PACKAGES, IMPORT AND SCANNER CLASS IN JAVA

Velocity Notes

JULY 27, 2022

VELOCITY

1.1

Java Packages

A package is nothing but a physical folder structure (directories) that contains a group of related classes, interfaces, and sub-packages according to their functionality.

In java we have several built-in packages, for example when we need user input, we import a package like this:

```
import java.util.Scanner
```

Here:
→ **java** is a top-level package
→ **util** is a sub package
→ and **Scanner** is a class which is present in the sub package util.

Why we need packages?

Suppose imagine, if you are having a large number of files in your project that is deployed on server, now the code is released on production server. And now there are bugs in some files then how you can reach to that file? without packages it is very difficult. If you have packages then it will get very easy to go specific folder and find that file. That's why packages come into picture.

Advantages

Reusability- we can place the common code into one folder and reuse it.

Maintenance- if any new developer/tester joined your company then it will be easy to find the file which they wanted.

Name Conflicts: We can define two classes with the same name in different packages so to avoid name collision, we can use packages.

Types of packages in Java

As mentioned in the beginning of this guide that we have two types of packages in java.

1) User defined package: These are the packages that are defined by the user.

2) Built-in package: These are the packages that are already defined in java. like java.io.*, java.util.*, java.lang.* etc are known as built-in packages.

Syntax:

com.wipro.jpmorgan.insurance.policy.education

Here,

Package are generally starts with com folder.

Wipro is your company name.

jpmorgan is your client name.

insurance is your project name.

policy is your module name.

education is your sub-module name.

Note- All alphabets are starts with small case letters only.

Import Statement is java

Java has an import statement that allows you to import an entire package or use only certain classes and interfaces defined in the package.

When to use?

When we want to use one class within another class then go for import statement.

Example- suppose we have two different classes Test & Example in different packages.

```
package com.velocity;

public class Test {

	//method or variable
	public void m1() {
		System.out.println("this is the m1 method");
	}
}

package com.wipro.jpmorgan;

public class Example {

	public static void main(String[] args) {

		Test test= new Test();
	}
}
```

In the example class, we are calling the method of test class, so we need to use the import statement here. Otherwise, it will give compile time error

To resolve this issue, we need to import the highlighted line that is Import `import 'Test'(com.velocity)` by just clicking on it.

Different ways for import-

```
import com.velocity.Test; //correct
import com.velocity.*;  //correct- it will import the all the
classes.

import com.velocity; //wrong
```

