

Leksion 3

Variablat. Tipet e te dhenave

Operatoret

Përmbytja

1.3 Variablat (Variables)

Gjuha e programimit Java përcakton llojet e mëposhtme të variablave:

Variablat e instancave (Fushat jo-statike). Teknikisht, objektet ruajnë gjendjet e tyre individuale në "fusha jo-statike", dmth fushat të deklaruara pa fjalën *static*. Fushat jo-statike njihen gjithashtu si variabla të shembujve sepse vlerat e tyre janë unike për secilën instancë të një klase (për çdo objekt, me fjalë të tjera); ShpejtësiaAktuale e një biçiklete është e pavarur nga ShpejtësiaAktuale e një tjetre

Variablat e Klases (Fushat Statike). Një variabël e klasës është çdo fushë e deklaruar me modifierin *static*; kjo i tregon kompiluesit se ekziston pikërisht një kopje e këtij variabli, pavarësisht se sa herë thërritet klasa. Një fushë që përcakton numrin e marsheve për një lloj të caktuar biçikletash mund të shënohet si statike, meqënëse konceptualisht i njëjti numër ingranazhi do të zbatohet në të gjitha rastet. Kodi statik `int numGears = 6;` do të krijonte një fushë të tillë statike. Përveç kësaj, fjalakycë **final** mund të shtohet për të treguar se numri i marsheve nuk do të ndryshojë kurrë.

Variablat lokale. Njelloj me atë se si një objekt e ruan gjendjen e tij në fusha, një metodë shpesh do të ruajë gjendjen e saj të përkohshme në variabla lokale. Sintaksa për deklarimin e një variabli lokale është e ngjashme me deklarimin e një fushe (për shembull, `int count = 0;`). Nuk ka fjalëkalim të veçantë që përcakton një ndryshore si lokale; se përcaktimi vjen tërësisht nga vendndodhja në të cilën është deklaruar variabli - e cila është midis hapjes së hapjes dhe mbylljes së një metode. Si e tillë, variablat vendorë janë të dukshme vetëm për metodat në të cilat ato deklarohen; ata nuk janë të arritshëm nga pjesa tjetër e klasës.

Parametrat. Ju keni parë tashmë shembuj të parametrave, si në klasën e Bicikletës dhe në metodën kryesore të kodit "Hello World!". Kujto se nënshkrimi për metodën kryesore është **public static void main (String [] args)**. Këtu, variabli **args** është parametri i kësaj metode. Gjëja e rëndësishme për t'u mbajtur mend është se parametrat klasifikohen gjithmonë si "variabla" dhe jo "fusha". Kjo vlen edhe për konstruktet e tjera të pranimit të parametrave (siç janë konstruktoret dhe përpunuesit e përjashtimeve) që do të mësoni më vonë.

Emërtimi

Çdo gjuhë programimi ka grupin e vet të rregullave dhe konventave për llojet e emrave që ju lejohet të përdorni dhe gjuha e programimit Java nuk është ndryshe. Rregullat dhe konventat për emërtimin e variablave tuaj mund të përmbliken si më poshtë:

- Emrat e variablave janë të ndjeshme ndaj shkronajve të mëdha, të vogla apo simboleve të ndryshme (case-sensitive). Emri i një variabli mund të jetë çdo identifikues ligjor - një sekuencë e pakufizuar e shkronjave dhe shifrave Unicode, duke filluar me një shkronjë, shenjën dollare "\$" ose karakterin nënvizues "_". Konventa, megjithatë, është që gjithmonë të fillojë emrat e variablave me një shkronjë, jo "\$" ose "_". Përveç kësaj, karakteri i shenjës së dollarit, sipas konventës, kurrë nuk përdoret. Mund të gjeni disa situata ku emrat e gjeneruar automatikisht do të përmbajnë shenjën e dollarit, por emrat tuaj të variablave duhet gjithmonë ta shmangin përdorimin e saj. Një konventë e ngjashme ekziston për karakterin

nënvizues; ndërkohë që është teknikisht e ligjshme për të filluar emrin e ndryshuesit tuaj me "_", kjo praktikë është e dekurajuar. Hapësira e bardhë nuk lejohet.

