Leksion nr.2 Programin në Java|



Leksion 1

Struktura e programeve, tipet e të dhënave në Java dhe operatorët

# Struktura dhe fjalët kyçe të programeve në Java

* 1. **Struktura e programeve në Java**

Një program në Java ka këtë sintaksë:

public class EmriIKlasës{

public static void main(String[] args){

<statements> // një ose më shumë statement-e

}

}

Shembull: Programi i mëposhtëm do afishojë mesazhin Hello World! në console.

1. public class HelloWorld{
2. public static void main(String[] args){
3. // Ky program printon mesazhin Hello World!
4. System.out.println("Hello World!"); 5 }

6 }

Numrat e rreshtave janë vendosur për qëllime reference por nuk jane pjesë e programit.

Rreshti 1 nis një klasë të re ose e thënë ndryshe përcakton një klasë. Klasat janë një koncept thelbësor në Java. Në Java çdo program përbëhet nga një ose më shumë klasa. Çdo klasë ka një emër. Emrat e klasave fillojnë me shkronjë të madhe dhe nëse përbëhen nga dy ose më shumë fjalë, këto fjalë nuk ndahen me hapësira ndërmjet tyre. Në këtë rast emri i klasës është HelloWorld që përbëhet nga fjalët Hello dhe World.

Rreshti 2 përcakton metodën main. Jo çdo klasë në Java është një program. Në mënyrë që një klasë të përcaktojë një program, domethënë që klasa të ekzekutohet, duhet që ajo të përfshijë një metodë main. Metoda main siguron kontrollin e rrjedhës së programit. Interpretuesi i Javës ekzekuton programin duke thirrur metodën main. Fjala “public” tregon që kjo metodë (routine) mund të thirret nga jashtë programit. Interpretuesi i Javës që thërret metodën main është jashtë programit. Një klasë në Java përbëhet nga një ose më shumë metoda. Emrat e metodave në Java fillojnë me shkronjë të vogël si në këtë rast ku emri i metodës është main. Nëse në emrin e metodës përfshihen disa fjalë atëherë fjala e parë nis me shkronjë të vogël kurse fjalët e tjera nisin me shkronjë të madhe, p.sh. public void llogaritSipëfaqe(). Fjala e rezervuar static tregon që metoda main nuk operon mbi ndonjë objekt. Kjo metodë duhet të jetë gjithmonë statike sepse ajo fillon të ekzekutohet përpara se programi të mund të krijojë objekte.

Parametri String[] args është një pjesë e kërkuar e metodës main.

Rreshti 3 është një koment që jep informacion rreth programit. Komentet ndihmojnë programuesit të përshkruajnë dhe kuptojnë programin. Ato nuk janë statement-e programimi dhe për këtë arsye injorohen nga kompilatori. Në programet Java mund të përdoren tre lloj komentesh:

* + Komente rreshti: një koment rreshti paraprihet nga dy slash-e (//).
  + Komente paragrafi: Një koment paragrafi përfshihet ndërmjet /\* dhe \*/ në një ose disa rreshta.
  + Komente javadoc: fillojnë me /\*\* dhe mbarojnë me \*/. Ato përdoren për të dokumentuar klasa, të dhëna, dhe metoda. Ato mund të futen në një skedar HTML duke përdorur komandën javadoc të JDK.

Rreshti 4 përfshin statement-in System.out.println të metodës main. Ky statement printon mesazhin “Hello Wold!” në console. Çdo statement në Java përfundon me një pikëpresje (;). System është një klasë në paketën java.lang, out është një variabël statik brenda klasës System dhe println është një metodë që ndiqet nga kllapat () që përcaktojnë parametrin e metodës. System.out është një objekt. Sekuenca e karaktereve të përfshira brenda thonjëzave (“ “) quhet një *string*. Përdorimi i thonjëzave bëhet që kompilatori ta kuptojë që ka të bëjë me tekst dhe mos përpiqet ta interpretojë stringun si një instruksion programi (p.sh me System.out.println “main”; kompilatori të kuptojë që është sekuenca e karaktereve m a i n dhe jo metoda main). Për të printuar vlera numerike nuk përdoren thonjëzat.

