

جامعة طرابلس كلية تقنية المعلومات

البرمجة الشيئية

Object Oriented Programming  
(with Java)

ITGS211

المحاضرة الرابعة

الفصل الدراسي: ربيع 2022

1

1

الحلقات LOOPS في JAVA

2

2

# الحلقات LOOPS في JAVA

كثيراً ما نحتاج في البرنامج إلى تكرار أمر معين موجه إلى الحاسب عدداً من المرات، وتوفر لنا لغة الجافا عدة وسائل تمكن المبرمج من أداء هذا التكرار، وعادة ما تسمى هذه الوسائل بالحلقات التكرارية ويوجد العديد من الحلقات التكرارية وهي :

- **while Loop**
- **do-while Loop**
- **for Loop**

3

3

## الحلقة while

- في هذه الحلقة التكرارية نحتاج إلى الشرط فقط وطالما كان هذا الشرط متحققاً استمرت الحلقة في التكرار **والشكل العام لها:**

```
while (condition)
{
    Statement (s) ; // loop-body
}
```

- حلقة **while** تسمى حلقة الاختبار القبلي (*pretest*)، مما يعني أنه سيتم اختبار قيمة الشرط قبل تنفيذ الحلقة. لذلك فإنها ستنفذ 0 أو أكثر من المرات.

**صفر ( 0 ) من المرات : إذا كان الشرط غير محقق فإن الحلقة لن تنفذ.**

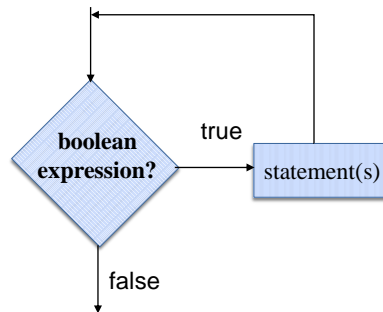
4

4

## الحلقة while

- يجب الحرص على جعل الشرط يصل إلى false في مكان ما من الحلقة حتى تنتهي الحلقة. والا فإنه سيدخل في تكرار لانهايتي وهذا ما يسمى infinite loops.

### while loop flowchart



5

5

## مثال

```

public class Java_WhileLoop {
    public static void main(String[] args) {
        int m= 1;
        while (m <= 5){
            System.out.println("m= " + m);
            m++;
        }
        System.out.println("*****");
    }
}
  
```

Loop body

```

m= 1
m= 2
m= 3
m= 4
m= 5
*****
  
```

6

6

بالإمكان اعادة المثال السابق (باستخدام جملة مركبة فيها الشرط وتغيير العداد بين القوسين ( ) ، وسنحصل على نفس النتائج كما يلي:

```
public static void main(String[] args) {
    int m= 0;
    while (++m <= 5)
        System.out.println("m= " + m); //Loop body
    System.out.println("*****");
}
```

ماذا لو تم استبدال  $(++m)$  بـ  $(m++)$ ؟

```
int m= 0;
while (m++ <= 5)
    System.out.println("m= " + m); //Loop body
System.out.println("*****");
```

7

7

بالإمكان اعادة المثال السابق (باستخدام جملة مركبة فيها الشرط وتغيير العداد بين القوسين ( ) ، وسنحصل على نفس النتائج كما يلي:

```
public static void main(String[] args) {
    int m= 0;
    while (++m <= 5)
        System.out.println("m= " + m); //Loop body
    System.out.println("*****");
}
```

```
m= 1
m= 2
m= 3
m= 4
m= 5
*****
```

ماذا لو تم استبدال  $(++m)$  بـ  $(m++)$ ؟

```
int m= 0;
while (m++ <= 5)
    System.out.println("m= " + m); //Loop body
System.out.println("*****");
```

```
m= 1
m= 2
m= 3
m= 4
m= 5
m= 6
*****
```

8

8

مثال :

```
public class Selectswitch2 {
    public static void main(String[] args) {
        int i=0;
        while(i<=10){
            System.out.print (i+",");
            ++i;
        }
    }
}
```

run:  
0,1,2,3,4,5,6,7,8,9,10,BUILD SUCCESSFUL (total time: 0 seconds)

```
public class Selectswitch2 {
    public static void main(String[] args) {
        int i=0;
        while(i++<=10){
            System.out.print (i+",");
        }
    }
}
```

run:  
1,2,3,4,5,6,7,8,9,10,11,BUILD SUCCESSFUL (total time: 0 seconds)

لماذا هذا الفرق في النتائج؟؟

??