- Karakteret e mëvonshme mund të jenë shkronja, shifra, shenja të dollarit ose karaktere nënvizuese. Konventat (dhe kuptimi i përbashkët) zbatohen edhe për këtë rregull. Kur zgjedhni një emër për variablat tuaj, përdorni fjalë të plota në vend të shkurtimeve të fshehta. Duke bërë kështu, do të lehtësojë leximin dhe kuptimin e kodit tuaj. Në shumë raste do të bëjë edhe kodin tuaj të dokumentuar vetë; fusha me emrin ritëm, shpejtësi dhe veshje, për shembull, janë shumë më intuitive se versionet e shkurtuara, të tilla si s, c dhe g. Gjithashtu mbani në mend se emri që zgjidhni nuk duhet të jetë një fjalekyce ose fjalë e rezervuar.
- Nëse emri që zgjidhni është vetëm një fjalë, shkruaj atë fjalë në të gjitha shkronjat e vogla. Nëse përbëhet nga më shumë se një fjalë, vendosni me te madhe shkronjën e parë të çdo fjale. Emrat gearRatio dhe currentGear janë shembuj kryesorë të kësaj konvente. Nëse variabli juaj mban një vlerë konstante, siç është static final NUM_GEAR = 6, konventa ndryshon pak, duke vendosur me shkronje te madhe çdo shkronje dhe ndan fjalët pasardhese me karakterin nënvizues. Sipas konventës, karakteri i nënshkrimit nuk përdoret kurrë diku tjetër.

Sic pamë tek program mësipër, variablat përdoren për të ruajtur vlerat që mund të përdoren më vonë në program. Quhen variabla sepse vlerat e tyre mund të ndryshojnë. Mund ti vendosim çfarëdo numri variablave rreze dhe sipërfaqe dhe vlerat e tyre mund të ndryshojnë.

```
// Llogarit sipërfaqen e parë

rreze= 1.0;

sipërfaqja= rreze * rreze * 3.14159;

System.out.println("Sipërfaqja është " + sipërfaqe + " për
rreze " + rreze);

// Llogarit sipërfaqen e dytë

rreze = 2.0;

sipërfaqja= rreze * rreze * 3.14159;

System.out.println("sipërfaqja është" + sipërfaqe+ " për rreze
" + rreze);
```

Variablat përfaqësojnë të dhëna të një tipi të caktuar. Deklarimi i variablave i tregon kompilatorit se sa hapsirë të ruajë në memorie për variablat bazuar në tipin e tyre të të dhënave. Sintaksa e deklarimit të variablave është:

datatype emriVariablit;

Këtu paraqiten disa shembuj të deklarimit të variablave:

```
int count; // Deklaro rreze për të qenë një variable integer;

double rreze; // Deklaro rreze për të qenë një variable
double;
```

```
double normaInteresit; // Deklaro normaInteresit për të qenë  
një variable double;
```

Nëse variablat janë të të njëjtit tip mund të deklarohen së bashku si:

```
datatype variable1, variable2, ..., variablen;
```

Variablat janë të ndara nga (,)

Variablat shpesh kanë vlera fillestare. Mund të deklarojmë një variabël dhe e inicializojmë atë në një hap.

```
p.sh. int count = 1;
```

Kjo është ekuivalente me këto dy rreshta:

```
int count;  
x = 1;
```

Ose mund të shkruajmë:

```
int i = 1, j = 2;
```

1.4 Caktimi i deklaratave dhe caktimi i shprehjeve (Assignment Statements and Assignment Expressions)

Pas deklarimit të një variable, mund ti shënjojmë një vlerë asaj duke përdorur assignment statement. Sintaksa e përdorur për të është e tillë:

```
variable = shprehje;
```

Një expression(shprehje) përfaqëson një llogaritje(veprim) që përfshin vlera, variabla dhe operatorë të marrë së bashku. Me poshtë japim një shembull:

```
System.out.println(x = 1);
```

