

People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
University M'hamed Bouguerra - Boumerdes
Institute of Electrical and Electronics Engineering
Electronics Department



Option: Computer engineering
LAB REPORT n°2
OCTOBER 27, 2022

Title

EE423: Advanced Programming/ Working with Classes, Objects and Methods

Authored by :
Agli Wafa
Zidane Aymen

Instructor :
Dr.A Zitouni

Session : 2022/2023

Contents

1	Introduction	1
2	Exercise 01: <i>if-else-then</i>	1
3	Exercise 02: <i>Integer to Binary conversion</i>	3
4	Exercise 03: <i>Using Math Library</i>	3
5	Exercise 04: <i>Loops</i>	4
6	Exercise 05: <i>Bit-wise operators</i>	5
7	Exercise 06: <i>Count set bits</i>	7
8	Exercise 07: <i>Single Number</i>	8
9	Exercise 08: <i>Get First Set Bit</i>	8
10	Exercise 09: <i>Printing and loops</i>	9
11	Conclusion	10

1 Introduction

Java compiler executes the code from top to bottom. The statements in the code are executed according to the order in which they appear. However, Java provides statements that can be used to control the flow of Java code. Such statements are called **control flow statements**. It is one of the fundamental features of Java, which provides a smooth flow of program.

In this Lab, we are going to learn how to utilize the different control statements to solve different problems, as well as how to use the various operators and instructions offered by JAVA.

Tools and Software:

1. A PC with ECLIPSE IDE V8.
2. Online LaTeX Editor for writing the report.

2 Exercise 01: *if-else-then*

```
1 /* Guessing a number game */
2 import java.util.Random; //To use the class Random from the package util
3 import java.util.Scanner; //To use the class Scanner from the package util
4 public class Game {
5     public static void main(String[] args) {
6         Scanner sca= new Scanner(System.in); //creating object with reference sca from Scanner
7         Random r= new Random(); //Creating object from random
8         int rand= r.nextInt(1000)+1; //using r Object to generate random number between 1 and
           1000
9         int a; //the input
10        int max= 1000, min= 1, tries=0; //intialization of local variables
11        do {
12            do { //insuring the input a between min and max
13                System.out.println("guess a number between " +min + " and " + max );
14                a = sca.nextInt();
15                if(a<min || a>max) {System.out.println("out of the range");}
16            }while(a<min || a>max) ;
17            tries++; //increment number of tries in each iteration
18            if(a<rand) { //if the input a is less than rand
19                min=a; //change the min to a
20                System.out.println("Bigger");
21            } else if(a>rand) { //if the input a is bigger than rand
22                max=a; //change the max to a
23                System.out.println("Smaller");
24            } else // a == rand
25            {System.out.println("Yes! You ve got the right answer after " +tries + " tries");}
26        } while(a!=rand);}}
```

Results and Explanation:

After running the program, the user will be asked to enter a number between 1 and 1000, when she/he enters the number the program keeps asking for another number but in a smaller range (since we used the value max and min which will vary each time the loop repeats), until the user guesses the right number. while the user is entering a number outside min or max an error message appears.

```

Javadoc Declaration Console × Debug
<terminated> Game [Java Application] C:\Users\agli wafa\p2\pool\plu
guess a number between 1 and 1000
10001
out of the range
guess a number between 1 and 1000
500
Bigger
guess a number between 500 and 1000
800
Bigger
guess a number between 800 and 1000
500
out of the range
guess a number between 800 and 1000
900
Bigger
guess a number between 900 and 1000
960
Bigger
guess a number between 960 and 1000
980
Smaller
guess a number between 960 and 980
970
Smaller

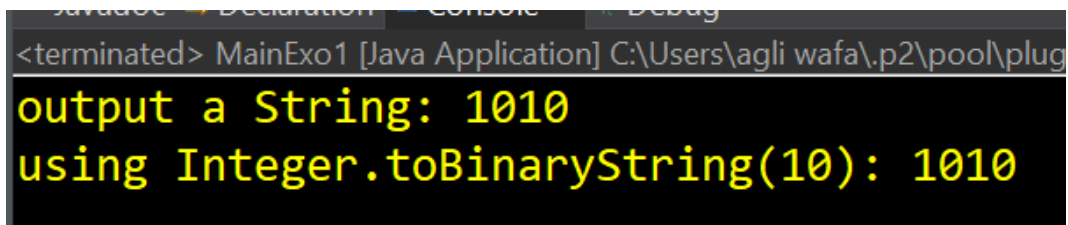
guess a number between 960 and 970
967
Bigger
guess a number between 967 and 970
968
Bigger
guess a number between 968 and 970
969
Yes! You've got the right answer after 9 tries

