Tutorial Link https://course.testpad.chitkara.edu.in/tutorials/Java: Iteration or Looping/5aa3ebba5ead875d7faaa8dd

TUTORIAL

Java: Iteration or Looping

Topics

while loop

Looping helps in repeatedly execution of a block subject to a condition. Java support three forms of loops while, for and do while. The while loop executes a statement (which may be a block statement) while a condition is true. The general form is

```
while (condition)
{
statement
}
```

The while loop will never execute if the condition is false at the outset. For example, following is the program to print the table of 2 with while loop:

Java

```java
class Main
{
  public static void main (String[] args)
  {
    int i=1;
    while  (i<=10)
    {
      System.out.println("2 * " + i + " = " + (2*i));
      i = i+ 1;
    }
  }
}

```

do while loop

A while loop tests at the top. Therefore, the code in the block may never be executed. If you want to make sure a block is executed at least once, you will need to move the test to the bottom. You do that with the do/while loop. Its syntax looks like this:

```
do
{
statement
} while (condition);
```

This loop executes the statement (which is typically a block) and only then tests the condition. It then repeats the statement and retests the condition, and so on. So the above program can be rewritten as below:

Java

```
class Main
{
  public static void main (String[] args)
  {
    int i=1;
    do
    {
      System.out.println("2 * " + i + " = " + (2*i));
      i = i + 1;
    } while  (i<=10);
  }
}

```

Determinate (for) loop

The for loop is a general construct to support iteration that is controlled by a counter or similar variable that is updated after every iteration. the following loop prints the numbers from 1 to 10 on the screen.

```
for (int i = 1; i <= 10; i++)
            System.out.println(i);
```

The first slot of the for statement usually holds the counter initialization. The second slot gives the condition that will be tested before each new pass through the loop, and the third slot explains how to update the counter. You can also write a loop that count down:

```java
class Main
{
  public static void main (String[] args)
  {
    for (int i = 15; i > 0; i--)
    {
      System.out.println("Next Minute starts in . . . " + i);
    }
    System.out.println("Congrats!");
  }
}

```

Be careful about testing for equality of floating-point numbers in loops. A for loop that looks like

```
for (double x = 0; x != 10; x += 0.1)
{
            // statements
}
```

may never end. Because of roundoff errors, the final value may not be reached exactly. For example, in the loop above, x jumps from 9.99999999999998 to 10.09999999999998 because there is no exact binary representation for 0.1.

When you declare a variable in the first slot of the for statement, the scope of that variable extends until the end of the body of the for loop.

```
for (int i = 1; i <= 10; i++)
{
// statements
}
// i no longer defined here
```

In particular, if you define a variable inside for statement, you cannot use the value of that variable outside the loop. Therefore, if you wish to use the final value of a loop counter outside for loop, be sure to declare it outside the loop header as follows:

```
int i;
for (i = 1; i <= 10; i++)
{
// statements
}
// i still defined here
```