



محاضرات العملي لمقرر

البرمجة الشيئية

ITGS211

الدرس العملي رقم (2)

إعداد الأستاذة :ملاك ددش



الحلقات في جاافا:

نستخدم الحلقات (Loops) بهدف تكرار نفس الكود عدة مرات

أنواع الحلقات في جاافا

1. الحلقة for

طريقة إستخدامها

```
for( initialization; condition; increment أو decrement )  
{  
    // statements  
}
```

مثال حول الحلقة for:

```
public class Main {  
  
    public static void main(String[] args) {  
  
        // هنا قمنا بإنشاء حلقة for تتألف من 10 دورات. في كل دورة تطبع قيمة العداد المستخدم فيها  
        for( int i=1; i<=10; i++ )  
        {  
            System.out.println( i );  
        }  
  
    }  
  
}
```

سنحصل على النتيجة التالية عند التشغيل

```
1  
2  
3  
4  
5  
6  
7  
8  
9  
10
```

2. الحلقة **while**:

نستخدم الحلقة **while** إذا كنا نريد تنفيذ الكود عدة مرات، و لكننا لا نعرف كم مرة بالتحديد. هذه الحلقة تتوقف عن تكرار نفسها إذا لم يعد الشرط الموضوع فيها يتحقق.

هنا كأننا نقول: "طالما أن الشرط لا يزال يتحقق إستمر في تكرار الكود".

طريقة إستخدامها:

```
initialization;  
  
while( condition )  
{  
    // statements  
  
    increment أو decrement;  
}
```

مثال حول الحلقة **while** :

```
public class Main {  
  
    public static void main(String[] args) {  
  
        // هنا قمنا بتعريف المتغير الذي استخدمناه كعداد في الحلقة  
        int i=1;  
  
        // هنا أنشأنا حلقة while تظل تنفذ الأوامر الموضوعه فيها طالما أن قيمة العدد لا تزال أصغر أو تساوي 10  
        while( i<=10 )  
        {  
            // في كل دورة سيتم طباعة قيمة العداد ثم إضافة 1 عليها  
            System.out.println( i );  
            i++;  
        }  
  
    }  
  
}
```



سنحصل على النتيجة التالية عند التشغيل

```
1  
2  
3  
4  
5  
6  
7  
8  
9  
10
```

الحلقة do while :

نستخدم الحلقة while إذا كنا نريد تنفيذ الكود عدة مرات، و لكننا لا نعرف كم مرة بالتحديد. هذه الحلقة تتوقف عن تكرار نفسها إذا لم يعد الشرط الموضوع فيها يتحقق.

هنا كأننا نقول: "نفذ الكود و إن كان الشرط لا يزال يتحقق فقم بتنفيذه من جديد".

إذاً الفرق الوحيد بينها وبين الحلقة while أنها تنفذ مرة واحدة على الأقل لأنها تتأكد من الشرط بعد تنفيذ الأوامر و ليس قبلهم.

طريقة إستخدامها :

```
initialization;  
  
do{  
    // statements  
    increment أو decrement;  
}  
While( condition );
```



مثال حول الحلقة do while :

```
public class Main {  
  
    public static void main(String[] args) {  
  
        // هنا قمنا بتعريف المتغير الذي استخدمناه كعداد في الحلقة في  
        int i=1;  
  
        // هنا أنشأنا حلقة while تظل تنفذ الأوامر الموضوعة فيها طالما أن قيمة العدد لا تزال أصغر أو تساوي 10  
        do  
        {  
            // في كل دورة سيتم طباعة قيمة العداد ثم إضافة 1 عليها  
            System.out.println( i );  
            i++;  
        }  
        while( i<=10 );  
  
    }  
  
}
```

سنحصل على النتيجة التالية عند التشغيل

```
1  
2  
3  
4  
5  
6  
7  
8  
9  
10
```

جمل التحكم مع الحلقات في جافا:

نستخدم جمل التحكم (Control Statements) للتحكم في سير تنفيذ الحلقات و مع جملة الشرط switch.

