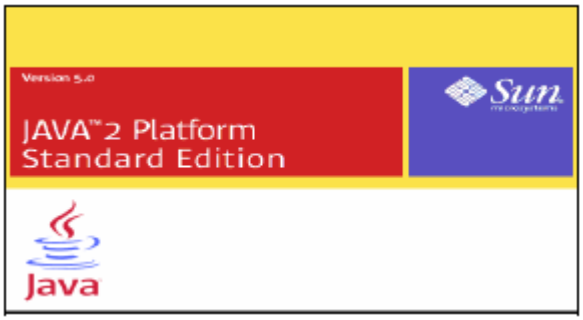
الباب الأول

البرمجة بإستخداميباستخدام لغة الجافا

JAVA



البرمجة بإستخداميباستخدام لغة الجافا
Java Language

مقدمة :

تعتبر لغة الجافا من اللغات القوية جداً في مجال انشاءإنشاء التطبيقات المختلفة سواء كانت هذه التطبيقات

تعمل منفردة على اجهزةأجهزة الكمبيوتر الشخصي أو تطبيقات الأنترنتإنترنت أو التطبيقات المختلفة للأجهزةأجهزة المحمولة , مثل الموبايل والمفكرات الألكترونيةالإلكترونية وهكذا.

-ولقد قامت شركة صن (Sun Microsystems) Sun microsystems)) بأختراعياأختراع وتطوير هذه اللغة.اللغة. وأصبحت شركةصن مملوكة لشركة أوراكل وبالتالي انتقلت ملكية الجافا لأوراكل . وكان الهدف عند أختراعاختراع لغة الجافا هو عمل لغة **قادرة على** برمجة نظم التشغيل لجميع الأجهزةالأجهزة من حاسبات عملاقة (mainframes) إلى الأجهزةالأجهزة الصغيرة مثل مشغلات MP3 ولقد اختارت الشركة صورة فنجان القهوة لتمثيل هذه اللغة .

1.1 أسس البرمجة بإستخداميباستخدام لغة الجا

قبل البدء في عملية البرمجة (أي كتابة البرنامج المطلوب تنفيذه) بلغة الجافا لابد من توافر العدة اللازمة (Tool Kit).

وهذه العدة عبارة عن البرامج اللازمة لعملية كتابة البرنامج نفسه ونقول أننا كتبنا برنامج بلغة الجافا.

بعد ذلك تأتي عملية الترجمة لهذا البرنامج وهي ما نطلق عليها عملية الترجمة (compiling). (compiling)

والحقيقة فأنه يوجد أكثر من طريقة لكتابة برامج الجافا وترجمتها نوجز منها :

1-

2- استعمال المكتبة (**JDK**) وهي اختصار JAVA DEVELOPMENTE KIT من انتاج شركة صن مع أي محرر نصوص وليكن برنامج NotePad الموجود في الويندوز.

3-

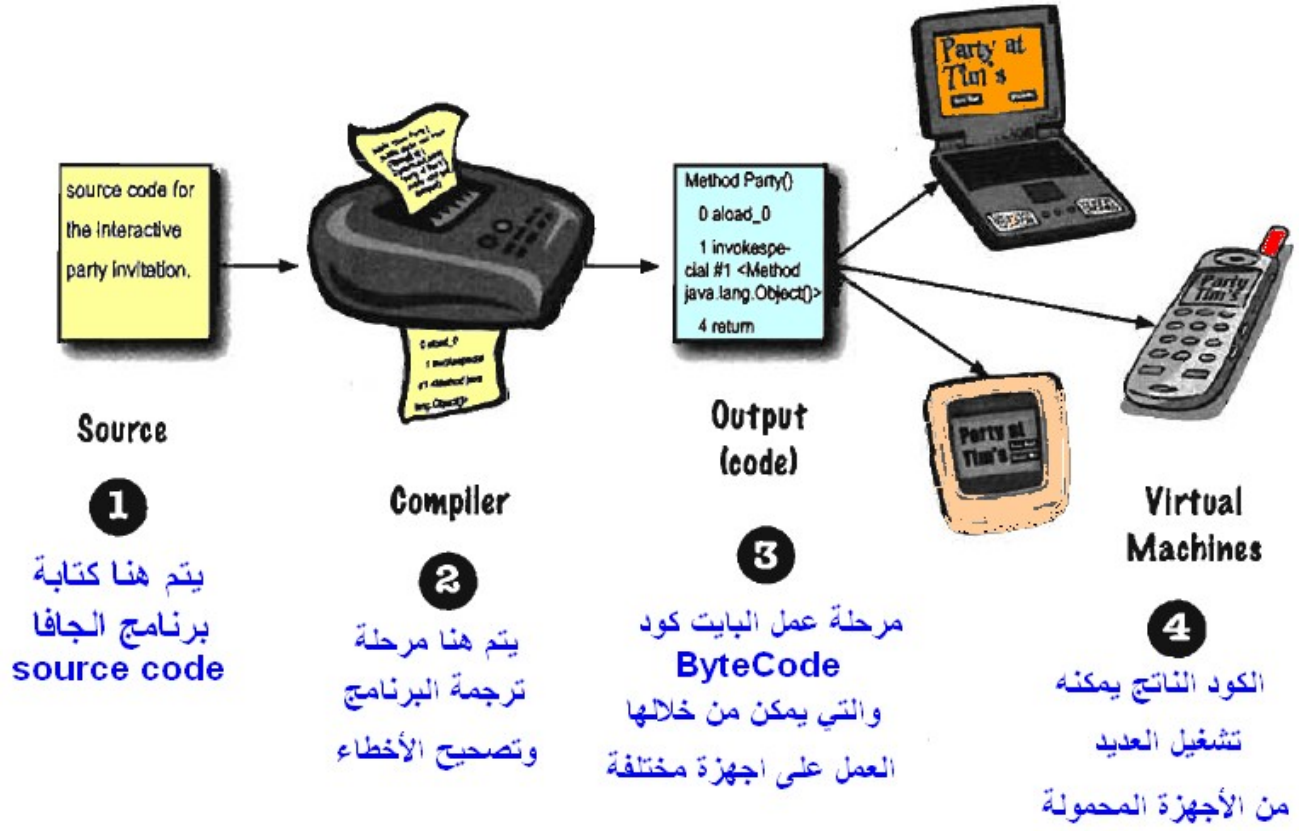
4- استعمال برامج وسيطة تسهل عملية الكتابة والترجمة وتصحيح الأخطاء مثل برنامج

5-

6- (..... - NetBeans – Jcreator). (... NetBeans – Jcreator).

وسوف نتناول في الجزء الخاص بالمعمل كيفية تثبيتكيفية تثبيت هذه البرامج على جهاز الحاسب وكيفية التعامل معها.

والشكل (1-1) يبين كيفية عمل لغة الجافا .



شكل (1-1)

ويوجد عدة نسخ للغة الجافا هي :-

- J2SE: هي اختصار ل Java 2 Standard Edition يتم يتم من خلالها دراسة اللغة وإنشاء التطبيقات المختلفة لتشغيلها على جهاز الحاسب desktop Application)-)-). وسوف تكون هي موضوع دراستنا في هذا الكتاب.الكتاب
- J2EE:هي: هي اختصار ل Java 2 Enterprise Edition وهي وهي تزودنا بالتطبيقات الكبيرة على مستوى الشركات الكبيرة.

- J2ME:هي: هي اختصار ل Java 2 Micro Edition فهي تخص ال الأجهزة اللاسلكية (wireless devices) (wireless devices) بشكلبشكل عام يعني على أجهزةأجهزة المحمول وغيرها.

1-1-1مميزات لغة الجافا

1- لغة الجافا غير مرتبطة بأنظمة التشغيل المختلفة Java Is Platform Independent .

2- تعتمد على أأسلوب برمجة الأهداف Object Oriented Programming .

3-انشاءإنشاء برامج ذات واجهة مستخدم .

4- تصميم برمجيات تستفيد من كل مميزات الأنترنتإنترنت Java Applet .

وفيما يلي شرح لأهم مميزات لغة الجافا كما ذكرناها في النقاط السابقة :

1- لغة الجافا غير مرتبطة بأنظمة التشغيل المختلفة Java Is Platform Independent

ومعنى ذلك إنه يمكن نقل البرامج (المكتوبة بلغة الجافا) بسهولة من نظام تشغيل إلى آخر.

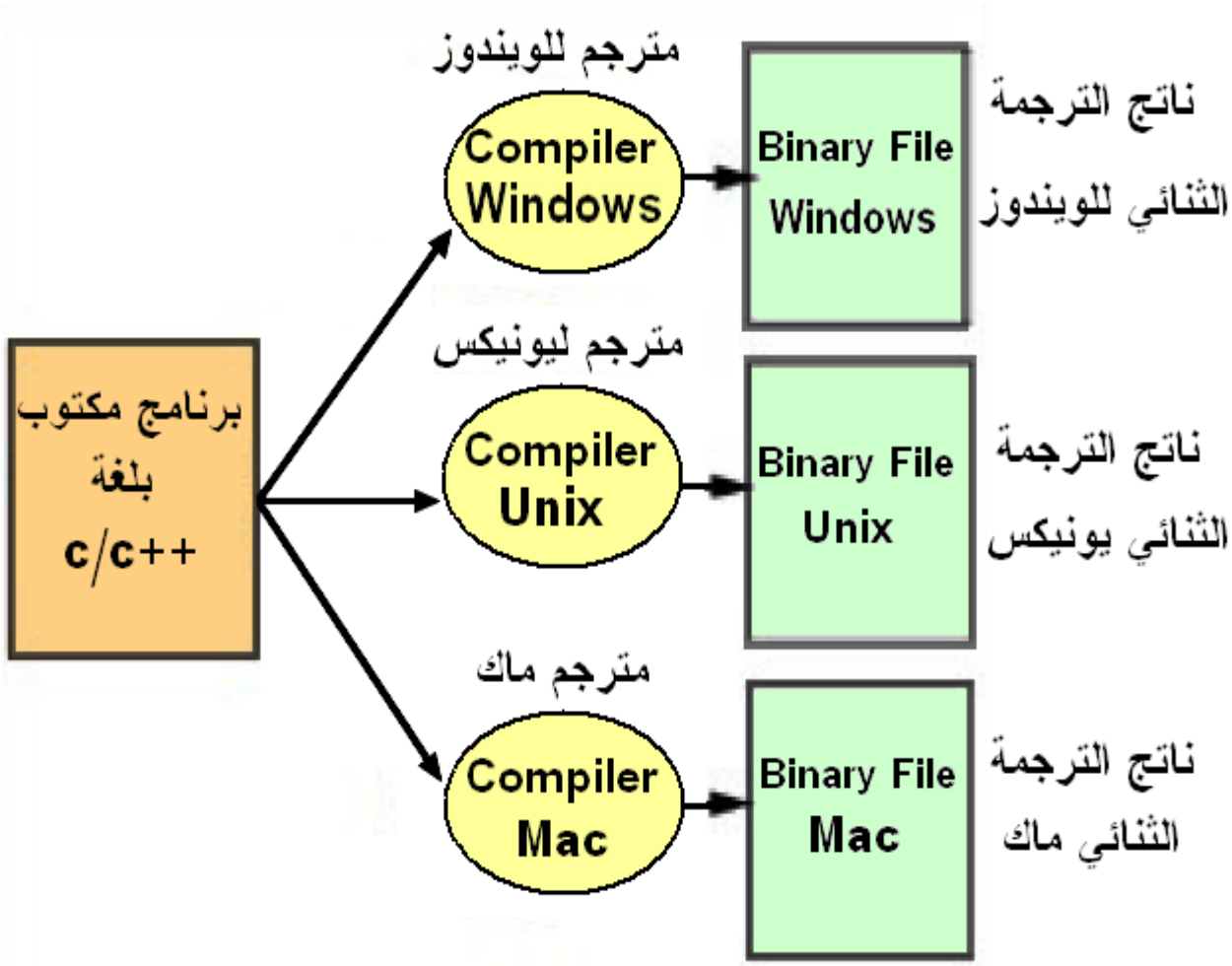
وفي المقابل يمكن القول إنه **لا** يمكن تشغيل برنامج WORD مثلا والخاص بنظام تشغيل ويندوز(WINDOWS) على جهاز حاسب آخر يعمل بنظام تشغيل مختلف مثل يونيكس (UNIX) أو نظام تشغيل لينيكس (LINUX) أو ايأي نظام تشغيل آخر غيرنظامغير نظام WINDOWS والمستخدم مع أجهزةالحاسباتأجهزة الحاسبات المختلفة. ويرجع ذلك لأن برنامج WORD بشكل عام مكتوب بلغة

(C++/C) والتي تعطي ملف من نوع EXEنوع EXE خلال عملية تسمى عملية الترجمة COMPILATION وبذلك يكون الملفيكون الملف الناتج مرتبطاً ارتباطاً كلياً بنظام التشغيل.التشغيل.

أما بالنسبة للغة الجافا فالوضع مختلف حيث يوجد وسيط بين البرنامج وبين نظام التشغيل وهذا الوسيط يسمى (Byte Code Interpreter) أي الترجمة على مستوى البايت.

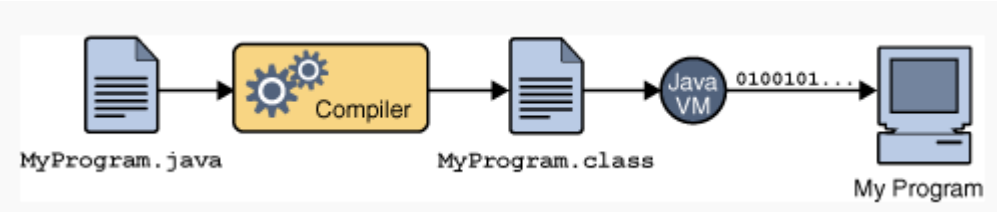
وكذلك يمكن تسميته بالآلة التخيلية للجافا (Java Virtual Machine).

ويوضح الشكل (1-2) خطوات تشغيل برنامج مكتوب بلغة (cأو c++) .

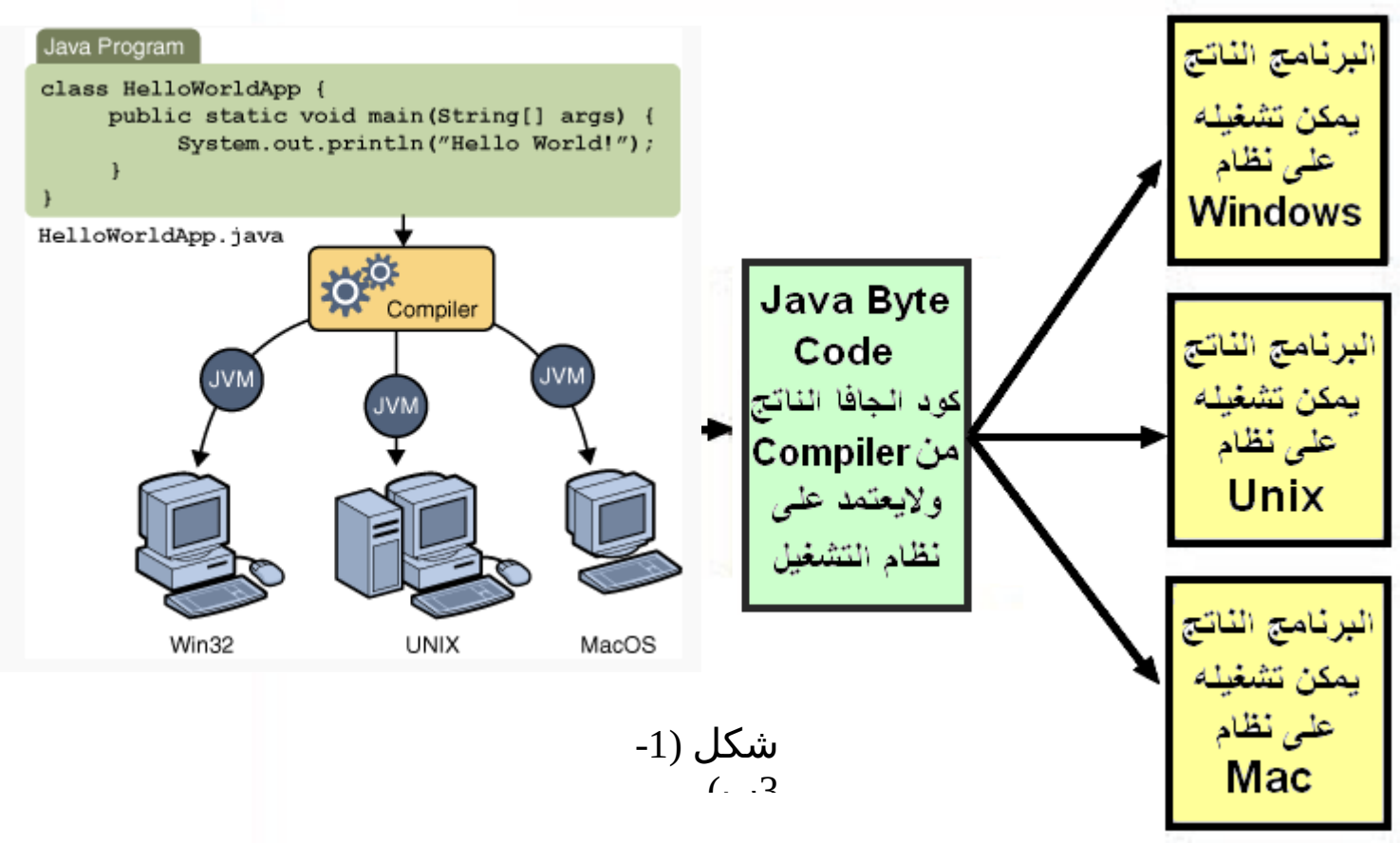


الشكل (1-2)

كما يوضح شكل (1-3أ) و شكل (1-3ب) و شكل (1-3ج)خطوات تشغيل برنامج مكتوب بلغة الجافا .



شكل (1-3أ)



شكل (1-

شكل (1-3ج)

2- **تعتمد لغة الجافا على أسلوب برمجة الأهداف Object Oriented Programming**

حيث وفرت كثيركثيرا من الجهد الذي كان يبذل بإستخداماستخدام البرمجة التقليدية . فقد كانت البرمجة التقليدية توفر للمبرمج مكتبة من الدوال إضافة إلى تركيب تقليدي للبرنامج وعلى المبرمج أن يستعمل الدوال مع تركيب البرنامج لإنشاء التطبيقات المختلفة مما يضطره لكتابة السطور الكثيرة أكثر من مرة ؛ و لقد كانت وحدة بناء البرنامج هي الدالة function . في حين أتت البرمجة بواسطة الأهداف بفكرة جديدة هي إنشاء عناصر متكاملة تحتوي على بيانات ودوال هي أساس إنشاء البرنامج . وبالتالي أصبحت وحدة بناء البرنامج وحدة كبيرة هي الفصيلة أو الفئة Class أو العنصر Object مما سهل واختصر الكثير من الوقت والجهد.والجهد.