Është ekuivalent me

```
x = 1;  
System.out.println(x);
```

Mënyra e shprehjes mëposhtë është gjithashtu e saktë:

```
i = j = k = 1;
```

Është ekuivalent me:

```
k = 1;  
j = k;  
i = j;
```

1.5 Named constants

Vlerat e variablave mund të ndryshojnë gjatë ekzekutimit të një program, por një konstante përfaqëson të dhëna që janë të pandryshueshme. Përshëmbull π është një konstante e deklarimi bëhet I tillë:

```
final datatype CONSTANTNAME = VALUE;
```

Një konstante duhet të deklarojete që në fillim. Fjala Final është një nga fjalët kyce të Java-s për deklarimin e një konstanteje. Përshëmbull deklarojmë π si konstante dhe shkruajme:

LLogarisim sipërfaqen e rrethit:

```
public class Siperfaqja {  
    public static void main(String arg[]){  
  
        double rrezja;                //deklaron variablin  
rrezja  
        double siperfaqja;            //deklaron variablin  
siperfaqja  
        double pi;                    //deklaron variablin pi  
  
        pi = 3.14159;                //inicializon  
variablin pi  
        rrezja = 20;                 //inicializon variablin  
rrezja  
  
        //formula për të gjetur sipërfaqen e rrethit  
        siperfaqja = pi * rrezja * rrezja;  
  
        // printo në ekran shkrimin, rrezen dhe sipërfaqen  
        System.out.println("Siperfaqja e rrethit me rreze: "  
+ rrezja + ", eshte: " + siperfaqja);  
  
    }    // mbaron metoda main  
}        // mbaron klasa Siperfaqja
```

Cfarë janë Operatorët?

Operatorët janë simbole speciale që kryejnë operacione specifike të parapërcaktuara.

Operatorët në tabelën e mëposhtme janë renditur sipas përparësisë. Operatorët me përparësi më të lartë vlerësohen më parë se ata që kanë përparësi më të ulët. Operatorët në të njëjtën linjë kanë përparësi të barabartë. Kur operatorët me përparësi të barabartë ndodhen në të njëjtën shprehje, të gjithë operatorët binare përveç operatorit të caktimit (=), vlerësohen nga e majta në të djathtë; Operatori i caktimit vlerësohet nga e djathta në të majtë.

Përmbledhje për Operatorët

Më poshtë janë listuar shumica e llojeve të operatorëve që suportohen nga JAVA.

=	<i>operatori i caktimit</i>
+	<i>operatori i mbledhjes (edhe për të lidhur stringje)</i>
-	<i>operatori i zbritjes</i>
*	<i>operatori i shumezimit</i>
/	<i>operatori i pjestimit</i>
%	<i>operatori modul</i>
+	<i>vlere pozitive</i>
-	<i>vlere negative</i>
++	<i>inkremental</i>
--	<i>dekremental</i>
!	<i>logjik (operatori mohues për tipin Boolean)</i>
==	<i>e barabartë me (kushtezor)</i>
!=	<i>jo e barabartë me (kushtezor)</i>
>	<i>me e madhe se</i>
>=	<i>me e madhe baraz me</i>
<	<i>me e vogël se</i>
<=	<i>me e vogël baraz me</i>
&&	<i>kushtezimi dhe</i>
	<i>kushtezimi ose</i>
?:	<i>shkurtim për shprehjen if-then-else</i>
instanceof	<i>krahason një object me një type specifike</i>
&	<i>dhe</i>
^	<i>perjashtues</i>
	<i>perfshires</i>

Operatorët e krahasimit për Java-n jepen në tabelën më poshtë:

Tabela 3.1 Operatorët e krahasimit

Ermi i operatorit	Shembull	Rezultati
< më pak se	rreze < 0	false
<= më pak se ose baraz me	rreze <= 0	false
> më i madh se	rreze > 0	true

<code>>=</code> më i madh se ose baraz me	<code>rreze >= 0</code>	<code>true</code>
<code>==</code> I barabartë me	<code>rreze == 0</code>	<code>false</code>
<code>!=</code> I ndryshëm nga	<code>rreze != 0</code>	<code>true</code>

Rezultati i shprehjeve është një vlerë Boolean: **true** ose **false**.