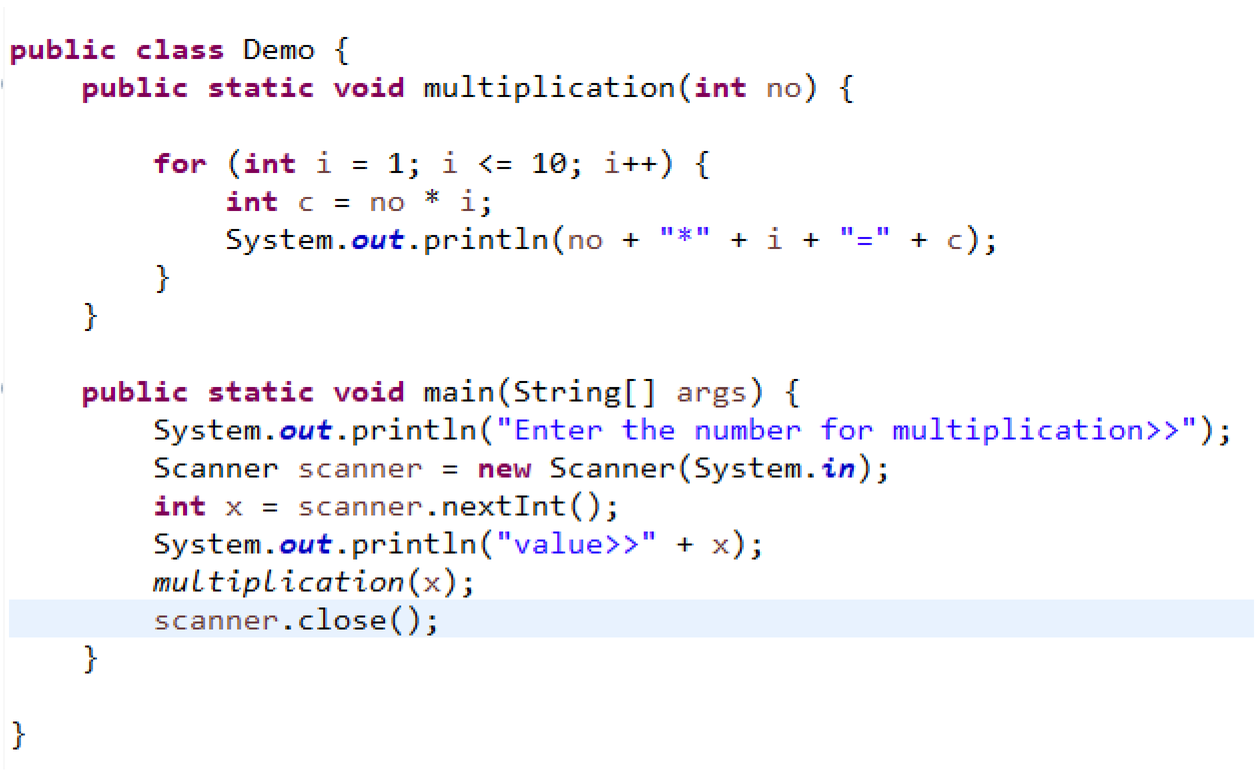
Scanner class in java

Scanner is a class in java.util package used for obtaining the input of the primitive types like int, double, etc. and strings. It is the easiest way to read input in a Java program.

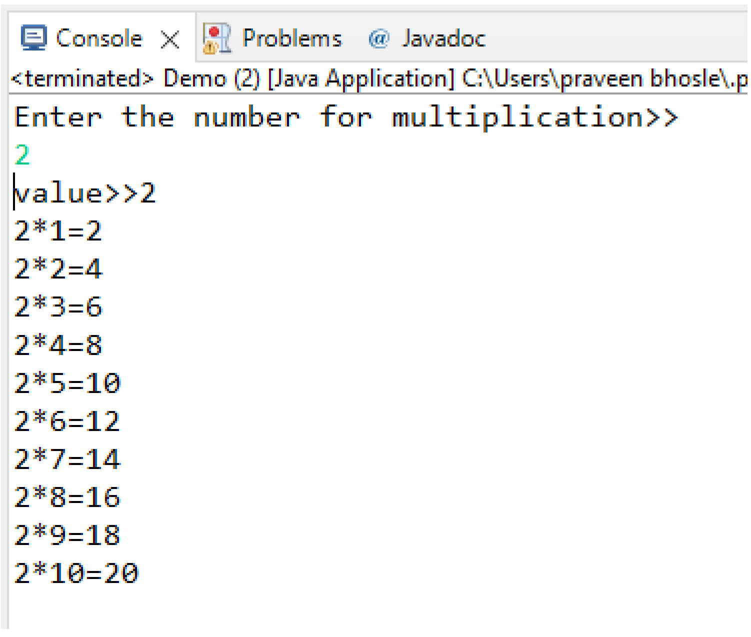
- To create an object of Scanner class, we usually pass the predefined object System.in.
- The System.in parameter is used to take input from the standard input. It works just like taking inputs from the keyboard.
- To read numerical values of a certain data type, the method to use is nextXYZ().

byte	nextByte()	It scans the next token of the input as a byte.
double	nextDouble()	It scans the next token of the input as a double.
float	nextFloat()	It scans the next token of the input as a float.
int	nextInt()	It scans the next token of the input as an Int.
long	nextLong()	It scans the next token of the input as a long.
short	nextShort()	It scans the next token of the input as a short.
boolean	nextBoolean()	It scans the next token of the input into a boolean value and returns that value.
String	nextLine()	It is used to get the input string that was skipped of the Scanner object.
String	next()	It is used to get the next complete token from the scanner which is in use.
boolean	hasNext()	It returns true if this scanner has another token in its input.