9

9

## الحلقات اللانهائية Infinite Loops

- يجب الحرص على جعل الشرط يصل إلى false في مكان ما من الحلقة حتى تنتهي الحلقة. والا فإنه سيدخل في تكرار لانهاية وهذا ما يسمى infinite loops.
- كما في المثال أدناه فإن قيمة X تساوي 20 وهي فعلا أكبر من الصفر وبما أن قيمتها لا تتغير فهي ستبقى دائما أكبر من الصفر

```
int x = 20;
while(x > 0)
{
    System.out.println("x is greater than 0");
}
```

10

10

## الحلقات اللانهائية Infinite Loops

لحل المشكلة والخروج من الحلقة اللانهائية يجب اضافة جملة  
تغير قيمة X لتقل (مثلا X--) ويصبح الشرط غير صحيح false

```
int x = 20;
while(x > 0)
{
    System.out.println("x is greater than 0");
    x--;
}
أو
int x = 20;
while(x-- > 0)
{
    System.out.println("x is greater than 0");
}
```

11

11

```
public static void main(String[] args) {
    int i=0;
    while (i++<=10);
    {
        System.out.print(i + " , ");
    }
}
```

تنبيه

Empty statement after 'while' in main  
(Alt-Enter shows hints)

الناتج

Output - JavaApplication2 (run)

run:  
12 , BUILD SUCCESSFUL

12

12

## اكتب برنامج لطباعة جدول الضرب من 1 إلى 10 باستخدام حلقة

: while

```

public class Selectswitch2 {
    public static void main(String[] args) {
        int j=1;
        while(j<=10){
            System.out.println ();
            int i=1;
            while(i<=10){
                System.out.print("\t"+ i*j);
                ++i;
            }
            ++j;
        }
    }
}

```



جدول 1	1	2	3	4	5	6	7	8	9	10
2	2	4	6	8	10	12	14	16	18	20
3	3	6	9	12	15	18	21	24	27	30
4	4	8	12	16	20	24	28	32	36	40
5	5	10	15	20	25	30	35	40	45	50
6	6	12	18	24	30	36	42	48	54	60
7	7	14	21	28	35	42	49	56	63	70
8	8	16	24	32	40	48	56	64	72	80
9	9	18	27	36	45	54	63	72	81	90
جدول 10	10	20	30	40	50	60	70	80	90	100

13

13

## الحلقة do-while

- حلقة do-while وتسمى أيضاً حلقة do هي حلقة الاختبار البعدي (*post-test*)، مما يعني أنه سيتم تنفيذ حلقة قبل اختبار الشرط (*condition*). بمعنى آخر لا بد من تنفيذ الجملة/الجملة البرمجية التي بالحلقة على الأقل مرة واحدة.
- الشكل العام لها:

```

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

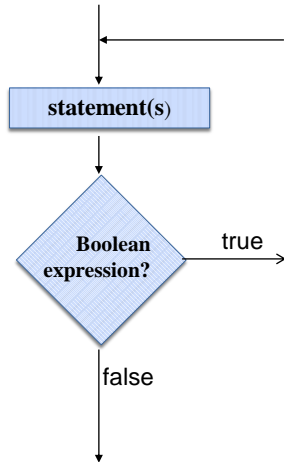
```

لاحظ أن الشرط ينتهي بفاصلة منقوطة (;)

14

14

## The do-while Loop Flowchart



15

15

## مثال (1) do-while

```

1 package dowhile1;
2 public class Dowhile1 {
3     public static void main(String[] args)
4     {
5         int i=0, sum=0;
6         do{
7             sum+=1;
8             System.out.println("i="+i+"&sum="+sum);
9             i+=2;
10        }while(i<=10);
11    }
12 }
13
  
