Java Constructors

Topics : JAVA
Written on April 07, 2023

In Java, a constructor is a special method that is called when an object is created. Its main purpose is to initialize the instance variables of the object. Constructors have the same name as the class and do not have a return type.

There are two types of constructors in Java:

1. Default Constructor: If no constructor is defined in a class, Java automatically provides a default constructor that takes no arguments. It initializes all instance variables to their default values. For example, if a class has an instance variable of type int, the default constructor initializes it to 0.

2. Parameterized Constructor: A parameterized constructor is a constructor that takes one or more parameters. It is used to initialize the instance variables of an object with specific values. This allows you to create objects with different initial values.

Example of a constructor in Java:

```
public class Car {
    private String make;
    private String model;
    private int year;

    // Parameterized Constructor
    public Car(String make, String model, int year) {
        this.make = make;
        this.model = model;
        this.year = year;
    }

    // Getters and Setters
    public String getMake() {
        return make;
    }

    public void setMake(String make) {
        this.make = make;
    }
```

```
    public String getModel() {
        return model;
    }

    public void setModel(String model) {
        this.model = model;
    }

    public int getYear() {
        return year;
    }

    public void setYear(int year) {
        this.year = year;
    }
}
```

In this example, the `Car` class has a parameterized constructor that takes three parameters: `make`, `model`, and `year`. It uses the `this` keyword to refer to the instance variables of the object being created and sets their values based on the constructor parameters. The class also has getters and setters to access and modify the instance variables.

Here are some important things to know about constructors in Java:

1. Constructors are called automatically when an object is created using the new keyword.
2. Constructors can be overloaded, which means that a class can have multiple constructors with different parameter lists.
3. If a class doesn't define any constructors, a default constructor is created automatically by the compiler.
4. A constructor can call another constructor using the this() keyword. This is called a constructor chaining.
5. Constructors can have access modifiers (public, private, protected), but they cannot be abstract, final, static or synchronized.
6. Constructors can throw exceptions, just like regular methods.

Example :

```
import java.util.*;

public class Aryatechno {
public static void main(String[] args) {

Person p = new Person("john roy", 19);


String name = p.getName();
```

```
System.out.println("Name : "+name);

int age = p.getAge();

System.out.println("Age : "+age);


}



}


public class Person {
private String name;
private int age;

public Person(String name, int age) {
this.name = name;
this.age = age;
}

public String getName() {
return name;
}

public int getAge() {
return age;
}
}
```

Output :

```
Name : john roy
Age : 19
```