وسوف نتحدث على هذه النقطة بالتفصيل في الباب الثاني .

3-**انشاءإنشاء برامج ذات واجهة مستخدم رسومية .**

يعتبر بناء واجهة المستخدم الرسومية من الأجزاء الهامة في البرنامج . حيث أن هذه الواجهات تعطي البرنامج شكلا معيناً , كما أن استخدام مفاهيم وأجزاء موحدة في بناء الواجهات للعديد من البرامج المختلفة يعطي المستخدم قدرا كبيرا من الراحة اثناء استخدام البرامج , كما أنه يقلل كثيرا من الوقت المستخدم لتعلمها . وقد تعرفنا في السنوات السابقة وأثناء استخدامنا للحاسب على واجهات رسومية كثيرة . مثل واجهات الويندوز والمستكشف للإنترنت وغيرها .

إن الأجزاء الرسومية الموجودة في لغة الجافا مرتبطة مباشرة مع الإمكانيات الرسومية للجهاز الذي يعمل عليه البرنامج . وبذلك فإن الواجهات الرسومية الموجودة في الجافا سوف تظهر بأشكال متباينة على الأجهزةأجهزة المختلفة . أي أننا عندما نقوم بكتابة برنامج يقوم بعمل زر على نظام الويندوز فأن هذا

الزر يأخذ نفس شكل الزر المستخدم في نظام الويندوز . ولكن اذا تم كتابته في نظام تشغيل آخر فإنه يأخذ شكل يتناسب مع نظام التشغيل المستخدم .

4- تصميم برمجيات تستفيد من كل مميزات الأنترنتإنترنت Java Applet .

وهيو نوع من التطبيقاتمن التطبيقات الذيالتي صممت خصيصا للانترنتللإنترنت. حيث يقوم المطور (Developer) باعدادبإعداد هذا البرنامج Applet ثم يتم استدعاءه من خلال ملف HTML بشرط HTML بشرط تحميل برنامج Applet على Server على الخادم (Server) الموجودالموجود عليه ملف HTML .HTML .HTML فيتم عرض هذا التطبيق من خلال صفحة الانترنت عندما يستعدي المستخدم هذه الصفحة.الصفحة.

1-1-2الشكل العام لبرنامج الجافا

البرنامج الآتي يبين الشكل العام لبرنامج الجافا ولا يهمنا هنا فهم كل جزئية في البرنامج فهذا سوف يتم في الدروس التالية :

ويقوم هذا البرنامج بطباعة جملة (Welcome to my World).-

وعند حفظ هذا البرنامج كما سنعرف لاحقا لابد وأن يتم تسمية الملف بأسمياسم Welcome.java. وكذلك يجب ان نراعي جيدا أن لغة الجافا هي لغة حساسة بالنسبة للأحرف فمثلا حرف (A) لايساويلا يساوي حرف (a) .

ويمكن تمثيل الهيكل العام لبرنامج الجافا بالشكل(1-4) التالي :

شكل (1-4)

1.2 أنواع البيانات والمعاملات

كما سبق وأن درسنا في الصف الثاني في لغة C++ الثوابت والمتغيرات والمعاملات فإنه في لغة الجافا لايوجدلا يوجد اختلافاً كبيراً فيماعما درسناه سابقا .

1-2-1 أولاً :حروف لغة الجافا

تتألف حروف لغة الجافا مما يلي :

1- الحروف الأبجدية (Letters) وهي الحروف الكبيرة (Capital Letters) من A إلى Z وكذلك الحروف الصغيرة (Small Letters) من a إلى z-z.

2- الأرقام العددية (Digits) من 0 إلى 9-9.

3- الحروف الخاصة (Special Characters) وهي تلك الحروف التي ليست بأعداد أو بحروف أبجدية ولكنها تكون على هيئة رموز كالآتي:كالآتي:

(+,-,//,>,<,$,#,%,(),||,!,[],!,=,,;,”,…..)وتعد هذه الرموز بأنواعها المادة الخام التالتي تتكون منها مفردات لغة الجافا.

1-2-2 ثانيا الثوابت والمتغيرات Constants & variables:

أولا : الثوابت Constants:

وهى عبارة عن قيم ثابتة يراد الإحتفاظالاحتفاظ بها طوال البرنامج ولا تتغير قيمتها أبداً.

وتنقسم الثوابت في لغةالجافالغة الجافا إلى:-

1- ثوابت عددية Numeric Constants

2- ثوابت رمزية Non-numeric Constants

1- الثوابت العددية:

يمكن تمثيل الثوابت العددية في لغة الجافا كالآتي:-

(أ)-الثابت العددي الصحيح : integer

- هو عبارة عن عدد مكون من الأرقام من (0 إلى 9).
- لا يحتوي على فاصلة عشرية.
- يمكن أن يحوى الإشارة (+ أو -).

ومن أمثلة الثابت العددي الصحيح (0، 12، 1000، -20،.....).

كما يمكن تصنيف الأعداد الصحيحة في لغة الجافا حسب طولها والسعة التخزينية لها في الذاكرة كما يلي :-

- الثوابت الصحيحة (19679، 40000) تسمى ثوابت صحيحة طويلة –long int.
- الثوابت (-16، 90 ، 55) تسمى ثوابت صحيحة قصيرة short int–.
- الثوابت (20000 ، 967) تسمى ثوابت صحيحة بدون إشارة unsigned int-.

والفرق بين الثوابت الطويلة والقصيرة هو في عدد الوحدات التخزينية المطلوبة لكل نوع في الذاكرة، فالثوابت الطويلة تأخذ حيزاً أكبر، والقصيرة توفر عدد الوحدات التخزينية المستعملة، أما الثوابت الصحيحة بدون إشارة فإن استعمالها يوفر وحدة تخزينية تستعمل للإشارة.

(ب)-الثابت العددي الحقيقي Floating Constant

- هو عدد مكون من الأرقام من (0 إلى 9)
- يجب أن يحتوي على فاصلة عشرية
- يمكن أن يحوي الإشارة (+، -)

ومن أمثلة الثوابت العددية الحقيقية

(421.5، 10.55، -67.99، 000000)

2- الثوابت الرمزية Non-Numeric:

وهى عبارة عن رموز اللغة وتتكون من الحروف والأرقام وتكون بين علامتي تنصيص أو اقتباس.

ومن الأمثلة على الثوابت الرمزية ما يلي:-

(" name" - "Khaled" - "12345" - "30+40")

ونلاحظ أن هناك ثوابت رمزية تحتوي على أرقام وهى في الحقيقة قيم حسابية لا يمكن إجراء أي عملية حسابية عليها بل يتم طبع الأرقام أو الرموز كما هي.

وإذا أردنا أن نضع قيمة سوف تظل ثابتة داخل البرنامج في مكان في الذاكرة فأننا نستخدم العبارة final للإعلان أن هذه القيمة ستظل ثابتة طوال تنفيذ البرنامج مثل:-

```
final int TABLE_SIZE = 41;

final float PI = 3.14159;
```

ويجب مراعاة أن اسم الثوابت constants يكون بالأحرف الكبيرة كاملاً و يفصل بين الكلمات كما يتم في المتغيرات مع ملاحظة أن الثوابت يتم تعريفها على أنها final .

مثال لأسماء الثوابت :

EXIT_ON_CLOSE

MY_NAME

CLASS_VERSION

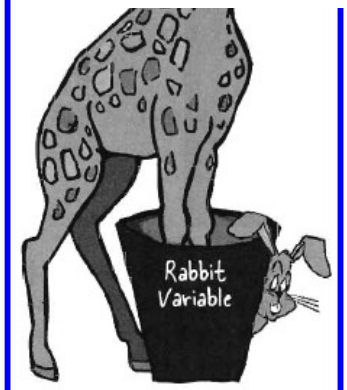
MATH_PI

LEFT

CENTER

BOTTOM

TOP



ثانيا : المتغيرات Variables:

وهي عبارة عن أسماء تحجز مواقع ففي الذاكرة حتى يتمكن البرنامج من تخزين البيانات فيها.

أو بمعنى آخر يمكن القول أن المتغيرات هي عبارة عن وعاء يمكن تحميله بقيمة وهذا الوعاء يتغير حجمه حسب القيمة التي سوف توضع فيه. أي أن هذه الأوعية متغيرة الحجم حسب قيمة ماتحملهما تحمله والشكل (1-5) يوضح هذه الفكرة:

ونلاحظ في هذا الشكل أنه إذا كان هناك وعاء على حجم الأرنب فأنه لايستطيعلا يستطيع بأي حال من الأحوال أن يحمل شيء كبير بحجم الزرافة مثلا . لذا لابد من عمل وعاء كبير يستطيع حمل الزرافة.الزرافة. ومن هذه الفكرة نستطيع القول أننا يجب أن نحدد حجم مخزن الذاكرة بالسعة المناسبة حسب حجم المتغير الذي سيتم تحميله بها.

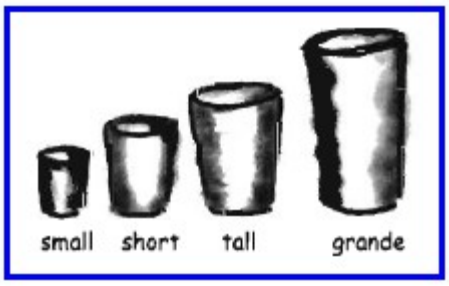
وعلى هذا الأساس نستطيع القول بأنه هناك أحجام مختلفة من الأوعية كما يظهرها الشكل (1-5).

شكل (1-5)

قواعد تسمية المتغيرات:المتغيرات:

- يمكن أن يبدأ الأسماالاسم بالحرف أو الشرطة السفلية (_) under score أو علامة الدولار ($) ولكن لايمكنلا يمكن بدء التسمية برقم ولكن يمكن أن نضع رقم بعد الحرف.

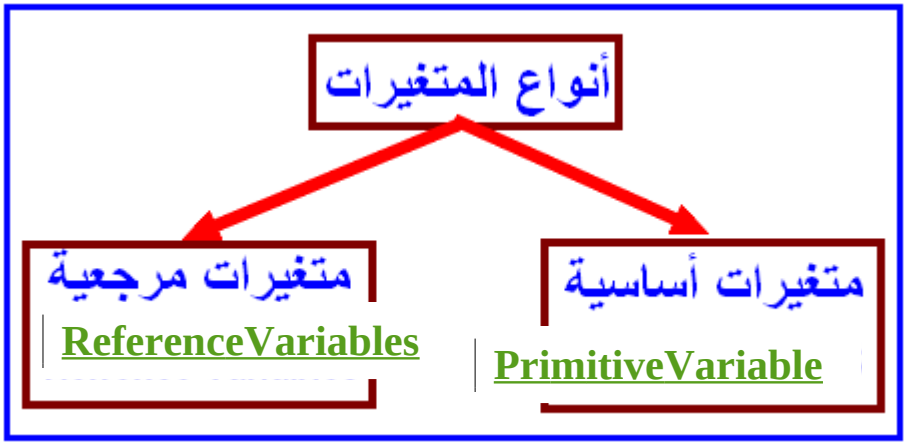
- لايمكنلا يمكن تسمية المتغير بأحديإحدى الكلمات المحجوزة



هذه للغة الجافا والجدول شكل (1-6) يبين الكلمات .

- اسم المتغيرات variables يكون بالاحرفيالأحرف الصغيرة لكل الأحرف ويلاحظ عدم وجود أقواس.

ويمكن تقسيم المتغيرات بصفة عامة إلى نوعين أساسين:أساسين:



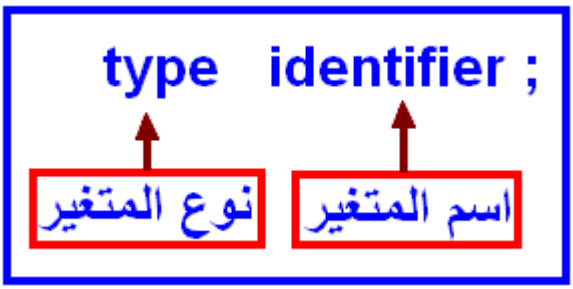
وسوف نتناول النوع الأول فقط بالشرح في هذا الجزء أما النوع الثاني فسوف يتم شرحه في الباب الثاني بشيء من التفصيل.

وتنقسم المتغيرات بدورها إلى نوعين:-

1- متغيرات رمزية (حرفية).

2- متغيرات عددية.عددية.

ولابد قبل استخدام المتغير من الأعلانالإعلان عنه ويتم ذلك كالتالي-:



أ- المتغيرات الحرفية Char:

وتتضمن الحروف بكافة أشكالها والرموز والفراغات (مسافة فارغة)

مثل:

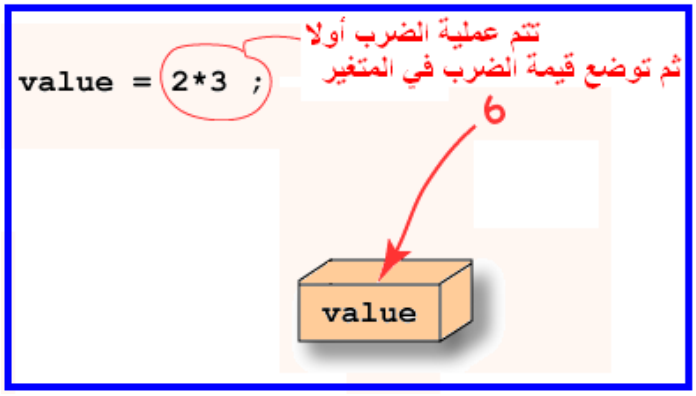
```
char a,b;
a= 'a';    char var1;
b=' ';      var1=' ' ;
```

2- المتغيرات العددية Numeric Variables :

أ) المتغيرات العددية الصحيحة Integer:

تتضمن قيم عددية صحيحة يمكن أن تأخذ قيمة تصل إلى 32767 وتكتب على الشكل التالالتالي:

```
;int a;     a=100
;int value ;  value=2*3
```



ب) المتغيرات العددية الحقيقية: Floating Point

تتضمن جميع الأعداد الحقيقية وتكتب على الشكل التالالتالي:

```
float x;      x=5.2;
```

كما يجوز تعريف المتغير وتخصيص قيمة له في نفس السطر كالتالي :كالتالي: **float x= 5.2**

وهنا يجب علينا الأنتباهالانتباه لجملة الأعلانالإعلان والتخصيص السابقة float x= 5.2 , فهنا رغم اننا أننا عرفنا المتغير على انه أنه متغير من نوع float حقيقي أي يقوم الحاسب بحجز مكان له في الذكرة مقداره 32 بت حسب الجدول شكل (1-7)(1 إلا أن الحاسب يعتبره من النوع double اي أي يحجز له مكان 64 بت .بت. والمبرمج الذكي هو الذي يجعل برنامجه يحتل اقلأقل مساحة في الذاكرة.الذاكرة.

وللتغلب على المشكلة السابقة يتم الأعلانالإعلان والتخصيص كالتالي:-

```
float x= 5.2 f
```

أي يتم وضع حرف (f) بعد الرقم لكي يتم حجز مكان له في الذاكرة مقداره 32 بت وبذلك نكون قد وفرنا في الذاكرة المستخدمة.المستخدمة.

ج) المتغيرات العددية الحقيقية الطويلة Double:

هتهي نفس المتغيرات العددية الحقيقية ولكن يمكن تمثيلها فتفي خمسة عشرة خانة

وتكتب على الشكل التالالتالي:

Double double x;

ويجب الإعلان عن المتغيرات في بداية البرنامج .

والجدول شكل(1-6) الآتي يوضح تطبيق بعض قواعد تسمية المتغيرات والأعلانوالإعلان عنها-:

```
int myPay, yourPay; // OK
long good-by ;  // اسم متغير خطأ لأنه يحتوي على "-"الشرطة
short shrift = 0;  // OK
double bubble = 0, toil= 9, trouble = 8 // خطأ لأنه لاينتهي بـ (;)
byte the bullet ;   // تعريف خطأ يحتوي على مسافات
int  double;   // تعريف خطأ يحتوي على كلمة محجوزة double
char thisMustBeTooLong ;  //تعريف صحيح ولكن اسم المتغير طويل جدا

int  8ball;   // تعريف خطأ اسم المتغير يبدأ برقم
float a=12.3; b=67.5; c= -45.44; //تعريف خطأ لأنه يستخدم الفاصلة المنقوطة
```

الجدول شكل(1-6)

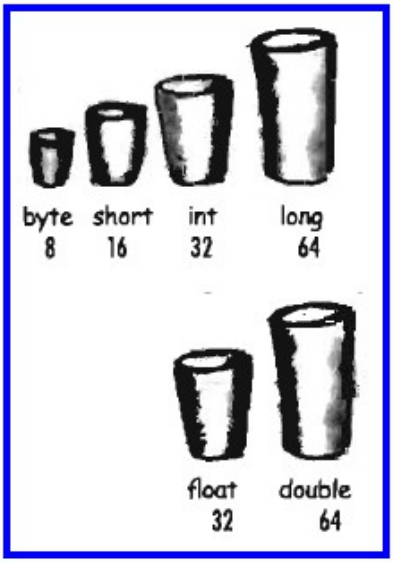
والجدول شكل (1-7) الآتي يبين أنواع البيانات والمتغيرات في لغة الجافا :

◄ انواع المتغيرات :

نوع المتغير	المتغير	الحجم
المتغيرات الصحيحة	byte	8 bit
	short	16 bit
	int	32 bit
	long	64 bit
المتغيرات الكسرية	float	32 bit
	double	64 bit
المتغيرات النصية	char	16 bit
	String	-
المتغير المنطقي	boolean	1 bit

الجدول شكل (1-7)

ومن الجدول السابق يمكن تمثيل أحجام البيانات في الذاكرة كما في الشكل (1-8):



الشكل (1-8)

وحتى الآن كل أنواع المتغيرات التي تم شرحها تسمى متغيرات أساسية primitive type وسوف نتعرف على ~~primative~~ نوع آخر في الباب الثاني يسمى المتغيرات المرجعية Reference type .

