

OOP

Java

Lec 3

Mohammed

المعاملات

OPERATORS

المعاملات

المعاملات هي عبارته عن رموز خاصه تستخدم في :
الدوال الرياضية
عمليات الإسناد
المقارنات المنطقية

مثال :

Examples of operators:

$3 + 5$ // uses + operator

$14 + 5 - 4 * (5 - 3)$ // uses +, -, * operators

مجموعات المعاملات

هناك خمس مجاميع للمعاملات وهي :

المعاملات الرياضية

وهي معاملات الجمع والطرق والقسمه والضرب وغيرها

+	add
-	subtract
*	multiply
/	divide
%	modulo (remainder)
^	exponent (to the power of)

معاملات الإسناد

معامل الإسناد هو =، ويعمل على إسناد قيمه للمتغير

`var = expr ;`

Examples:

`x = x + 5;` is equivalent to `x += 5;`

`y = y * 7;` is equivalent to `y *= 7;`

معاملات الزيادة والنقصان

Increment/Decrement Operators

`++` هو معامل الزيادة، `--` هو معامل النقصان

For example:

`i = i + 1;` can be written as:

`++i;` or `i++;`

Decrement :

`i = i - 1;` can be written as:

`--i;` or `i--;`

المعاملات المقارنة

المعاملات المنطقية تقارن بين قيمتين ، والنتاج هو قيمه منطقية انا true أو false اعتمادا على الجدول الاتي :

OperationIs true when
$a > b$	a is greater than b
$a \geq b$	a is greater than or equal to b
$a == b$	a is equal to b
$a != b$	a is not equal to b
$a \leq b$	a is less than or equal to b
$a < b$	a is less than b

المعاملات المقارنة

```
int x = 3;
```

```
int y = 5;
```

```
boolean result;
```

```
1) result = (x > y);
```

now result is assigned the value **false** because
3 is not greater than 5

```
2) result = (15 == x*y);
```

now result is assigned the value **true** because the
product of
3 and 5 equals 15

المعاملات المنطقية

تتعامل مثل معاملات المقارنة حسب الجدول الاتي :

Symbol	Name
&&	AND
	OR
!	NOT

```
boolean x = true;  
boolean y = false;  
boolean result;  
Let result = (x && y);
```

now result is assigned the value **false**

دالة الإخراج في لغة الجافا

System.out.print

وهي من الدوال الهامة في لغة الجافا وهي تقوم بطباعة المخرجات سواء كانت عددية أو حرفية .

```
public class HelloEgypt
{

    public static void main ( String [] s)
    {
        System.out.print("Hello World ");

    }

}
```

دالة الإخراج في لغة الجافا

الحرف الخاص	الوصف
\n	سطر جديد. يضع المؤشر في بداية السطر التالي
\t	مسافة أفقية. تحريك المؤشر مسافة معينة إلى النقطة التالية في السطر
\r	carriage return. يضع المؤشر في بداية السطر الحالي ولا يتقدم إلى السطر التالي ، وأي حرف يطبع يتم طباعته على حرف سابق تم كتابته في نفس السطر
\\	شرطة خلفية. إظهار "\" في الخرج
\"	علامة تنصيص مزدوجة. إظهار علامة التنصيص المزدوجة

الادخال في لغه الجافا

عند الحاجه لعمل برنامج تفاعلي، فاننا نحتاج إلى عمليه ادخال البيانات من المستخدمين.

تتم عملية الادخال في الجافا عن طريق استدعاء كلاسات ودوال الادخال من حزم الادخال

هذه الكلاسات والحزم المكتبية تم توفيرها في الجافا عن
Java API (application programming
interface)

الادخال في لغه الجافا

Scanner

هو اسم الكلاس الذي يقوم بعملية الادخال.
وهو موجود في الحزمة البرمجيه `java.util`

فعند استخدام احد هذه الكلاسات لازم أن يتم استدعاء
هذه الحزم في أول البرنامج
يتم هذا الاستدعاء لكل بكج أو حزمة برمجية باستثناء
الخدمة `java.lang` ، فإنه يتم استدعائها تلقائياً

الادخال في لغه الجافا

الاستدعاء بطريقتين: اما استدعاء كل الكلاسات في الحزمة
عن طريق *

`import java.util.*;` <-this statement provides
access to every class in the “util” package.

أو استدعاء فقط الكلاس المحدد واستخدام الدوال الموجودة
فيه فقط

`import java.util.Scanner;` <-this statement
provides access to only the “Scanner” class
that’s in the “util” package.