Për shembull, statement-i:

System.out.println(3 + 4);

printon numrin 7. Metoda println printon një string ose një numër dhe fillon një rresht të ri.

Në program vihet re përdorimi i kllapave. Një çift kllapash në një program formon një bllok. Në Java çdo bllok fillon me një kllapë hapëse ({) dhe mbaron me një kllapë mbyllëse (}). Kujdes: çdo kllapë hapëse duhet të ketë një kllapë mbyllëse që lidhet me të. Sa herë që shtypni një kllapë hapëse menjëherë shtypni një kllapë mbyllëse që të parandaloni krijimin e gabimeve nga mungesa e kllapave. Çdo klasë ka një *bllok klase* që grupon të dhënat dhe metodat e klasës. Çdo metodë ka një *bllok metode* që grupon statement-et në metodë.

Java është *case sensitive* që do të thotë që main dhe Main kanë kuptime të ndryshme në Java.

Emri i skedarit të programit duhet të jetë i njëjtë me emrin e klasës, i shoqëruar nga prapashtesa .java. Për shembull: Nëse emri i klasës është “HelloWorld”, atëherë skedari ruhet si “HelloWorld.java”.

### Identifikuesit, fjalët e rezervuara dhe modifikuesit

*Identifikuesit*. Gjithë komponentet në Java kërkojnë emra. Emrat e përdorur për klasat, variablat dhe metodat quhen identifikues. Gjithë identifikuesit duhet të fillojnë me një shkronjë (A deri Z ose a deri z), me ($) ose (\_). Pas karakterit të parë identifikuesit mund të kenë çdo kombinim karakteresh. Identifikuesit janë case sensitive. Shembuj identifikuesish: age, $salary, \_value, 1\_value.

*Fjalët e rezervuara* ose *fjalëtçel*ës*.* Janë fjalë të rezervuara nga Java për përdorime specifike. Ato kanë një kuptim specifik për kompilatorin dhe nuk mund të përdoren si identifikues. Për shembull, kur kompilatori ndesh fjalën class, ai kupton që fjala pas kësaj fjale është emri i klasës. Fjalët e rezervuara në kodin që pamë më sipër janë class, public, static, dhe void. Në tabelën e mëposhtme paraqiten fjalët e rezervuara në Java.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| abstract | assert | boolean | break | byte | case |
| catch | char | class | const | continue | default |

2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Leksion nr.2 Programin në Java | | | | | |
| do | double | else | enum | extends | false |
| final | finally | float | for | goto | if |
| implements | import | instanceof | int | interface | long |
| native | new | null | package | private | protected |
| public | return | short | static | strictfp | super |
| switch | synchronized | this | throw | throws | transient |
| true | try | void | volatile | while |  |

*Modifikuesit*. Modifikuesit janë fjalëçelës që përdoren për të ndryshuar kuptimin e një përcaktimi. Ekzistojnë dy kategori modifikuesish në Java:

* + - Modifikuesit e aksesit: përcaktojnë nivelin e aksesimit për variablat, klasat, metodat,dhe konstruktorët. Katër nivelet e aksesit janë:

1. Të aksesueshme nga paketa (default). Nuk nevojiten modifikues.
2. Të aksesueshme vetëm nga klasa (private).
3. Të aksesueshme nga bota, pra aksesohet nga çdo klasë tjetër (public).
4. Të aksesueshme nga paketa dhe nga nënklasat jashtë paketës sipas hierarkisë (protected).
   * + Modifikuesit e jo-aksesit: përdoren për arritjen e shumë funksionaliteteve të tjera.
5. Modifikuesi *static* përdoret për krijimin e variablave dhe metodave në Java.
6. Modifikuesi *final* përdoret për finalizimin e implementimeve të klasave, metodave dhe variablave. Nuk lejon që përmbajtjet e tyre të modifikohen.
7. Modifikuesi *abstract* përdoret për krijimin e klasave dhe metodave abstrakte.
8. Modifikuesit *synchronized* dhe *volatile*, përdoren për thread-et.

