

Aritmetika dhe variablat

Aritmetika dhe variablat

Objektivat:

- ⑥ Të futet nocioni i tipit të së dhënës, i cili siguron konsistencë gjatë llogaritjeve.
- ⑥ Të futen variablat kompjuterike dhe zbatimet tradicionale të tyre.
- ⑥ Të jepet një mënyrë e thjeshtësuar por efektive për t'i dhënë të dhënat hyrëse programit.

Aritmetika e numrave të plotë

```
/** Llogarit sasinë e parave të imëta. */
public class Total
{ public static void main (String[] args)
  { System.out.print("Për 4 gjysma, 2 njëzetqindarka, ");
    System.out.print("asnjë dhjetëqindarkë, 1 pesëqindarkë ");
    System.out.println("dhe 6 qindarka,");
    System.out.print("shuma totale është ");
    System.out.println((4 * 50) + (2 * 20) + (0 * 10)
                       + (1 * 5) + (6 * 1));
  }
}
```

Aritmetika e numrave të plotë – Vazhdim

Operator: simbol, si $*$ ose $+$, i cili shënon një operacion aritmetik.

Operand: argument i një operatori, p.sh., 1 në $6 * 1$.

Aritmetika e numrave të plotë – Vazhdim

```
public class Test
{ public static void main(String[] args)
  { System.out.println(1 + ((2 - 3) * 4) + 5); }
}
```

$$\begin{aligned} & 1 + ((2 - 3) * 4) + 5 \\ \Rightarrow & 1 + ((-1) * 4) + 5 \\ \Rightarrow & 1 + (-4) + 5 \\ \Rightarrow & -3 + 5 \\ \Rightarrow & 2 \end{aligned}$$

Madhësitë e emërtuara: variablat

```
/** Llogarit sasinë e parave të imtëa
 * mbështetur në vlerat e emërtuara me variabla. */
public class Total
{ public static void main (String[] args)
  { int gjysma = 4;
    int njezeta = 2;
    int dhjeta = 0;
    int pesa = 1;
    int qindarka = 6;
```

Madhësitë e emërtuara: variablat – Vazhdim

```
System.out.println("Për këto sasi metelikësh:");
System.out.print("Gjysma = ");
System.out.println(gjysma);
System.out.print("Njëzetqindarka = ");
System.out.println(njezeta);
System.out.print("Dhjetëqindarka = ");
System.out.println(dhjeta);
System.out.print("Pesëqindarka = ");
System.out.println(pesa);
System.out.print("Qindarka = ");
System.out.println(qindarka);
System.out.print("Shuma totale është ");
System.out.println( (gjysma * 50) + (njezeta * 20)
    + (dhjeta * 10) + (pesa * 5) + (qindarka * 1) );
}
}
```

Madhësitë e emërtuara: variablat – Vazhdim

Variabël (ndryshore): qelizë e cila përmban një vlerë primitive të ruajtur (si, p.sh., një numër të plotë) ose një adresë të një objekti.

Deklarim: forma e urdhërit që konstrukton variabël, p.sh.,
`int gjysma = 4;`

Inicializim: urdhëri i cili i jep variablës vlerën e parë, initiale të saj.

Tipi i të dhënave: bashkësi e emërtuar, ose „lloj“ i vlerave.

Madhësitë e emërtuara: variablat – Vazhdim

Fjala kyçe në Java për tipin e numrave të plotë është `int`.

Literal: vlera aktuale e një tipi, p.sh. numrat e plotë si 1 dhe 115 janë literalë të tipit `int`.

Referencim i variablës: sjellja në shprehje e një kopjeje të vlerës së ruajtur në variabël.

Verifikimi i tipit të të dhënave: kontrolli të cilin e bën kompilatori për të siguruar se tipi i të dhënave të variablës është i pranueshëm për kontekstin e referencimit të variablës.

