

Java - Variable Types

O variabilă ne oferă un spațiu de stocare numit pe care programele noastre le pot manipula. Fiecare variabilă din Java are un tip specific, care determină dimensiunea și aspectul memoriei variabilei; gama de valori care pot fi stocate în memoria respectivă; și setul de operații care pot fi aplicate variabilei.

Trebuie să declarați toate variabilele înainte de a le utiliza. Urmează forma de bază a unei declarații de variabile:

```
data type variable [ = value][, variable [ = value] ...] ;
```

Aici *data type* este unul dintre tipurile de date Java și *variable* este numele variabilei. Pentru a declara mai multe variabile de tipul specificat, puteți utiliza o listă separată de virgule.

Următoarele sunt exemple valide de declarație și inițializare a variabilelor în Java:

```
int a, b, c;           // Declares three ints, a, b, and c.
int a = 10, b = 10;    // Example of initialization
byte B = 22;           // initializes a byte type variable B.
double pi = 3.14159;   // declares and assigns a value of PI.
char a = 'a';          // the char variable a is initialized with value 'a'
```

Acest proiect va explica diferite tipuri de variabile disponibile în limbajul Java. Există trei tipuri de variabile în Java :

- Variabile locale (local variables)
- Variabile de instanță (instance variables)
- Variabilele clasă / statice (class/static variables)

Local Variables

- Variabilele locale sunt declarate în metode, constructori sau blocuri.
- Variabilele locale sunt create atunci când metoda, constructorul sau blocul este introdus și variabila va fi distrusă odată ce iese din metodă, constructor sau bloc.
- Modificatorii de acces nu pot fi folosiți pentru variabile locale.
- Variabilele locale sunt vizibile numai în cadrul metodei, constructorului sau blocului declarat.
- Variabilele locale sunt implementate la nivel de stivă la nivel intern.
- Nu există o valoare implicită pentru variabilele locale, deci variabilele locale ar trebui declarate și o valoare inițială trebuie atribuită înainte de prima utilizare.

Exemplu:

Aici, vârsta este o variabilă locală. Aceasta este definită în cadrul metodei `PupAge()`, iar sfera acesteia este limitată numai la această metodă.

```
public class Test {  
    public void pupAge() {  
        int age = 0;  
        age = age + 7;  
        System.out.println("Puppy age is : " + age);  
    }  
  
    public static void main(String args[]) {  
        Test test = new Test();  
        test.pupAge();  
    }  
}
```

Aceasta va produce următoarea eroare la compilarea ei:

Output:

```
Test.java:4:variable number might not have been initialized  
    age = age + 7;  
          ^  
1 error
```

Instance Variables

- Variabilele de instanță sunt declarate într-o clasă, dar în afara unei metode, constructor sau orice bloc.
- Când un spațiu este alocat pentru un obiect din grămadă, se creează un slot pentru fiecare valoare variabilă de instanță.
- Variabilele de instanță sunt create atunci când un obiect este creat cu utilizarea cuvântului cheie „nou” și distrus atunci când obiectul este distrus.
- Variabilele de instanță conțin valori la care trebuie să fie făcute referințe de mai multe metode, constructor sau bloc, sau părți esențiale ale stării unui obiect care trebuie să fie prezente în întreaga clasă.
- Variabilele de instanță pot fi declarate la nivel de clasă înainte sau după utilizare.
- Modificatorii de acces pot fi furnizați pentru variabile de exemplu.
- Variabilele de instanță sunt vizibile pentru toate metodele, constructorii și blocurile din clasă. În mod normal, se recomandă ca aceste variabile să fie private (nivel de acces). Cu toate acestea, vizibilitatea pentru subclase poate fi dată pentru aceste variabile cu utilizarea modificatoarelor de acces.
- Variabilele de instanță au valori implicite. Pentru numere, valoarea implicită este 0, pentru Booleans este falsă, iar pentru referințele la obiect este nulă. Valorile pot fi alocate în timpul declarației sau în cadrul constructorului.
- Variabilele de instanță pot fi accesate direct apelând numele variabilei din interiorul clasei. Cu toate acestea, în cadrul metodelor statice (atunci când variabilelor de instanță li se oferă accesibilitate), acestea trebuie apelate folosind numele complet calificat. `ObjectReference.VariableName`.

Exemplu:

```
import java.io.*;
public class Employee {

    // this instance variable is visible for any child class.
    public String name;

    // salary variable is visible in Employee class only.
    private double salary;

    // The name variable is assigned in the constructor.
    public Employee (String empName) {
        name = empName;
    }

    // The salary variable is assigned a value.
    public void setSalary(double empSal) {
        salary = empSal;
    }

    // This method prints the employee details.
    public void printEmp() {
        System.out.println("name : " + name );
        System.out.println("salary :" + salary);
    }

    public static void main(String args[]) {
        Employee empOne = new Employee("Ransika");
        empOne.setSalary(1000);
        empOne.printEmp();
    }
}
```

Acest lucru va produce următorul rezultat:

Output:

```
name : Ransika
salary :1000.0
```

Class / Static Variables

- Variabilele de clasă cunoscute și sub numele de variabile statice sunt declarate cu cuvântul cheie static într-o clasă, dar în afara unei metode, constructor sau bloc.
- Ar exista o singură copie a fiecărei variabile de clasă pe clasă, indiferent de câte obiecte sunt create din ea.
- Variabilele statice sunt rareori utilizate, în afară de a fi declarate constante. Constanțele sunt variabile care sunt declarate publice / private, finale și statice. Variabilele constante nu se schimbă niciodată de la valoarea lor inițială.
- Variabilele statice sunt stocate în memoria statică. Este rar să se utilizeze variabile statice, altele decât cele declarate finale și utilizate ca constante publice sau private.
- Variabilele statice sunt create când programul pornește și distruge atunci când programul se oprește.
- Vizibilitatea este similară cu variabilele de instanță. Cu toate acestea, majoritatea variabilelor statice sunt declarate publice, deoarece acestea trebuie să fie disponibile pentru utilizatorii clasei.
- Valorile implicite sunt aceleași cu variabilele de instanță. Pentru numere, valoarea implicită este 0; pentru booleani, este fals; iar pentru referințe la obiect, este nul. Valorile pot fi alocate în timpul declarației sau în cadrul constructorului. În plus, valorile pot fi atribuite în blocuri de inițializare statică speciale.
- Variabilele statice pot fi accesate apelând cu numele clasei `ClassName.VariableName`.
- Când se declară variabilele de clasă ca publice statice finale, atunci numele de variabile (constante) sunt toate majuscule. Dacă variabilele statice nu sunt publice și finale, sintaxa de denumire este aceeași cu variabilele de instanță și locale.

Exemplu:

```
import java.io.*;
public class Employee {

    // salary variable is a private static variable
    private static double salary;

    // DEPARTMENT is a constant
    public static final String DEPARTMENT = "Development ";

    public static void main(String args[]) {
        salary = 1000;
        System.out.println(DEPARTMENT + "average salary:" + salary);
    }
}
```

Live Demo

Acest lucru va produce următorul rezultat:

Output:

```
Development average salary:1000
```

Atentie! - Dacă se accesează variabilele dintr-o clasă din afară, constanta trebuie accesată ca Employee.DEPARTMENT