جملة التحكم break في جافا

تعريف الجملة break:

الجملة break تستخدم في الحلقات و في الجملة switch. بمجرد ان تنفذ الجملة break فإنها توقف ال scope بأكماله و تخرج منه و تمسحه من الذاكرة ثم تنتقل للكود الذي يليه في البرنامج.

مثال حول جملة التحكم break:

في المثال التالي قمنا بتعريف حلقة كانت ستطبع جميع الأرقام من 1 إلى 10 لولا أننا استخدمنا الجملة break لجعل الحلقة تتوقف عندما تصبح قيمة العداد تساوي 6

```
public class Main {  
  
    public static void main(String[] args) {  
  
        // هنا قمنا بإنشاء حلقة for تتألف من 10 دورات. في كل دورة تطبع قيمة العداد المستخدم فيها  
        for( int i=1; i<=10; i++ )  
        {  
  
            // في كل دورة سيتم فحص قيمة العداد و بمجرد أن تصبح تساوي 6 سيتم إيقاف الحلقة نهائياً  
            if( i == 6 ) {  
                break;  
            }  
  
            System.out.println( i );  
        }  
  
    }  
}
```

سنحصل على النتيجة التالية عند التشغيل

```
1  
2  
3  
4  
5
```

إذاً الجملة break جعلت الحلقة تتوقف عندما أصبحت قيمة العداد تساوي 6

جملة التحكم continue في جافا

تعريف الجملة continue

أمثلة حول جملة التحكم continue

في المثال التالي قمنا بتعريف حلقة تطبع جميع الأرقام من 1 إلى 10 ما عدا الرقم 3. استخدمنا الجملة continue لجعل الحلقة تتجاوز الدورة الثالثة في الحلقة. أي لن يتم تنفيذ أمر الطباعة عندما تصبح قيمة العداد 3.

```
public class Main {  
  
    public static void main(String[] args) {  
  
        // هنا قمنا بإنشاء حلقة for تتألف من 10 دورات. في كل دورة تطبع قيمة العداد المستخدم فيها  
        for (int i=1; i<=10; i++)  
        {  
            // كل دورة سيتم فحص قيمة العداد. عندما تصبح تساوي 3 سيتم الانتقال إلى الدورة التالية في الحلقة بدون تنفيذ أمر الطباعة الموضوع بعدها  
            if (i == 3) {  
                continue;  
            }  
  
            System.out.println(i);  
        }  
  
    }  
}
```

سنحصل على النتيجة التالية عند التشغيل

```
1  
2  
4  
5  
6  
7  
8  
9  
10
```

إذاً الجملة continue جعلت الحلقة تتجاوز الدورة الثالثة, لذلك لم تطبع الرقم 3 لأنها لم تنفذ أمر الطباعة في الدورة الثالثة.



في المثال التالي قمنا بتعريف حلقة تطبع جميع الأرقام المفردة من 1 إلى 10 .
إستخدمنا الجملة `continue` لجعل الحلقة تتجاوز كل دورة تكون فيها قيمة العداد `i` عبارة عن عدد مزدوج.

المثال الثاني :

```
public class Main {  
  
    public static void main(String[] args) {  
  
        // هنا قمنا بإنشاء حلقة for تتألف من 10 دورات. في كل دورة تطبع قيمة العداد المستخدم فيها  
        for (int i=1; i<=10; i++)  
        {  
            // دورة سيتم فحص قيمة العداد. في حال كانت مزدوجة سيتم الانتقال إلى الدورة التالية في الحلقة بدون تنفيذ أمر الطباعة الموضوع بعدها  
            if (i%2 == 0) {  
                continue;  
            }  
  
            System.out.println(i);  
        }  
  
    }  
}
```

سنحصل على النتيجة التالية عند التشغيل

```
1  
3  
5  
7  
9
```

إذاً الجملة `continue` جعلت الحلقة تتجاوز كل دورة كانت فيها قيمة العداد عبارة عن عدد مزدوج.



