

Teorijas datne "Mainīgie, operatīvi, Scanner klase" paredzēts programmēšanas pamatu apguvei mācību gada sākumā. Zemāk aprakstīts, kas ir mainīgie, kāds ir to mērķis programmā un tā īpašības. Pievienotajās tabulās var iepazīties ar mainīgo tipiem un to raksturojumu, deklarēšanas veidiem un mainīgo nosaukumu veidošanas noteikumiem.

Datu ievade, lietojot Scanner klasi (valodā Java) ir paredzēta, lai lietotājs var ievadīt datus no tastatūras, tādā veidā programmas padarot universālākas. Skolēniem jāzina kā piesaistīt Scanner klasi programmai un norādīt kāda tipa datus lietotājs varēs ievadīt. Zemāk ir dots paraugs programmai (pēc K. Veisa materiāliem), kurā attēlota gan datu ievade no tastatūras, gan mainīgo deklarēšana, gan aritmētiskie aprēķini.

Mainīgie

Lai programma varētu apstrādāt saņemto informāciju jeb datus, tos pirms tam ir nepieciešams saglabāt operatīvajā atmiņā. Mainīgais ir noteikta tipa dati, kas tiek glabāti operatīvajā atmiņā. Lai varētu izmantot mainīgo, vispirms programmas kodā jāietver komanda, kas to izveido. Pēc tam mainīgajam var piešķirt vērtību. Ja nepieciešams, tā var tikt mainīta daudzas jo daudzas reizes. Piemēram, ja kādas datorspēles programmai ir jāskaita spēlētāja iegūtie punkti, ir jāizveido mainīgais un jāpiešķir tam sākuma vērtība nulle. Tālāk programmas kodā noteiktos gadījumos mainīgā vērtību var palielināt vai samazināt. Mainīgajiem ir divas īpašības: datu tips un vārds.

Datu tipi

Izveidojot jeb deklarējot mainīgo, ir jānorāda, kāds būs mainīgā datu tips, piemēram, vai tajā tiks glabāts vesels skaitlis, decimāldaļa, simbols, simbolu virkne vai patiesumvērtība. Izmantojot veselo skaitļu mainīgo, papildus var norādīt, vai tajā tiks glabāti pozitīvi un negatīvi skaitļi vai tikai pozitīvi skaitļi un nulle.

Datu tips	Raksturojums
byte	8 bitus ¹ gari veseli skaitļi ar zīmi
short	16 bitus gari veseli skaitļi ar zīmi
int	32 bitus gari veseli skaitļi ar zīmi
long	64 bitus gari veseli skaitļi ar zīmi
float	32 bitus gari decimāldaļskaitļi

double	64 bitus gari decimāldaļskaitļi ar dubultu precizitāti
boolean	Loģiskais datu tips, kuram ir tikai 2 vērtības: false un true.
char	Rakstzīme, kurai atbilst 16 bitus garš Unicode sistēmas kods

Mainīgā deklarācijas vispārēja forma ir:

Datu tips identifikators= vērtība;

Byte vecums=15;

int x= 10,y= 10;

double pi = 3.14159;

string virkne="Helloworld!"

Mainīgo vārdu veidošana

Mainīgā vārds drīkst saturēt tikai mazos un lielos latīņu alfabēta burtus, ciparus un apakšsvītras. Mainīgā vārdam jā sākas ar mazo burtu vai apakšsvītru. Ir daži vārdi, kas, atbilstot iepriekš dotajiem noteikumiem, tomēr nevar tikt izmantoti mainīgo vārdiem, jo ir rezervēti īpašiem nolūkiem, piemēram, int, if, return, include, break. Programmēšanas valodas C++ un Java ir reģistrjutīgas. Tas nozīmē, ka, piemēram, mainīgo vārdi "burtuskaitis" un "burtuSkaitis" nav viens un tas pats. Mainīgā vārdam vajadzētu paskaidrot, kādus datus tajā ir paredzēts glabāt. Piemēram, "vards", "burtuSk", "sum" un "istaisBurts" ir piemēroti mainīgo vārdi. To vietā izmantot "a", "b", "c", "d" nav ieteicams, jo šādi neko neizsakoši mainīgo vārdi apgrūtina programmas koda lasīšanu.