Examples :-1



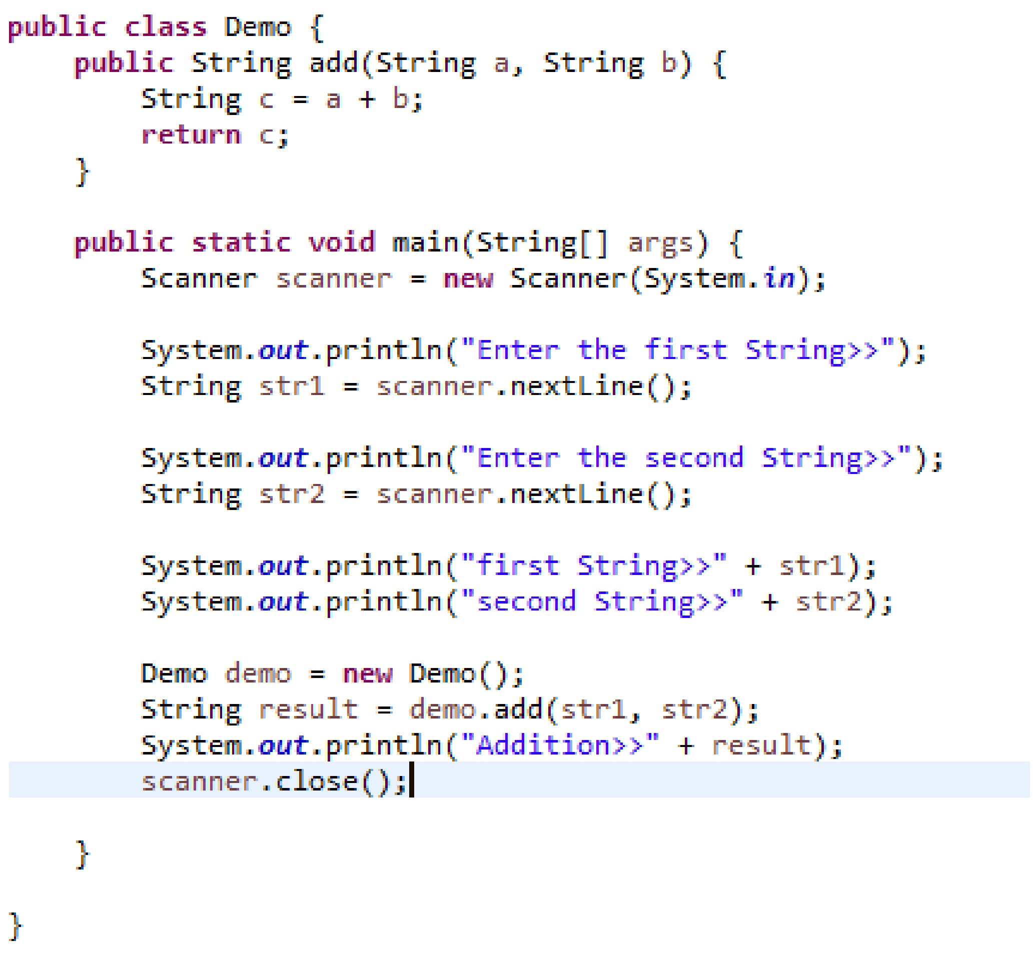
```java
public class Demo {
    public static void multiplication(int no) {

        for (int i = 1; i <= 10; i++) {
            int c = no * i;
            System.out.println(no + "*" + i + "=" + c);
        }
    }

    public static void main(String[] args) {
        System.out.println("Enter the number for multiplication>>");
        Scanner scanner = new Scanner(System.in);
        int x = scanner.nextInt();
        System.out.println("value>>" + x);
        multiplication(x);
        scanner.close();
    }

}
```



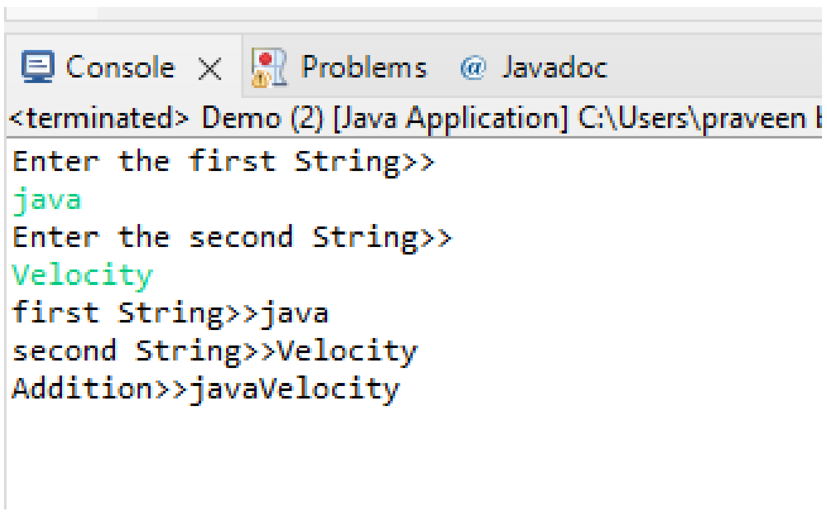
<terminated> Demo (2) [Java Application] C:\Users\praveen bhosle\.p

```
Enter the number for multiplication>>
2
value>>2
2*1=2
2*2=4
2*3=6
2*4=8
2*5=10
2*6=12
2*7=14
2*8=16
2*9=18
2*10=20
```

Example 2



```java
public class Demo {
    public String add(String a, String b) {
        String c = a + b;
        return c;
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.println("Enter the first String>>");
        String str1 = scanner.nextLine();

        System.out.println("Enter the second String>>");
        String str2 = scanner.nextLine();

        System.out.println("first String>>" + str1);
        System.out.println("second String>>" + str2);

        Demo demo = new Demo();
        String result = demo.add(str1, str2);
        System.out.println("Addition>>" + result);
        scanner.close();

    }

}
```



```
<terminated> Demo (2) [Java Application] C:\Users\praveen
Enter the first String>>
java
Enter the second String>>
Velocity
first String>>java
second String>>Velocity
Addition>>javaVelocity
```