Shëmbull – Operatori Caktimit

Ky shëmbull shpjegon mënyren se si funksionon operatori i caktimit. Do të kuptoni që vlera kalon gjithmonë nga e djathta në të majtë. Në këtë shëmbull do të ndërojnë vlerat dy variablat a dhe b, dhe për ta realizuar këtë na nevojitet një variabël i tretë, që po e quajmë temp.

```
public class OperatoriCaktimit {
    public static void main(String[] args) {
        int a = 5, b = 7, temp = 0;
        System.out.println("a = " + a + "; b = " + b + "; temp = " + temp);
        temp = a; //i jap temp-it vleren e a-se
        a = b;     //i jap a-se vleren e b-se
        b = temp;  //i jap b-se vleren e temp
        System.out.println("a = " + a + "; b = " + b + "; temp = " + temp);
    }
}
```

Console

a = 5; b = 7; temp = 0

a = 7; b = 5; temp = 5

Shëmbull – Operatori ?: (if-then-else)

Operatori ?:, është shkurtim i shprehjes if-then-else. Për shëmbull, shprehjen e mëposhtme:

```
if(x>0){
    y=1;
}
else{
    y=-1;
```



```
}
```

mund ta shkuajme ndryshe si:

```
y = (x>0) ? 1 : -1;
```

Kuptohet që të dyja shprehjet kanë të njejtin funksion (pra bëjnë të njetën punë). Sintaksa e këtij operatori është:

```
shprehja-booleane ?shprehja1 :shprehja2;
```

Nëse shprehja-booleane është true (pra nëse kushti plotësohet), atëherë kthehet shprehja1, përndryshe (nëse kushti nuk plotësohet) do të kthehet nga kodi shprehja2.

Shembull – Gjej nr max midis dy numrave

Në këtë shembull do të shikojmë përdorimin e operatorit :?

```
public class Max {  
    public static void main(String[] args) {  
        int num1=3,num2=7;  
        int max = (num1 > num2) ? num1 : num2;  
        System.out.println(max);  
    }  
}
```

Console

7

Variabli i ndërtuar nga ne **max**, do të mbajë kushin dhe do të jetë ai që do të na kthejë rezultatin.

Shembull – TekCift

Ndryshe nga shëmbulli paraardhës, në këtë shëmbull do të kuptojmë se përdorimi i një variabli për të ruajtur shprehjen-kusht dhe rezultatin nuk është i detyrueshëm. Ne mund ta shkruajmë kushtin dhe të marrim rezultatin direct në output si më poshtë:

```
import java.util.Scanner;  
public class TekCift{
```

```

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.println("Fut nje nr te plote");
        int num = input.nextInt();
        System.out.println(( num % 2 == 0) ? "eshte cift"
        : "eshte tek");
    }
}

```

Console

Fut nje nr te plote

9

eshte tek

4.2 Operatorët e shkurtuar

Simboli i operatorit	Emri i operatorit	Shembull	Ekujvalente
+=	Mbledhje barazim	x += 4;	x = x + 4;
-=	Zbritje barazim	x -= 4;	x = x - 4;
*=	Shumezim barazim	x *= 4;	x = x * 4;
/=	Pjestim barazim	x /= 4;	x = x / 4;
%=	Modul barazim	x %= 4;	x = x % 4;

Shembull–ShortOperators

Ky shëmbull i thjeshtë do t'ju tregojë se si mund të shkruhen gjatë dhe shkurt operatorët që kryejnë të njejtën punë. Në koment është mënyra e gjatë e shkrimit:

```

public class ShortOperators {
    public static void main(String[] args)
    {
        double x;
        x = 7;
        x += 4; //x=x+4
        System.out.println("+=: " +x);
    }
}

```

```

        x = 7; //i rijep x-it vleren 7, perndryshe do
ishte 11
        x -= 4; //x=x-4
System.out.println("--: " +x);
        x = 7;
x *= 4; //x=x*4
System.out.println("*=: " +x);
        x = 7;
x /= 4; //x=x/4
System.out.println("/=: " +x);

        x = 7;
x %= 4; //x=x%4
System.out.println("%=: " + x);
    }
}

```

Console

```

+=: 11.0
-=: 3.0
*=: 28.0
/=: 1.75
%=: 3.0

```

Shohim qe numrat e plote jane afishuar si (nrInt.0). Kjo sepse ne i kemi deklaruar si double.