Plašāks info: https://likta.lv/wp-content/uploads/2018/12/Programmesanas_gramata_e-versija.pdf

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.

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

Aritmētiskas izteiksmes tiek veidotas, operandus saistot ar aritmētiskajiem operatoriem.

Operators	Darbība	Izteiksmes piemērs
+	Summēšana	a+b
-	Atņemšana (starpība)	a-b
*	Reizināšana	a*b
/	Parastā dalīšana, ja vismaz viens operands nav vesela skaitļa tipa	a/b ja a=11 un b=4f (float literālis), tad rezultāts ir 2.75
/	Veselu skaitļu dalīšana bez atlikuma, ja abi operandi ir vesela skaitļa tipa	a/b Ja a=11 un b=4 (integer literāļi), tad rezultāts ir 2
%	Veselā atlikuma iegūšana	a % b Ja a=11 un b:=4, tad rezultāts ir 3

Veidojot aritmētiskas izteiksmes, jāņem vērā, ka aritmētiskās izteiksmes var saturēt apaļās iekavas; vispirms tiek pildītas darbības, kas ir apaļajās iekavās, tad reizināšana un dalīšana, un tikai tad summēšana un atņemšana.

Datu ievade, lietojot klasi Scanner

Popularitāti ir ieguvusi datu ievade, lietojot klasi **Scanner**, kas ietilpst pakotnē **java.util**. 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.

Simbolu secība	Skaidrojums
<code>\ "</code>	Izvada uz ekrāna pēdiņas
<code>\t</code>	Izveido tabulācijas atstarpi
<code>\n</code>	Pārvieto kursoru uz jaunu rindu
<code>\\</code>	Izvada uz ekrāna reverso slīpsvītru (\)
<code>\b</code>	Dzēš rakstzīmi pa kreisi
<code>\'</code>	Izvada uz ekrāna apostrofu

Tālāk dots datu ievades piemērs ar skaidrojumiem.

```
package p_2n_1; //Pakotnes p_2n_1 deklarēšana
import java.util.Scanner; //Scanner klases piesaistīšana programmai. Šo rindiņu
raksta pirms galvenās klases deklarēšanas
public class P_2N_1{ //Deklarē klasi ar nosaukumu P_2N_1.
public static void main(String[] args){ //Deklarē galveno metodi
Scanner ievade= new Scanner(System.in); //Deklarē Scanner klases objektu ar
nosaukumu ievade, norādot, ka tas būs jauns objekts, kas nodrošina datu ievadi
notastatūras
System.out.println("Ievadiet veselu skaitli:"); //Izvada uz ekrāna tekstu
int skaitlis = ievade.nextInt(); //Deklarē int tipa mainīgo skaitlis un inicializē to,
skaitļa vērtību ievadot no tastatūras
System.out.println("Ievadīts skaitlis: " +skaitlis);}} //Galvenās metodes un klases
beigas.
```

Veicot datu ievadi, tiek veikta datu atbilstības pārbaude. Ja ievadāmie dati neatbilst norādītajam datu tipam programmā, tad tiek realizēta kāda no iespējām: ja iespējams, tiek veikta datu pārveidošana jeb konvertēšana atbilstoši norādītajam datu tipam. Piemēram, dots šāds priekšraksts:

```
floatskaitlis = ievade.nextFloat();
```

Ievadot 10, kas pēc noklusējuma ir int tipa skaitlis, to pārveido float tipa skaitlī 10.0.