# Variablat dhe tipet e të dhënave në Java

### Variablat

Variblat janë lokacione të rezervuara në kujtesë për të ruajtur vlera. Kjo do të thotë që kur ju krijoni një variabël ju rezervoni hapësirë në kujtesë. Kur deklarohet një variabël, ai duhet të ketë të përcaktuar përkrahë tij tipin e të dhënës. Tipi i të dhënës përcakton vlerat që variabli mund të marrë dhe cilat operacione mund te kryhen me të. Duke u bazuar në tipin e të dhënave të variablave, sistemi i operimit alokon kujtesë dhe vendos se çfarë do të ruhet në kujtesën e rezervuar. Kështu, duke caktuar tipe të ndryshme të dhënash për variablat ju mund të ruani numra të plotë, numra dhjetorë ose karaktere në këto variabla.

**Kujdes!** Në gjuhën e programimit Java, ju duhet fillimisht ta deklaroni një varibël dhe më pas ta përdorni atë.

### Deklarimi i një variabli në Java:

Tipi\_i\_të\_dhënës emër\_variabli [ = vlerë][, emër\_variabli [=vlerë] ...]

; ku kllapat katrore [] tregojnë se seksioni brenda tyre është opsional.

int x; // Deklarohet x si një variabël int, numër i plotë;

3

double rrezja; // Deklarohet rrezja si një variabël double, numër dhjetor me dy shifra pas presjes dhjetore; char a; // Deklarohet a si një variabël i tipit karakter;

Emrat e variablave në Java fillojnë me shkronjë të vogël. Nëse në emrin e variablit përfshihen disa fjalë atëherë fjala e parë nis me shkronjë te vogël kurse fjalët e tjera nisin me shkronjë të madhe për shembull: double fitimiVjetor;

### Inicializimi i një variabli:

Për të inicializuar një variabël, ju duhet ti jepni atij një vlerë. Pra dhënia e vlerës fillestare quhet inicializim i variablit.

x = 1; // i jepet vlera 1 x-it;

rrezja = 1.0; // i jepet vlera 1.0 rrezes; a = 'A'; // i jepet vlera 'A' a-së;

### Deklarim dhe inicializim në një hap

int x = 1; double d = 1.4; **Konstantet**

Forma e përgjithëshme e deklarimit të konstanteve:

final tipi\_i\_të\_dhënës EMRI\_I\_KONSTANTES = Vlera; Shembuj:

final double PI = 3.14159; final int SIZE = 3;

### Tipet e të dhënave në Java

Ekzistojnë dy tipe të dhënash në Java:

1. Tipe të dhënash primitive
2. Tipe të dhënash objekt/referenciale

### Tipet e të dhënave primitive

Java suporton tetë tipe të dhënash primitive.

### byte

* + - * Është 8 bit
      * Vlera minimale është -128 (-2^7)
      * Vlera maksimale është 127 (inkluzive)(2^7 -1)
      * Vlera default është 0
      * Shembull: byte a = 100 , byte b = -50
      * Përdoret për të kursyer hapësirë në array të mëdha, kryesisht në vend të integer-ave, meqenëse një byte është katër herë më e vogël se një byte.

4

### short

* + - * Është 16-bit.
      * Vlera minimale është -32,768 (-2^15)
      * Vlera maksimale është 32,767(inkluzive) (2^15 -1)
      * Vlera default është 0.
      * Shembull : short s= 10000 , short r = -20000
      * Përdoret për të kursyer kujtesë si tipi byte i të dhënave. Një short është dy herë më e vogël se një int.

### int

* Është 32-bit.
* Vlera minimale është - 2,147,483,648 (-2^31).
* Vlera maksimale është 2,147,483,647 (inclusive) (2^31 -1).
* Vlera default është 0.
* Shembull: int a = 100000, int b = -200000.
* Përdoret si një tip default i të dhënave për vlerat integrale përderisa nuk përbën ndonjë shqetësim në lidhje me kujtesën.

