



Getter and Setter in Java

Last Updated : 22 Jun, 2023

In Java, Getter and Setter are methods used to protect your data and make your code more secure. Getter and Setter make the programmer convenient in setting and getting the value for a particular data type.

Getter in Java: Getter returns the value (accessors), it returns the value of data type int, String, double, float, etc. For the program's convenience, the getter starts with the word "get" followed by the variable name.

Setter in Java: While Setter sets or updates the value (mutators). It sets the value for any variable used in a class's programs. and starts with the word "set" followed by the variable name.

Syntax

```
class ABC{
    .... private variable;
    ....
    .... public void setVariable(int x){
        .... this.variable=x;
    .... }
    ....
    .... public int getVariable{
        .... return variable;
    .... }
}
```

Note: In both getter and setter, the first letter of the variable should be capital.

Examples of Getter and Setter in Java

Example 1:

Java

```
// Java Program to Illustrate Getter and Setter

// Importing input output classes
import java.io.*;

// Class 1
// Helper class
class GetSet {

    // Member variable of this class
    private String name;

    // Method 1 - Getter
    public String getName() { return name; }

    // Method 2 - Setter
    public void setName(String N)
    {

        // This keyword refers to current instance itself
        this.name = N;
    }
}

// Class 2
// Main class
class GFG {

    // Main driver method
    public static void main(String[] args)
    {
        // Creating an object of class 1 in main() method
        GetSet obj = new GetSet();

        // Setting the name by calling setter method
        obj.setName("Geeks for Geeks");
        // Getting the name by calling getter method
        System.out.println(obj.getName());
    }
}
```

Output

Geeks for Geeks

Getter and Setter give you the convenience of entering the value of the variables of any data type by the requirement of the code. Getters and setters let you manage how crucial variables in your code are accessed and altered. It can be seen in the program discussed below as follows:

Example 2

Java

```
// Java Program to Illustrate Getter and Setter

// Importing input output classes
import java.io.*;

class GetSet {
    // Member variable of this class
    private int num;

    // Method 1 - Setter
    public void setNumber(int number)
    {
        // Checking if number is between 1 to 10 exclusive
        if (number < 1 || number > 10) {

            throw new IllegalArgumentException();
        }
        num = number;
    }

    // Method 2 - Getter
    public int getNumber() { return num; }
}

// Class 2
// Main class
class GFG {
    // Main driver method
    public static void main(String[] args)
    {
```

```
GetSet obj = new GetSet();

// Calling method 1 inside main() method
obj.setNumber(5);

// Printing the number as setter above
System.out.println(obj.getNumber());
}
}
```

Output

5

Explanation of the above program:

Here we can see that if we take a value greater than 10 then it shows an error, By using the setNumber() method, one can be sure the value of a number is always between 1 and 10. This is much better than updating the number variable directly.

***Note:** This could be avoided by making the number a private variable and utilizing the setNumber method. Using a getter method, on the other hand, is the sole way to read a number's value.*



sachi...



37

Next Article

Program to convert Array to Set in Java

Similar Reads

Advantages of getter and setter Over Public Fields in Java with Examples

Providing getter and setter methods to access any class field in Java can at first look pointless and meaningless, simply because you can make the field public,...

2 min read

Spring - Injecting Literal Values By Setter Injection

Spring IoC (Inversion of Control) Container is the core of Spring Framework. It creates the objects, configures and assembles their dependencies, manages thei...

4 min read

How to Build a Simple Alarm Setter App in Android?

In this article, we are going to see how to build a much interesting app named Alarm Setter. Alarm plays a vital role in our day-to-day life. Nowadays alarm ha...

6 min read

Spring - Dependency Injection by Setter Method

Dependency Injection is the main functionality provided by Spring IOC(Inversion of Control). The Spring-Core module is responsible for injecting dependencies...

3 min read

Spring - Injecting Objects by Setter Injection

Spring IoC (Inversion of Control) Container is the core of Spring Framework. It creates the objects, configures and assembles their dependencies, manages thei...

5 min read

Spring - Setter Injection with Dependent Object

Dependency Injection is the main functionality provided by Spring IOC(Inversion of Control). The Spring-Core module is responsible for injecting dependencies...

3 min read

Spring - Setter Injection with Non-String Collection

Dependency Injection is the main functionality provided by Spring IOC(Inversion of Control). The Spring-Core module is responsible for injecting dependencies...

3 min read

Spring - Setter Injection with Map

Spring is an open-source lightweight framework that allows Java EE developers to build simple, reliable, and scalable enterprise applications. This framework is...

5 min read

Spring - Setter Injection with Non-String Map

Dependency Injection is the main functionality provided by Spring IOC(Inversion of Control). The Spring-Core module is responsible for injecting dependencies...

3 min read

Spring - Setter Injection vs Constructor Injection

Spring Framework is built on top of servlets and one of its main features is Dependency Injection (DI) it's one of the most common principles of modern...

7 min read

Article Tags : [Java](#) [Java Programs](#)

Practice Tags : [Java](#)



Corporate & Communications Address:-
A-143, 9th Floor, Sovereign Corporate
Tower, Sector- 136, Noida, Uttar Pradesh
(201305) | Registered Address:- K 061,
Tower K, Gulshan Vivante Apartment,
Sector 137, Noida, Gautam Buddh
Nagar, Uttar Pradesh, 201305



Company

[About Us](#)
[Legal](#)
[In Media](#)
[Contact Us](#)
[Advertise with us](#)
[GFG Corporate Solution](#)

Languages

[Python](#)
[Java](#)
[C++](#)
[PHP](#)
[GoLang](#)
[SQL](#)