

Java Constructor and Class Initializers

By

Sunil Kumar(Master of Sc.)

Bangalore, India

1. Agenda

- Establishing Initial State
- Field Initializers
- Constructors
- Constructor Chaining and visibility
- Initialization blocks
- Initialization and Construction Order

Establishing Initial State

- In last slide we have studied that class is made up of STATE and EXECUTABLE code. When an object is created, it is expected to be in a useful state. Java provide default state which is useful. But often the default state established by Java is not enough. The object may need to set values or execute code.

Mechanism for establishing Initial State

- Java provide 3 mechanism for establishing initial state.
 - Field Initializers
 - Constructors
 - Initialization Blocks

Field Initial State

- In case of variable we explicitly assigned value. Initializing variable and initializing field is 2 different thing. A field's initial state is established as part of object construction. Fields receive a zero value by default. Different data type fields will have different values.

Byte Short Int Long	Float Double	Char	Boolean	Reference Type
0	0.0	'\u0000'	False	Null

Field Initial State Contd..

- Sometimes this default initial state is not acceptable as part of requirement.
- Java provide field Initializers which allows programmer to specify a field's initial value as part of declaration.
 - Can be a simple assignment
 - Can be an equation
 - Can reference other fields
 - Can be a method call

Field Initial State Contd..

- We can see through an example.

```
public class Earth{  
    long circumferenceInMiles = 24901;  
    //long circumferenceInKilometers = (long)24901 * 1.6d;  
    long circumferenceInKilometers = (long)(circumferenceInMiles * 1.6d);  
    long circumferenceInKilometers =Math.round(circumferenceInMiles* 1.6d);  
}
```

- So we have seen that these field Initializers are very powerful where we can assign using simple assignment, a method call or a field reference.

Constructor

- A constructor in Java is a block of code similar to a method that's called when an instance of an object is created. A constructor doesn't have a return type. The name of the constructor must be the same as the name of the class. We can say, it is a special type of method that is used to initialize the object. Remember that a constructor is not a method but a executable code used during object creation to set initial state. Constructor is invoked at the time of object creation. It constructs the values i.e. provides data for the object that is why it is known as constructor.

Constructor Contd..

- Thus a constructor is a executable code where we must have to name it same as CLASS.
 - Have no return type
 - Every class has at least one constructor
- From our last example flight class

```
public class Flight{  
private int passenger;  
private int seats;  
    public Flight(){  
        seats = 150;  
        passenger = 0;  
    }  
}
```

Constructor Contd..

- In the last example, declaration of passenger =0 is not required as by default field value is initialized to 0. Also seats can be declared in the above during declaration. So we don't really require a constructor here.
- What happened when we don't really need an explicit constructor.
- We will try to understand this using an example.

Constructor Contd..

```
public class Passenger{
    private int checkedBags;
    private int freeBags;
    //accessors and mutators elided for clarity
    private double perBagFee;
    public Passenger(){
    }
    public Passenger (int freeBags){
        this.freeBags = freeBags;
    }
}
```

```
Passenger bob = new Passenger();
bob.setCheckedBags(3);

Passenger jane = new Passenger(2);
jane.setCheckedBags(3);
```

Constructor Contd..

- In case of requirement like this, where we don't require an explicit constructor
 - Java provides a default constructor with no argument
- A class can have a multiple constructors
 - Each with a different parameter list
- Once explicit constructor is declared, we have to explicitly declared the definition of default constructor as well, else program will have an error.