الادخال في لغه الجافا

```
import java.util.Scanner;
public class GetANumber
{
    public static void main(String args[])
    {
        Scanner input= new Scanner(System.in);
    }
}
```

```
import java.util.Scanner;
public class GetANumber
{
    public static void main(String args[])
    {
        Scanner keyboard = new Scanner(System.in);

        int num = keyboard.nextInt();
    }
}
```

```
import java.util.Scanner;
public class GetAName
{
    public static void main(String args[])
    {
        Scanner keyboard = new Scanner(System.in);

        String name = keyboard.next();
    }
}
```


Scanner

frequently used methods

Name	Use
nextInt()	returns the next int value
nextDouble()	returns the next double value
nextFloat()	returns the next float value
nextLong()	returns the next long value
nextByte()	returns the next byte value
nextShort()	returns the next short value
next()	returns the next one word String
nextLine()	returns the next multi word String

```
import java.util.Scanner;
```

عبارات التحكم والحلقات

LOOPS

If Statements

تعمل على المقارنه بين القيم، وارجاع قيمة منطقية ، فإذا كانت عبارة الشرط صحيحة يتم الدخول إلى عبارة if ، وإذا كانت خاطئة فإنه لا يتم الدخول إليها

الصيغة العامة

```
if (expression) {  
    statement;  
}  
next_statement;
```

مثال

```
if (grade == 'A')  
    System.out.println("You got an A.");  
  
else if (grade == 'B')  
    System.out.println("You got a B.");  
  
else if (grade == 'C')  
    System.out.println("You got a C.");  
  
else  
    System.out.println("You got an F.");
```

Switch Statement

For example:

```
switch (expression) {
```

```
    case value1:  
        statement1;  
        break;
```

```
    case value2:  
        statement2;  
        break;
```

```
    default:  
        default_statement;  
        break;
```

```
}
```

Switch Statement

```
switch (grade) {  
    case 'A':  
        System.out.println("You got an A.");  
        break;  
    case 'B':  
        System.out.println("You got a B.");  
        break;  
    case 'C':  
        System.out.println("You got a C.");  
        break;  
    default:  
        System.out.println("You got an F.");  
}
```

الحلقات Loops

الحلقات تعمل على تنفيذ الكود مرات عديدة للحصول على النتيجة المحددة

يوجد ثلاثة أنواع للحلقات في الجافا

There are three types of loops in Java:

for loops

while loops

Do-while loops

For loop

الصيغة العامة لحلقة التكرار for هي :

```
for (initialization_expression;  
    loop_condition; increment_expression) {  
  
    //statement  
  
}
```


Example

```
int limit = 5;  
int sum = 0;  
for(int i = 1; i<=limit; i++){
```

```
/* initialization_expression  
   loop_condition  
   increment_expression */  
  
   // sum = sum + i;  
   sum += i;  
}
```

Double
tap
to add
text

i = 1	sum = 1
i = 2	sum = 3
i = 3	sum = 6
i = 4	sum = 10
i = 5	sum = 15
i = 6	

- What is the value of sum ? **15**

The while Statement

الصيغة العامه لها :

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

إذا كان الشرط صحيح، فيتم تنفيذ الحلقة، إذا لم يكن صحيح فيتم تجاهل الحلقة والتنفيذ من أول تعليمة بعد الحلقة
while

Example

```
int i = 1;
```

```
while ( i < 10 ) {  
    System.out.println(i);  
    i++;  
}
```

```
System.out.println("done");
```

- What is the output?

~~1~~, ~~2~~, ~~3~~, ~~4~~, ~~5~~, ~~6~~, ~~7~~, ~~8~~, ~~9~~, ~~10~~

1
2
3
4
5
6
7
8
9
done

Java do...while loop

يتم تنفيذ هذه الحلقة مره واحده على الاقل

```
int i = 0;  
do {  
    System.out.println(i);  
    i++;  
}  
while ( i < 10 );
```

for, while, and do-while

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

```
do {  
    // loop body  
    statement(s);  
} while (loop-continuation-condition)
```

```
for (initial-action; loop-continuation-condition; action-after-each-iteration) {  
    // loop body  
    statement(s);  
}
```

المصفوفات في الجافا

ARRAYS IN JAVA

المصفوفات

عبارة عن مجموعة من القيم المرتبطة مع بعضها البعض
وتكون من نفس النوع

مثال

المصفوفة من النوع الرقمي `int`، تخزن فقط بيانات من
النوع الرقمي

المصفوفة تمتلك نوع بيانات، واسم، وحجم

يتم الإشارة لكل قيمة مخزنه في المصفوفة باسم عنصر
element
ويتم الوصول إلى هذه القيم عن طريق العنوان index

تعريف المصفوفات

الصيغة العامة لتعريف المصفوفة هي :

```
datatype[] name;
```

For example :

```
int[] prices;
```

```
String[] people;
```

Can alternatively use the form:

```
datatype name[];
```

```
int prices[];
```

ويتم تحديد حجم المصفوفة كالاتي :

```
name = new type[size];
```

```
prices = new int[3];
```

```
people = new String[4];
```

استخدام المصفوفات

يتم الدخول الي اي عنصر في المصفوفة واسناد القيم عن طريق فهرس المصفوفة أو index

```
prices[0] = 6;  
prices[1] = 80;  
prices[2] = 10;
```

```
String[] people;  
people = new String[4];  
people[0] = "Alice";  
people[1] = "Bilha";  
people[2] = "Chris";  
people[3] = "David";
```

Any Questions?