```

3 Exercise 02: *Integer to Binary conversion*

```
1 public class BinaryConvers2 {
2     public static String toBinary(int i) {
3         int rest;
4         String str = ""; //String initialization
5         do {
6             rest = i % 2; //find the rest of the division using modulo
7             str = rest + str; //concatination of rest in the beginning of str rest may
            takes 0 or 1
8             i = i / 2; //deviding i by 2 and assign it to new i
9         } while (i != 0);
10
11         return str; // return the binnary number
12     }
13     public static void main(String[] args) {
14         String result = toBinary(10); //call the method and pass argument i=10
15         System.out.println(result);
16         System.out.println(Integer.toBinaryString(10)); //for comparision
17     }
18 }
```

Results and Explanation:



```
<terminated> MainExo1 [Java Application] C:\Users\agli wafa\.p2\pool\plug
output a String: 1010
using Integer.toBinaryString(10): 1010
```

The table bellow shows the execution of toBinary method when $i = 10$.

Iteration	$i = i/2$	$rest = i\%2$	$str = rest + str$
0	10	/	/
1	5	0	0
2	2	1	10
3	1	0	010
4	0	1	1010

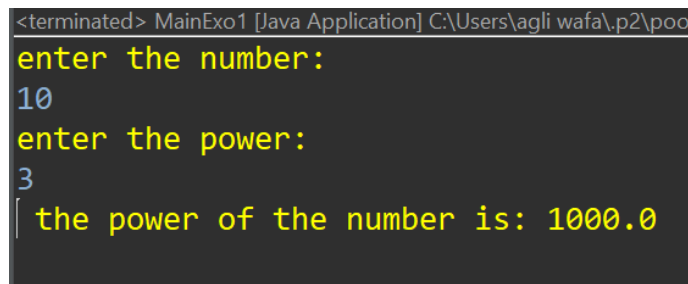
4 Exercise 03: *Using Math Library*

```
1 import java.util.Scanner; //import Scanner class
2 public class PowerFunction {
3     public static double power(double nb, int pw) {
4         double pow = 1; //initialization of pow to 1
5         for (int i = 1; i <= pw; i++) {
6             pow = pow * nb;
7         }
8     }
9 }
```

```

8
9     return pow;
10 }
11
12 public static void main(String[] args) {
13     Scanner sca = new Scanner(System.in);
14     System.out.println("enter the number: ");
15     double nb = sca.nextDouble();
16     System.out.println("enter the power: ");
17     int pw = sca.nextInt();
18     System.out.println(" the power of the number is: " + power(nb, pw));}

```



```

<terminated> MainExo1 [Java Application] C:\Users\agli wafa\.p2\poo
enter the number:
10
enter the power:
3
the power of the number is: 1000.0

```

Supposing the user enters nb=4 and pw = 3; the power method runs like bellow:

Iteration	nb	pow = pow * nb
0	4	1
1	4	4
2	4	16
3	4	64

5 Exercise 04: *Loops*

```

1 import java.util.Scanner;
2
3 public class SumOfOddNum {
4     public static int sumSquaredOdd(int n) {
5         int sum = 0;
6         int j = 1;
7         for (int i = 0; i < n; i++) {
8             //we force an int return because sum is type of int and Math.pow(j, 2) is double
9             sum = (int)(sum + Math.pow(j, 2));
10            j+=2;
11        }
12        return sum;
13    }
14    public static void main(String[] args) {
15        Scanner sca = new Scanner(System.in);
16        System.out.println("enter a number: ");
17        int i = sca.nextInt();
18        //call sumSquaredOdd

```

```

19     int result = sumSquaredOdd(i);
20     System.out.print("the first sum of odd numbers in power 2 is " + result);
21
22 }
23 }

```

<terminated> MainExo1 [Java Application] C:\Users\agli wafa\.p2\pool\plugins\org.eclipse.

enter a number:
5
the first sum of odd numbers in power 2 is 165

Iteration	j	sum
/	1	0
0	3	$0+1^2$
1	5	$0+1^2+3^2$
2	7	$0+1^2+3^2+5^2$
3	9	$0+1^2+3^2+5^2+7^2$
4	11	$0+1^2+3^2+5^2+7^2+9^2 = 165$

6 Exercise 05: *Bit-wise operators*

```

1 import java.util.Scanner;
2 public class BitwiseOperators {
3
4     public static String toBinary(int x) {
5         // change x to binary and replace spaces by 0
6         String s = String.format("%16s", Integer.toBinaryString(x)).replaceAll(" ", "0");
7         int l = s.length()-1; //length of our binary
8         String s2= ""; //new string
9         while(s2.length()<20){//while s2 doesnt have 19 chars
10             s2 = s.charAt(l)+s2; //s2 = s[l]
11             if(l%4==0){ //if l is Multiple of 4 we add space to s2
12                 s2 = " "+s2;
13             }
14             l--;
15         }
16         return s2;
17     }
18 }
19 public static void printBitwiseOperators(int a, int b, int c) {
20
21     System.out.println("-----");
22     System.out.println("a in binary " + toBinary(a));
23     System.out.println("b in binary " + toBinary(b));
24     System.out.println("c in binary " + toBinary(c));
25     System.out.println("-----");

```

```

26     int Or= a|b;
27     System.out.println(" a|b      :   " + toBinary(Or));
28     int And= a & b;
29     System.out.println(" a & b    :   " + toBinary(And));
30     int Xor= a ^ b;
31     System.out.println(" a ^ b    :   " + toBinary(Xor));
32     int Not= ~a ;
33     System.out.println(" ~a       :   " + toBinary(Not));
34     int sotl= b<<1;
35     System.out.println(" b<<1     :   " + toBinary(sotl));
36     int sttl= b<<2;
37     System.out.println(" b<<2     :   " + toBinary(sttl));
38     int sotr= b>>1;
39     System.out.println(" b>>1     :   " + toBinary(sotr));
40     int ks= c>>1;
41     System.out.println(" c>>1     :   " + toBinary(ks));
42     int is= c>>>1;
43     System.out.println(" c>>>1    :   " + toBinary(is));
44 }
45 public static void main(String[] args) {
46     Scanner sca= new Scanner(System.in);
47     System.out.println("enter the a: ");
48     int a= sca.nextInt();
49     System.out.println("enter the b: ");
50     int b= sca.nextInt();
51     System.out.println("enter the c: ");
52     int c= sca.nextInt();
53     printBitwiseOperators( a,  b,  c);}}

```

```

<terminated> MainExo1 [Java Application] C:\Users\agli wafa\.p2\
enter the a:
10
enter the b:
12
enter the c:
-10
-----
a in binary  0000 0000 0000 1010
b in binary  0000 0000 0000 1100
c in binary  1111 1111 1111 0110
-----
a|b      :   0000 0000 0000 1110
a & b    :   0000 0000 0000 1000
a ^ b    :   0000 0000 0000 0110
~a       :   1111 1111 1111 0101
b<<1     :   0000 0000 0001 1000
b<<2     :   0000 0000 0011 0000
b>>1     :   0000 0000 0000 0110
c>>1     :   1111 1111 1111 1011
c>>>1    :   1 1111 1111 1111 011