1-2-3 العمليات الحسابية والمنطقية في لغة الجافا

❖ الجدول شكل(1-9) التالي يبين أهم العمليات الحسابية في لغة الجافا :

◄ المعاملات الحسابية :

المعامل	الوصف	مثال بلغة جافا
+	جمع	X = A + B
-	طرح	X = A - B
*	ضرب	X = A * B
/	قسمة	X = A / B
+=	جمع ثم إسناد	(A+=B) = (A = A + B)
-=	طرح ثم إسناد	(A-=B) = (A = A - B)
*=	ضرب ثم إسناد	(A*=B) = (A = A * B)
/=	قسمة ثم إسناد	(A/=B) = (A = A / B)
%=	باقي القسمة	(A%=B) = (A = A % B)
++	زيادة بمقدار واحد	(A++) = (A = A + 1)
--	نقصان بمقدار واحد	(A--) = (A = A - 1)
%	باقي القسمة	X = A%B

الجدول شكل (1-9)

وهذه المعاملات قد تمت دراستها بأستفاضةباستفاضة في منهج الصف الثاني-.

الجدول شكل (1-9)

والجدول شكل (1-10) التالي يبين العمليات المنطقيةالمنطقية:

◄ المعاملات المنطقية :

المعاملات بالشكل الرياضي	شكل العمليات في لغة الجافا	مثال على الشرط	معنى الشرط
=	=	x==y	x تساوي y
≠	!=	x!=y	x لاتساوي y
>	>	x>y	x اكبر من y
<	<	x<y	x اصغر من y
≥	>=	x>=y	x اكبر من او تساوي y
≤	<=	x<=y	x اصغر من او تساوي y
And	&& او &	if (x ==1 & y== 1)	اذا تحقق كلا الشرطين
Or	|| او |	if (x = =1 | y== 1)	اذا تحقق احد الشرطين
Xor	^	if (x == 1 ^ y== 1)	-

الجدول شكل (1-10)

- **أما الجدول التالي شكل (1-11) فهو يبين الكلمات المحجوزة في لغة الجافا :-**

الكلمات المحجوزة في الجافا

public	abstract	finally	boolean	return
float	break	short	for	Static
if	byte	case	implements	Super
catch	switch	import	char	int
synchronized	instanceof	this	class	Throw
continue	interface	default	long	else
throws	native	true	transient	New
do	package	double	null	try
private	extends	while	protected	Final
volatile	void	false	--------	--------

الجدول شكل (1-11)

ومعنى الكلمات المحجوزة أي انهاأنها هي الكلمات والأوامر التي تعبر وتستخدم في لغة الجافا ولا يجوز استخدامها في غير ذلك كأسماء لمتغيرات مثلا ولذلك فهي محجوزة لمفردات اللغة فقط .

1-2-4 دالة الأخراجالإخراج في لغة الجافا

System.out.print

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

ولتوضيح عمل هذه الدالة سوف يتم دراسة بعض الأمثلة :

مثال(1)المطلوب عمل برنامجبرنامجا يقوم بطباعة العبارة Hello .Egypt .Egypt

```
} puplic public class  HelloEgypt

Public pubblic static void main ( String
} s[ ] )

; System.out.print("Hello  Egypt")

{
   {
```

شرح البرنامج

- **السطر الأول**

```
public class HelloEgypt {
```

وهذا هو السطر الأول في البرنامج وهو يتكون من: من:

```
puplic public
```

تسبق تعريف الفئة أو الكلاسالفصيلة Class وهي تعني أن هذه الفئة أو الكلاسالفصيلة عامة أي

يمكن لأي فئة اخرأخرى في البرنامج استخدام عناصر هذه الفئة .لأن برنامج الجافا قد يتكون

من أكثر من فئة class .class.

```
class
```

وهنا يتم بداية الكلاسالفصيلة

HelloEgypt وهذا هو اسم الكلاسالفصيلة ولقد تم تسميته هنا بالأسمياالاسم الذي نريده .

☚ ولابد هنا أن نشير إلى نقطة هامة جدًا!! وهي أنه عند حفظ ملف الجافا لابد أن يتم حفظه بنفس اسم الكلاسالفصيلة وبنفس شكل الحروف والمسافات وفي مثالنا هذا سيكون الأسمياالاسم HelloEgypt.java .

{ وهذا هو قوس بداية تعريف الكلاسالفصيلة .

- **السطر الثاني** Public static void main (String s[ ]) {

وهنا سوف يتم شرح الكلمات المطلوبة حاليا والباقي سوف نتناولها فيما بعد:بعد:

void ولقد سبق دراسة هذه الدالة-الكلمة في لغة c++ ومعناها أن الدالة بعد تنفيذ البرنامج لن تعود بأي قيم.قيم.

main وهي تشبه الدالة الرئيسية في لغة C++ وهي تعتبر نقطة البداية لوظيفة الكلاسالفصيلة main method .

(String s[ ]) الجملة الموجودة داخل قوسي البداية للدالة main وهي String s[ ] تعني مصفوفة من النوع الحرفي وتسمى s لتخزين جملة الطباعة في البرنامج .

وكما قلنا من قبل أن لغة الجافا هي لغة حساسة لحالة الأحرف لذلك يجب ملاحظة أن حرف **S** في كلمة String يجب أن يكون حرفاً كبيراً (capital letter) وإلا سيعطى البرنامج خطأ عند الترجمة .

- **السطر الثالث** `System.out.print ("Hello Egypt") ;`

`System.out.print` وهذا هو أمر الطباعة في لغة الجافا وسوف نتناوله بالتفصيل في الأمثلة القادمة .

وهنا يجب أن نلاحظ أن حرف S يجب أن يكون كبير (Capital letter)

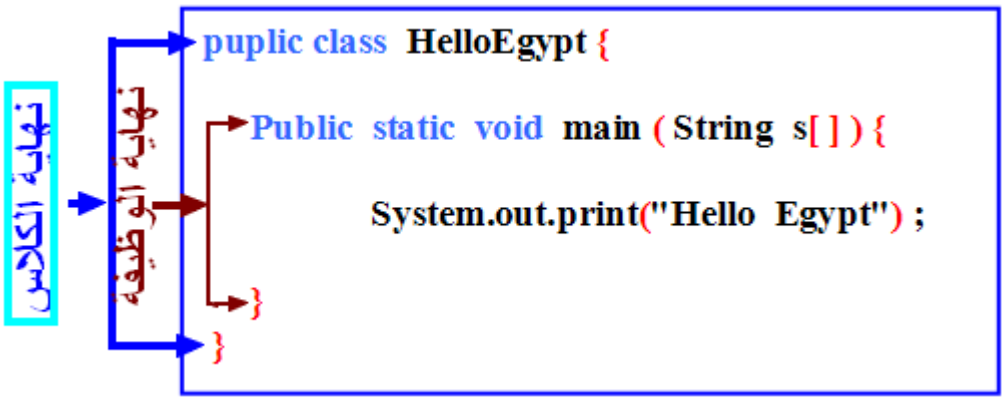
`("Hello Egypt")` وهذا هو النص المراد طباعته ويجب أن يوضع بين علامتي تنصيص (" ") وقوسين

`;` لابد وأن تنتهي كل جملة (سطر أو أمر) بعلامة (;)

```
}
}
```

بعد ذلك يتم إنهاء البرنامج بقوسي النهاية حيث يمثل القوس الأول نهاية الوظيفة methodemethod

للدالة main والقوس الآخر يمثل نهاية البرنامج الكلاسالفصيلة class .
والشكل الآتي يبين ذلك :



وكما ذكرنا سابقاً ان وظيفة هذا البرنامج هو طباعة الجملة Hello Egypt .

ولتشغيل هذا البرنامج يجب أن يتم له عملية ترجمة أولاً وسيتم ذلك بعدة طرق سوف نذكرها بالتفصيل في الجزء الخاص بالمعمل.

كما ذكرنا سابقا يجب تسمية البرنامج بالأسمياالاسم الموجود أمام جملة class وهو Hello Egypt .

خطوات تنفيذ البرنامج

1- نجري له عملية ترجمة كالآتي :كالآتي:

```
javac HelloEgypt.java
```

إذا لم يكن هناك أخطاء لاتظهرلا تظهر أي رسالة ومعنى ذلك أن البرنامج صحيح لغويا وهنا يتم عمل ملف كلاس أي HelloEgypt.class .

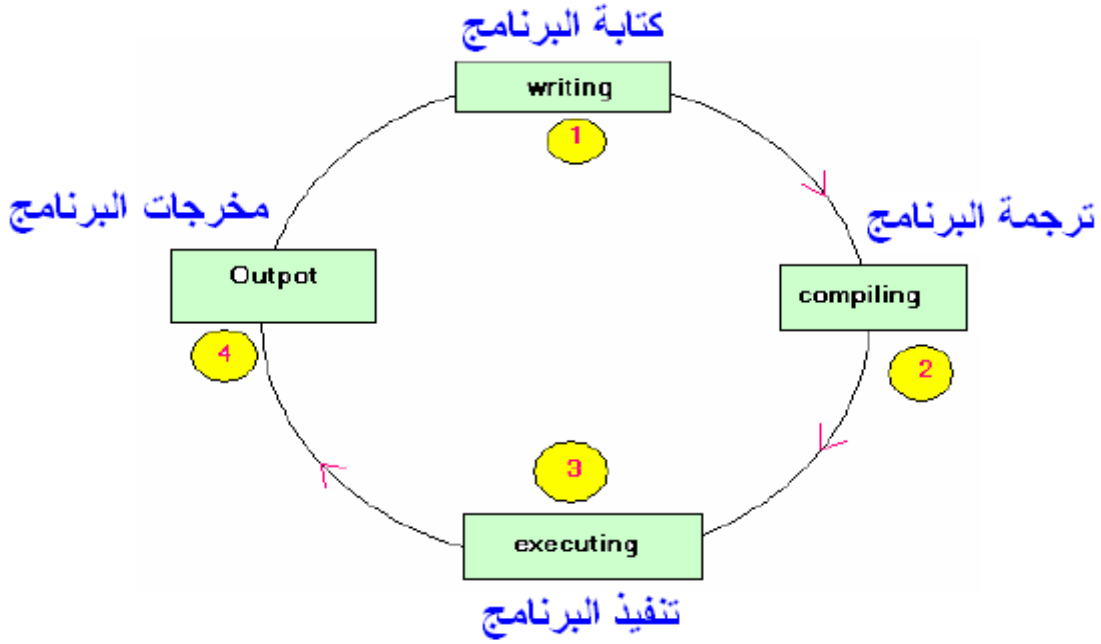
2- بعد ذلك نقوم بعملية تشغيل البرنامج كالآتي :كالآتي:

```
Java HelloEgypt
```

فتظهر على الشاشة عبارة Hello Egypt

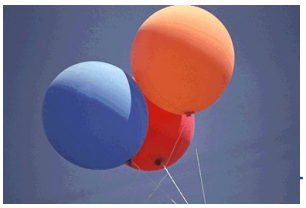
وبذلك يكون قد تم تنفيذ البرنامج.

وكما هو معروف فأن خطوات كتابة وتنفيذ أي برنامج يمكن أن تكون كما بالشكل (1-12)



شكل(1-12)

ويمكن إضافة متسلسلات الهروب مع جملة الطباعة للحصول على عدة أشكال من المخرجات



كما هو موضح بالجدول شكل (1-13)

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

شكل (1-13)

أمثلة على جملة الطباعة

مثال(2): ماهي مخرجات البرنامج التالي :

```
} puplicpublic class  Welcome

} pPublic  static  void  main ( String  s[ ] )

      3   ; System.out.print("Welcome to")

      4    ; System.out.print("Egypt")

          {
{
```

نلاحظ انهأنه عند تنفيذ هذا البرنامج ستظهر العبارة (**Welcome to Egypt**) على سطر واحد

وذلك تبعا لعبارتي الطباعة في السطر (3،4)

☚أما اذا اردنا أن تكون المخرجات على سطرين مختلفين فيتم اضافة حرفي (ln) على العبارة print ومعناها الأنتقالالانتقال إلى سطر جديد (new line) ويتم ذلك في السطر الثالث كالآتي :

```
3   ; System.out.println ("Welcome to")

      4    ; System.out.print("Egypt")
```

فتكون مخرجات البرنامج كالتالي :كالتالي:

```
Welcome to
Egypt
```

☚ويمكن تنفيذ نفس شكل المخرجات السابقة بسطر واحد وذلك عن طريق اضافة (n\) وتعني الأنتقالالانتقال إلى سطر جديد ويكون شكل البرنامج كالتالي :

```
3 ; System.out.print ("Welcome to\n Egypt")
```

فتكون شكل المخرجات كالشكل السابق:

```
Welcome to
Egypt
```

وفي هذه الحالة يتم الغاء السطر رقم (4)

☚أما اذا أردنا طباعة عدة أسطر متتالية بأمر طباعة واحد فيتم ذلك بتكرار (n\) كالتالي:

```
; System.out.print ("One\n Two \n Three \n Four")
```

فتكون المخرجات كالتالي:كالتالي :كالتالي:

```
One
Two
Three
Four
```

☚أما اذا أردنا أن تكون المخرجات على مسافات أفقية متساوية فأننا نستخدم (t\) كالتالي:

```
; System.out.print ("One\t Two \t Three \t Four")
```

وتكون المخرجات كالتالي :كالتالي:

```
One    Two    Three    Four
```

مثال (3): أكتب برنامج يقوم بجمع العددين (5+16)

ويتم ذلك بكتابة الأرقام المراد جمعها داخل اقواسأقواس جملة print ولكن من دون علامتي تنصيص

لأن علامتي التنصيص تكون دائما لطباعة الحروف وحتى اذا تم كتابة ارقام داخل علامتي التنصيص فأنها تعامل معاملة الحروف ايأي لايمكنلا يمكن اجراء ايأي عمليات حسابية عليها .

ويكون شكل عبارة print كالتالي :كالتالي:

```
; System.out.print (5+16)
```

ويكون الناتج (21) .

1-2-5 التعليقات ✍ Comment

إن أي مبرمج يحتاج في بعض الأحيان إلى اضافة بعض التعليقات والملاحظات الخاصة به و التي لا يتم تنفيذها في البرنامج ولكن فقط تذكره بالغرض من الأوامر التي يقوم بكتابتها .

ويمكن تعريف التعليقات كالآتي :

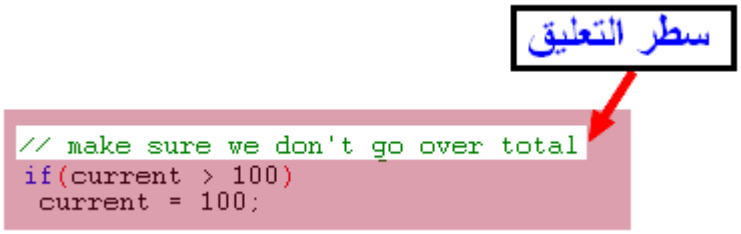
انهأنها الأسطر التي يتجاهلها مترجم الجافا، و لكنها تجعل البرنامج أسهل قراءة للمبرمج نفسه. بعبارة اخرأخرى، انهأنها مجموعة الملاحظات التي يضعها المبرمج في برنامجه لتسهيل قراءته.

والتعليقات في الجافا هي نفسها التعليقات الموجودة في لغة C++ كما سبق دراسته .

ومن أنواع التعليقات في الجافا :

1- التعليق بسطر واحد

ويكون هذا السطر مسبوقا بعلامتي (//) كالآتي :



أو يمكن كتابة التعليقات بجانب أسطر البرنامج كما يلي :

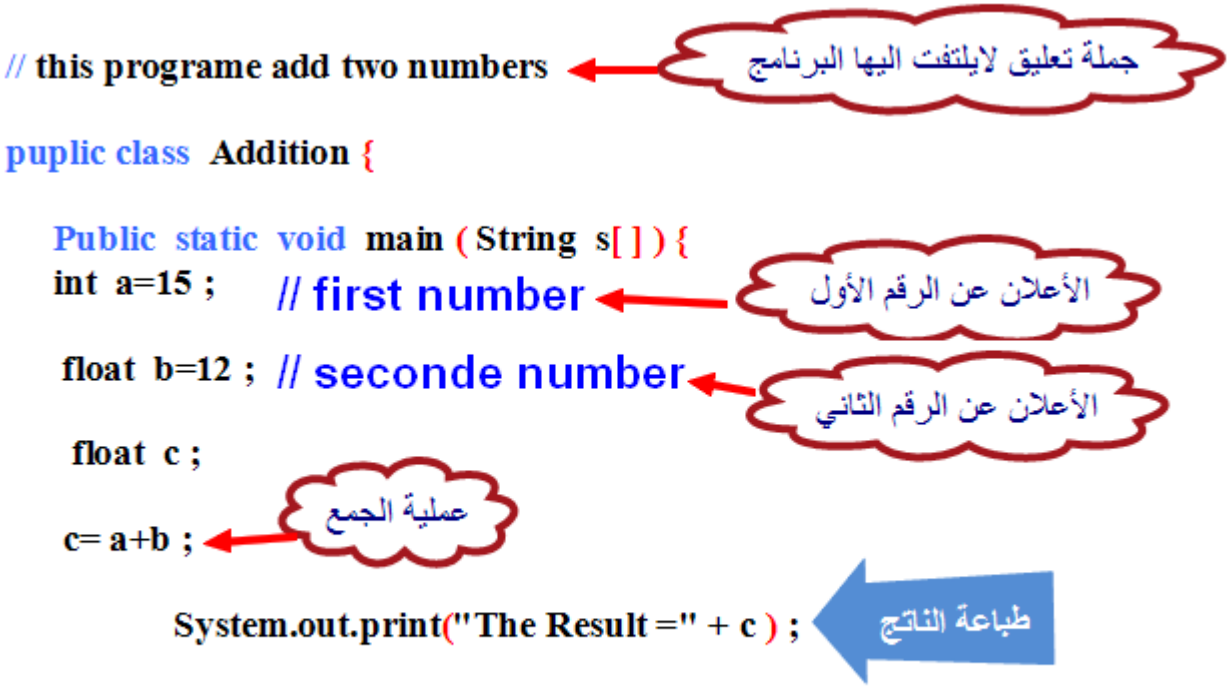
```
g.fillArc(0, 0,          // start
capWidth, height,        // size
90, 180);                // angle
```

2- التعليق بعدة أسطر

وفي هذه الحالة يمكن كتابة تعليق مكون من عدة أسطر كما يلي:يلي:
ويكون التعليق بين علامتي (/* التعليق */)

```
/*
   see if we are serializing or deserializing.
   The ability to deserialize or serialize allows
   us to see the bidirectional readability and writeability
 */

if (args.length == 1) {
    if (args[0].equals("-d")) {
deserialize = true;
    } else if (args[0].equals("-s")) {
```

مثال (4): أكتب برنامج يقوم بجمع عددين أحدهما صحيح والآخر حقيقي.