الشروط في جافا:

أنواع جمل الشرط في جافا:

- جملة الشرط if
- جملة الشرط else
- جملة الشرط else if

جمل الشرط بشكل عام :

```
if ( condition )
{
    // إذا كان الشرط صحيحاً نفذ هذا الكود
}

else if ( condition )
{
    // إذا كان الشرط صحيحاً نفذ هذا الكود
}

else
{
    نفذ هذا الكود في حال لم يتم التعرف على الكود في أي شرط
}
```

المثال الأول:

إذا كانت قيمة المتغير **S** أكبر من 5 سيتم طباعة الجملة: **S is bigger than 5**.

```
public class Main {  
  
    public static void main(String[] args) {  
  
        int S = 0;  
  
        if( S > 5 )  
        {  
            System.out.print("S is bigger than 5");  
        }  
  
    }  
  
}
```

سنحصل على النتيجة التالية عند التشغيل

المثال الثاني:

إذا كانت قيمة المتغير **S** أكبر من 5 سيتم طباعة الجملة: **S is bigger than 5**.

```
public class Main {  
  
    public static void main(String[] args) {  
  
        int S = 30;  
  
        if( S > 5 ) {  
            System.out.print("S is bigger than 5");  
        }  
  
    }  
  
}
```

سنحصل على النتيجة التالية عند التشغيل.

```
S is bigger than 5
```



جملة الشرط else

في اللغة العربية تعني "أي شيء آخر". وهي تستخدم فقط في حال كنا نريد تنفيذ كود معين في حال كانت نتيجة جميع الشروط التي قبلها تساوي **false**.

يجب وضعها دائماً في الأخير, لأنها تستخدم في حال لم يتم تنفيذ أي جملة شرطية قبلها.

إذاً, إذا نفذ البرنامج الجملة **if** أو **else if** فإنه سيتجاهل الجملة **else**.
و إذا لم ينفذ أي جملة من الجمل **if** و **else if** فإنه سينفذ الجملة **else**.

المثال الأول

إذا كانت قيمة المتغير **S** تساوي 5 سيتم طباعة الجملة: **S is equal 5**.

إذا كانت قيمة المتغير **S** لا تساوي 5 سيتم طباعة الجملة: **S is not equal 5**.

```
public class Main {  
  
    public static void main(String[] args) {  
  
        int S = 5;  
  
        if( S == 5 ) {  
            System.out.print("S is equal 5");  
        }  
  
        else {  
            System.out.print("S is not equal 5");  
        }  
  
    }  
  
}
```

سنحصل على النتيجة التالية عند التشغيل.

```
S is equal 5
```

المثال الثاني

إذا كانت قيمة المتغير **S** تساوي 5 سيتم طباعة الجملة: **S is equal 5**.

إذا كانت قيمة المتغير **S** لا تساوي 5 سيتم طباعة الجملة: **S is not equal 5**.

```
public class Main {  
  
    public static void main(String[] args) {  
  
        int S = 20;  
  
        if( S == 5 ) {  
            System.out.print("S is equal 5");  
        }  
  
        else {  
            System.out.print("S is not equal 5");  
        }  
  
    }  
}
```

سنحصل على النتيجة التالية عند التشغيل

```
S is not equal 5
```

جملة الشرط else if

جملة **else if** تستخدم إذا كنت تريد وضع أكثر من احتمال (أي أكثر من شرط).

جملة أو جمل الـ **else if** يوضعون في الوسط، أي بين الجملتين **if** و **else**.

مثال

1. إذا كانت قيمة المتغير **number** تساوي 1 سيتم طباعة الكلمة **one**
2. إذا كانت قيمة المتغير **number** تساوي 2 سيتم طباعة الكلمة **two**
3. إذا كانت قيمة المتغير **number** تساوي 3 سيتم طباعة الكلمة **three**
4. إذا كانت قيمة المتغير **number** أكبر أو تساوي 4 سيتم طباعة الجملة **four or greater**
5. إذا كانت قيمة المتغير **number** أصغر من 0 سيتم طباعة الجملة **negative number**



```
public class Main {  
  
    public static void main(String[] args) {  
  
        int number = 3;  
  
        if( number == 1 ) {  
            System.out.print("one");  
        }  
  
        else if( number == 2 ) {  
            System.out.print("two");  
        }  
  
        else if( number == 3 ) {  
            System.out.print("three");  
        }  
  
        else if( number >= 4 ) {  
            System.out.print("four or greater");  
        }  
  
        else {
```



```
System.out.print("negative number");
```

```
}
```

```
}
```

```
}
```

سنحصل على النتيجة التالية عند التشغيل

```
three
```



الجملة switch في جافا

تعريف الجملة switch

switch نستخدمها إذا كنا نريد إختبار قيمة متغير معين مع لائحة من الإحتمالات نقوم نحن بوضعها فيها, و إذا تساوت هذه القيمة مع أي إحتمال وضعناه سنتنفذ الأوامر التي وضعناها في هذا الإحتمال فقط.