```

```

run:
i=0&sum=0
i=2&sum=2
i=4&sum=6
i=6&sum=12
i=8&sum=20
i=10&sum=30
BUILD SUCCESSFUL (total time: 2 seconds)
  
```

i	sum	output	i+=2	(i<=10)
0	0	i=0 & sum=0	2	true
:				
8	20	i=8 & sum=20	10	(10<=10)→ true
10	30	i=10 & sum=30	12	(12<=10)→ false

i	sum	(i<=10)
0	0	true
2	2	True
4	6	True
6	12	True
8	20	True
10	30	True
12		false

16

16



## مثال (2) do-while

```
package dowhile1;
public class Dowhile1 {
    public static void main(String[] args) {
        int i=1;
        do{
            System.out.println("Ahmed");
            i++;
        }while(i<=5);
    }
}
```

```
run:
Ahmed
Ahmed
Ahmed
Ahmed
Ahmed
Ahmed
BUILD SUCCESSFUL (total time: 0 seconds)
```

17

17

## الحلقة (FOR LOOP)

- الحلقة for تعتبر جملة اختبار قبلي pre-test
- الحلقة for تسمح للمبرمج بإعطاء القيمة الابتدائية لدليل الحلقة واختباره وتغيير قيمته (بالزيادة أو النقصان) في نفس السطر.
- الشكل العام لها:

```
for(initialization; condition; update)
{
    statement(s);
}
Next statement;
```

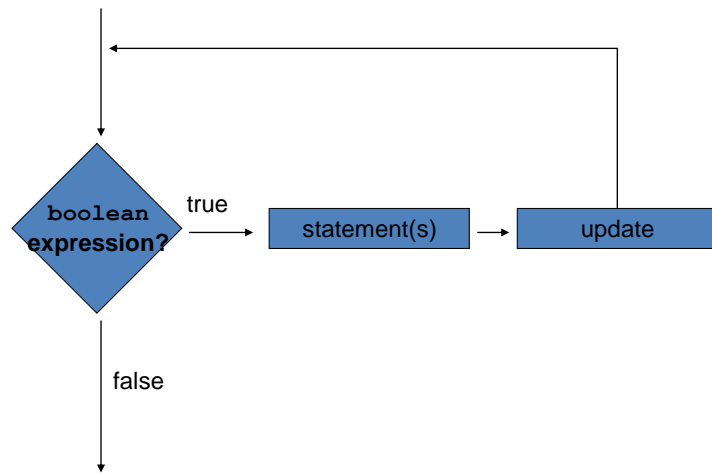
↓ القيمة الابتدائية التي يبدأ عندها العد  
 ↓ الشرط  
 ↑ مقدار الزيادة أو النقصان  
 ← الجمل المراد تكرارها

**ملاحظة :** جملة For لها قوس بداية ونهاية ويتم وضع الأقواس في حالة تكرار أكثر من جملة كما يمكن الاستغناء عن هذه الأقواس في حالة تكرار جملة واحدة