4.3 Operatorët e rritjes (++) dhe të zbritjes (--)

Simboli i operatorit	Emri i operatorit	Shembull	Ekujvalente
++	Rritës	x++;	x = x + 1;
--	Zbritës	x--;	x = x - 1;

Shembull ++ VS –

Ky shembull do t'ju tregojë se si operon operatori ++ dhe --, në një kod

```

public class RritesZbrites {
    public static void main(String[] args) {
        int y = 7,z = 11;
        y++;
    }
}

```

```

        z--;
        System.out.println("y = " + y + " dhe z = " + z);
    }
}

```

Console

y = 8 dhe z = 10

Karakteret speciale

Shenimi	Kuptimi
\t	Tab (hapesire)
\n	Rrjesht i ri
\'	Apostrof
\"	Thonjza dopjo
\\	backslash

Shembull –KarakteretSpeciale

Më poshtë janë treguar disa simulime rastesh për të kuptuar më mirë funksionin që kryen secili karakter special:

```

public class KarakteretSpeciale {
    public static void main(String[] args) {
        System.out.println("Ky eshte nje shkrim");
//teksti ynë
        System.out.println("Ky eshte nje shk\ttrim");    //
\t
        System.out.println("Ky eshte nje shk\nrim"); // \n
        System.out.println("Ky eshte nje \'shkrim\' "); //
\'
        System.out.println("Ky eshte nje \"shkrim\" "); //
\"
    }
}

```

Console

```
Ky eshte nje shkrim
Ky eshte nje shk    rim
Ky eshte nje shk
rim
Ky eshte nje 'shkrim'
Ky eshte nje "shkrim"
```

4.4 Operatori (==)

Një tjetër mënyrë për të bërë vleresime.

Shembull – Sa

Në shembullin e mëposhtëm, do të gjenerohen dy numra një shifror, dhe t'ju kërkohet nga kodi që ti mblidhni ata dhe të shkuani rezultatin në console. Pasi të keni shtypur enter, kodi do t'ju tregoj nëse përgjigja që juve kthyet ishte e saktë (true) apo e pasaktë (false).

```
public class Sa {
    public static void main(String[] args){
        // gjeneroj numrin e pare
        int nr1 = (int) (System.currentTimeMillis() % 10);
        //shoh nr1 qe gjeneron sistemi nga 1-9
        System.out.println(nr1);
        // gjeneroj numrin e dyte
        int nr2 = (int) (System.currentTimeMillis() * 7 %
10);

        //shoh nr2 qe gjeneron sistemi nga 1-9
        System.out.println(nr2);
        //krijoj skanerin
        Scanner input = new Scanner(System.in);
        //Afishoj shkrimin
        System.out.println("Shkruani rezultatin");
        //krijoj variablin rezultat
        int rezultat = input.nextInt();
        //Vlereso nese pergjigja eshte e sakte
        System.out.println((nr1 + nr2 == rezultat));
    }
}
```

Console

3

8

Shkruani rezultatin

11

true

Console

6

2

Shkruani rezultatin

7

false

Tipi i variablit String

Është një datatype i cili mban një string.

Shembull –Tipi String

Afishimi dhe manipulimi i variablit të tipit String nëpërmjet operatorve (+) dhe (+=):

```
public class TipiString {  
    public static void main(String[] args) {  
        //krijon variablin mesazh te tipit String  
        String mesazh = "Ky eshte nje mesazh ne sting";  
        System.out.println(mesazh);  
    }  
}
```

Console

Ky eshte nje mesazh ne sting

Për të lidhur shkrimin mund të përdorni operatorin (+) si më poshtë:

```
String mesazh = "Ky" + "eshte" + "nje mesazh ne sting";
```

Dhe do të kemi të njejtin output.