```


7 Exercise 06: *Count set bits*

```
1 import java.util.Scanner;
2
3 public class CountSetBits2 {
4     public static int countSetBits(int i) {
5         int cout=0;
6         while(i!=0) {
7             System.out.println(i +"\t" +Integer.toBinaryString(i));
8             System.out.println(i-1 +"\t" +Integer.toBinaryString(i-1));
9             int j=i & (i-1); //j holds bitwise & between i and i-1
10            if (j==0) {///if j in the last iteration
11                System.out.println(i+"&" + (i-1) +"\t" +Integer.toBinaryString(i)+" (" +j+")"+
12                " END");
13            }else
14                System.out.println(i+"&" + (i-1) +"\t" +Integer.toBinaryString(i)+" (" +j+")");
15            i= j;
16            cout++;
17            System.out.println();
18            System.out.println();
19        }
20        return cout;
21    }
22    public static void main(String[] args) {
23        Scanner sca =new Scanner(System.in);
24        System.out.println("Enter an integer ");
25        int n=sca.nextInt();
26        System.out.println(countSetBits(n));
27    }
28 }
29 }
```

```
<terminated> MainExo1 [Java Application] C:\Users\agli v
Enter an integer
125
125      1111101
124      1111100
125&124 1111101 (124)

124      1111100
123      1111011
124&123 1111100 (120)

120      1111000
119      1110111
120&119 1111000 (112)

112      1110000
111      1101111
112&111 1110000 (96)
```

```
96      1100000
95      1011111
96&95   1100000 (64)

64      1000000
63      1111111
64&63   1000000 (0) END

6
```

8 Exercise 07: *Single Number*

• Solution 01:

```
1 public class SingleNumber {
2
3 public static void main(String[] args) {
4     int[] nums = { 4, 1, 2, 9, 1, 4, 2 };
5     //initialize every num by false as if every num has an occurrence
6     boolean [] found = {false, false,false,false,false,false,false};
7     for(int i =0; i<nums.length; i++) {
8         for(int j=i+1; j<nums.length; j++) {
9             if((nums[i]^0)==nums[j]) {if occurrence of i is j
10                 found[i] = true; //set found[i] to true
11                 found[j] = true; //set found[j] to true
12             } } }
13     //search for false index in found and return it's equivalent in nums
14     for (int i =0; i< found.length; i++) {
15         if (found[i]==false)
16             System.out.print(nums[i]);
17     }}}
```

• Solution 02:

```
1     int countOccurence = 0; //every num has two copies so we look for num that
2     owns only one copy
3     for(int i =0; i<nums.length; i++) {
4         for(int j=0; j<nums.length; j++) {
5             if((nums[i]^0)==nums[j]) {
6                 countOccurence ++;
7             }
8         }
9         if (countOccurence == 1) System.out.print(nums[i]);
10        countOccurence =0; //when j==nums.length reinetialize countOccurence
11    }
```

9 Exercise 08: *Get First Set Bit*

```
1 public class GetFirstSetBit {
2     public static void main(String[] args) {
3         Scanner sca =new Scanner(System.in);
4         int n=sca.nextInt();
5         int i=1;
6         int count = 1;
7         while((n&i)==0) {
8             i=i<<1;
9             count++;
10        }
11        System.out.println("Input n = "+n+"(0b"+Integer.toBinaryString(n)+") "+ "Output:"
12        + count);
13    }
14 }
```

```
<terminated> MainExo1 [Java Application] C:\Users\agli wafa\.  
18  
Input n = 18(0b10010)  Output:2
```

10 Exercise 09: *Printing and loops*

- **Line method:**

```
1 static void line(int n) {  
2     for(int i = 0; i<n; i++) //n ==> *  
3         System.out.print("*");  
4 }  
5
```