18

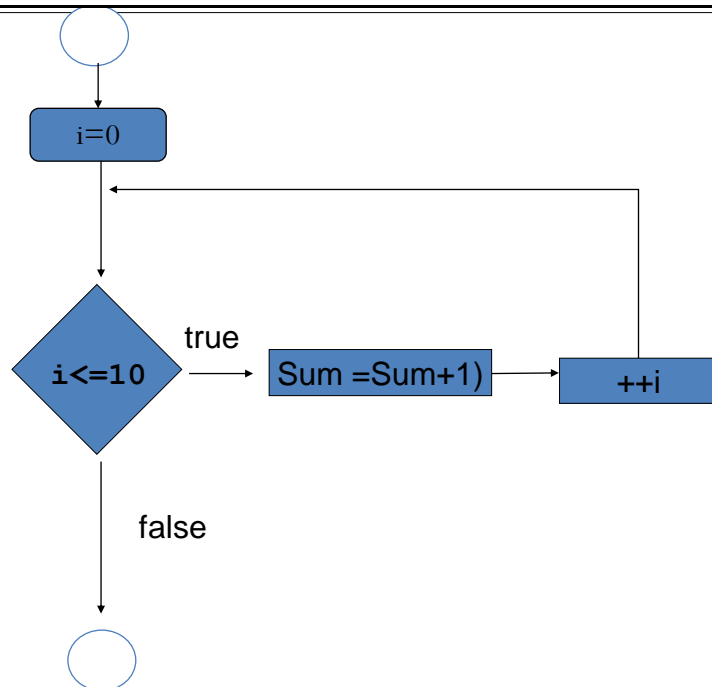
18

## The for Loop Flowchart



19

19



20

20

```

public class ForLoop {
    public static void main(String[] args) {
        int num;
        System.out.println("Number" + "\t\t" + "squer");
        System.out.println("-----");

        for (num=1; num<=5; ++num)
            System.out.println(num + "\t\t" + num*num);
    }
}

```

Output - ForLoop (run) ☒

```

run:
Number      squer
-----
1           1
2           4
3           9
4          16
5          25
BUILD SUCCESSFUL (total time: 0 seconds)

```

21

21

• برنامج يقوم بالعد من (1 إلى 20)

```

package forloop1;
public class ForLoop1 {
    public static void main(String[] args) {
        int a;
        for (a=1;a<=20;++a){
            System.out.print(a + ",");
        }
    }
}

```

```

run:
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,BUILD SUCCESSFUL (total time: 0 seconds)

```

أعد كتابة البرنامج باستخدام for & if للحصول على نفس النتيجة السابقة

22

22

- برنامج يقوم بالعد من (1) إلى (x) على أن يتم ادخال قيمة نهاية العد من لوحة المفاتيح ؟

```
import java.util.Scanner;

public class ForLoop1 {
    public static void main(String[] args) {
        int a;
        Scanner in =new Scanner(System.in);
        int x=in.nextInt();
        for (a=1;a<=x;++a){
            System.out.print(a +",");
        }
    }
}
```

```
run: 6
1,2,3,4,5,6,BUILD SUCCESSFUL (total time: 2 seconds)
```

يتم اولا إدخال قيمة x

23

23

- برنامج يطبع مجموع الأعداد الزوجية من X إلى y ؟

```
package sum1;
import java.util.Scanner;
public class Sum1 {
    public static void main(String[] args) {
        Scanner input =new Scanner(System.in);
        int x = input.nextInt();
        int y = input.nextInt();
        int i,sum=0;
        for(i=x;i<=y;i++){
            if (i%2==0)
                sum=sum+i;
        }
        System.out.println("sum="+ sum);
    }
}
```

```
run:
10
20
sum=90
```



24

أعد كتابة البرنامج السابق بحيث يطبع مجموع الأعداد الفردية.

24

## الحلقات المتداخلة Nested Loops

- Like `if` statements, loops can be nested.
- If a loop is nested, the inner loop will execute all of its iterations for each time the outer loop executes once.

```
for(int i = 0; i < 10; i++)
    for(int j = 0; j < 10; j++)
        loop statements;
```

25

25

## برنامج يطبع جدول الضرب من جدول 1 إلى جدول 10 ؟

```
package for3;
public class For3 {
    public static void main(String[] args) {
        int i,j;
        for (j=1;j<=10;++j){
            System.out.println();
            for (i=1;i<=10;++i)
                System.out.print("\t"+ i*j);
        }
    }
}
```

```
run:
جدول 1 → 1 2 3 4 5 6 7 8 9 10
          2 4 6 8 10 12 14 16 18 20
          3 6 9 12 15 18 21 24 27 30
          4 8 12 16 20 24 28 32 36 40
          5 10 15 20 25 30 35 40 45 50
          6 12 18 24 30 36 42 48 54 60
          7 14 21 28 35 42 49 56 63 70
          8 16 24 32 40 48 56 64 72 80
          9 18 27 36 45 54 63 72 81 90
جدول 10 → 10 20 30 40 50 60 70 80 90 100
```

