

DATU IEVADE, FORMATĒŠANA, IZVADE 20.nodarbība

BEZ IEVADES UN IZVADES

- Programma bez ievades un izvades:

```
public static void main(String[] args) {  
    // programma aprēķina taisnstūra laukumu  
    int platums = 5;  
    int augstums = 10;  
    int laukums = platums * augstums;  
}
```

- Programma neko neizvada.

PROGRAMMA AR IZVADI: DATU TIPS INT

- Programma izvada int tipa datus:

```
public static void main(String[] args) {  
    // programma aprēķina taisnstūra laukumu  
    int platums = 5;  
    int augstums = 10;  
    int laukums = platums * augstums;  
    System.out.println( "L = " + laukums );  
}
```

- Programma izvada: L = 50

PROGRAMMA AR IZVADI: DATU TIPS DOUBLE

- Programma izvada double tipa datus:

```
public static void main(String[] args) {  
    // programma aprēķina taisnstūra laukumu  
    double platums = 5.1;  
    double augstums = 10.2;  
    int laukums = platums * augstums;  
    System.out.println( "L = " + laukums );  
}
```

- Programma izvada: L = 52.019999999999996

IZVADES FORMATĒŠANA

- Izmanto metodi `System.out.format(...)`
- Metodes parametri:
 - pirmais parametrs norāda formātu,
 - pārējie parametri norāda vērtības.
- Piemērs:

```
int i = 12;
System.out.format( "Skaitlis = %d", i );
```
- Izvada: Skaitlis = 12

FORMĀTA NORĀDĪTĀJI

- Sākas ar % (procentu) zīmi.
- Biežāk lietotie norādītāji:
 - %d – vesels skaitlis (int),
 - %f – decimāls skaitlis (float),
 - %s – simbolu virkne (String),
 - %c – simbols (char),
 - \n – jaunas rindas simbols,
 - ...

FORMĀTA NORĀDĪTĀJU PIEMĒRS

- Programma:

```
int i = 12;
double d = 12.34;
String t = "teksts";
char c = 'A';
System.out.format( "%d, %f, %s, %c", i, d, t, c );
```

- Izvada: 12, 12.340000, teksts, A

FORMĀTA NORĀDĪTĀJU PARAMETRI

- %[flags][width][.precision]conversion
jeb
- %[karogs][platums][.precizitāte]tips

FORMĀTA NORĀDĪTĀJU PARAUGI I

- Veseliem skaitļiem:
 - %6d (platums 6 simboli)
 - %6d (platums 6 simboli)
 - %06d (6 simboli, sākumā nulles)
- Izvadīs:
 - 35
 - 2345
 - 000035

FORMĀTA NORĀDĪTĀJU PARAUGI II

- Decimālie skaitļi:
 - %5.2f (kopējais platums 5 simboli)
 - %9.1f (platums 9 simboli)
 - %9.3f (platums 9 simboli, ieskaitot punktu un mīnusa zīmi)
- Izvadīs:
 - 5.12
 - 456.3
 - -456.346

FORMĀTA NORĀDĪTĀJU PARAGI III

- Decimālo skaitļu noapaļošana (6.987):
 - `%5.0f`
 - `%5.1f`
 - `%5.2f`
- Izvadīs:
 - `7`
 - `7.0`
 - `6.99`

KĻŪDU «ĶERŠANA»

- Ja norādīti formātam neatbilstoši dati, tad Java notiek izņēmums (exception):
 - `System.out.format("%4d\n", 5.28);`
 - Exception in thread "main"
java.util.IllegalFormatException: d != java.lang.Double
- Izlabo kļūdaino kodu.
- Var izmantot `try`, `catch` bloku:

```
try {
    System.out.format( "%4d\n", 5.28 );
} catch ( Exception e ) {
    System.out.println( "Kļūda" );
}
```

JA PLATUMS PAR MAZU

- Piemēram:
 - `System.out.format("%3.1f\n", 12345.6789);`
- Izvadīs:
 - 12345.7

PLATUMU VAR NENORĀDĪT

- Decimālos skaitļos var norādīt precizitāti, nenorādot platumu:
 - `System.out.format("%.1f\n", 12345.6789);`
- Izvadīs:
 - 12345.7

RAKSTĪŠANA DATNĒ

```
try {
    PrintWriter raksta = new PrintWriter (
        new FileWriter("test.txt"));
    raksta.format("Skaitļi %5d, %5d", 123, 456 );
    raksta.close();
} catch( Exception e ) {
    System.out.println( "Kļūda rakstot failu" );
}
```

SIMBOLU VIRKNES VEIDOŠANA

- Var formatēt arī teksta virknes (bez izvades):

```
int min = 15;
int max = 30;
String intervals =
    String.format("[%d;%d]", min, max);
System.out.println(intervals);
```
- Izvadīs: [15;30]

DATU IEVADE

- Datu ievadišanai var lietot `Scanner()` klasi.

- Scanner prot skenēt no:
 - datnes,
 - konsoles,
 - simbolu virknes,
 - utt.
- Scanner pēc noklusējuma izmanto atdalītājus – atstarpes un pārejas jaunā rindā simbolus.

SKENĒŠANA NO SIMBOLU VIRKNES

```
String teksts = "Teksts 123 ar 456 skaitļiem";
Scanner scan = new Scanner(teksts);

String s1 = scan.next();
int i = scan.nextInt();
String s2 = scan.next();
int j = scan.nextInt();
String s3 = scan.next();

System.out.format("%s\n%d\n%s\n%d\n%s", s1, i, s2, j, s3);

scan.close();
```

IEVADES KĻŪDU KONTROLE I

```
Scanner scan = new Scanner(System.in);
int i = scan.nextInt();
System.out.println("Ievadīts="+i);
scan.close();
```

- Ja ievada tekstu nevis skaitli, tad ir kļūda:

- Exception in thread "main" java.util.InputMismatchException

IEVADES KĻŪDU KONTROLE II

```
boolean ok = false;
while( !ok ) {
    try {
        Scanner scan = new Scanner(System.in);
        int i = scan.nextInt();
        System.out.println("Ievadīts="+i);
        scan.close();
        ok = true;
    } catch (Exception e) {
        System.out.println("Ievadi veselu skaitli!");
    }
}
```

IEVADES LOKALIZĀCIJA

- Scanner izmanto sistēmā izvēlēto decimālatdalītāju (parasti punktu vai komatu).
- Lai vienmēr lietotu punktu, var norādīt ASV lokalizāciju:
 - Scanner scan = new Scanner(System.in);
 - scan.useLocale(Locale.US);