Madhësitë e emërtuara: variablat – Vazhdim

```
/** Llogarit sasinë e parave të imëta. */  
public class Total  
{ public static void main (String[] args)  
  { int gjysma = 4;  
    int njezeta = 2;  
    int dhjeta = 0;  
    int pesa = 1;  
    int qindarka = 6;
```

Madhësitë e emërtuara: variablat – Vazhdim

```
System.out.println("Për këto sasi metelikësh:");
System.out.println("Gjysma = " + gjysma);
System.out.println("Njëzetqindarka = " + njezeta);
System.out.println("Dhjetëqindarka = " + dhjeta);
System.out.println("Pesëqindarka = " + pesa);
System.out.println("Qindarka = " + qindarka);
int total = (gjysma * 50) + (njezeta * 20)
            + (dhjeta * 10) + (pesa * 5) + (qindarka * 1);
System.out.println("Shuma totale është " + (total / 100)
                  + " Euro dhe " + (total % 100) + " cent");
    }
}
```

Ndryshoret mund të ndryshojnë: ndarja e vlerës

Total

```
public static void main(String[] args)
{ int gjysma == 4
  int njezeta == 2
  int dhjeta == 0
  int pesa == 1
  int qindarka == 6
    >System.out.println("Për këto sasi metelikësh:");
  System.out.println("Gjysma = " + gjysma);
  ...
}
```

Ndryshoret mund të ndryshojnë: ndarja e vlerës – Vazhdim

Ndarje vlere: forma e urdhërit e cila vë një vlerë të re në variabël.

Shembuj:

```
int money = 0;  
money = 100;  
money = money + 50; // money += 50;
```

Ndryshoret mund të ndryshojnë: ndarja e vlerës – Vazhdim

```
/** Llogarit sasinë e metelikeve. */
public class MakeChange
{ public static void main (String[] args)
  { int euros = 3;
    int cents = 46;
    int money = (euros * 100) + cents;
    System.out.println("Gjysma = " + (money / 50));
    money = money % 50; // money %= 50
    System.out.println("Njëzetqindarka = " + (money / 20));
    money = money % 20;
    System.out.println("Dhjetëqindarka = " + (money / 10));
    money = money % 10;
    System.out.println("Pesëqindarka = " + (money / 5));
    money = money % 5;
    System.out.println("Qindarka = " + money);
  }
}
```

Ndryshoret mund të ndryshojnë: ndarja e vlerës – Vazhdim

MakeChange

```
public static void main(String[] args)
{ int euros == 3
  int cents == 46
  int money == 346
  >System.out.println("Gjysma = " + (money / 50));
  money = money % 50;
  System.out.println("Njëzetqindarka = " + (money / 20));
  money = money % 20;
  ...
}
```

Ndryshoret mund të ndryshojnë: ndarja e vlerës – Vazhdim

MakeChange

```
public static void main(String[] args)
{
    int euros == 3
    int cents == 46
    int money == 46
    ...
    >System.out.println("Njëzetqindarka = " + (money / 20));
    money = money % 20;
    ...
}
```


Ndryshoret mund të ndryshojnë: ndarja e vlerës – Vazhdim

Semantika e urdhërave të ndarjeve të vlerës, të cilët mund të kenë paraqitje cirkulare, si

`money = money % 50; // money %= 50`

konsiston nga tre hapat vijues:

1. Alokohet variabla (e emërtuar `money`).
2. Llogaritet rezultati i shprehjes në anën e djathtë të shenjës së barazimit (`money % 50`), duke referencuar vlerën të cilën për momentin e përmban variabla.
3. Rezultati nga hapi paraprak vendoset në qelizën e variablës (`money`) si vlerë e re e saj.

Operatorët e inkrementimit dhe dekrementimit

Operatori i inkrementimit (rritjes) ++: rrit për 1 vlerën e operatorit.

Forma postfikse:

```
int count = 10;  
count++; // sikur count = count + 1
```

Forma prefikse:

```
++count; // sikur count = count + 1
```

Operatorët e inkrementimit dhe dekrementimit – Vazhdim

Operatori i dekrementimit (zvogëlimit) --: zvogëlon për 1 vlerën e operatorit.