كل إحتمال نضعه يسمى **case**.

أنواع المتغيرات التي يمكن إختبار قيمتها باستخدام هذه الجملة هي:

enum - char - short - byte - int - String

طريقة تعريفها:

```
switch(expression) {  
  
    case value:  
        // Statements  
        break;  
  
    case value:  
        // Statements  
        break;  
  
    default:  
        // Statements  
        break;  
  
}
```



المثال الأول

سنقوم باختبار قيمة المتغير x و الذي نوعه `int`.
سنضع عدة حالات و كل حالة تطبع شيء معين.

```
public class Main {  
  
    public static void main(String[] args) {  
  
        int x = 2;  
  
        switch( x ) { // اختبار قيمة المتغير x  
  
            case 1: // في حال كانت تساوي 1 سيتم تنفيذ أمر الطباعة الموضوع فيها  
                System.out.println("x contain 1");  
                break;  
  
            case 2: // في حال كانت تساوي 2 سيتم تنفيذ أمر الطباعة الموضوع فيها  
                System.out.println("x contain 2");  
                break;  
  
            case 3: // في حال كانت تساوي 3 سيتم تنفيذ أمر الطباعة الموضوع فيها  
                System.out.println("x contain 3");  
                break;  
  
        }  
    }  
}
```



default: // في حال كانت لا تساوي أي قيمة من القيم الموضوعه سيتم تنفيذ أمر //
الطباعة الموضوع فيها

```
System.out.println("x contain a different value");
```

```
}
```

```
}
```

```
}
```

سنحصل على النتيجة التالية عند التشغيل

```
x contain 2
```



المثال الثاني

سنقوم باختبار قيمة المتغير **x** و الذي نوعه **int**.
سنضع عدة حالات و كل حالة تطبع شيء معين.

```
public class Main {
```

```
public static void main(String[] args) {
```

```
int x = 5;
```

```
switch( x ) { // إختبر قيمة المتغير x
```

```
case 1: // في حال كانت تساوي 1 سيتم تنفيذ أمر الطباعة الموضوع فيها //
```

```
System.out.println("x contain 1");
```



```
break;
```

```
case 2: // في حال كانت تساوي 2 سيتم تنفيذ أمر الطباعة الموضوع فيها //
```

```
System.out.println("x contain 2");
```

```
break;
```

```
case 3: // في حال كانت تساوي 3 سيتم تنفيذ أمر الطباعة الموضوع فيها //
```

```
System.out.println("x contain 3");
```

```
break;
```

```
default: // في حال كانت لا تساوي أي قيمة من القيم الموضوعه سيتم تنفيذ أمر  
الطباعة الموضوع فيها
```

```
System.out.println("x contain a different value");
```

```
}
```

```
}
```

```
}
```

سنحصل على النتيجة التالية عند التشغيل

```
x contain a different value
```





مثال

سنقوم باختبار قيمة المتغير **x** و الذي نوعه **int**.

سنضع ثلاث حالات ينفذون نفس الأوامر.

```
public class Main {  
  
    public static void main(String[] args) {  
  
        int x = 3;  
  
        switch( x ) { // اختبار قيمة المتغير x  
  
            case 1: // في حال كانت تساوي 1 أو 2 أو 3 سيتم تنفيذ أمر الطباعة  
            case 2:  
            case 3:  
                System.out.println("x contain 1 or 2 or 3");  
                break;  
  
            default: // في حال كانت لا تساوي أي قيمة من القيم الموضوعة سيتم تنفيذ أمر الطباعة الموضوع فيها  
                System.out.println("x contain a different value");  
  
        }  
  
    }  
  
}
```

سنحصل على النتيجة التالية عند التشغيل

```
x contain 1 or 2 or 3
```