Gjithashtu mund ta lidhni atë edhe me numra nëpërmjet këtij operatori:
String mesazh = "Leksioni" + "2"; dhe String mesazh = "Leksioni" + 2; do të japin të njejtin

Output:

Console

Leksioni2

Ndërsa me operatorin (+=)mund ta lidhni tekstin edhe pjese te tjera teksti si më poshte:

```
public class TipiString {
    public static void main(String[] args) {
        String mesazh = "Leksioni2 - ";
        mesazh += "Ky eshte nje mesazh ne sting";
        System.out.println(mesazh);
    }
}
```

**1.6 Tipet e të dhënave numerike dhe veprimet
(Numeric Data Types and Operations)**

Tipet e të dhënave numerike jepen në tabelen 2.2

Tabelë përmbledhese 2.2

Name	Range	Storage Size
byte	-2^7 (−128) to 2^7-1 (127)	8-bit signed
short	-2^{15} (−32768) to $2^{15}-1$ (32767)	16-bit signed
int	-2^{31} (−2147483648) to $2^{31}-1$ (2147483647)	32-bit signed
long	-2^{63} to $2^{63}-1$ (i.e., −9223372036854775808 to 9223372036854775807)	64-bit signed
float	Negative range: $-3.4028235E + 38$ to $-1.4E-45$ Positive range: $1.4E-45$ to $3.4028235E + 38$	32-bit IEEE 754
double	Negative range: $-1.7976931348623157E + 308$ to $-4.9E-324$ Positive range: $4.9E - 324$ to $1.7976931348623157E + 308$	64-bit IEEE 754

1.7 Veprimet numerike

Tabela 2.3

TABLE 2.3 Numeric Operators

<i>Name</i>	<i>Meaning</i>	<i>Example</i>	<i>Result</i>
+	Addition	34 + 1	35
-	Subtraction	34.0 - 0.1	33.9
*	Multiplication	300 * 30	9000
/	Division	1.0 / 2.0	0.5
%	Remainder	20 % 3	2

Shembull: ShfaqKohen.java

```
import java.util.*;

public class SecMin {
    public static void main(String[] args) {
        Scanner FutSec = new Scanner(System.in);

        System.out.println("Fusni nje numer te plote
per sekondat: ");

        int sec = FutSec.nextInt();
        int min = sec / 60;
        int SecTeMbetura = sec % 60;
        System.out.println(sec + " perbejne " + min + "
minuta dhe " + SecTeMbetura + " sekonda");
    }
}
```

1.8 Problem: Shfaqja e Orës Aktuale (Displaying the Current Time)

Problemi është krijimi i një program që paraqet kohën aktuale në GTM (Greenwich Mean Time) në formatin: orë: minut: second p.sh 13:19: 08

Metoda `currentTimeMillis` në klasën `System` kthen kohën e tanishme në milisekonda që prej kohës 00: 00: 00 në 01: 01: 1970. Mund të përdorim këtë metodë për të marrë kohën e tashme dhe duke llogaritur sekondat, minutat dhe orët aktuale

1. Arrihet gjetja e milisekondave totale që prej 01: 01: 1970 duke thërritur `System.currentTimeMillis()` (e.g., `1203183086328` milliseconds).
2. Gjejmë totalin e sekondave `totalSeconds` duke pjesëtur `totalMilliseconds` me 1000.
3. Llogarisim sekondat aktuale nëpërmjet `totalSeconds % 60`.
4. Llogarisim minutat totale duke pjesëtur `totalSeconds` me 60
5. Llogarisim minutën aktuale nëpërmjet `totalMinutes % 60`
6. Marrim orët totale `totalHours` duke pjesëtur `totalMinutes` me 60
7. Llogarisim orët aktuale nga `totalHours % 60`

Shembull: Gjetja e kohes aktuale.

```
public class KohaAktuale {
    public static void main(String[] args) {
        long totalMiliSec = System.currentTimeMillis();
        long totalSec = totalMiliSec / 1000;
        long secAktuale = (int)(totalSec % 60);
        long totalMin = totalSec / 60;
        long minAktuale = totalMin % 60;
        long totalOre = totalMin / 60;
        long oraAktuale = totalOre % 24;
        System.out.println(oraAktuale + ":" + minAktuale +
            ":" + secAktuale + " GMT");
    }
}
```