ويلاحظ في السطر الأخير للبرنامج أنه تم كتابة (C+) . وذلك لطباعة محتويات المخزن (C) أمام علامة (=) وعند تنفيذ البرنامج سوف تكون المخرجات ~~كالتالي:~~كالتالي:

The Result = 27

مثال (5) ما هو ناتج مخرجات البرنامج ~~التالي:~~التالي:

في هذا المثال تم استخدام عدة ~~اشياء~~أشياء منها :

- الطرق المختلفة ~~للأعلان~~للإعلان عن المتغيرات.
- العمليات الحسابية المختلفة.
- جملة الطباعة

```
public class ArithOper
{
        public static void main(String arg[])
        {
            int a=15;
            int b=4;
            int x,y,z,v,u;          // جمل الأعلان والتخصيص

            x=a+b;
            y=a-b;
            z=a*b;                  // جمل المعاملات والمؤثرات الحسابية
            v=a/b;
            u=a%b;

            System.out.println("a+b="+x);
            System.out.println("a-b="+y);
            System.out.println("a*b="+z);
            System.out.println("a/b="+v);
            System.out.println("a%b="+u);
        /* جمل
           طباعة
           المخرجات
        */
        }
}
```

وتكون مخرجات البرنامج على الشكل التالي: التالي:

```
a + b = 19
a  - b  =11
a  * b = 06
a / b   = 3
a %b = 3
```

مثال (6) ماهي مخرجات البرنامج التالي :

هذا البرنامج تطبيق على المؤثرات الأحادية

```
public class UnaryOper
{
    public static void main(String arg[])
    {
        int a,b,i,j;
        i=j=5;
        a=i++ * 3; // الزيادة بعد العملية الحسابية
        b=++j *3; // الزيادة قبل العملية الحسابية
        System.out.println("a = "+a+"\n"+"b = "+b);
    }
}
```

وتكون مخرجات البرنامج على الشكل:الشكل:

```
A=15
B=18
```

مثال (7) ما

```
public class UnaryOper1
{
    public static void main(String arg[])
    {
        int x1,x2,z=10;
        x1=z--;   // x1=10 هنا مازالت قيمة
        System.out.println("x1 = "+x1);
        x2=--z;  // x2=8 هنا أصبحت قيمة
        System.out.println("x2 = "+x2);

    }
}
```

ونلاحظ في هذا المثال أن قيمة x1 لازالت تساوي 10 ولا يتم انقاص الواحد منها الاإلا بعد الخروج من هذه الخطوة وتصبح قيمة z الجديدة z=9

ويكون ناتج خرج البرنامج كالتالي :كالتالي:

```
X1=10
X2=8
```

1-2-6دالة الأدخالالإدخال Input

بطبيعة الحال لايخلولا يخلو أي برنامج ذو فائدة من جملة الأدخالالإدخال , فهي الجملة التي تربط البرنامج بالعالم الخارجي وهي الوسيلة الوحيدة التي يستطيع فيها المستخدم ادخال إدخال القيم عن طريق لوحة المفاتيح للحاسب , حتى يقوم بمعالجة هذه القيم سواء كان البرنامج (برنامج حسابات – قاعدة بيانات .....) .

والحقيقة أن لغة الجافا تحتوي على أكثر من طريقة لأدخالالإدخال البيانات منها ماهوما هو مناسب لتطبيقات الويندوز ومنها ماهوما هو مناسب لبرامج الدوس (ConsolConsole ApplecationApplication) وهي البيئة التي سيتم تنفيذ برامجنا من خلالها في هذا المنهج .

وعبارة الأدخالالإدخال التي سوف نستخدمها هنا هي العبارة (Scanner) وهي عبارة عن كلاس فصيلة من كلاسات فصائل لغة الجافا وهي موجودة في مكتبة تسمى (java.util) ولابد لأستخداملاستخدام عبارة الأدخالالإدخال (Scanner)

أن نستدعيها من مكتبات لغة الجافا ويتم ذلك كالتالي :

```
; Import.java.util
```

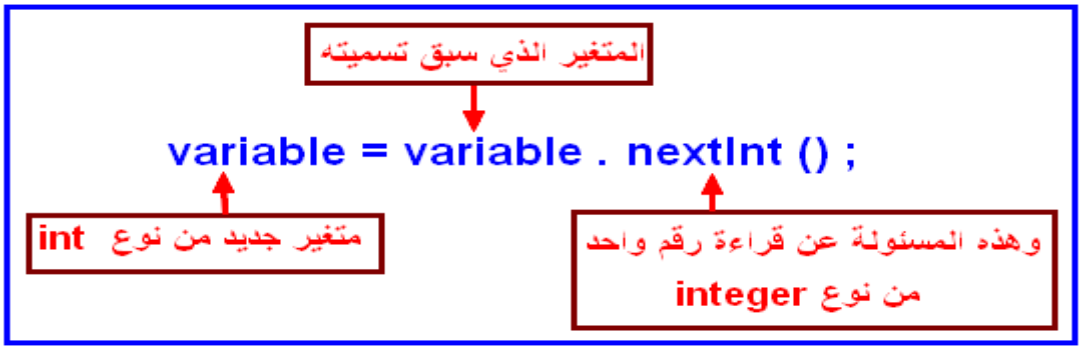
وبذلك يمكننا استخدام عبارة الأدخالالإدخال (Scanner) في البرنامج المطلوب .

ويكون ذلك بحجز مخزن لمتغير في الذاكرة ليحتوي الرمز المدخل عن طريق لوحة المفاتيح كالتالي :كالتالي:

```
Scanner variable = new Scanner(System.in) ;
```

اي اسم متغير
(a , b , sum ,.....)

ثم بعد ذلك يتم كتابة العبارة التالية :



والمثال التاليوالمثال التالي يوضح طريقة عمل عبارة الأدخالالإدخال ((Scanner

مثال (8)

أكتب برنامج لجمع رقمين على أن يتم ادخالإدخال الرقمين من لوحة المفاتيح ثم طباعة الناتج على الشاشة.

```
1 import java.util.Scanner  ;   ← استدعاء المكتبة التي تحتوي على جملة الأدخال
2  public class magdy{
3 public static void main(String s[] ){
4 int  a,b,sum;   ← الأعلان عن ثلاث متغيرات
5 Scanner Keyboard=new Scanner(System.in);   ← تهيئة البرنامج لعملية الأدخال
   6  System.out.println("Enter first number");
   7  a=Keyboard.nextInt();   ← هنا يتم ادخال الرقم الأول وتخزينه في المتغير a
   8  System.out.println("enter seconde number");
   9  b=Keyboard.nextInt();   ← هنا يتم ادخال الرقم الثاني وتخزينه في المتغير b
  10 sum=a+b;
  11 System.out.println("the sum is="+sum);
    }
}
```

شرح البرنامج

1- في السطر **الأول** تم استدعاء المكتبة التي تحتوي على جملة الأدخالالإدخال Scanner

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

2- أما في السطر **الرابع** فقد تم الأعلانالإعلان عن ثلاث متغيرات من النوع integer

وهي المتغير a لتخزين الرقم الأول والمتغير b لتخزين الرقم الثاني والمتغير sum لتخزين ناتج عملية الجمع.الجمع

```
4 int  a,b,sum;
```

3- أما السطر الخامس فهو يعمل على تهيئة الحاسب لأستقبالالاستقبال مدخلات من لوحة المفاتيح ولقد تم تسمية مخزن مؤقت تم تسميته Keyboard أو يمكن تسميته بأي اسم ويتم فيه تخزين القيمة المدخلة مؤقتا تمهيدا لنقلها لمتغير اخر سيكون هنا a أو b ولاحظ كلمة (System.in) أصبح بجوارها كلمة in دلالة على عملية الأدخالالإدخال

```
5 Scanner Keyboard=new Scanner(System.in);
```

4-أما السطر السادس فوظيفته هي طبع رسالة على الشاشة تخبر المستخدم بأدخالبإدخال الرقم الأول-.

5- أما في السطر السابع فيتم ادخالالإدخال الرقم الأول ثم يخزن مؤقتاً في المخزن Keyboard ثم ننقل أو نخصص القيمة الموجودة في المخزن Keyboard وهي هنا الرقم الأول ونضعها في المخزن a-.

```
7 a=Keyboard.nextInt();
```

6- أما في السطر الثامن فهو يكرر العملية لطلب الرقم الثاني-.الثاني.

7- وفي السطر التاسع يتم ادخالالإدخال الرقم الثاني كما سبق ولكن يتم تخزينه هذه المرة في المخزن b.

8- أما السطر العاشر فيتم فيه عملية الجمع ووضع الناتج في المخزن sum -.s.um

9- وفي السطر الحادي عشر يتم طباعة قيمة الجمع على الشاشة-.الشاشة.

ملاحظات هامة عن

1- يجب الملاحظة جيدا أن هناك كلمات لابد وأن يكتب الحرف الأول منها بحروف كبيرة ~~Cabital~~Capital ~~letter~~letter مثل الكلمات في هذا المثال (System , Scanner , nextInt) .

2- يجب أن تكون اسماء المتغيرات واضحة حتى يتم فهم البرنامج جيدا.

3- بالنسبة للأرقام المدخلة يجب أن تكون من النوع الصحيح فقط integer و هذا يكون في مثالنا فقط ~~لانا~~لأننا طلبنا منه ذلك في برنامجنا وذلك في السطر السابع عن طريق عبارة

(**nextInt**) فالحروف الثلاثة ذات اللون الأحمر (**Int**) والتي جاءت بعد كلمة next هي المسئولة عن المدخلات يجب أن تكون من النوع الأرقام الصحيحة وهي لها عدة حالات:

- **حالات العبارة (next) :**

والجدول -شكل (1-13) التالي يوضح الحالات المختلفة للعبارة (**next**)

شكل عبارة next	مثال EXAMPLE
Int_Variable = Object_Name.nextInt()	int number; ادخال عدد صحيح number = keyboard.nextInt();
Double_Variable = Object_Name.nextDouble()	double cost; ادخال عدد ذو دقة مضاعفة cost = keyboard.nextDouble();
String_Variable = Object_Name.next()	String word; ادخال حرفيات word = keyboard.next();
String_Variable = Object_Name.nextLine()	String line; ادخال حرفيات line = keyboard.nextLine();

الجدول شكل (1-13)

تشغيل البرنامج السابق

عند تشغيل البرنامج السابق يظهر الآتي :

```
Enter first number
20
enter second number
30
The sum is = 50
```

1- السطر الأول يطلب منك ادخالإدخال الرقم الأول وهنا تم ادخالإدخال العدد 20.

2- السطر الثاني يطلب منك ادخالإدخال الرقم الثاني وهنا يتم ادخالإدخال العدد 30.

3- أما السطر الأخير فيظهر النتيجة وهي حاصل الجمع 50. 50.

ويجب ملاحظة انهأنه عند ادخالإدخال الرقم الثاني يجب الضغط على مفتاح Enter أو ترك مسافة- واحدة-.

مثال (9)

أكتب برنامج تقوم من خلاله بأدخالإدخال اسمك فيطبع عبارة ترحيب بك

```
import java.util.Scanner;
   public c Cairo igdy{
   public static void main(String s[] ){

Scanner Keyboard=new Scanner(System.in);
System.out.println("Enter your name");

6  String a=Keyboard.next();   ← next أنظر الجدول السابق

System.out.println("Welcome\t"+a);

    }
}
```

ونلاحظ هنا أن عبارة الأدخال الإدخال لم تتغير كثيرا عن البرنامج السابق والذي تم فيه ادخال الإدخال الأرقام, إلا اختلافا بسيطاً في السطر السادس.

وقد تم عمل متغير حرفي من نوع String هو المتغير a والذي يتم فيه تخزين الحروف المدخلة من لوحة المفاتيح كما نلاحظ تغير العبارة next ولقد كتبت منفردة بدون أي اضافات (أنظر الجدول السابق الذي يوضح وظائف next) .

تشغيل البرنامج

عند تشغيل البرنامج يطلب منك ادخال الإدخال أسمك فنقوم بأدخال الإدخال الأسم الاسم من لوحة المفاتيح فيقوم بعد ذلك بطباعة عبارة الترحيب كالتالي :

```
Enter your name
Cairo
Welcome Cairo
```

1.3 جمل الأختيار الاختيار Selection Statements

ويطلق عليها ايضا أيضًا جمل التحكم أو جمل اتخاذ القرار ولقد سبق دراسة هذه الجمل في الصف الثاني وهي لاتختلف تختلف كثيرا عن الجمل الموجودة في لغة الجافا.

1.3.1 جملة الشرط if statement

- **الصيغة الأولى لجملة if**

تأخذ الجملة if الصيغة العامة التالية :

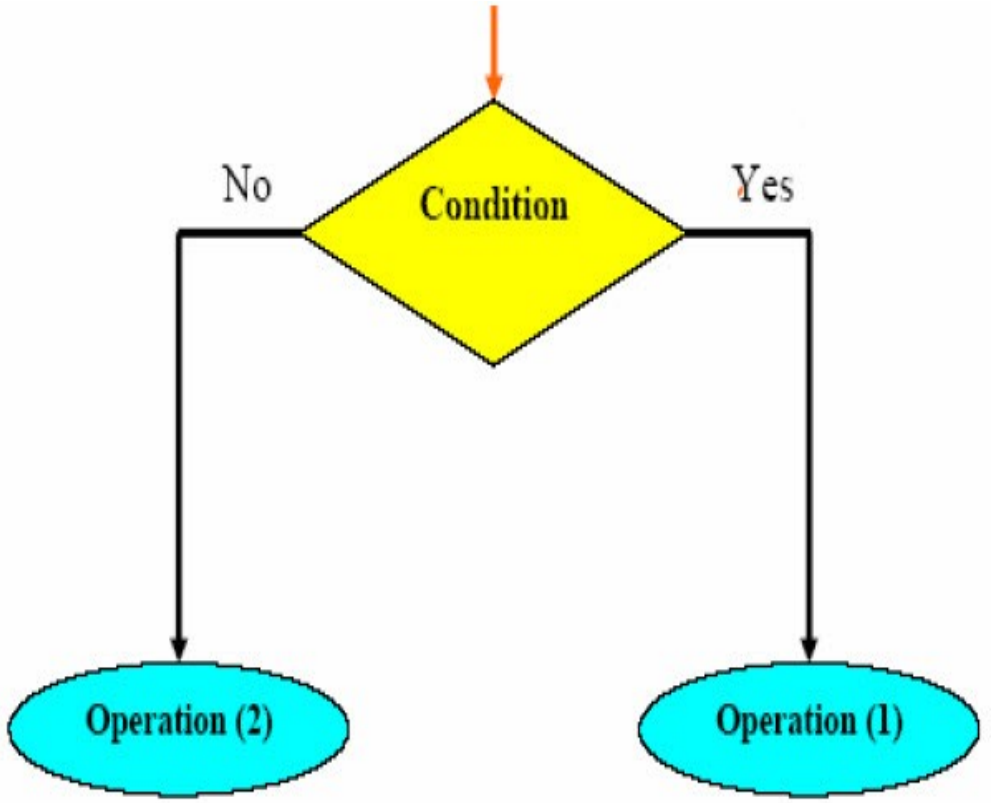
```
الشرط
if (expression) {
        statement1; // جملة1
        statment2;  // جملة2
          }
```

ومعنى هذه العبارة أنه اذا كان الشرط الذي تقوم الجملة (if) بأختبارهياختباره صحيحا فقم بتنفيذ الجملة التي بين القوسين . وفي حالة عدم صحة الأختبارالاختبار فلا تقم بتنفيذ جملة (if) وانما وإنما استمر في تنفيذ بقية جمل البرنامج من بعد تخطي جملة (if) .

وفي حالة تنفيذ جملة واحدة فقط بعد جملة (if) فأنه فإنه يمكن الأستغناءالاستغناء عن الأقواس وفي هذه الحالة تنتهي جملة الشرط بالفاصلة المنقوطة (;) كما يلي :

```
الشرط
if (expression)
        statement1;
تنتهي جملة الشرط عند الفاصلة المنقوطة
```

ويمكن تمثيل عبارة if بالشكل (1-14) التالي :



شكل (1-14)

وكما نعرف دائما أن فإن الشرط يجب أن يكون شرطا منطقيا ولفهم طريقة عمل جملة if ندرس الأمثلة التالية:-

مثال (10)

المطلوب كتابة برنامج يقوم بفحص رقم يتم ادخاله من لوحة المفاتيح وليكن)

```
import java.util.Scanner;
public class magdy{

public static void main(String s[] ){
int x;
  Scanner Keyboard=new Scanner(System.in);
  System.out.println("Enter Number X");
   x=Keyboard.nextInt();
```

هنا يتم ادخال الرقم X

```
if(x>0)
        System.out.println("X is positive");
    }
  }
```

هنا تتم عملية المقارنة X

ونلاحظ هنا أن جملة if أنتهتانتهت بأول فاصلة منقوطة قابلتها ايأي بعد عبارة الطباعة ونلاحظ هنا انناأننا لم نستخدم الأقواس في جملة if لأننا لم نكتب غير سطر واحد فقط بعد عبارة if وهي جملة الطباعة أما لو كتبنا أكثر من جملة يجب تنفيذها عند تحقق الشرط في هذه الحالة يجب اضافة الأقواس ويتم تنفيذ البرنامج كما يلي :

```
Enter Number X
5
X is positive
```

مثال (11)

كرر نفس المثال السابق مع إستخداماستخدام أكثر من سطر في جملة if كالأتيكالآتي-:

```
import java.util.Scanner;
public class magdy{

public static void main(String s[] ){
int x;
  Scanner Keyboard=new Scanner(System.in);
  System.out.println("Enter Number X");
   x=Keyboard.nextInt();
```

هنا يتم ادخال الرقم X

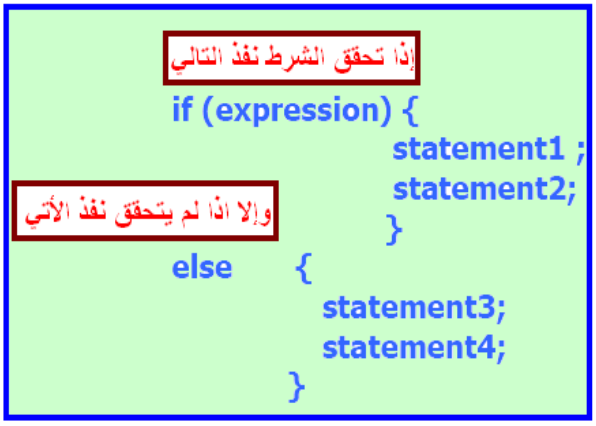
```
if(x>0){
        System.out.println("X is positive");
        System.out.println("X is not negative"); }
   }
}
```

نلاحظ أن جملة الشرط احتوت على أكثر من سطر ولذلك وضعنا الأقواس

وتكون مخرجات البرنامج كالتالي:

```
Enter Number X
10
X is positive
X is not negative
```

- **الصيغة الثانية لجملة if**



مثال (12)

نفذ البرنامج السابق بحيث يطبع عبارة (X is positive) في حالة إذا كانت موجبة وإلا يطبع عبارة (X is negative) في حالة إذا كانت X ~~سالبة~~.سالبة.