26

26

```

public static void main(String[] args) {
    for(int i=0 ; i<=5;i++)
    {
        for(int j=0 ; j<i;j++)
            System.out.print("*");
        System.out.println();
    }
}

```

Run

```

*
**
***
****
*****
BUILD SUCCESSFUL

```

عدل في البرنامج أعلاه لتحصل على الشكل التالي

```

*1
**2
***3
****4
*****5

```

27

27

## جملة أقطع break

- تستخدم لتغيير مسار البرنامج والانتقال من داخل الحلقة الى خارج الحلقة.

```

package break1;
public class Break1 {
    public static void main(String[] args) {
        int i,sum=0;
        for(i=0;i<=10;i=i+2){
            sum=sum+i;
            System.out.println("sum="+ sum);
        }
        System.out.println("break in java");
    }
}

```

```

sum=0
sum=2
sum=6
sum=12
sum=20
sum=30
break in Java

```

28

28

## جملة أقطع break

- تستخدم لتغيير مسار البرنامج والانتقال من داخل الحلقة الى خارج الحلقة.

```
package break1;
public class Break1 {
    public static void main(String[] args) {
        int i,sum=0;
        for(i=0;i<=10;i=i+2){
            sum=sum+i;
            if(sum >10)break;
            System.out.println("sum="+ sum);
        }
        System.out.println("break in java");
    }
}
```

```
sum=0
sum=2
sum=6
break in Java
```

29

29

## جملة أقطع break

```
1 package continue1;
2 public class Continue1 {
3     public static void main(String[] args) {
4         double a,s=0;
5         int m,n=7;
6         for(a=1;a<=n;a+=0.5){
7             if(a==2.5){
8                 s+=n;
9                 break;
10            }
11            else
12                s+=a;
13        }
14        System.out.println("s----->>" +s);
15    }
16 }
17 }
```

```
run:
s----->>11.5
```

30

30

## جملة الاستمرار continue

- مهمتها الاستمرار في توجيه التحكم إلى نهاية الحلقة التكرارية والرجوع إلى بدايتها واكمال تنفيذ الحلقة حتى نهايتها.

```
public class Continue1 {
    public static void main(String[] args) {
        double a,s=0;
        int m,n=7;
        for(a=1;a<=n;a+=0.5){
            if(a==2.5){
                s+=n;
                System.out.println("SI---->>" +s);
                continue;
            }
            else
                s+=a;
        }
        System.out.println("S---->>" +s);
    }
}
```

```
SI---->>11.5
```

```
S---->>56.5
```

31

31

```
public class break_Ex {
    public static void main(String[] args) {
        for(int i=0; i<=10; i++)
        {
            if(i==5)break;
            System.out.print(i + " ");
        }
    }
}
```

```
run:
0 1 2 3 4 BUILD SUCCESSFUL (total time: 0 seconds)
```

ما ناتج البرنامج عند استبدال جملة break بجملة الاستمرار continue؟

```
public class break_Ex {
    public static void main(String[] args) {
        for(int i=0; i<=10; i++)
        {
            if(i==5)continue;
            System.out.print(i + " ");
        }
    }
}
```

```
run:
0 1 2 3 4 6 7 8 9 10 BUILD SUCCESSFUL (total time: 0 seconds)
```

32

32



مثال لتوضيح الفرق بين جملة الاستمرار **continue** وجملة القطع **break**  
 أكتب برنامج بلغة جافا يقوم بحساب متوسط درجات الطلبة المسجلين في المجموعة 1  
 (عملي) لمقرر ITGS211 وليكن عددهم 5 طلبة على سبيل المثال.

```
package javaapplications;
/** @author Nahed **/
import java.util.Scanner;
public class WhileExm {
    public static void main(String[] args) {
        Scanner in=new Scanner(System.in);
        int grd,sum=0,std_no=0;
        float Avg;
        while(std_no<5)
        {
            System.out.print(" أدخل درجة الطالب رقم ");
            System.out.println(std_no+1);
            grd=in.nextInt();
            sum+=grd;
            std_no++;
        }
        System.out.println("sum=" +'\t' + sum +'\t' + "std_no=" +'\t' + std_no);
        System.out.println("متوسط درجات الطلبة يساوي" + (Avg=sum/std_no));
    }
}
```