Forma postfikse:

```
count--; // sikur count = count - 1
```

Forma prefikse:

```
--count; // sikur count = count - 1
```

Operatorët e inkrementimit dhe dekrementimit – Vazhdim

Operatorët e inkrementimit dhe dekrementimit mund të përdoren përbrenda një shprehjeje:

```
int numApples = 5;  
int numOranges = 10;  
int numFruit = 0;
```

```
numFruit = numApples++ + numOranges; // numFruit == 15  
                                         // numApples == 6
```

```
numFruit = ++numApples + numOranges; // numApples == 7  
                                         // numFruit == 17
```

Operatorët e inkrementimit dhe dekrementimit – Vazhdim

```
int numApples = 5;  
int numOranges = 10;  
int numFruit = 0;
```

```
numFruit = numApples-- + numOranges; // numFruit == 15  
                                         // numApples == 4
```

```
numFruit = --numApples + numOranges; // numApples == 3  
                                         // numFruit == 13
```

Aritmetika e numrave thyesorë

Fjala kyçe në Java për tipin e numrave thyesorë është `double`.

```
int c = 22;  
double p = 3.14159;  
double t = 4.7E17;  
double d = 9.0;
```

Një veti e rëndësishme e operatorëve themelorë aritmetikë `+`, `-`, `*`, `/`:

- ⑥ Kur zbatohen mbi dy operandë të tipit `int`, rezultati është i tipit `int`.
- ⑥ Kur së paku njëri nga operandët është i tipit `double`, rezultati është i tipit `double`.

Aritmetika e numrave thyesorë – Vazhdim

```
/** Konverton vlerën e temperaturës. */  
public class FahreheitToCelsius  
{ public static void main(String[] args)  
  { int f = 70;  
    double c = (5.0/9.0) * (f - 32);  
    System.out.println("Për gradë Fahrenheit " + f + ",");  
    System.out.println("gradë Celsius " + c);  
  }  
}
```

Aritmetika e numrave thyesorë – Vazhdim

Ekziston klasa `Math` e cila përmban metoda për llogaritje me numra thyesorë:

```
double num = 1E-2;
```

```
System.out.println("Rrënja e " + num  
    + " është " + Math.sqrt(num));
```

```
System.out.println("Kubi i " + num  
    + " është " + Math.pow(num, 3));
```

```
System.out.println("Logaritmi natyror i " + num  
    + " është " + Math.log(num));
```


Aritmetika e numrave thyesorë – Vazhdim

Shembull:

```
int i = 1;  
double d = i;
```

Shembull jokorrekt:

```
double d = 1.6;  
int i = d;
```

Programuesi mund të shfrytëzojë *shprehjen e konvertimit*
(*cast expression*):

```
double d = 1.6;  
int i = (int)d; // Vlera e cunuar
```

Operatorët *OP* =

Sintaksa:

NDRYSHORE OP = SHPREHJE

Semantika e njëjtë sikurse e ndarjes së vlerës:

NDRYSHORE = NDRYSHORE OP (SHPREHJE)

Shembuj:

```
count += 5;    // sikur count = count + 5  
money %= 50;   // sikur money = money % 50
```

Tipi boolean

Tipi i të dhënave boolean konsiston nga dy vlerat logjike: true dhe false.

```
double c = 22.0;
double f = 77.0;
double fConverted = (5.0/9.0) * (f - 32);
System.out.print(c + " Celsius më nxehtë se "
    + f + " Fahrenheit? ");
System.out.println(c > fConverted);
```

Operatorët e krahasimit (të cilët kthejnë rezultat të tipit boolean): <, >, <=, >=, ==, !=.

```
boolean b = ((3 * 2) == (5 + 1));
```

Përparësia e operatorëve (operator precedence)

++ postfix, -- postfix
- unar, ++ prefix, -- prefix
*, /, %
+, -
<, <=, >, >=
==, !=
operatorët <i>OP</i> =

Tabela 1. Përparësia e operatorëve themelorë

Shembull:

```
b = 3 * 2 == 5 + 1;
```

Stringjet, karakterët dhe operacionet me ta

Emri në Java i tipit të të dhënave për stringje është String.

```
String name = "Filan Fisteku";  
System.out.println("Unë quhem " + name);  
System.out.println("dhe jam një Pentium " + (5 - 1));
```

Stringjet janë objekte në Java dhe operacionet me stringje janë metoda; p.sh.,

- ⑥ Krahasoni në barazim i stringjeve s1 dhe s2: `s1.equals(s2)`; p.sh.,

```
String s = "tungjatjeta";  
System.out.println(s.equals("tungjat" + "jeta"));  
System.out.println(s.equals(s));
```

- ⑥ Gjatësia e stringut s: `s.length()`; p.sh.,

```
String s = "tungjatjeta";  
int l = s.length(); // l == 11
```

Stringjet, karakterët dhe operacionet me ta – Vazhdim

Simbolet individuale përbrenda një stringu quhen *karakterë*.