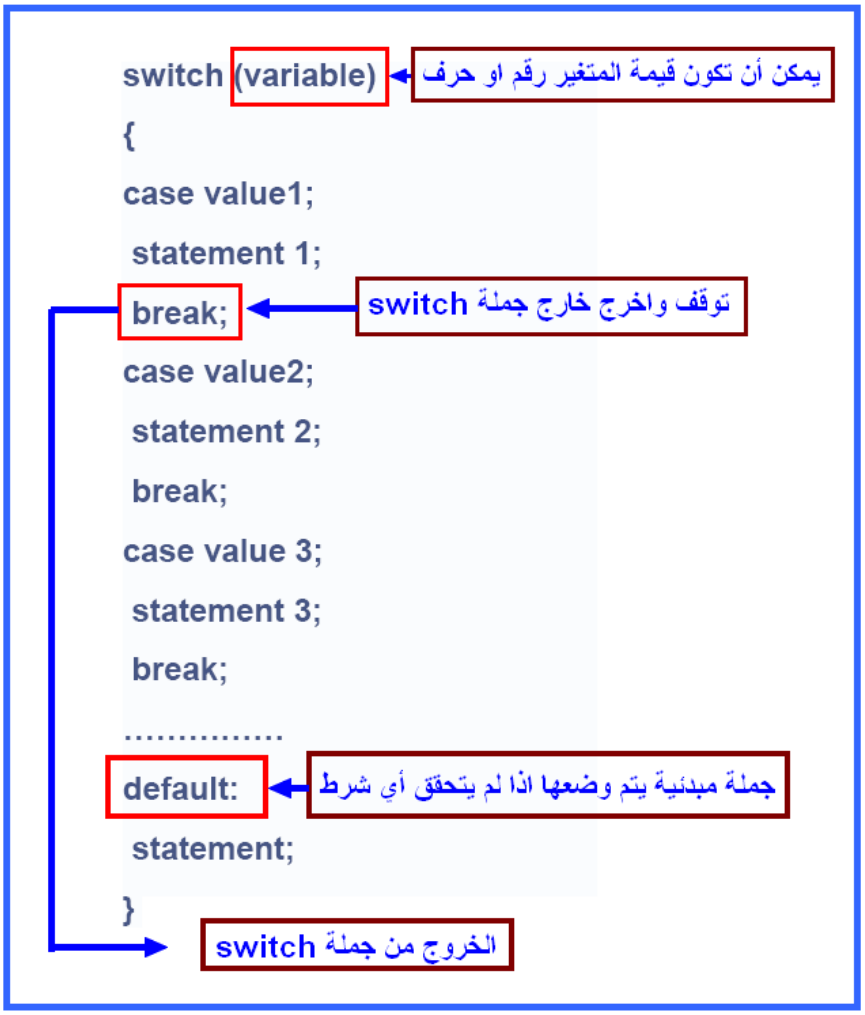
ويتم ذلك ~~بأستخدام~~باستخدام عبارة IF~~-IF~~ –الكاملة كالآتي:-

```
import java.util.Scanner;
public class magdy{

public static void main(String s[] ){
int x;
  Scanner Keyboard=new Scanner(System.in);
  System.out.println("Enter Number X");
   x=Keyboard.nextInt();  <-- هنا يتم ادخال الرقم X

  if(x>0){
         System.out.println("X is positive"); }
else  {

      System.out.println("X is not negative");  }

        }
       }
```

و يمكننا في البرنامج السابق حذف الأقواس الموجودة بعد جملة if , وكذلك حذف الأقواس الموجودة بعد else وذلك لوجود جملة طباعة واحدة بعد كلاً كلاً لامنهما.

1.3.2 جملة switch

تستخدم عبارة if if إذا كان جواب الشرط عبارة عن إحتماليناحتمالين أو ثلاثة إحتمالاتاحتمالات على الأكثر, أما إذا زاد عدد الأحتمالاتالاحتمالات على ذلك فمن الأفضل إستخداماستخدام عبارة switch switch وصيغتها العامة كالآتي:



ولكي نفهم كيفية عمل جملة switch نجري المثال التالي :التالي:

مثال (13)
لنفرض انناأننا نريد ان يطبع الحاسب جملة ترحيب معينة إذا تم الضغط على أحد الأرقام في لوحة المفاتيح .
خطوات البرنامج :

```
import java.util.Scanner;
public class magdy{

public static void main(String s[] ){
int a;
Scanner Keyboard=new Scanner(System.in);
System.out.println("Enter a Number ");
 a=Keyboard.nextInt();

switch(a)
            {
   case 1:    في حالة اذا كان الرقم 1
        System.out.println("welcome");
     break;
    case 2:    في حالة اذا كان الرقم 2
          System.out.println("how are you");
     break;
                في حالة اذا كان الرقم 3
    case 3:
           System.out.println("good morning");
       break;
   default:

         System.out.println("good by"); }
      }
 }
```

وعند تنفيذ هذا البرنامج فأنه تحدث إحدى الحالات الآتية:الآتية:

1- إذا تم ~~ادخال~~الإدخال الرقم (1) ~~فإ~~أنه يطبع العبارة الأولى Welcome ثم يجد عبارة break فيخرج خارج جملة switch وينتهي البرنامج .

2- إذا تم ~~ادخال~~الإدخال الرقم (2) ~~فأنه~~ فإنه يطبع العبارة الثانية how are you ثم يجد عبارة التوقف break فيخرج خارج جملة switch وينتهي البرنامج .

3- وهكذا في حالة ~~ادخال~~الإدخال الرقم (3) ~~فأنه~~ فإنه يطبع الجملة الثالثة ثم break ثم يخرج .

4- أما في حالة ~~ادخال~~الإدخال أي رقم غير موجود في البرنامج وليكن (4) مثلا, فإن البرنامج يطبع العبارة الموجودة في جملة default ثم ينتهي البرنامج .

5- يجب مراعاة أن جملة ~~switch~~ switch لها قوسي بداية ونهاية.

6- يجب دائماً أن تنتهي كل حالة case من حالات switch بالعبارة break.

☺ حاول -أن تقوم ~~بألغاء~~بإلغاء هذه العبارة وتجربة البرنامج ... ماذا تجد ؟؟

☺ جرب ~~ادخال~~الإدخال حرف بدلا من الرقم ماذا تجد ؟؟

1.4 جمل الدوران

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

1- الحلقة (~~For~~ for Loop)

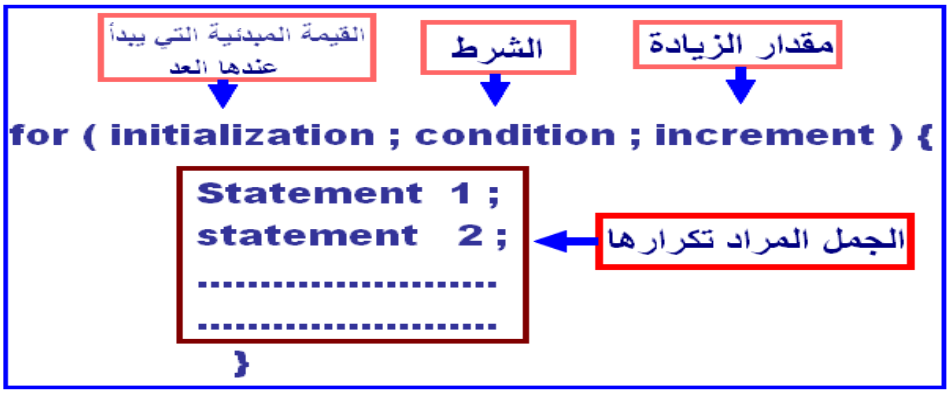
2- الحلقة (while loop)

3- الحلقة (do-while Loop)

1- 1-4الحلقة (for loop)

تستخدم الحلقة for– for لتكرار أمر معين (أو مجموعة من الأوامر) عددا من المرات.

والصيغة العامة لهذه الحلقة كالتالي :



نلاحظ أن هذه الحلقة تتكون من ثلاث أقسام هي :هي:

1- القيمة الإبتدائيةالابتدائية initialization .

نضع في هذا الجزء متغير ونعطيه القيمة الإبتدائيةالابتدائية التي يبدأ منها التكرار.

2- الشرط Condition

هنا نضع الشرط الذي يتوقفالذي يتوقف عنده العد.

3- مقدار الخطوة increment

هنا نضع مقدار الزيادة في حالة العد التصاعدي أو النقصان في حالة العد التنازلي.التنازلي.

ونلاحظ هنا أن جملة for لها قوس بداية وقوس نهاية ويتم وضع الأقواس في حالة تكرار أكثر من جملة كما يمكن الاستغناء عن هذه الأقواس في حالة تكرار جملة واحدة.

مثال (14)

نفذ برنامج بلغة الجافا يقوم بالعد من (1 إلى 20) .

خطوات البرنامج كالتالي :

```
import java.util.Scanner;
public class magdy{

public static void main(String s[] ){
 int a;
     for ( a=1;a<=20;++a)

          System.out.println(a) ;
       }
  }
```

جملة for

في هذا البرنامج تم استخدام متغير a من نوع integer فيتم زيادة قيمته كل مرة بمقدار واحد. والقيمة الابتدائية –له داخل الحلقة a=1 حتى يصل العد إلى 20 . تنتهي الحلقة وينتهي البرنامج

و في كل خطوة زيادة يتم طباعة قيمتها على الشاشة عن طريق أمر الطباعة.

وتكون الأرقام في شريط تحت بعضها على الشاشة. لماذا ؟؟

ثم فكر كيف يمكن طباعة المخرجات متجاورة أو على سطر واحد بينها مسافات متساوية.

مثال (15)

المطلوب عمل عداد تصاعدي يبدأ العد من القيمة (1) حتى القيمة (x) على أن يتم ~~ادخال~~الإدخال قيمة نهاية العد من لوحة المفاتيح .

وفي هذا البرنامج استخدمنا عبارة ~~الأدخال~~الإدخال كما استخدمناها في البرامج السابقة, وعن طريقها تم ~~ادخال~~الإدخال رقم نهاية العد وتم وضعه في المتغير (x) ثم وضعناه في الجزء الخاص بالشرط في الحلقة .

ونلاحظ هنا أن المتغير (a) قد تم تعريفه واعطاؤه قيمة ابتدائية داخل الحلقة ((int a=1.

خطوات البرنامج :

```
import java.util.Scanner;
public class magdy{

public static void main(String s[] ){
 int x;
 Scanner Keyboard=new Scanner(System.in);
 System.out.println("Enter a Number ");
  x=Keyboard.nextInt();
    for (int a=1;a<=x;++a)

        System.out.println(a) ;
     }
 }
```

هنا يتم ادخال رقم نهاية العد

جملة for

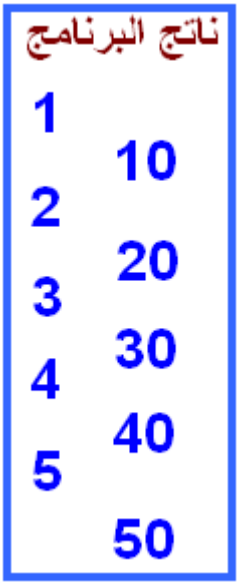
مثال (16)

عمل برنامج يوضح تكرار أكثر من جملة داخل الحلقة for~~ .for~~.

```
import java.util.Scanner;
public class magdy{

public static void main(String s[] ){
 int a;
     for ( a=1;a<=5;++a){
            System.out.println(a) ;
              System.out.println("\t"+a*10) ;
              }

        }
  }
```

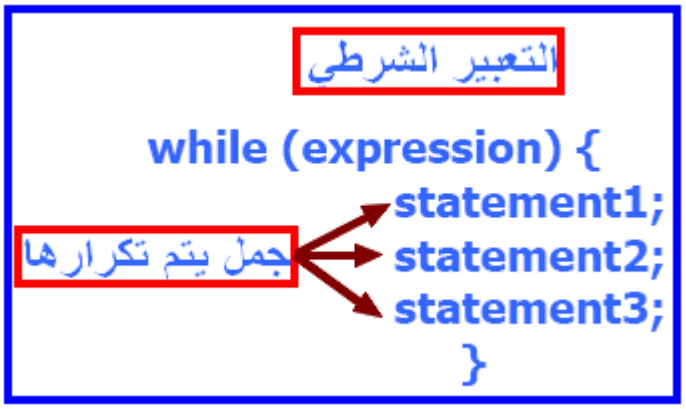
طباعة اكثر من جملة داخل الحلقة

وفي هذا البرنامج تم استخدام ~~اكثر~~أكثر من جملة يراد تكرارها داخل الحلقة ولذلك تم استخدام قوسي بداية ونهاية للحلقة وهذا البرنامج يقوم بطباعة المخرجات كالتالي :



2-4-1-الحلقة (while loop)

في هذه الحلقة التكرارية نحتاج إلى الشرط فقط وطالما كان هذا الشرط متحققا استمرت الحلقة في التكرار والصيغة العامة لها كالآتي:



ونلاحظ هنا أن الشرط يأتي ~~أولاً~~ أولاً قبل تنفيذ ~~الحلقة~~ الحلقة.

مثال (17)

أكتب برنامج يقوم بعملية العد من (0 إلى 10) ~~بأستخدام~~باستخدام الحلقة while loop مع طباعة النتائج على الشاشة.

```
import java.util.Scanner;
public class magdy{

public static void main(String s[] ){
int a=0;        <-- القيمة المبدئية للعداد

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

جملة while

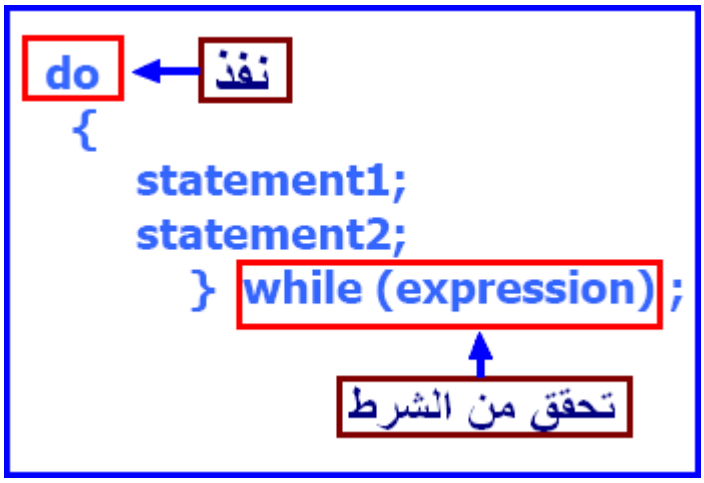
ونلاحظ في هذا البرنامج أنه لابد من إعطاء قيمة ابتدائية للعداد int (a=0) .

ونلاحظ كذلك أنه في جملة while while لابد من وجود الأقواس-, لأن الجملة بطبيعتها تتكون من أكثر من سطر-.

كذلك يتم زيادة قيمة (a) بمقدار واحد عن طريق الصيغة (a++) وبعد ان أن يتم زيادة قيمة a بمقدار واحد يتم التحقق من الشرط كل مرة وستكون نتيجة البرنامج طباعة الأعداد من (0 إلى 10) .

3-4-1-الحلقة (do – while)

تختلف هذه الحلقة عن الحلقات السابقة في مكان وضع الشرط، الشرط حيث يكتب الشرط بعد العبارات المطلوب تكرارها وتكون صيغتها العامة كالتالي:



بإمكاننا القول أن الحلقة (do-while) تعني قم بالدخول في الكتلة do وقم بتنفيذ الأوامر. وفي حالة الإنتهاءالانتهاء قم بأختباراختبار التعبير الشرطي الموجود في آخر الكتلة, وفي حالة صحة التعبير قم بالرجوع مرة أخرى إلى مكان الكلمة do.

مثال (18)

يمكن تطبيق نفس المثال السابق في حلقة while وهو البرنامج الذي يقوم بالعد من (0 إلى 10) ولكن هذه المرة باستخدام الحلقة (do-while) كالآتي :

```
import java.util.Scanner;
public class magdy{

public static void main(String s[] ){
 int a=0;

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

تحقق من الشرط

ونلاحظ هنا أنه في كل مرة يقوم البرنامج بالتحقق من الشرط في نهاية ~~الحلقة~~ الحلقة، وهذا يعني أنه إذا لم يتحقق الشرط فسوف يتم تكرار الحلقة مرة واحدة فقط.

1-5 الدوال ~~methode~~methods

وهي عبارة عن طرق ودوال من تعريف ~~(تصميم)~~ المبرمج(تصميم) أو تكون جاهزة في البرنامج. والغرض منها هو تسهيل عملية البرمجة في الأشياء ~~التالي~~التالي تتكرر أكثر من مرة في البرنامج.

- **الهدف من الدوال**

في حالة تكرار مجموعة من سطور الأوامر أكثر من مرة في مواضع مختلفة في البرنامج فإن أوامر التكرار لن تكون ذات ~~منفعة~~ منفعة. ولذلك يتم كتابة هذه الجمل منفصلة عن البرنامج الرئيسي .

- **مزايا استخدام الدوال**

1- عدم الحاجة إلى تكرار التعليمات داخل البرنامج حيث يتم ~~انشاء~~إنشاء الدالة مرة واحدة ويمكن استدعائها أكثر من مرة عند الحاجة اليها .

2- باستخدام الدوال يصبح البرنامج أكثر وضوحا.

3- باستخدام الدوال الجاهزة يمكن توفير الكثير من الوقت ~~والجهد~~ والجهد.

- **هناك نوعان من الدوال يمكن ~~استخدامهما~~ استخدامهما:**

1- دوال جاهزة يمكن أن توفرها لغة الجافا .

2- دوال يمكن تعريفها عن طريق ~~المستخدم~~ المستخدم.

1- 5-1 دوال جاهزة يمكن ان توفرها لغة الجافا .

