Lecture 8: Java Loops

By Ahmed E. Khaled
Java Note
package welcomejava;
import java.util.Scanner;

public class WelcomeJava {

    public static void sayHello(){

        //here is the main method
        System.out.println("This is a method");
    }
}

public static void main(String[] args) {

    //here is the main method
    System.out.println("This is the start point of the program");
}
}
Recap to previous lecture!

- How to define method in Java?
- How to pass value to method?
- What is a java loop?
Lecture 8 : Java Loops Cont

• Continue Java loops
• Math Library and String object.
public static int sumBoth ( int num1 , int num2 ){
    int result = num1 + num2;
    return result;
}

public static void main(String[] args) {

    int x = 100;
    int y = 200;
    int z = sumBoth ( x , y );
    System.out.println("result = " + z );

    int m = 300;
    int n = 500;
    System.out.println("result = " + sumBoth ( m , n ) );
}
public static void incrementValue(int num) {
    System.out.println("User Passed value "+num);
    num = num + 2;
    System.out.println("the function updated the value to "+num);
}

public static void main(String[] args) {
    int x = 5;
    System.out.println("Value of x is "+x);
    incrementValue(x);
    System.out.println("Value of x is "+x);
    x = 12;
    incrementValue(x);
    System.out.println("Value of x is "+x);
}
public class WelcomeJava {

    public static int num1 = 10;

    public static void changeGlobal() {
        int num3 = 30;
        System.out.println(num3);
        System.out.println(num1);

        num1 = 100;
    }

    public static void main(String[] args) {
        int num2 = 20;
        System.out.println(num2);

        System.out.println(num1);
        changeGlobal();
        System.out.println(num1);
    }
}
Java Loops

• Repeat a given set of statements over and over, of course not forever (*infinite loop*).

• A one-time execution of a loop body is referred to as an *iteration* (or repetition) of the loop.

• During this course, we will check different types of java Loops:
  1. While loop.
  2. For loop.
  3. Do while.
  4. Nested Loops
The While Loop

• While loop will repeat a statement over and over, but only so long as a specified condition remains true. A while loop has the form:

```java
while ( boolean Expression ) {
    Instruction
    Instruction
    ....
    Instruction
}
```
int count = 0;
while (count < 5) {
    System.out.println("Hello Java its "+ count);
    count ++;
}

Guess the output!
int number = 6;

while ( number > 1 ) {

    System.out.println("this is number " + number + " !!");

    number = number - 1; // Go on to the prev number.
}

System.out.println("Done!");
Guess the output!

```c
int sum = 0,
Int i = 1;
while (i < 10) {
    sum = sum + i;
}
```
For Loops

For loop can be easier to construct and easier to read.

For (initialization; continuation expression; step value){
  Statement(s)
}

For Loops

```java
for ( int count = 0; count < 5; count ++ ) {
    System.out.println(count);
}
```
Guess the output!

for(int i = 0 ; i < 5 ; i++){
    System.out.println("In first for loop with iteration " + i);
}

for(int i = 0 ; i <= 5 ; i++){
    System.out.println("In second for loop with iteration " + i);
}

for(int i = 1 ; i < 5 ; i++){
    System.out.println("In third for loop with iteration " + i);
}
for(int i = 5; i > 1; i--){
    System.out.println("In forth for loop with iteration " + i);
}

for(int i = 1; i < 5; i = i + 2){
    System.out.println("In fifth for loop with iteration " + i);
}
Do While loop

• It may be more convenient to test the boolean expression at the end of a loop, instead of at the beginning (as done in while loop.

• The do..while statement is very similar to the while statement, except that the word “while,” along with the condition that it tests, has been moved to the end.

     do {
         Play a Game
         Ask user if he wants to play another game
         Read the user’s response
     } while ( the user’s response is yes );
//while loop
int count1 = 0;
while( count1 < 5 ){
    System.out.println("In while loop with iteration number" + count1);
    count1 = count1 + 1;
}

//Do while loop
int count2 = 0;
do{
    System.out.println("In Do-while loop with iteration number" + count2);
    count2 = count2 + 1;
}while( count2 < 5 );
```java
int data = 0;
int sum = 0;

do {
    System.out.print("Enter an integer, 0 to exit");
    data = input.nextInt();
    sum = sum + data;
} while (data != 0);

System.out.println("The sum is " + sum);
```
Nested for loops

Nested loops consist of an *outer* loop and one or more *inner* loops.

Each time the outer loop is repeated, the inner loops are reentered, and started anew.

```cpp
OuterLoop()
{
  InnerLoop()
  {
  }
}
```
for(int i = 0; i < 3; i++){
    System.out.println("Outer Loop with iteration " + i);
    for(int j = 0; j < 5; j++){
        System.out.println("Inner loop with iteration " + j + " and outer of " + i);
    }
}
Outer Loop with iteration 0
Inner loop with iteration 0 and outer of 0
Inner loop with iteration 1 and outer of 0
Inner loop with iteration 2 and outer of 0
Inner loop with iteration 3 and outer of 0
Inner loop with iteration 4 and outer of 0
Outer Loop with iteration 1
Inner loop with iteration 0 and outer of 1
Inner loop with iteration 1 and outer of 1
Inner loop with iteration 2 and outer of 1
Inner loop with iteration 3 and outer of 1
Inner loop with iteration 4 and outer of 1
Outer Loop with iteration 2
Inner loop with iteration 0 and outer of 2
Inner loop with iteration 1 and outer of 2
Inner loop with iteration 2 and outer of 2
Inner loop with iteration 3 and outer of 2
Inner loop with iteration 4 and outer of 2
for(int i = 0 ; i < 3 ; i++){
    System.out.println("Outer Loop with iteration " + i);
    for(int j = 0 ; j < 5 ; j++){
        System.out.println("Inner loop with iteration " + j + " and outer of " + i);
        for(int k = 0 ; k < 10 ; k++){
            System.out.println("Inner of Inner with iteration " + k + ", Inner "+ j + " and outer of " + i);
        }
    }
}
Break Vs. Continue

- Break, is a way to break out of the middle of any loop and jump out of the loop.

```java
Int sum = 0;
Int item = 0;
while (item < 5) {
    item++;
    sum = sum + item;
    if (sum >= 6) break;    // jump out of loop
} // continue here after break
```
• Continue statement tells the computer to skip the rest of the current iteration of the loop then jumps back to the beginning of the loop and continues with the next iteration.

```
Int sum = 0;
Int item = 0;
while (item < 5) {
    item++;
    if (sum >= 6) { continue; }
    sum = sum + item;
}
// continue here the loop is done
```
Build Program

• Note the difference:
  – System.out.println(“hello world!”);
  – System.out.println();
  – System.out.print(“hello world”);
  – System.out.println(“!”);
Build Program

• Using loop that iterates from 1-5, build program that prints the square value of the iteration number.

• Using loops, build program that prints the following:
  
  *******

Done!
Build Program

• Use nested loop to print the multiplication table 1x1 to 9x9

• Using loops, define method that takes print a value of m for n times.