33

33

```
run:
أدخل درجة الطالب رقم : 1
30
أدخل درجة الطالب رقم : 2
40
أدخل درجة الطالب رقم : 3
50
أدخل درجة الطالب رقم : 4
30
أدخل درجة الطالب رقم : 5
50
sum=    200    std_no= 5
متوسط درجات الطلبة يساوي: 40.0
```

34

34

أعد كتابة البرنامج مع التعديل التالي:

أكتب برنامج بلغة جافا يقوم بحساب متوسط درجات الطلبة المسجلين في المجموعة 1 (عملي) لمقرر ITGS211 وليكن عددهم 5 طلبة على سبيل المثال مع الخروج من البرنامج في حال كانت الدرجة (-1).

```
while(std_no<5)
{
    System.out.print(" أدخل درجة الطالب رقم ");
    System.out.println(std_no+1);
    grd=in.nextInt();
    if (grd==-1)
    {
        System.out.println("القيمة المدخلة غير مقبولة");
        break;
    }
    sum+=grd;
    std_no++;
}
System.out.println("sum=" +'\t' + sum +'\t' + "std_no=" +'\t' + std_no);
System.out.println("متوسط درجات الطلبة يساوي" + (Avg=sum/std_no));
```

35

35

```
run:
1 : أدخل درجة الطالب رقم
30
2 : أدخل درجة الطالب رقم
40
3 : أدخل درجة الطالب رقم
-1
القيمة المدخلة غير مقبولة
sum= 70      std_no= 2
متوسط درجات الطلبة يساوي: 35.0
BUILD SUCCESSFUL (total time: 14 seconds)
```

36

36

أعد كتابة البرنامج مع التعديل التالي:  
أكتب برنامج بلغة جافا يقوم بحساب متوسط درجات الطلبة المسجلين في المجموعة 1  
(عملي) لمقرر ITGS211 وليكن عددهم 5 طلبة على سبيل المثال مع عدم قبول  
الدرجة (-1).

```
while(std_no<5)
{
    System.out.print(" أدخل درجة الطالب رقم ");
    System.out.println(std_no+1);
    grd=in.nextInt();
    if (grd==-1)
    {
        System.out.println("القيمة المدخلة غير مقبولة");
        continue;
    }
    sum+=grd;
    std_no++;
}
System.out.println("sum=" +'\t' + sum +'\t' + "std_no=" +'\t' + std_no);
System.out.println("متوسط درجات الطلبة يساوي" + (Avg=sum/std_no));
```

37

37

```
run:
1 : أدخل درجة الطالب رقم
30
2 : أدخل درجة الطالب رقم
40
3 : أدخل درجة الطالب رقم
-1
القيمة المدخلة غير مقبولة
أدخل درجة الطالب رقم : 3
50
4 : أدخل درجة الطالب رقم
-1
القيمة المدخلة غير مقبولة
أدخل درجة الطالب رقم : 4
|
```

38

38

<b>For</b>	<pre>for(int x = 10; x &lt; 20; x = x+1) {     System.out.print("value of x : " + x );     System.out.print("\n"); }</pre>	<b>Output</b>
<b>while</b>	<pre>int x = 10; while( x &lt; 20 ) {     System.out.print("value of x : " + x );     x++;     System.out.print("\n"); }</pre>	
<b>Do_while</b>	<pre>int x = 10; do{     System.out.print("value of x : " + x );     x++;     System.out.print("\n"); }while( x &lt; 20 );</pre>	

value of x : 10  
 value of x : 11  
 value of x : 12  
 value of x : 13  
 value of x : 14  
 value of x : 15  
 value of x : 16  
 value of x : 17  
 value of x : 18  
 value of x : 19

39

39

انتهت المحاضرة



الفائدة تبدأ بالسؤال ثم النقاش ...

40

40