Piemērs

Mainīgajiem a un b tiek piešķirtas veselu skaitļu vērtības un realizēta aritmētisko darbību izpilde. Programmas pirmkodam jāatbilst šādām prasībām:

- tiek veikta mainīgo deklarēšana, inicializācija, vērtības piešķiršana mainīgajiem;
- tiek veikti aritmētiskie aprēķini ar int tipa skaitļiem;
- tiek realizēta parastā dalīšana, lietojot float tipa mainīgo;
- tiek realizēta dažāda tipa datu izvade uz ekrāna.

```
public class Piemers1 {

    public static void main(String[] args) {

        int a, summa; //Deklarē int tipa mainīgos a un summa
        a = 11; //Piešķir mainīgajam a vērtību.
        int b = 4; //Deklarē int tipa mainīgo b un piešķir tam vērtību
        (inicializē mainīgo).

        System.out.println("Skaitlis a = " + a + " un skaitlis b = " + b);
        //Izvada uz ekrāna tekstu, kas ietverts pēdīnās, un mainīgo a un b vērtības
        summa = a + b; //Piešķir mainīgajam summa izteiksmes a + b vērtību
        System.out.println("Skaitļu a un b summa ir: " +summa); //Izvada uz
        ekrāna tekstu, kas ietverts pēdīnās, un mainīgā summa vērtību
        System.out.println("Skaitļu " + a + " un " + b +" summa ir: " + (a +
        b)); //Izvada uz ekrāna tekstu, kas ietverts pēdīnās, mainīgo a un b vērtības un
        izteiksmes(a+b) vērtību
        System.out.println("Skaitļu "+ a + " un " + b +" starpība ir: " +
        (a-b)); //Izvada uz ekrāna tekstu, kas ietverts pēdīnās, mainīgo a un b vērtības
        un izteiksmes(a-b) vērtību
        System.out.println("Skaitļu " + a + " un "+b +" reizinājums ir: " +
        (a * b)); //Izvada uz ekrāna tekstu, kas ietverts pēdīnās, mainīgo a un b vērtības
        un izteiksmes(a * b) vērtību
        System.out.println("Veselu skaitļu " + a + " un " + b +" dalījums
        bez atlikuma ir: " + (a / b)); //Izvada uz ekrāna tekstu, kas ietverts pēdīnās,
        mainīgo a un b vērtības un izteiksmes(a / b) vērtību
        System.out.println("Veselu skaitļu " + a + " un " + b +" dalījuma
        atlikums ir: " + (a % b)); //Izvada uz ekrāna tekstu, kas ietverts pēdīnās,
        mainīgo a un b vērtības un izteiksmes(a % b) vērtību
        float dalijums,c; //Deklarē float tipa mainīgos.c = 4.0f; //Piešķir
        mainīgajam c vērtību (float tipa literālis).
        c = 4.0f;
        dalijums = a/c; //Piešķir mainīgajam dalijums izteiksmes a / c
        vērtību.

        System.out.println("Skaitļu " + a + " un " + c +" parastais dalījums
        ir: " + dalijums); //Izvada uz ekrāna tekstu, kas ietverts pēdīnās, mainīgo a un c
        vērtības,un mainīgādalijums vērtību.

    }
}
```

Piemērs(2)

Tiek ievadīts vesels pozitīvs divciparu skaitlis un tiek aprēķināta ciparu summa. Lai iegūtu skaitļa pirmo ciparu, jālieto veselu skaitļu dalīšana (skaitlis/10). Lai iegūtu otro ciparu, jālieto veselā atlikuma iegūšana (skaitlis%10).

Programmas pirmkodam jāatbilst šādām prasībām:

- tiek realizēta vesela skaitļa ievade, lietojot klases Scanner iespējas;
- tiek veikta mainīgo deklarēšana;
- tiek aprēķināta divciparu skaitļa ciparu summa;
- tiek realizēta dažāda tipa datu izvade uz ekrāna