### long

* + Është 64-bit.
  + Vlera minimale është -9,223,372,036,854,775,808 (-2^63).
  + Vlera maksimale është 9,223,372,036,854,775,807 (inkluzive) (2^63 -1).
  + Vlera default është 0L.
  + Shembull: int a = 100000L, int b = -200000L.
  + Ky tip përdoret kur nevojitet një rang më i gjerë se int.

### float

* + Është me preçizion të rangut të parë, që do të thotë për numrat me një shifër dhjetore pas presjes.
  + Është 32-bit.
  + Vlera default është 0.0f.
  + Shembull: float f1 = 234.5f

### double

* + Është me preçizion të dyfishtë, që do të thotë për numrat me dy shifra pas presjes dhjetore
  + Është 64-bit.
  + Vlera default është 0.0d.
  + Shembull: double d1 = 123.4
  + Përdoret si si një tip default i të dhënave për vlera dhjetore.

### boolean

* + Përmban vetëm 1-bit informacion.
  + Ka vetëm 2 vlera të mundshme : true dhe false.
  + Vlera default është false.
  + Shembull: boolean one = true

5

### char

* + Është 16-bit.
  + Vlera minimale është '\u0000' (ose 0).
  + Vlera maksimale është '\uffff' (ose 65,535 inkluzive).
  + Përdoret për të ruajtur çdo lloj karakteri.
  + Shembull. char shrkonjaA = 'A'

Ndryshe, tipet primitive të të dhënave mund të ndahen si më poshtë:

1. *Numra të plotë*

byte (1 byte)



short (2 bytes)

int (4 bytes)

long (8 bytes)

1. *Numra me presje dhjetore*

float (4 bytes)



double (8 bytes)

1. *Tekst*

char (2 bytes)

1. *Llogjike*

 boolean (1 byte) (true/false)

### Tipet e të dhënave referenciale

E vetmja mënyrë për të aksesuar një objekt janë variablat referencialë. Një variabël reference deklarohet për të qenë e një tipi specifik dhe tipi i tij nuk mund të ndryshohet më pas. P.sh Employee, Animal, etj. Variablat referencial mund të deklarohen si variabla statikë, variabla të instancës, parametra metode ose variabla lokalë. Vlera dafult e çdo variabli referencial është null. Një variabël referencial mund të përdoret për t’iu referuar çdo objekti.

Shembull: Animal animal = new Animal("giraffe");

# Operatorët

Java ofron një grup operatorësh për manipulimin e të dhënave. Pjesa më e madhe e tyre janë të shpjeguar më poshtë:

### Operatorët aritmetikë

Përdoren në shprehje matematikore në të njëjtën mënyrë siç përdoren në algjebër. Më poshtë listohen operatorët aritmetikë. Supozojmë se variabli integer a mban vlerën 10 dhe variabli b mban vlerën 20.

|  |  |  |  |
| --- | --- | --- | --- |
| **Operatori** | **Veprimi** | **Shembull** | **Rezultati** |
| **+** | Mbledhje | a + b | 30 |
| **-** | Zbritje | a - b | -10 |
| **\*** | Shumëzim | a \* b | 200 |
| **/** | Pjestim | b / a | 2 |

6

|  |  |  |  |
| --- | --- | --- | --- |
| Leksion nr.2 |  |  | Programin në Java |
| **%** | Mbetje pjestimi | b % a | 0 |
| **++** | Inkrementim | b++ | 21 |
| **--** | Dekrementim | b-- | 19 |

### Operatorët relacionalë

Kryejnë kontrolle në lidhje me vlerat e variablave. Këta operatorë japin si rezultat një vlerë true ose false. Më poshtë janë listuar operatorët relacionalë me shpjegimet përkatëse:

== e barabartë me (kushtëzor)

!= jo e barabartë me (kushtëzor)

> më e madhe se

>= më e madhe e barabartë me

< më e vogël se

<= më e vogël e barabartë me

### Operatorët logjikë

Më poshtë janë të listuar operatorët logjikë: && Operatori logjik AND

|| Operatori logjik OR

! Operatori logjik NOT

### Operatorët e caktimit

Operatori Shembull

= c = a + b

## += c += a

-= c -= a

## \*= c \*= a

/= c /= a

%= c \*= a

Shpjegimi

do të transferojë vlerën e a + b tek c

## është ekuivalente me c = c + a është ekuivalente me c = c - a është ekuivalente me c = c \* a është ekuivalente me c = c / a është ekuivalente me c = c % a