مثل الدوال الرياضية بأنواعها والجدول شكل (1-15) الآتي يبين الدوال الحسابية الجاهزة في لغة الجافا :

اسم الدالة	وصف الدالة	مثال
abs(x)	القيمة المطلقة لـ x.	Math.abs(6.2) → 6.2 Math.abs(-2.4) → 2.4
ceil(x)	تقرّب x إلى أقل عدد صحيح ليس اقل من x.	Math.ceil(5.1) → 6 Math.ceil(-5.1) → -5
floor(x)	تقرب x إلى أكبر عدد صحيح ليس أكبر من x.	Math.floor(5.1) → 5 Math.floor(-5.1) →-6
max(x,y)	أكبر قيمة من x و y.	Math.max(7,6) → 7
min(x,y)	أقل قيمة من x و y.	Math.min(-7,-8) → -8
pow(x,y)	x مرفوعة للأس y.	Math.pow(6,2) →6^2 → 36
sqrt(x)	الجذر التربيعي لـ x.	Math.sqrt(9) → $\sqrt{9}$ → 3
random()	تكوّن رقم عشوائي بين الصفر والواحد.	Math.random() → 0.23121

شكل (1-15)

الجدول شكل (1-16) يبين المكتبات الخاصة بالجافا وما تقدمه هذه المكتبات من خدمات .

المكتبة	الخدمات / الفئات	التطبيقات
java.util	تحتوي هذه الباقة على فئات تمثل هياكل بيانات عامة الإستعمالالاستعمال مثل الصفوف و المجموعات و غيرها.	البرمجة الخوارزمية العادية.
java.io	تحتوي هذه الباقة على فئات تتصرف في عمليات تصدير و توريد البيانات.	البرامج التي تتطلب معاملة فورية مع المستخدم.

java.lang	تحتوي هذه الباقة على الفئات المتعلقة بتنفيذ البرنامج و مراقبته، بما فيها الفئات التي تعالج أخطاء التنفيذ و بعض الفئات العامة. نظرا لأهميتها، فإن هذه الباقة يقع توريدها ضمنيا في كل برنامج.	كل البرامج/ كل التطبيقات.
java.math	تحتوي هذه الباقة على فئات تقوم بعمليات حسابية، بأي دقة يطلبها المستخدم.	التطبيقات الهندسية و تطبيقات الرياضيات.
java.sql	تحتوي هذه الباقة على فئات تختص في عمليات على قواعد البيانات.	تطبيقات التصرف التي تتطلب قواعد بيانات.
java.awt	تحتوي هذه الباقة على فئات تختص في الرسم و في إنجاز واجهات رسومية.	تطبيقات تتطلب واجهات رسومية مع المستخدم.
java.swing	تمدد هذه الباقة إمكانيات و قدرات الباقة السابقة.	تطبيقات تتطلب واجهات رسومية مع المستخدم.
java.security	تحتوي هذه الباقة على فئات تختص في تنفيذ إجراءات أمنية في البرنامج، مثل مراقبة المستخدمين و صيانة الوارد و غير ذلك.	تطبيقات تتطلب إجراءات أمنية.

شكل (1-16)

ويمكن استدعاء الدوال بكتابة اسم الكلاسالفصيلة(الفئة) متبوعاً بنقطة بعدها اسم الطريقة ثم قائمة المعاملات داخل أقواس دائرية كما يلي:

```
Class_Name.method_Name(Argument List)
```

فمثلا إذا اردنا الحصول على الجذر التربيعي للعدد(25) فيمكن كتابة الصيغة كالتالي:

```
System.out.print(Math.sqrt
```

تقوم هذه الجملة باستدعاء الدالة (sqrt) الموجودة في الكلاس الفصيلة (Math) والتي تأخذ معامل واحد من نوع (Double) ونتيجة تنفيذ هذه الجملة سيكون طباعة (5.0) .

مثال (19) :

المطلوب عمل برنامج يستقبل قيمة من لوحة المفاتيح ثم يقوم بإيجاد الجذر التربيعي ومربع هذا الرقم وذلك بإستخداماستخدام الدوال الجاهزة في لغة الجافا .

خطوات البرنامج كالآتي :

```
import java.util.Scanner;
public class Math{

public static void main(String s[] ){
double number;

Scanner Number=new Scanner(System.in);
System.out.println("Enter a Number");

number=Number.nextDouble();

System.out.println("The square Root ="+Math.sqrt(number));
System.out.println("The Squar number is ="+Math.pow(number,2));

}}
```

جمل طباعة الدوال الرياضية

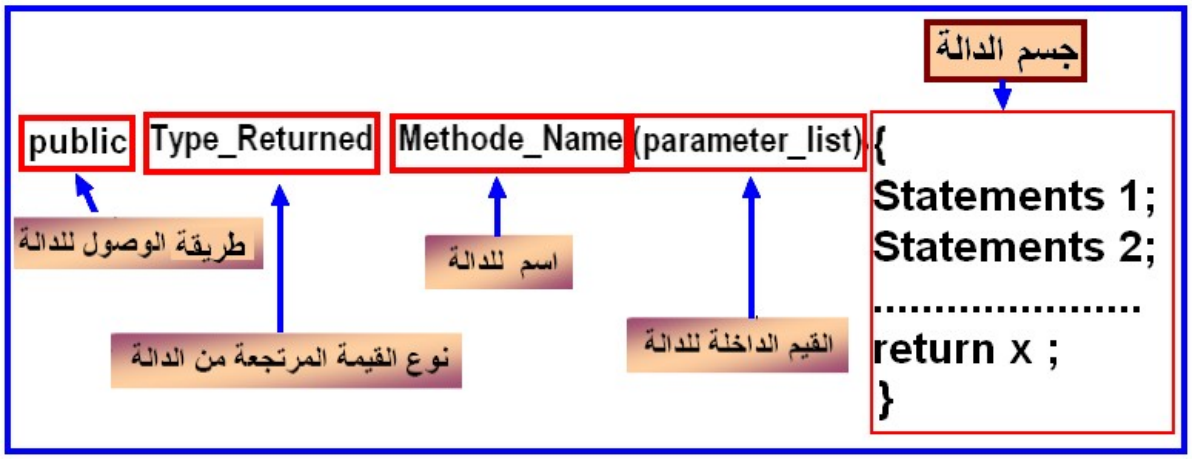
في هذا البرنامج تم الأعلانالإعلان عن متغير من نوع (double number) double.
ثم يتم ادخالالإدخال الرقم من لوحة المفاتيح وفي سطري الطباعة. السطر الأول يقوم بطباعة الجذر التربيعي أما السطر الثاني فيقوم بطباعة مربع هذا الرقم .

وعند تنفيذ البرنامج تظهر المخرجات كالتالكالتالي :

```
Enter a Number
9
The square Root is = 3.0
The Square number is =81.0
```

2-5-1- الدوال يمكن تعريفها عن طريق المستخدم :

الشكل العام للدالة



وفيما يلي شرحيلي شرح الشكل العام للدالة:-

✓ **طريقة الوصول للدالة**

ويوجد ثلاث طرق نذكرها كالتالي :كالتالي:

1- : public أي عامة أي تستطيع الوصول اليها من خارج الكلاسالفصيلة ومن خارج البرنامج أيضا .أيضا.

2- private Private : أي خاصة أي خاصة فتستطيع الوصول للدالة من داخل الكلاسالفصيلة فقط ,فقط. ولا يمكن ان تصل اليها من خارج الكلاس .الفصيلة.

3- protected Protected : أي محمي ,محمي. ايأي انك أنك تستطيع الوصول للدالة من داخل الكلاسالفصيلة أو من خارج الكلاسالفصيلة (وهذا يدعم موضوع الوراثة).

أما عبارة static Static التي نجدها في معظم البرامج فهي من أجل إخبار المترجم أن هذه الدالة من نوع ثابت أي يتعرف عليها المترجم قبل الدخول للدالة الرئيسية .الرئيسية.

✓ **وهناك نوعان من الدوال كالآتي** :

1- نوع يعود بقيمة .

وفي هذا النوع لابد من استخدام العبارة return كالشكل العام الذي رأيناه سابقا.

وكمثال على الدالة التي تعود بقيمة:بقيمة :

```
Public public int
( )getDay
}
;return day
```

وهنا نرى أن الدالة المعرفة تعود بالتاريخ وهو قيمة ولذلك تم وضع عبارة .return .return

2- نوع لايعودلا يعود بقيم method methode.-void

ويكون تعريف هذا النوع كالأتيكالآتي :

```
Public public void method_
name(parameter_list)
}
<list of statements>
```

ونلاحظ انهأنه في هذا النوع لم يتم استخدام عبارة return. return.
وكمثال لهذا النوع:النوع :

```
( )Public public void writeoutput
}
;System.out.println(month +" "+day + " " + year)
{
```

وهنا نلاحظ ان أن عبارة الطباعة لاتعودلا تعود بأي قيم للبرنامج الرئيسي ولذلك تم استخدام void ايأي دالة لاتعودلا تعود بقيم وكذلك لم يتم استخدام العبارة return .

✓ **العبارة return**

وتوجد في نهاية الدالة وهي تجعل البرنامج يعود في مساره بعد انتهاء تنفيذ الدالة.الدالة.

والصيغة العامة لهذه الدالة كالتالي :

```
Return return
```

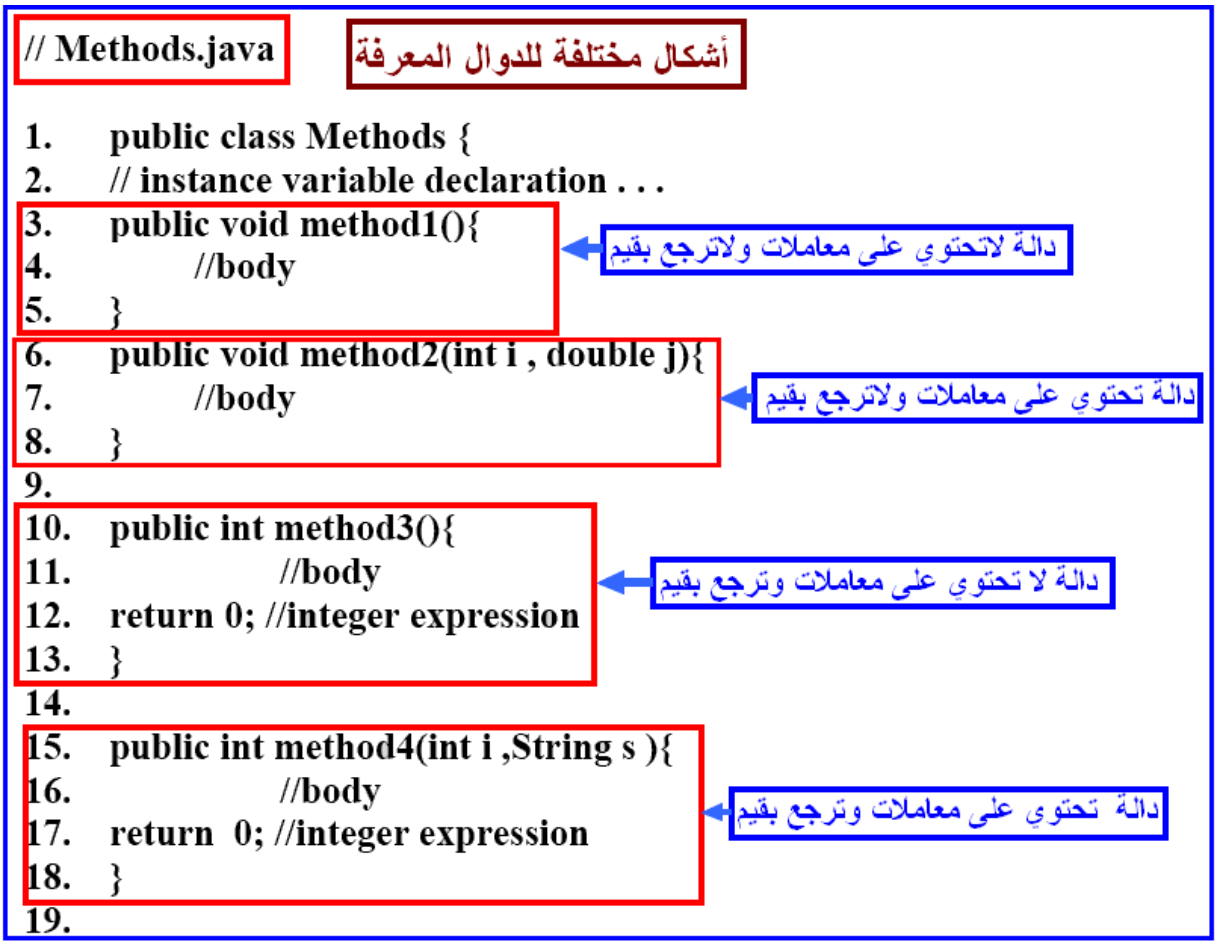
وكمثال على هذه العبارة

```
Public public int
(  )getYear
}
; return year
```

 لاحظ

استخدام العبارة return بدون أي اقواسأقواس في الدوال من نوع void يمكن أن يتسبب في إنهاء البرنامج في الحال .

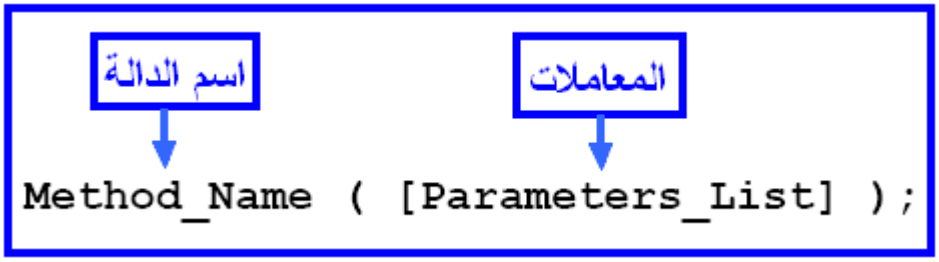
والشكل التالي (1-17) يوضح اشكال أشكال الدوال التي يمكن تعريفها :



شكل(1-17)

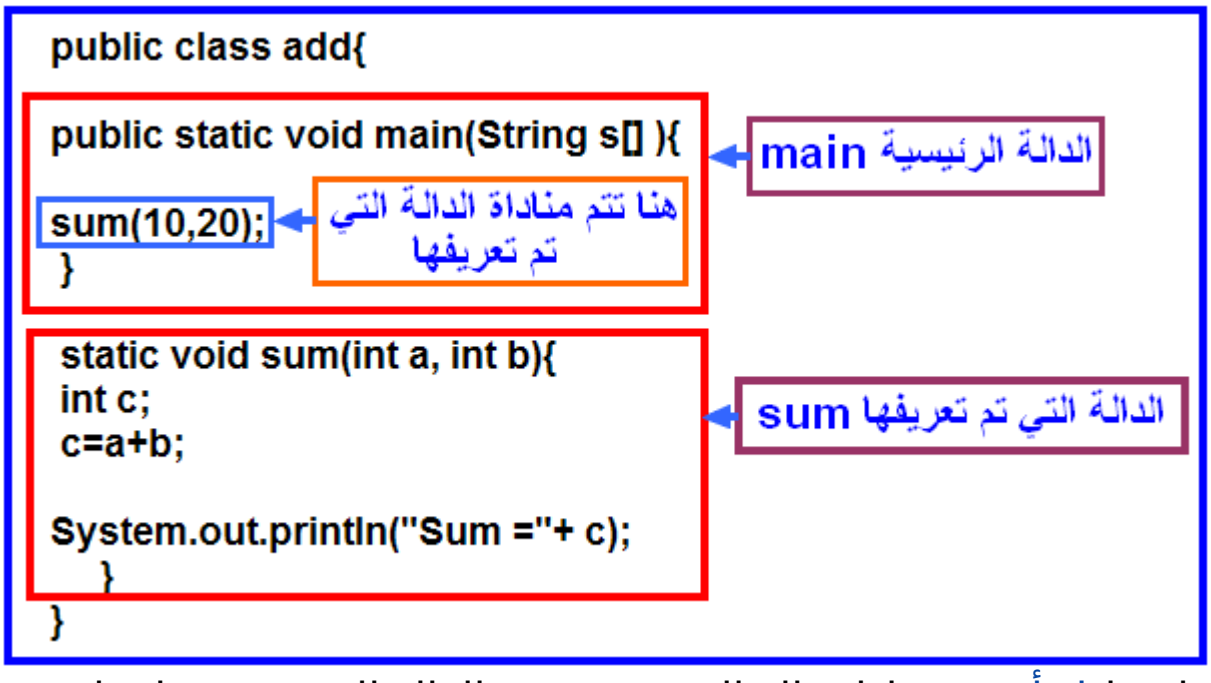
كيفية استدعاء الدوال داخل البرنامج

يمكن استدعاء الدالة داخل أي مكان في البرنامج عن طريق كتابة اسمها وارسال قيم المعاملات إن وجدت . والصيغة العامة ~~لأستدعاء~~لاستدعاء الدالة كالآتي :



مثال (20) :

كيفية كتابة دالة بسيطة تقوم بعملية الجمع وطريقة استدعاؤها.~~ استدعائها~~.



ونلاحظ ~~انهأنه~~ تم مناداة دالة الجمع sum في الدالة الرئيسية مع اسناد معاملات لها .

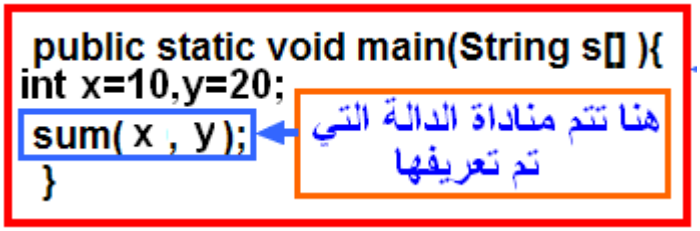
Sum(10,20)

ونلاحظ في جسم الدالة الفرعية التي تقوم بعملية الجمع أنها من نوع void. أي ~~لاتعودلا تعود~~ للدالة الرئيسية بأي قيم~~-~~, بل ينتهي دورها بمجرد طباعة ناتج الجمع. ونلاحظ كذلك أنها ~~لاتنتهيلا تنتهي~~ بعبارة return -. لماذا؟

ويلاحظ أن معاملات الدالة المستدعاة عبارة عن متغيرين (a,b) من النوع int كما تم تعريف متغير آخر داخل جسم الدالة الفرعية وهو (c) ليتم تخزين ناتج الجمع به .

وعند تشغيل البرنامج يتم طباعة ناتج الجمع وهو هنا (30) .