- **square_fill method:**

```
1 static void square_fill(int n) {  
2     for(int i = 0; i<n; i++) {  
3         for (int j = 0; j < n; j++)  
4             System.out.print("*"); // * 0,0 to n,n  
5         System.out.println(); //when i,n back to line  
6     }  
7 }  
8
```

- **square_no_fill method:**

```
1 static void square_no_fill(int n) {  
2     int i, j;  
3     for (i = 1; i <= n; i++){  
4         for (j = 1; j <= n; j++){  
5             if (i == 1 || i == n || j == 1 || j == n)  
6                 System.out.print("*");  
7             else  
8                 System.out.print(" ");  
9         }  
10        System.out.println(); // when i,n back to line  
11    }  
12 }  
13
```

- **triangle:**

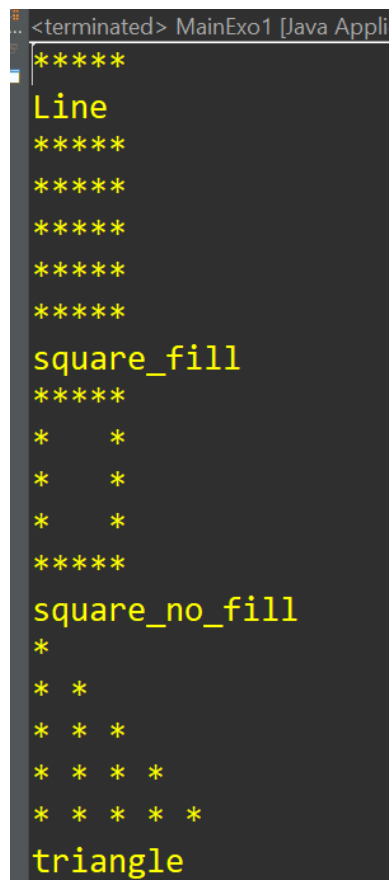
```
1 static void triangle(int n) {  
2     int i, j;  
3     for(i=0; i<n; i++){//for each row  
4         for(j=0; j<=i; j++){//for each column  
5             System.out.print("* ");  
6         }  
7         System.out.println(); // ending line after each row  
8     }  
9 }  
10
```

- **triangle_centered**

```

1  static void triangle_centered(int n) {
2      for (int i=0; i<n; i++){
3          for (int j=n-i; j>1; j--){ //we have n-i space
4              System.out.print(" ");
5          }
6          for (int j=0; j<=i; j++ ){
7              System.out.print("* ");
8          }
9          System.out.println();
10     }
11 }

```



```

*****
Line
*****
*****
*****
*****
*****
square_fill
*****
*   *
*   *
*   *
*****
square_no_fill
*
* *
* * *
* * * *
* * * * *
triangle

```



```

triangle
  *
 * *
* * *
* * * *
* * * * *
triangle_centered

```

11 Conclusion

1. Notices from exo 01:

- **Random** in java returns a random number between [0,1], if we want to specify the range we must apply the following format **`random.nextInt(max - min + 1) + min`**
- Unlike other loops ***do-while*** executed at least once.

2. *Notices from exo 05:*

- **Integer.toBinnary(x)** method of java returns a String of 32bits when the integer is negative or when we apply `Integer.toBinnary(x)`, if we want to get specific number of bits 16 in this example we can make a condition for that.
- **String.format** works like printf in c, example : `String.format("%s",Integer.toBinnary(10));` output would be 1010. And, `String.format("%16s",Integer.toBinnary(10)).replaceAll(" ", "0");` output would be 00000000000001010
- in order to have spaces between chars every 4 bits we used a loop over the string that copy the bit and checks if bit index is multiple of 4 using modulus
- Bit-wise operators works on binary digits or bits of input values. We can apply these to the integer types – long, int, short, char, and byte.