HowToDoInJava

Constructors in Java

Last Updated: January 25, 2022   By: Lokesh Gupta   Java Object Oriented Programming   Java Constructors, Java OOP

Java constructors are special methods (without return type) which allow you to fully initialize the object state before it can be used by other classes inside application. Constructors in java are invoked using `new` keyword.

Let's learn about constructors in more depth.

Table of Contents

What are Constructors in Java

Constructors are special method like (no exactly methods) constructs which helps programmer in writing object initialization code, before the object is available for use by other objects in the application.

Whenever application needs a new instance of any class, JVM allocates a memory area inside heap. Then JVM executes the invoked constructor (class can have multiple constructors) and initialize the object state. Inside constructor, you can access all object attributes and assign them to their default values or any desired values.

> Read More: Java Memory Model

Types of Constructors

1. **Default Constructor (no-arg constructor)**
   In case, programmer does not provide any constructor in class definition – JVM provides a default constructor to the class in runtime.

   Programmer also can override default constructor in class. Let's look at the syntax.

   ```
   public class Employee
   {
     public Employee() {

     }
   }
   ```

   In default constructor, name of the constructor MUST match the class name and it should not have any parameters.

2. **Parameterized Constructor via Constructor Overloading**
   As stated above, there can be multiple constructors inside a class. This is possible by having overloaded constructors. In **constructor overloading**, you can pass list of arguments as per requirements i.e. how many ways a class can be initialized.

   ```
   public class Employee {
     private String firstName;
     private String lastName;
   ```

```
    public Employee() { //constructor 1

    }

    public Employee(String firstName) { //constructor 2

    }

    public Employee(String firstName, String lastName) { //constructor 3

    }
}
```

In above class, we have defined 3 constructors to handle 3 situations – how the employee object might be required to create by the application i.e. without name, with first name only and with first and last name both.

```
Employee employee1 = new Employee();
Employee employee2 = new Employee("Lokesh");
Employee employee3 = new Employee("Lokesh", "Gupta");
```

Constructor Rules

There are few mandatory rules for creating the constructors in java.

1. Constructor name MUST be same as name of the class.
2. There cannot be any return type in constructor definition.
3. There cannot be any return statement in constructor.
4. Constructors can be overloaded by different arguments.
5. If you want to use `super()` i.e. super class constructor then it must be first statement inside constructor.

Constructor Chaining

In java, it's possible to call other constructors inside a constructor. It's just like method calling but without any reference variable (obviously as instance is fully initialized as of now).

Now we can call constructors of either same class or of parent class. Both uses different syntax.

Call same class constructor

To call other constructors from same class, use **this** keyword. For example,

```
public Employee() {

}

public Employee(String firstName) {
  this();    //calling default constructor
}

public Employee(String firstName, String lastName) {
  this(firstName);  //calling constructor with single argument of String type
}
```

Call super class constructor

To call constructors from super or parent class, use **super** keyword. The usage of super keyword is similar to `this` keyword – only difference is that `super` refers to super class and `this` refers to current instance.

```
public Employee() {
  //refer to Object class constructor
  //as it is parent class for every class
  super();
}
```

Private Constructors

Sometimes you want to protect the constructor from being called by other classes. Altogether you want that nobody should be able to create a new instance of the class.

Why anybody would want that? Well, it's necessary for singleton pattern. In singleton, an application wants to have one and only one instance of any class.

A common singleton class definition looks like this:

```
public class DemoSingleton implements Serializable
{
    private static final long serialVersionUID = 1L;

    private DemoSingleton() {
        // private constructor
    }

    private static class DemoSingletonHolder {
        public static final DemoSingleton INSTANCE = new DemoSingleton();
    }

    public static DemoSingleton getInstance() {
        return DemoSingletonHolder.INSTANCE;
    }

    protected Object readResolve() {
        return getInstance();
    }
}
```

That's all about **constructors in java**. Drop me your questions in comments section.

Happy Learning !!

Was this post helpful?

Let us know if you liked the post. That's the only way we can improve.

Yes

No

Recommended Reading:

1. Java Cloning – Deep and Shallow Copy – Copy Constructors
2. Checked exceptions thrown in initializer blocks can be declared by the constructors
3. Interface vs Abstract Class in Java
4. Encapsulation vs Abstraction in Java
5. Overloading vs Overriding in Java
6. Java Access Modifiers
7. Java Instance Initializer Blocks
8. Guide to Abstraction
9. Object Oriented Programming
0. Association, Aggregation and Composition

twilio.com
Sponsored
The future of customer engagement is now
twilio
State of Customer Engagement Report 2022

Join 7000+ Awesome Developers

Get the latest updates from industry, awesome resources, blog updates and much more.

Email Address

Subscribe

** We do not spam !!*

3 thoughts on “Constructors in Java”

Dazel

January 14, 2020 at 9:04 am

Please also add, If one declare a parameterized constructor then user should declare default constructor (as now JVM will not create default constructor by default).

Reply

priya

May 4, 2018 at 10:00 am

Constructor is a special method in Java which is used to initialize the object. It looks like a normal method however it is not.It is really important concept in java. Many thanks for sharing this.

Reply

priya

November 6, 2017 at 11:59 am

When the constructor is invoked using the new operator, the types must match those that are specified in the constructor definition. this is very informative post, many thanks for sharing.

Reply

Leave a Comment

Name *

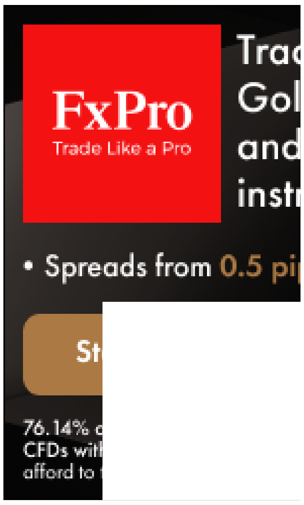
Email *

Website

☐ Add me to your newsletter and keep me updated whenever you publish new blog posts

Post Comment

Search …


FxPro
Trade Like a Pro
• Spreads from 0.5