كما يمكن تغيير القيم الداخلة إلى الدالة الفرعية بمتغيرات (x,y) كالتالي :

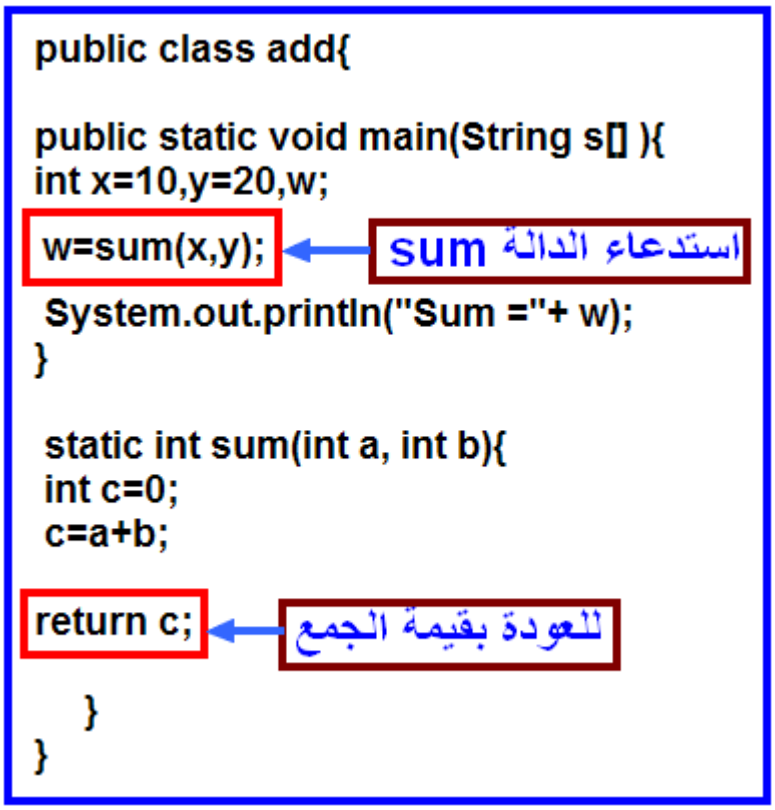


ولا يتم تغيير شيء في الدالة الفرعية .

مثال (21):

استخدام دالة تعود بقيم .

ولذلك لم نستخدم void وتم استخدام العبارة return .



ومن المؤكد عند تنفيذ البرنامج سوف يتم طباعة حاصل الجمع (30) .

1-6-المصفوفات (المنظومات) Array .

في الحقيقة وقبل أن نبدأ في شرح المصفوفات نسأل أنفسنا أولاً: لماذا استخدمت طريقة المصفوفات ؟

وللأجابةوللإجابة على هذا السؤال نرجع إلى تعريف المتغيرات .

فالمتغير كما هو معروف يستخدم في تخزين البيانات سواء كانت هذه البيانات حروفًاأ أم ارقاماًأرقامًا . فمثلا لو افترضنا أن هناك متغيرا من النوع الصحيح يسمى a)) وبه قيمة معينة فأننا كنا نعلن عنه هكذا :هكذا:

```
Int int a=3;
```

ولكن ماذا لوكنا سنتحدث مثلا عن درجات خمس طلاب وكل طالب له درجة معينة ففي هذه الحالة سوف نحتاج خمس متغيرات .ولو فرضنا أن الدرجات من النوع الصحيح فأننا سنعلن عن هذه ...عطيها قيماً كالتالي :

```
Int int
a1=80;

Int int
a2=90;

Int int
a3=60;
```

فهنا يمكننا فعلاً الإعلان عن خمس متغيرات واعطائهم القيمة المطلوبة.

ولكن ماذا نفعل لو أن هناك مائة طالب أو الف طالب مثلا ؟ هل سنعلن عن كل هذه المتغيرات في البرنامج ؟ فيمكن ان نتخيل حجم البرنامج وكيفية

فهمه وتصحيحه اذا تم الأعلانالإعلان بالطريقة العادية. ولذلك كله تم الأستعانةالاستعانة بالمصفوفات .

والمصفوفات تعتبر من نوع المتغيرات المرجعية RefrenceReference variables .

- **تعريف المصفوفة**

المصفوفة هي عبارة عن مخزن يحمل عدد محدد من القيم Values لمتغيرات Variables من نفس النوع type. وهذا النوع يمكن ان يكون (int , float , string ,....) ويتحدد سعة هذا المخزن (المصفوفة) عند الأعلانالإعلان عنها وبعد الأعلانالإعلان عن المصفوفة وتحديد طولها (عدد المتغيرات التي ستخزنها) فإن هذا الطول يظل ثابتا ولايمكنلا يمكن تحميل المصفوفة بعناصر اكثرأكثر من سعتها .

وكل عنصر في المصفوفة array يسمى element ويمكن الوصول لهذا العنصر في المصفوفة عن طريق فهرس رقمي index .

- **أنواع المصفوفات :**

يوجد نوعان من المصفوفات :

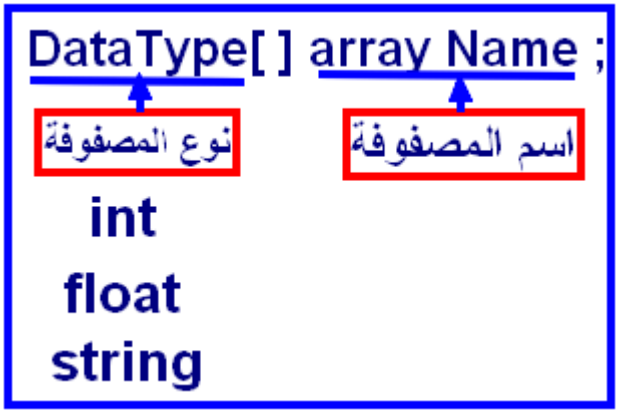
1- المصفوفة الأحادية :الأحادية: وهي مكونة من بعد واحد فقط.

2- المصفوفة متعددة الأبعاد: وهي مكونة من عدد من الصفوف والأعمدة (ليس شرطا ان تكون بعدين) .

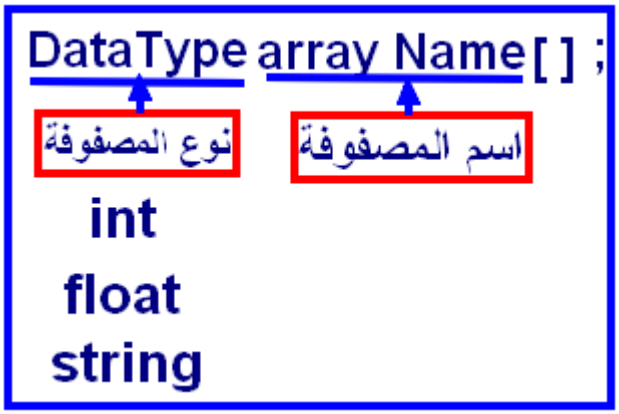
وسوف نتناول بالشرح , المصفوفة ذات البعد الواحد , والمصفوفة ذات البعدين .

1.6.1 المصفوفة ذات البعد الواحد.الواحد

والصيغة العامة للأعلانللإعلان عن المصفوفة ذات البعد الواحد كالتالي-:



أو يمكن الأعلانالإعلان عنها بنفس الصيغة السابقة مع وضع الأقواس بعد اسم المصفوفة كالتالي-:

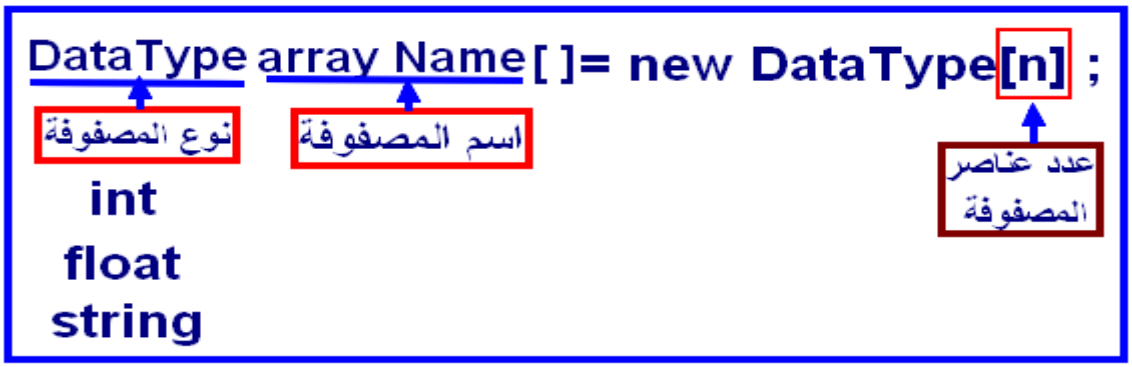


فمثلا يمكن الأعلانالإعلان عن المصفوفة ذات البعد الواحد كالتالي :

وسوف نستخدم الصيغة الثانية String name[]) float degree[] int ;

(; [ ]degree

وبعد الأعلانالإعلان عن المصفوفة لابد من تحديد عدد عناصرها ويتم ذلك كالتالي-:



فمثلا لعمل مصفوفة رقمية من النوع int خاصة بدرجات عشرة طلاب مثلاً مثلًا يتم ذلك كالتالي :كالتالي:

```
; [] Int int degree
;degree[ ]= new int[10]
```

أو يمكن الأعلانالإعلان عن المصفوفة وتحديد عدد عناصرها في سطر واحد كالتالي-:

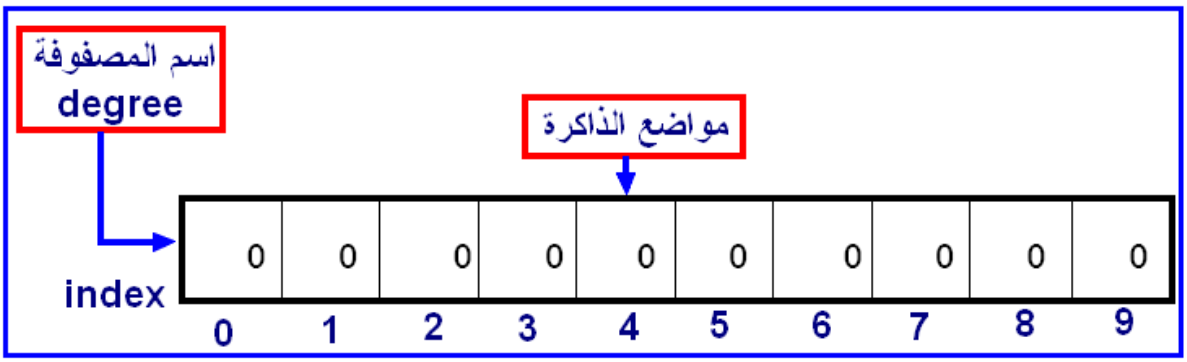
```
Int int degree[ ]=new
```

وهذا السطر يخبر الحاسب بحجز عشرة اماكن أماكن لمصفوفة ذات بعد واحد من النوع int وتسمى degree .

وكما قلنا سابقًا يمكن كتابة الصيغة السابقة كالتالي :كالتالي::

```
Intint[ ] degree=new
```

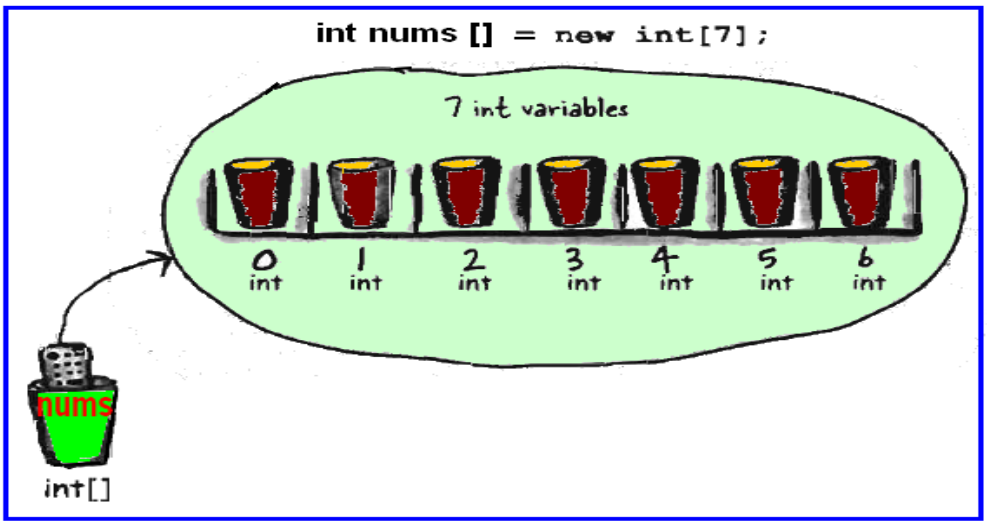
والحقيقة انهأنه بعد تحديد عدد عناصر المصفوفة يتم حجز 10 مواضع في الذاكرة لتخزين الأرقام الصحيحة التي سيتم إدخالها ويبدأ الترقيم في الذاكرة من الصفر كالتالي:كالتالي:



وكما قلنا ان عناصر المصفوفة عبارة عن متغيرات يتم تخزينها في الذاكرة , وأن المتغير عبارة عن وعاء يتم تخزين القيم به والمثال التالي يوضح هذا المفهوم :

```
int nums[]=new int
```

وهذا معناه حجز عدد (7) أماكن (أوعية) في الذاكرة تمهيداً لتخزين قيماً بها كالآتي:كالآتي:



و يمكن تخزين قيماً (أعداد صحيحة) في هذه الأوعية كالتالي:

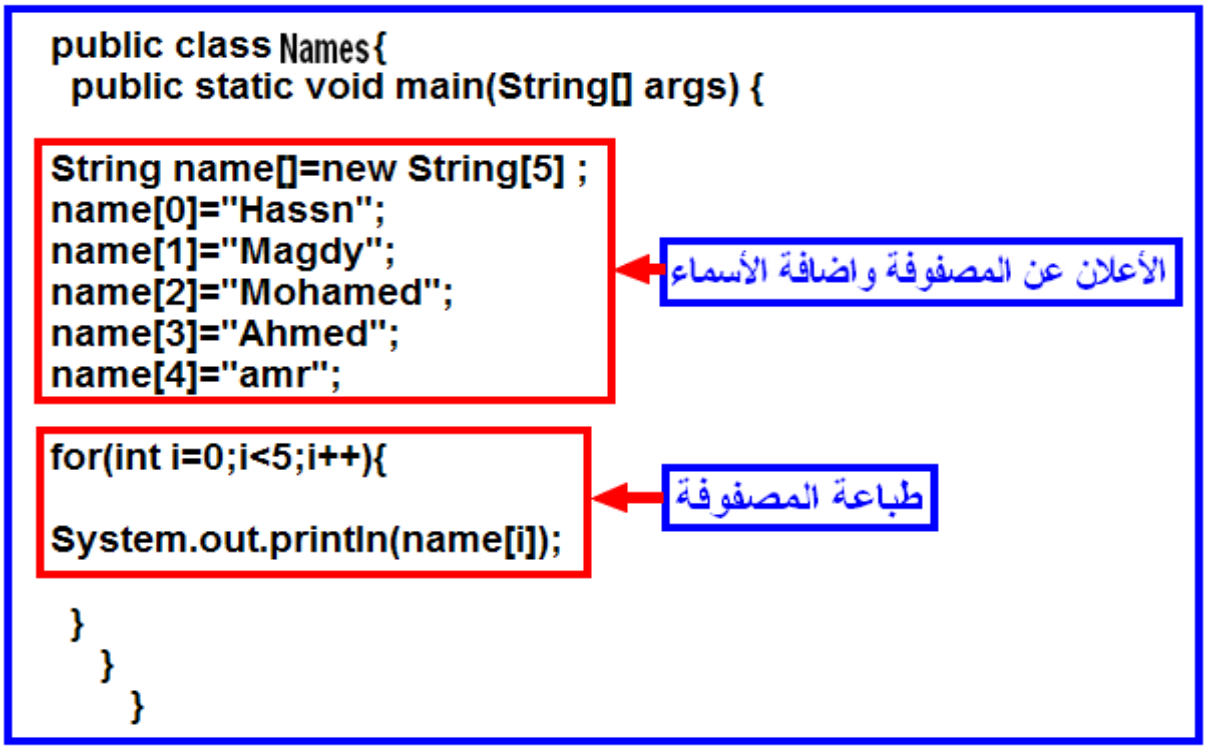
```
nums[0] = 6;
nums[1] = 19;
nums[2] = 44;
nums[3] = 42;
nums[4] = 10;
nums[5] = 20;
nums[6] = 1;
```

فمثلاً العنصر رقم (0) يمكن اعطاؤه الرقم (6) , ورقم (5) يمكن اعطاؤه القيمة (20) , وهكذا ....

ومن المؤكد أن المصفوفات لاتتعاملا تتعامل فقط مع الأرقام بل يمكنها تخزين الحروف والكلمات.

والبرنامج الآتي يبين ذلك:ذلك :

نفرض انناأننا نريد تخزين عدد (5) أسماء ثم طباعتهم فيتم عمل ذلك كالأتيكالآتي :

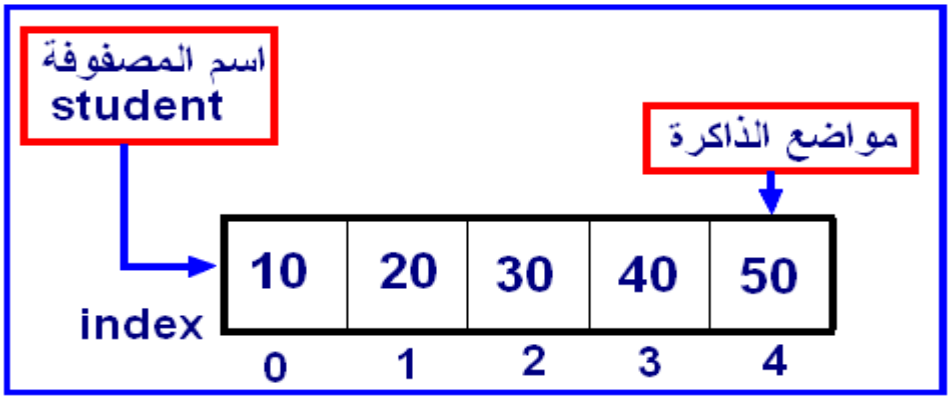


ونلاحظ هنا اننا أننا استخدمنا الحلقة التكرارية (for) لطباعة عناصر المصفوفة وتكون المخرجات كالتالي:

ويمكن اعطاء المصفوفة قيماً قيمًا ابتدائية كالتالي :

; Int int student[]={10,20,30,40,50}

فهذه المصفوفة تم اعطاؤها 5 قيم مسبقة ويتم تخزينها في الذاكرة كالتالي :



ولطبع الرقم 30 الموجود في الخانة 2 يتم كتابة الأمر التالي:التالي
;System.out.println(student[2])

ونلاحظ مما سبق أنه إذا لم نحدد قيمًاقيمًا ابتدائية للمصفوفة فيجب أن نستخدم كلمة (new) لحجز مواقع للمصفوفة كما أوضحنا سابقاأوضحنا سابقا.