Tipi i karakterëve në Java është `char`.

```
char c = 'a';
```

Literali karakter paraqitet duke u mbyllur në thonjëza të njëfishta, p.sh., `'J'`, `'a'`, `'2'`, `'&'`.

Java zbaton *Unicode* formatin për karakterë; p.sh.,

```
int i = 'a' + 1;    // 98
char c = (char)('a' + 1);    // 'b'
```

Stringjet, karakterët dhe operacionet me ta – Vazhdim

Karakteri	Semantika
\b	backspace
\t	tab
\n	rresht i ri (newline)
\r	return
\"	thonjëza e dyfishtë
\'	thonjëza e njëfishtë
\\	backslash

Tabela 2. Kodimi i karakterëve kontrollues

Stringjet, karakterët dhe operacionet me ta – Vazhdim

Metoda	Semantika	Tipi i rezultatit
<code>s1.equals(s2)</code>	Krahasimi në barazim i <code>s1</code> , <code>s2</code>	boolean
<code>s.length()</code>	Gjatësia e <code>s</code>	int
<code>s.charAt(i)</code>	Karakter i në pozitën <code>i</code> në <code>s</code>	char
<code>s.substring(i1,i2)</code>	Nënstringu i <code>s</code> nga pozita <code>i1</code> deri në <code>i2 - 1</code>	String
<code>s.toUpperCase()</code>	<code>s</code> në shkronja të mëdha	String
<code>s.toLowerCase()</code>	<code>s</code> në shkronja të vogla	String
<code>s.trim()</code>	<code>s</code> pa hapësirat përreth	String
<code>s1.indexOf(s2,i)</code>	Indeksin (pozitën) e <code>s2</code> në <code>s1</code> nga pozita <code>i</code>	int
<code>s1.compareTo(s2)</code>	Krahasimi (leksikografik) i <code>s1</code> , <code>s2</code>	int

Tabela 3. Disa metoda të klasës String

Stringjet, karakterët dhe operacionet me ta – Vazhdim

Për të manipuluar me karakterë individuale përbrenda një stringu shfrytëzohet metoda `charAt`; p.sh.,

```
char c = "abc".charAt(0); // 'a'
String s = "abc" + "de"; // "abcde"
c = s.charAt(2+1); // 'd'
boolean result = s.charAt(0) == 'a'; // true
c = "abc".charAt(3); // gabim
```

Kontrolli i tipit të të dhënave

Java kompilatori zbaton tipet e të dhënave që të:

- ⑥ sigurohet kompatibiliteti i operandëve ndaj operatorëve;
- ⑥ parashikohet tipi i rezultatit të shprehjes;
- ⑥ krijohen qeliza variablash të formatit përkatës;
- ⑥ imponohet kompatibiliteti i shprehjeve ndaj variablave në inicializime dhe ndarje vlerash.