import java.util.Scanner; //Scanner klases piesaistīšana programmai. Šo rindinu raksta pirms galvenās klases deklarēšanas.

```
public class Piemeri2 {

    public static void main(String[] args){

        System.out.println("Divciparu skaitla ciparu summas
aprekins");
        System.out.println("Ievadiet divciparu skaitli!");

        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.
        int divciparuSkaitlis = ievade.nextInt(); //Deklarē int tipa
mainīgo divciparuSkaitlis un inicializē to, skaitļa vērtību ievadot ar
tastatūru.

        System.out.println("Divciparu skaitlis = " +
divciparuSkaitlis); //Izvada uz ekrāna tekstu un mainīgā divciparuSkaitlis
vērtību.

        int pirmaisCipars, otraisCipars, ciparuSumma; //Deklarē int
tipa mainīgos.
        pirmaisCipars = divciparuSkaitlis / 10; //Piešķir mainīgajam
pirmaisCipars izteiksmes divciparuSkaitlis / 10 vērtību.
        otraisCipars = divciparuSkaitlis % 10; //Piešķir mainīgajam
otraisCipars izteiksmes divciparuSkaitlis % 10 vērtību.
        ciparuSumma = pirmaisCipars + otraisCipars; //Piešķir
mainīgajam ciparuSumma aprēķina izteiksmes vērtību.

        System.out.println("Divciparu skaitla" + divciparuSkaitlis + "
ciparu summa ir: " + ciparuSumma); //Izvada uz ekrāna tekstu, kas ietverts
pēdējās, un mainīgo vērtības.
    }

}
```

Piemērs(3)

Izstrādāt programmu ar nosaukumu Piemers3. Tiek realizēta teksta un decimāldaļ skaitļu ievade un aprēķināts izteiksmes $\frac{(x+y) \cdot (x-y)}{y-x}$ rezultāts.

```
import java.util.Scanner;//Scanner klases piesaistišana programmai.

public class Piemers3 {

    public static void main(String[] args) {

        Scanner ievade = new Scanner(System.in);// Deklarē Scanner
klases objektu

        System.out.println("Ievadiet savu vardu un uzvardu: ");
        String vardsUzvards = ievade.nextLine();//Deklarē String tipa
mainīgovardsUzvardsun inicializē to, vērtību ievadot ar tastatūru.
        System.out.println("Jusu vards un uzvards ir: " +
vardsUzvards);
        System.out.println("Ievadiet decimāldalskaitli x: ");
        float x = ievade.nextFloat(); //Deklarēfloattipa mainīgo x un
inicializē to, skaitļa vērtību ievadot ar tastatūru.
        System.out.println("Ievadits skaitlis x: " + x);
        System.out.println("Ievadiet decimāldalskaitli y : ");
        float y = ievade.nextFloat();

        System.out.println("Ievadits skaitlis Y: " + y);
        float izteiksmesRezultats;//Deklarē float tipa mainīgo
izteiksmesRezultats.
        izteiksmesRezultats = (x+y)*(x-y)/(y-x);//Mainīgajam
izteiksmesRezultats piešķir aprēķina rezultātu.
        System.out.println("Izteiksmes rezultats ir: "
+izteiksmesRezultats);//Izvada uz ekrāna tekstu un mainīgā
izteiksmesRezultats vērtību.

    }

}
```

Rezultātā:

```
Ievadiet savu vardu un uzvardu:
Jānis Bērziņš
Jusu vards un uzvards ir: Jānis Bērziņš
Ievadiet decimāldalskaitli x:
2,5
Ievadits skaitlis x: 2.5
Ievadiet decimāldalskaitli y :
2,
Ievadits skaitlis Y: 2.0
Izteiksmes rezultats ir: -4.5
```

Piemērs(4)

Programmas pirmkodam jāatbilst šādām prasībām:

- programmas pirmkoda 1. rindā izveidot komentāru, norādot programmas autora vārdu un uzvārdu;
- 1.rindas teksts satur pēdiņas;
- 2. rindā izvadīti skaitļi 10, 20 un 30, lietojot tabulācijas atstarpes;
- no 3. līdz 6. rindai ar vienu izvades priekšrakstu izvadīti burti A, B, C un D, katrs jaunā rindā, pie tam pirms burtiem B un C tiek izveidotas tabulācijas atstarpes.

```
public class Piemers4 {
    public static void main(String[]args) {
        System.out.println("Lugas\"Zelta zirgs\" autors ir Rainis");
        /*Lietojot simbolu secību \"panāk,
        ka uz ekrāna tiek izvadīts teksts \"Zelta zirgs\" pēdinās.*/

        System.out.println("10\t20\t30");
        /*Lietojot simbolu secību \t panāk, ka skaitļi
        * 10, 20 un 30 tiek izvadīti ar tabulācijas atstarpēm
        */
        System.out.print("A\n\tB\n\t\tC\n\n");
        /*Lietojot simbolu secību \n panāk, ka katrs burts
        * tiek izvadīts jaunā rindā. Vēl tiek lietota secība
        \ t, lai veidotu tabulācijas atstarpes. Piemēram, A\ n
        izvada A un pārvieto kursoru jaunā rindā, \tB\n
        izveido tabulācijas atstarpi, izvada B un pārvieto kursoru
        jaunā
        rindā utt. */
    }
}
```