### Operatori i kushtëzuar (?:)

Njihet si operatori ternar dhe është shkurtim i shprehjes if-then-else. Qëllimi i këtij operatori është të përcaktojë se cila vlerë do ti caktohet variablit. Operatori shkruhet si:

shprehja-booleane ? shprehja1 : shprehja2 ;

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

7

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

mund të shkuhet ndryshe: if ( x > 0 ) {

y = 1;

}

else {

y = -1;

}

### Operatori instanceof

Përdoret për të testuar nëse një objekt është një instancë e një tipi të caktuar ( klasës, nënklasës, ose ndërfaqes). Njihet ndryshe si operator krahasimi sepse krahason instancën me tipin. Ai kthen true ose false. Nëse aplikohet me ndonjë variabël që ka vlerën zero, ai kthen false.

Shembull:

public class Test {

public static void main(String[] args) {

Test t= new Test();

System.out.println(t instanceof Test); //kthen true

}

}

# Shembuj

Shembull1

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

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 vlerën e a-së

a = b; //i jap a-së vlerën e b-së

b = temp; //i jap b-së vlerën e temp

System.out.println("a = " + a + "; b = " + b + "; temp = " + temp);

}

}

Ky program afishon: a = 5; b = 7; temp = 0

8

a = 7; b = 5; temp = 5 Shembull 2

Në këtë shembull jepet një program që gjen maksimumin e dy numrave.

public class Max {

public static void main(String[] args) { int num1 = 2, num2 = 6;

int max = (num1 > num2) ? num1 : num2;

//nëse num1> num2 atëherë max = num1 në të kundërt max = num2 System.out.println( max );//printon vleren e variablit max }

}

Ky program afishon:

6

Shembull 3

Ky shembull paraqet një program që demostron përdorimin e karaktereve special dhe funksionin që kryen secili prej tyre. Backslash ( \ ) vendoset para karaktereve special në mënyrë që të bëjë të mundur printimin e tyre në console.

public class KarakteretSpeciale {

public static void main(String[] args) {

System.out.println("Ky është një tekst"); // afishon tekstin

System.out.println("Ky është një tek\tst"); // \t t qëndron për tab

System.out.println("Ky është një tek\nst"); // \n dmth kalo në rresht të ri

System.out.println("Ky është një \'tekst\' "); // \'

System.out.println("Ky është një \"tekst\" "); // \"

System.out.println("Ky është një tek\\st"); // \\

}

}

Ky program afishon në console: Ky është një tekst

Ky është një tek st Ky është një tek

st

Ky është një 'tekst' Ky është një "tekst" 9

Ky është një tek\st Shembull 4

Ky program llogarit sipërfaqen e rrethit

public class SipRrethi {

public static void main(String[] args) {

final double PI=3.14; // deklarimi i konstantes PI

double rrezja=5; // deklarimi i variablit rrezja dhe vlerëdhënia double sipërfaqja; //deklarimi i variablit sipërfaqe

sipërfaqja=PI\*rrezja\*rrezja; //llogaritja e sipërfaqes System.out.println("Sipërfaqja është: " + sipërfaqja); //afishimi i rezultatit

}

}

Ky program printon:

Sipërfaqja është: 78.5 Shembull 5

Ky program tregon si mund të manipulohet një varabël i tipit String për të lidhur tekstin.

public class String1 {

public static void main(String[] args) { String mesazh = "Ky është "; mesazh += "shembulli i fundit.";

System.out.println( mesazh );

}

}

Ky program afishon:

Ky është shembulli i fundit.

\*output-i i këtij programi është i njëjtë me atë të programit të mëposhtëm:

public class String1 {

public static void main(String[] args) {

String mesazh = "Ky është " + "shembulli i fundit."; System.out.println( mesazh );

}

}

10