Kontrolli i tipit të të dhënave – Vazhdim

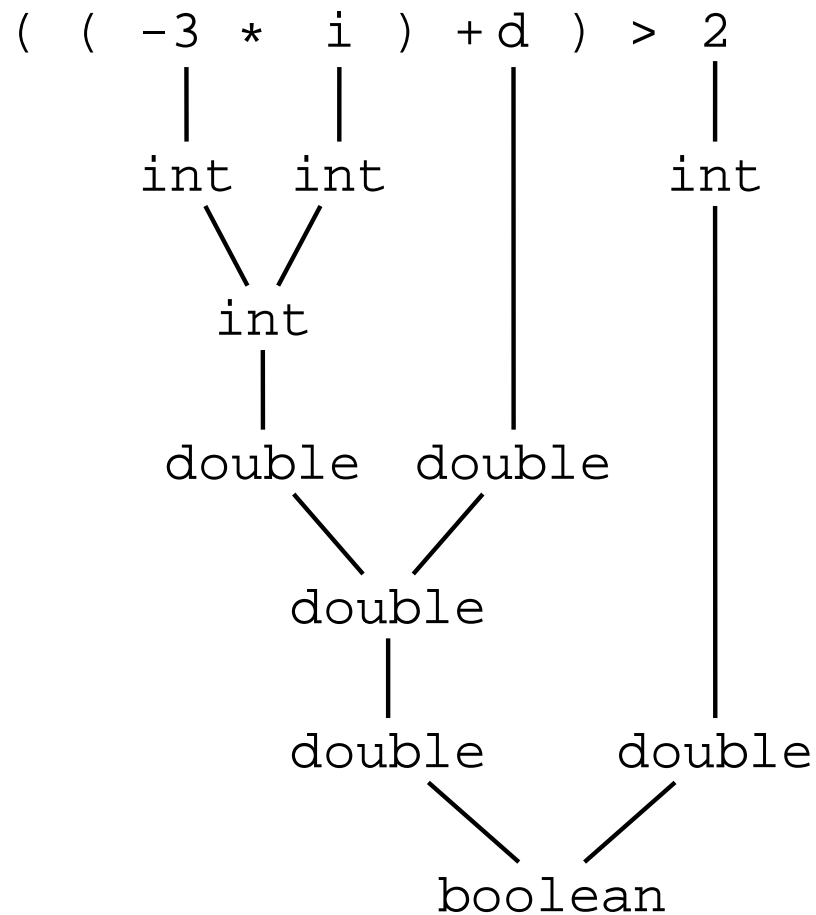


Figura 1. Druri i tipit të së dhënës

Kontrolli i tipit të të dhënave – Vazhdim

Ne Java ekzistojnë dy forma tipesh të të dhënave:

- ⑥ *tipet primitive*, si `int`, `double` dhe `boolean`
- ⑥ *tipet e referencës*, si `GregorianCalendar`

```
double d = 2.5;
```

```
GregorianCalendar c = new GregorianCalendar();
```

Tipi `String` është tip i referencës në Java, megjithëse përdoret sikurse tipet primitive të të dhënave.

```
String s = "abc";
```

Hyrja (input) përmes argumentëve programorë

```
/** Afishon gjatësinë e argumentit programor */
public class LengthOfName
{ public static void main(String[] args)
  { String name = args[0];
    int length = name.length();
    System.out.println("Emri " + name + " ka gjatësinë " + length);
  }
}
```

Konvertimi ndërmjet stringjeve dhe numrave dhe formatizimi

Konvertimi i stringjeve në numra të plotë:

```
String s = "20";
```

```
int i = new Integer(s).intValue();
```

class Integer konverton stringje numërorësh në numra të plotë; posedon metodën intValue.

Konvertimi ndërmjet stringjeve dhe numrave dhe formatizimi – Vazhdim

Konvertimi i stringjeve në numra thyesorë:

```
String approxPi = "3.14159";  
double d = new Double(approxPi).doubleValue();  
class Double konverton stringje numrash në numra  
thyesorë; posedon metodën doubleValue.
```

Konvertimi ndërmjet stringjeve dhe numrave dhe formatizimi – Vazhdim

Konvertimi i numrave në stringje:

```
double d = 3.14159;  
String s = d + "";
```


Konvertimi ndërmjet stringjeve dhe numrave dhe formatizimi – Vazhdim

Formatizimi i numrave:

```
import java.text.*;
public class Test
{ public static void main(String[] args)
  { DecimalFormat formatter = new DecimalFormat("0.00");
    double d = 100.0 / 3.0;
    String s = formatter.format(d);
    System.out.println(s + " Euro");
  }
}
```

class DecimalFormat gjendet në pakon java.text. Posedon metodën format(d) e cila kthen vlerën e formatizuar në decimale të numrit d, mbështetur në shabllonin e zbatuar në new DecimalFormat(*SHABLLON*).