Rezultātā:

```
<terminated> Piemers4 [Java Application] C:\Pro...
Lugas\"Zelta zirgs\" autors ir Rainis
10 20 30
A
B
C
```

Piemērs(5)

Programmas pirmkodam jāatbilst šādām prasībām:

- programmas pirmkoda 1. rindā izveidot komentāru//Programmas autors Vards Uzvars
- 1.rindas saturs izvadīts ar vienu izvades priekšrakstu;
- 2.rinda tukša;
- 3. rindas saturs izvadīts ar vienu izvades priekšrakstu;
- 4. rindas saturs izvadīts ar 4 izvades priekšrakstiem, no kuriem katrs izvada vienu vārdu;
- 5.rindas saturs izvadīts ar vienu izvades priekšrakstu;
- no 6.līdz8.rindaiizvadīts trīsstūris. Katru rindu izvada ar vienu izvades priekšrakstu, rindas saturu veidojot ar atstarpes taustiņu un rakstzīmi *
//Programmas autors Karlis Veiss
//Komentārs, kurā norāda savu vārdu uzvārdu.

```
public class Piemers5 //Klases deklarēšana

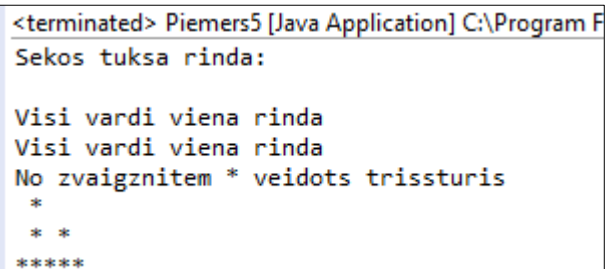
{ //Klases sākums

    public static void main(String[]args) //Galvenās metodes deklarēšana
    { //Galvenās metodes sākums
        System.out.println("Sekos tukša rinda:"); //Izvada tekstu "Sekos
tuksa rinda:" un pārvieto kursoru jaunā rindā.
        System.out.println(""); //Pārvieto kursoru jaunā rindā,
veidojot tukšu rindu
        System.out.println("Visi vardi viena rinda");
        System.out.print("Visi ");
        System.out.print("vardi ");
        System.out.print("viena ");
        System.out.print("rinda");
        /*Katrs priekšraksts izvada vienu vārdu. Tā kā kursori paliek
tajā pašā rindā, tad viss
        teksts tiek izvadīts vienā rindā.*/
        System.out.println("");

        System.out.println("No zvaigznitem * veidots trissturis"); //Izvada
tekstu "No zvaigznitem * veidots trissturis" un pārvieto kursoru jaunā rindā.
        System.out.println(" *"); //Ievada 2 atstarpes, tad zvaigznīti *, un
pāriet uz jaunu rindu.
        System.out.println(" * *"); //Ievada atstarpi, zvaigznīti*, atstarpi,
zvaigznīti*, un pāriet uz jaunu rindu.
        System.out.println("*****");

    } //Galvenās metodes beigas

} //Klases beigas
```



```
<terminated> Piemers5 [Java Application] C:\Program F
Sekos tuksa rinda:

Visi vardi viena rinda
Visi vardi viena rinda
No zvaigznitem * veidots trissturis
*
* *
*****
```

Info pēc K.Veisa materiāliem