

Private Constructors and Singleton Classes in Java

Difficulty Level : Medium Last Updated : 21 Jun, 2018

Let's first analyze the following question:

Can we have private constructors ?

As you can easily guess, like any method we can provide access specifier to the constructor. If it's made private, then it can only be accessed inside the class.

Do we need such 'private constructors' ?

There are various scenarios where we can use private constructors. The major ones are

1. Internal Constructor chaining
2. Singleton class design pattern

What is a Singleton class?

As the name implies, a class is said to be singleton if it limits the number of objects of that class to one.

We can't have more than a single object for such classes.

Singleton classes are employed extensively in concepts like Networking and Database Connectivity.

Design Pattern of Singleton classes:

The constructor of singleton class would be private so there must be another way to get the instance of that class. This problem is resolved using a class member instance and a factory method to return the class member.

Below is an example in java illustrating the same:

```
// Java program to demonstrate implementation of Singleton
// pattern using private constructors.
import java.io.*;

class MySingleton
{
    static MySingleton instance = null;
    public int x = 10;
```

```
// private constructor can't be accessed outside the class
private MySingleton() { }

// Factory method to provide the users with instances
static public MySingleton getInstance()
{
    if (instance == null)
        instance = new MySingleton();

    return instance;
}

// Driver Class
class Main
{
    public static void main(String args[])
    {
        MySingleton a = MySingleton.getInstance();
        MySingleton b = MySingleton.getInstance();
        a.x = a.x + 10;
        System.out.println("Value of a.x = " + a.x);
        System.out.println("Value of b.x = " + b.x);
    }
}
```

Output:

Value of a.x = 20

Value of b.x = 20

We changed value of a.x, value of b.x also got updated because both 'a' and 'b' refer to same object, i.e., they are objects of a singleton class.

This article is contributed by **Ashutosh Kumar Singh**. If you like GeeksforGeeks and would like to contribute, you can also write an article and mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above