, kas izvada uz ekrāna iekavās norādītos lielumus un pārvieto kursoru jaunā rindā;

`System.out.print()`, kas izvada uz ekrāna iekavās norādītos lielumus, bet kursors paliek tajā pašā rindā;

Lietojot metodes, jāievēro sekojošais:

- ✓ kā metodes argumentus iekavās var norādīt tekstu, kas ietverts pēdiņās, izteiksmes, mainīgo identifikatorus, literāļus;
- ✓ ja iekavās norāda vairākus argumentus, tad tos savā starpā saista ar operatoru +.
- ❖ **Literāļi (literals) ir dažāda tipa konstantās vērtības, kas tiešā veidā parādās programmas tekstā (vesela skaitļa literāļi, decimāldaļskaitļu literāļi, simbolu literāļi, virkņu literāļi).**

Datu izvade

Papildus teksta izvades iespējas nodrošina **speciālu simbolu secību (escape sequences)** lietošana. Speciālās simbolu secības veido 2 rakstzīmes, no kurām pirmā ir atpakaļsvītra (\). Dažu speciālo simbolu secību nozīme dota tabulā.

Simbolu secība	Raksturojums
\"	Izvada uz ekrāna pēdiņas
\t	Izveido tabulācijas atstarpi
\n	Pārvieto kursoru uz jaunu rindu
\\	Izvada uz ekrāna reverso slīpsvītru (\)

Datu ievade, lietojot klasi *Scanner*

Klase Scanner ļauj izveidot jaunu objektu, kas nodrošina dažāda tipa datu ievadi no tastatūras, lietojot šādas metodes:

nextInt() – integer tipa datu ievade;

nextFloat() – float tipa datu ievade;

nextDouble() – double tipa datu ievade;

nextLong() – long tipa datu ievade;

nextShort() – short tipa datu ievade;

next() – vārda ievade;

nextLine() – rakstzīmju virknes ievade;

nextBoolean() – Boolean tipa datu ievade.

Datu ievade, lietojot klasi *Scanner*(2)

```
import java.util.Scanner; //Scanner klases piesaistīšana objektam

public class Matemātika {
    public static void main(String[] args) {

Scanner ievade = new Scanner(System.in);/*Deklarē Scanner klases objektu ar nosaukumu ievade, norādot, ka tas būs jauns objekts, kas nodrošina datu ievadi no tastatūras. */

System.out.println("Ievadiet veselu skaitli: ");

int skaitlis = ievade.nextInt(); //Deklarē int tipa mainīgo skaitlis un inicializē to, skaitļa vērtību ievadot no tastatūras.

System.out.println("Ievadīts skaitlis: " + skaitlis);
    }
}
```


Aritmētiskie aprēķini

Lai programmā aprakstītu nepieciešamos aritmētiskos aprēķinus, parasti tiek lietoti šādi elementi:

- izteiksmes, kas apraksta veicamo darbību;
- piešķiršanas operators, ar kuru mainīgajam tiek piešķirta vērtība vai izteiksmes rezultāts.

1.1. Izteiksmes

Lai aprakstītu veicamās darbības, tiek veidotas izteiksmes, kas vienmēr satur divus elementus:

- lielumus, ar kuriem tiek veikta darbība;
- operāciju operatorus, kas norāda izpildāmo darbību.

Operators ir viena rakstzīme vai rakstzīmju pāris, kas norāda veicamo darbību ar vienu vai vairākiem mainīgajiem izteiksmē.

Aritmētiskie operatori

Operators	Darbība	Izteiksmes piemērs
+	Summēšana	$a+b$
-	Atņemšana(starpība)	$a-b$
*	Reizināšana	$a*b$
/	Dalīšana(ja viens no skaitļiem nav vesela skaitļa tipa)	a/b , ja $a=11$ un $b=4$, tad rezultāts ir 2.75
/	Veselu skaitļu dalīšana bez atlikuma	a/b , ja $a = 11$ un $b = 4$, tad rezultāts ir 2
%	Veselā atlikuma iegūšana	$a\%b$

Piešķiršanas operators

Piešķiršanas operators – vienādības zīme (=) ir viens no jebkuras programmēšanas valodas pamatoperatoriem.