Konvertimi i temperaturës me input

```
import java.text.*;
/** Konverton vlerën e temperaturës. */
public class FahrenheitToCelsius
{ public static void main(String[] args)
  { int f = new Integer(args[0]).intValue();
    double c = (5.0/9.0) * (f - 32);
    System.out.println("Për gradë Fahrenheit " + f + ",");
    DecimalFormat formatter = new DecimalFormat("0.0");
    System.out.println("gradë Celsius " + formatter.format(c));
  }
}
```

Konvertimi i temperaturës me input – Vazhdim

FahrenheitToCelsius

```
main(args[0] == "70" )  
{  
    >int f = new Integer(args0).intValue();  
    double c = (5.0/9.0) * (f - 32);  
    System.out.println("Për gradë Fahrenheit " + f + ",");  
    DecimalFormat formatter = new DecimalFormat("0.0");  
    System.out.println("gradë Celsius " + formatter.format(c));  
}
```

Konvertimi i temperaturës me input – Vazhdim

FahrenheitToCelsius

```
main(args[0] == "70" )  
{ int f == ?  
  >f = new Integer(args[0]).intValue();  
  double c = (5.0/9.0) * (f - 32);  
  ...  
}
```

Konvertimi i temperaturës me input – Vazhdim

FahrenheitToCelsius

```
main(args[0] == "70" )  
{ int f == ?  
  >f = new Integer("70").intValue();  
  double c = (5.0/9.0) * (f - 32);  
  ...  
}
```

Konvertimi i temperaturës me input – Vazhdim

FahrenheitToCelsius

```
main(args[0] == "70" )  
{ int f == ?  
  >f = a1.intValue();  
  double c = (5.0/9.0) * (f - 32);  
  ...  
}
```

a1 : Integer

```
përmban 70  
intValue()  
{ metodë e cila kthen 70 }
```

Konvertimi i temperaturës me input – Vazhdim

FahrenheitToCelsius

```
main(args[0] == "70" )  
{ int f == ?  
  >f = PRIT;  
  double c = (5.0/9.0) * (f - 32);  
  ...  
}
```

a1 : Integer

```
përmban 70  
intValue()  
{ >metodë e cila kthen 70 }
```

Konvertimi i temperaturës me input – Vazhdim

FahrenheitToCelsius

```
main(args[0] == "70" )  
{ int f == ?  
  >f = 70;  
  double c = (5.0/9.0) * (f - 32);  
  ...  
}
```


Konvertimi i temperaturës me input – Vazhdim

FahrenheitToCelsius

```
main(args[0] == "70" )  
{ int f == 70  
  >double c = (5.0/9.0) * (f - 32);  
  System.out.println("Për gradë Fahrenheit " + f + ",");  
  DecimalFormat formatter = new DecimalFormat("0.0");  
  System.out.println("gradë Celsius " + formatter.format(c));  
}
```

Diagnostifikimi i gabimeve në shprehje

Gabimet mund të kenë këto dy forma:

- ⑥ *Compile time errors*: gabimet sintaksore të cilat paraqiten gjatë kompilimit.
- ⑥ *Run time errors (Exceptions)*: gabimet të cilat paraqiten gjatë ekzekutimit, kur llogaritet rezultat jokorrekt dhe ekzekutimi ndalohe.

Gabimet sintaksore (compile time errors)

```
System.out.println( (1+2(*3 ) );  
string s = 3;  
System.out.println(3+true);
```

```
int i;  
System.out.println(2 * i);
```

```
int a = 1;  
double a = 2.5;  
System.out.println(a);
```

Gabimet sintaksore (compile time errors) – Vazhdim

```
System.out.println( (1+2)*3 );  
String s = "3";  
System.out.println(3+1);
```

```
int i = 0;  
System.out.println(2 * i);
```

```
int a = 1;  
double b = 2.5;  
System.out.println(a);
```

Përjashtimet (exceptions)

```
int i = 0;  
System.out.println(1 / i);  
  
i = new Integer(args[0]).intValue();  
System.out.println(1 / i);
```

Gabimet semantike

```
int x = 3;  
int y = 7;  
System.out.println(x = y);
```

Gabimet semantike – Vazhdim

```
int x = 3;  
int y = 7;  
System.out.println(x == y);
```

Fjalët kyçe dhe identifikatorët

Identifikator: emër variable, klase ose metode; varg i karakterëve, ku karakteri i parë është jonumëror.

Fjalët kyçe kanë domethënie të posaçme në Java dhe nuk mund të përdoren si identifikatorë.

Për stil konsistent, klasat emërtohen me identifikatorë që fillojnë me shkronjë të madhe, kurse variablat dhe metodat me shkronjë të vogël.