2.10 Operatorët e mosbarazimit (Shorthand Operators)

Tabela 2.4

TABLE 2.4 Shorthand Operators

<i>Operator</i>	<i>Name</i>	<i>Example</i>	<i>Equivalent</i>
<code>+=</code>	Addition assignment	<code>i += 8</code>	<code>i = i + 8</code>
<code>-=</code>	Subtraction assignment	<code>i -= 8</code>	<code>i = i - 8</code>
<code>*=</code>	Multiplication assignment	<code>i *= 8</code>	<code>i = i * 8</code>
<code>/=</code>	Division assignment	<code>i /= 8</code>	<code>i = i / 8</code>
<code>%=</code>	Remainder assignment	<code>i %= 8</code>	<code>i = i % 8</code>

1.9 Numeric Type Conversions

Shembull: TaksaShitje.java

```
import java.util.Scanner;

public class TaksaShitjes {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.print("Tregoni Shumen e blerjes: ");
        double shumaBlerjes = input.nextDouble();

        double taksa = shumaBlerjes * 0.06;

        System.out.println("Taksa e shitjes eshte " + (int)
            (taksa * 100) / 100.0);
    }
}
```

Tipi i te dhenave Boolean

Tipi i te dhenave Boolean deklarohet nje variable me vleren True (te vertete) ose False (te gabuar)

Java Operator	Mathematics Symbol	Name	Example (radius is 5)	Result
<	<	less than	radius < 0	false
<=	≤	less than or equal to	radius <= 0	false
>	>	greater than	radius > 0	true
>=	≥	greater than or equal to	radius >= 0	true
==	=	equal to	radius == 0	false
!=	≠	not equal to	radius != 0	true

1.10 Problem: Llogaritja e kredisë (Computing Loan Payments)

Problemi është të shkruajmë një program që llogarit kredimë. Kredia mund të jetë, kredi për makinë, kredi studentore ose për shtëpi. Programi bën të mundur që ti të fusësh përqindjen e interesit, numrin e viteve, shumën e kredisë dhe të llogarit koston mujore dhe totalin e pagesës. Formula për llogaritjen e koston mujore është:

$$\frac{\text{shuma kredisë} \times \text{interesi mujor}}{1 - \frac{1}{(1 + \text{interesi mujor})^{\text{nr. i viteve} \times 12}}} = \text{pagesa mujore}$$

Në formulë duhet llogaritur $(1 + \text{monthly Interest})^{\text{nr. i viteve} \times 12}$

kjo bëhet nëpërmjet klasës `Math`, p.sh. `Math.pow(a,b) = ab`

Shembull: LlogaritKredine.java

```
public class LlogaritKredine {
    public static void main(String[] args) {
        // Krijojmë Objektin Skaner
        Scanner input = new Scanner(System.in);
        // Vendosni interesin vjetor
        System.out.print("Vendosni interesin vjetor, psh, 8.5");
        double normaVjetoreInteresit = input.nextDouble();

        // Vendosni interesin mujor
        double normaMujoreInteresit = normaVjetoreInteresit /
1200;

        // Vendos një numër për vitet
        System.out.print("Vendos një numër të plotë për vite, psh
5");
        int numriViteve = input.nextInt();

        // Shkruani shumën e kredisë
        System.out.print("Jepni shumën e kredisë, psh.120000.95:
");
    }
}
```

```

    double shumaKredise = input.nextDouble();

    // Llogarit pagesen
    double pagesaMujore = shumaKredise * normaMujoreinteresit
/ (1- 1 / Math.pow(1 + normaMujoreinteresit, numriViteve
* 12));
    double pagesaTotale = pagesaMujore * numriViteve * 12;

    // Shfaq rezultatin
    System.out.println("Pagesa mujore eshte " + (int)(
pagesaMujore * 100) / 100.0);
    System.out.println("Pagesa totale eshte " + (int)(
pagesaTotale * 100) / 100.0);

}

}

```