Piešķiršanas operatora kreisajā pusē norādītajam mainīgajam tiek piešķirta tā labajā pusē esošā vērtība (skaitliska vai izteiksmes rezultāts), piemēram:

- vērtības piešķiršana mainīgajam: $x=5$;
- izteiksmes rezultāta piešķiršana mainīgajam: $x=a*b$;

Saliktie piešķiršanas operatori

(Assignment Operators)

Tiek ietoti, lai saīsinātu programmas tekstu. Piemēri:

Operators	Izteiksme	Ekvivalenta izteiksme
<code>+=</code>	<code>a +=b</code>	<code>a = a + b</code>
<code>-=</code>	<code>a -=b</code>	<code>a = a - b</code>
<code>*=</code>	<code>a *=b</code>	<code>a = a * b</code>
<code>/=</code>	<code>a /=b</code>	<code>a = a / b</code>
<code>%=</code>	<code>a%= b</code>	<code>a = a % b</code>

Viena mainīgā operatori

(Unary Operators)

Ietekmē mainīgā vērtību. Piemēri:

Operators	Izteiksme	Ekvivalenta izteiksme	Skaidrojums
++	a ++	$a = a+1$ vai $a+=1$	Vērtības palielināšana par 1
--	a--	$a = a-1$ vai $a-=1$	Vērtības samazināšana par 1
-	-a	$a=-a$	Zīmes maiņa uz pretējo
+	+a	$a=+a$	Zīmes saglabāšana

Salīdzināšanas operatori

(Relational Operators)

Lieto, lai salīdzinātu divu mainīgo vērtības. Izteiksmes rezultāts ir loģiskā tipa(boolean) vērtība: true vai false. Piemēri, ja $a=1$ un $b=2$, un rezultāts ir Rez.

Operators	Nozīme	Izteiksme	Skaidrojums
$=$	Vienāds ar	Rez = $a==b$	Rez = false, jo 1 nav vienāds ar 2
$!$	Nav vienāds ar	Rez = $a!=b$	Rez = true, jo 1 nav vienāds ar 2
$<$	Mazāks par	Rez = $a<b$	Rez = true, jo 1 ir mazāks par 2
$>$	Lielāks par	Rez = $a>b$	Rez = false, jo 1 nav lielāks par 2
$<=$	Mazāks vai vienāds ar	Rez = $a<=b$	Rez = true, jo 1 ir mazāks par 2
$>=$	Lielāks vai vienāds ar	Rez = $a>=b$	Rez = false, jo 1 nav vienāds vai lielāks par 2

Loģiskie operatori - ievads

(Logical operators)

Lieto, lai veidotu izteiksmes, kas satur vairākas salīdzināšanas izteiksmes. Loģiskās izteiksmes rezultāts ir loģiskā tipa vērtība: true(patiesa) vai false(nepatiesa).

Loģiskais operators	Nosaukums
&&	Loģiskais UN
	Loģiskais VAI
!	Loģiskais NĒ

Loģiskais UN

Tabulā dots izteiksmes ar loģisko UN rezultāts atkarībā no mainīgo a un b vērtības. Rezultāts ir true tikai tad, ja abu mainīgo vērtība ir true.

a = 5;

rezUn = (a >= 1) && (a <= 10);

Mainīgā rezUn vērtība ir true, jo:

- izteiksmes (a >= 1) vērtība ir true (5 ir lielāks par 1);
- izteiksmes (a <= 10) vērtība ir true (5 ir mazāks par 10);
- abu izteiksmju (a >= 1) un (a <= 10) vērtības ir true.

a	b	a && b
true	true	true
true	false	false
false	true	false
false	false	false

Apskatītā izteiksme nosaka, vai skaitlis a = 5 atrodas skaitļu intervālā no 1 līdz 10, ieskaitot intervāla galapunktus.