مثال (22)

المطلوب عمل مصفوفة ذات بعد واحد تحتوي على درجات خمس طلاب وطباعة الناتج على الشاشة.الشاشة

نفترض انأن درجات الطلاب (10، 20، 30، 40، 50) .و البرنامج كالتالي:كالتالي

```
import java.util.Scanner;
public class array{

public static void main(String s[] ){
    int student[]={10,20,30,40,50};

        for(int i=0;i<=4;++i)
              System.out.println(student[i]) ;

    }
}
```

الأعلان عن المصفوفة واسناد قيم لها

حلقة تكرارية لقراءة محتويات المصفوفة

ونلاحظ هنا أنه تم عمل حلقة بجملة for لقراءة محتويات المصفوفة وطباعتها على الشاشة.

ونلاحظ كذلك أن نهاية العداد هو العدد 4 على الرغم من كونهم 5 عناصر.

لماذا ؟

وضح ماذا يحدث لو جعلنا نهاية العد إلى الرقم 5 `

وتكون مخرجات البرنامج كالتالي:

```
10
20
30
40
50
```

مثال (23)

مطلوب كتابة برنامج يقوم بعمل مصفوفة حروف تقوم بطبع أيام الأسبوع على الشاشة كالآتي:

```
import java.util.Scanner;
public class array{

public static void main(String s[] ){

String student[]= {"saturday","sunday","monday","tuseday",
                   "wedensday","thrusday","friday"};

for(int i=0;i<=6;++i)
      System.out.println(student[i]) ;

    }
}
```

مصفوفة حرفية String

حلقة تكرارية لقراءة محتويات المصفوفة

ونلاحظ في هذا البرنامج أنه تم الأعلانالإعلان عن مصفوفة من النوع الحرفي String type لأن عناصر المصفوفة عبارة عن حروف.

ويجب أن نلاحظ أن كلمة String يجب أن يكتب أول حرف فيها بحرف كبير capital .capital.ثم يتم تنفيذ أمر الطباعة داخل الحلقة لطباعة أيام الأسبوع .الأسبوع. وهنا يتبادر إلى ذهننا سؤالاً سؤالاً:

ماذا لوقلت قيمة نهاية العد عن 6 ؟ ماذا لو اصبحت 4 مثلا ؟

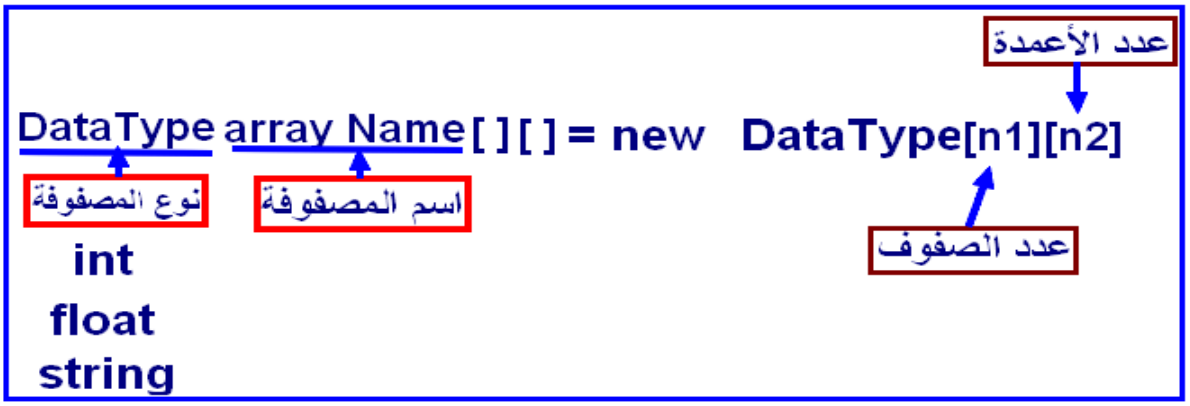
وماذا لو زادت هذه القيمة عن 6 ؟ ماذا لو اصبحت 8 مثلا ؟

ونترك لك عزيزي الطالب التفكير واستخلاص النتائج .النتائج.

1-6-2- المصفوفة متعددة الأبعاد (ذات البعدين) Multidimensional array

ويمكن القول بأن المصفوفة ذات البعدين هي عبارة عن جدول يحتوي على صفوف وأعمدة .

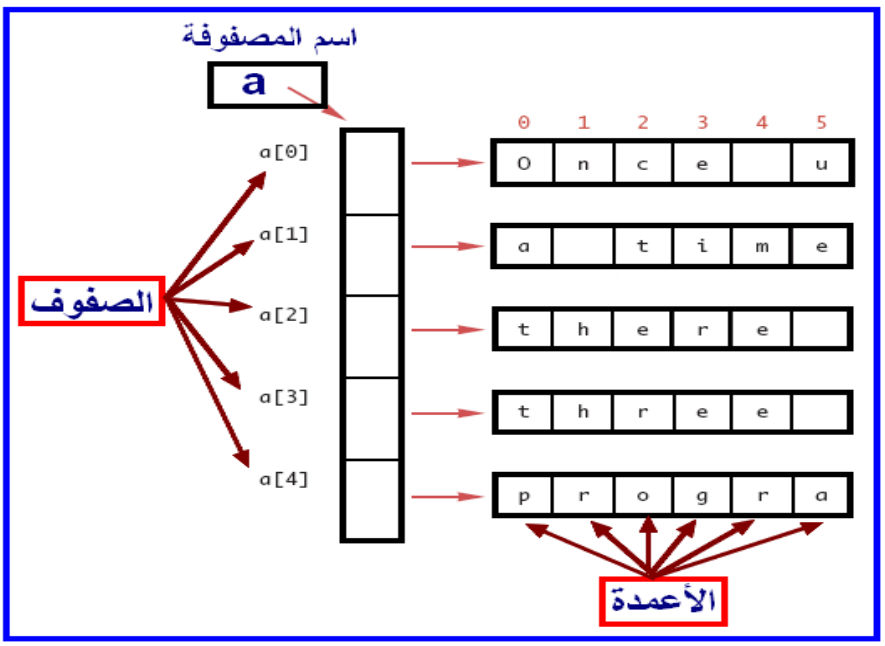
والصيغة العامة لهذه المصفوفة كالتالي :



فمثلاً اذا كان هناك مصفوفة تم تعريفها كالآتي :

ولفهم طريقة ترتيب العناصر في هذه المصفوفة نفرض أن لدينا مصفوفة(a) حروف ~~كالآتي:~~كالآتي:

; ~~Chr~~ char a[][]=new char[5][6]

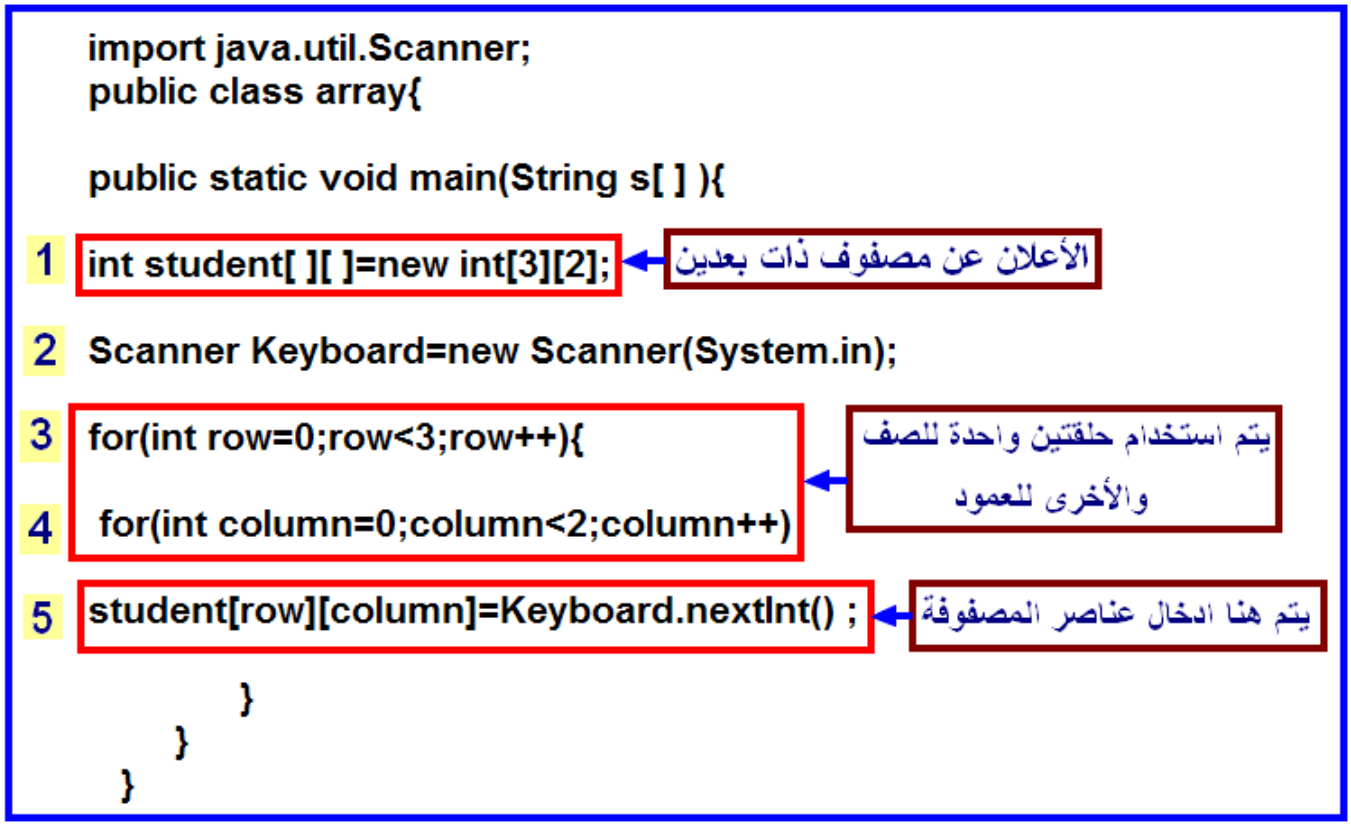


فإذا أردنا مثلاً مثلاً أن نعرف محتويات المصفوفة في الموقع [2][1]a)) فنجد أنه حرف (t).(وكذلك الموقع ([2][3]a) فنجد أنه الحرف (r) وهكذا.وهكذا. وبطبيعة الحال يمكن اعطاء هذا النوع من المصفوفات قيماً إبتدائيةابتدائية كما سبق ورأينا في المصفوفة ذات البعد الواحد , ولكننا هنا في المصفوفة ذات البعدين سوف نتعرف على كيفية إدخال القيم من لوحة المفاتيح .

✓ كيفية ادخالالإدخال العناصر للمصفوفة

لنفرض أن هناك مصفوفة ذات بعدين يراد فيها إدخال درجات 6 طلاب عن طريق لوحة المفاتيح يتم ذلك كالتالي-:

سنقوم بتسمية المصفوفة student وسوف نستخدم دالة الإدخال Scanner لإدخال قيماً صحيحة إلى هذه المصفوفة وهذا هو شكل البرنامج:البرنامج:



1- في السطر الأول يتم الأعلانالإعلان عن مصفوفة ذات بعدين من النوع int وعدد عناصرها 6 عناصر.

2- السطر الثاني سبق وتم شرحه في جملة الأدخالالإدخال.

3- السطر الثالث والرابع تم عمل حلقتين , الحلقة الأولى الخارجية للإشارة إلى رقم الصف row والحلقة الثانية للإشارة إلى رقم العمود row column.
وطبعا لاحظ ان الحلقة الخارجية قد تم إضافة اقواسأقواس لها . لماذا ؟

4- أما السطر الخامس فيتم استخدام دالة الإدخال كما شرحناها سابقاًسابقًا.
ولكننا هنا استخدمنا المصفوفة student كمتغير student كمتغير يتم تخزين عناصر المصفوفة المدخلة فيه.

وعند تشغيل البرنامج لن تظهر لك أي رسالة . لأننا لم نضف رسالة للأدخالللإدخال يمكنك انتأنت اضافتها

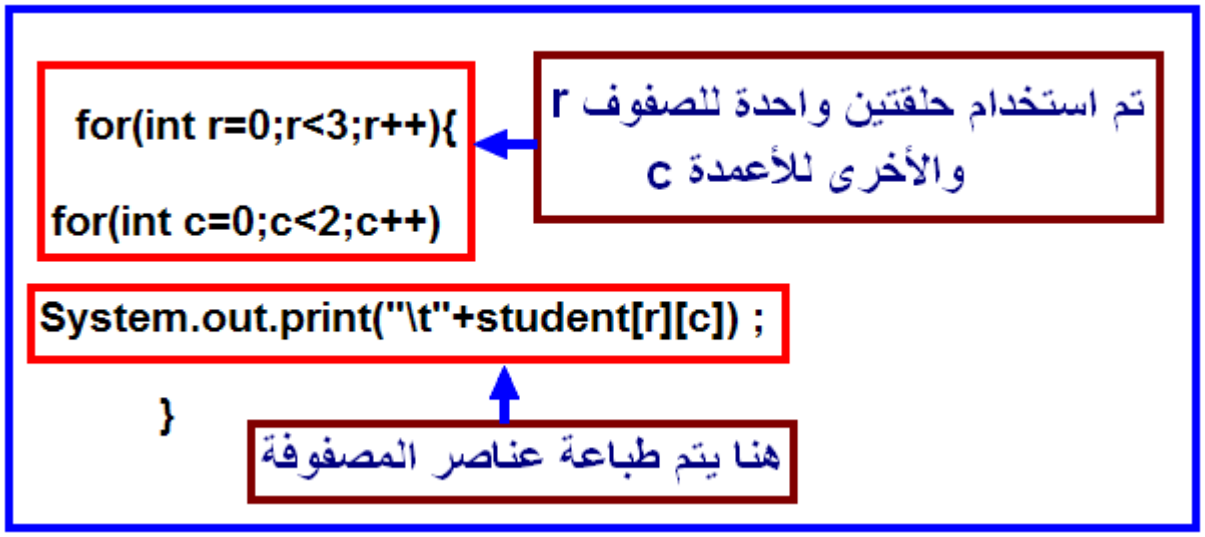
ولكن, سيظهر المؤشر في أقصى يسار الشاشة منتظرا الإدخال قيم عناصر المصفوفة

فيتم الإدخال 6 عناصر والضغط على مفتاح الإدخال كل مرة.

وبعد تمام الإدخال للستة عناصر تظهر علامة المحث للدوس دلالة على انتهاء الإدخال.

✓ كيفية قراءة عناصر المصفوفة

بعد أن يتم الإدخال عناصر المصفوفة واجراء أي عمليات عليها كالعمليات الحسابية مثلاً, يهمنا أن نرى المخرجات على الشاشة. وفي هذا البرنامج سوف نتعرف على كيفية قراءة عناصر المصفوفة وطباعة هذه العناصر كما هي على الشاشة دون أي تغيير, ويتم ذلك عن طريق البرنامج التالي:



ونلاحظ اننا استبدلنا جملة الطباعة print الإدخال في البرنامج السابق إلى بجملة الإدخال في البرنامج السابق print الطباعة. ونلاحظ اننا أضفنا علامة (t\) وذلك لتنسيق الطباعة على الشاشة. فتظهر عناصر المصفوفة على سطر واحد لماذا ؟.

وبينها مسافات متساوية لماذا؟

مما سبق يتبين لنا انهأنه لابد من جمع البرنامج الأول (ادخالالإدخال عناصر المصفوفة) والبرنامج الثاني

وهو طباعة عناصر المصفوفة مع بعضهما ليظهر برنامج واحد متكامل للأدخاللللإدخال والأخراجوالإخراج كالتالي

```
import java.util.Scanner;
public class magdy{
public static void main(String s[ ] ){

int student[ ][ ]=new int[3][2];
 Scanner Keyboard=new Scanner(System.in);
   for(int row=0;row<3;row++){

  for(int column=0;column<2;column++)

     student[row][column]=Keyboard.nextInt() ;

        }

  for(int r=0;r<3;r++){
        for(int c=0;c<2;c++)

System.out.print("\t"+student[r][c]) ;

           }
       }
   }
```

هنا يتم ادخال عناصر المصفوفة

هنا يتم طباعة عناصر المصفوفة

تطبيقات الباب الأول

1- ماهي الأصداراتالإصدارات المختلفة للغة الجافا ؟
2- ماهي مميزات لغة الجافا ؟
3- وضح مع الرسم كيف أن لغة الجافا لاتعتمدلا تعتمد على نظام التشغيل في الأجهزةأجهزة المختلفة .

4- قم بتظليل الأجاباتالإجابات الصحيحة فقط مما يأتي :

1 – تكتب رأس الدالة الرئيسية للبرنامج كالآتي :كالآتي:

```
○ A. public static void main(string[] args)
○ B. public static void Main(String[] args)
○ C. public static void main(String[] args)
○ D. public static main(String[] args)
○ E. public void main(String[] args)
```

2- أي العبارات الآتية صحيحة :

○ A كل سطر في البرنامج يجب أن ينتهي بفاصلة منقوطة
○ B كل جملة في البرنامج يجب أن تنتهي بفاصلة منقوطة
○ C كل سطر ملاحظة يجب أن ينتهي بفاصلة منقوطة
○ D كل دالة يجب أن تنتهي بفاصلة منقوطة
○ E كل كلاس (فئة) يجب أن تنتهي بفاصلة منقوطة

3 - أي العبارات الآتية تقوم بطباعة العبارة (Welcome to Java) .يمكن أن تختار أكثر من اختيار.

```
☐ A. System.out.println('Welcome to Java');
☐ B. System.out.println("Welcome to Java");
☐ C. System.println('Welcome to Java');
☐ D. System.out.print('Welcome to Java');
☐ E. System.out.print("Welcome to Java");
```

4- أذا أردنا ترجمة الملف المسمى (Test.java) فأننا نقوم بكتابة الآتي في سطر الأوامر :الأوامر:

```
○ A. java Test
○ B. java Test.java
○ C. javac Test.java
○ D. javac Test
○ E. JAVAC Test.java
```

5ـ - إذا افترضنا أن هناك كلاس فصيلة تمت تسميتها كما يلي :

} public class Test

{

فأنه فإنه بعد عملية الترجمة ينتج ملف بأسم :باسم:

```
○ A. Test.class
○ B. Test.doc
○ C. Test.txt
○ D. Test.java
○ E. Java أي اسم له امتداد
```

6- أي السطور الآتية لاتعتبرلا تعتبر سطور ملاحظات comment يمكن اختيار اكثرأكثر من اجابة :

- ☐ A. /** comments */
- ☐ B. // comments
- ☐ C. -- comments
- ☐ D. /* comments */
- ☐ E. ** comments **

7- أي من الكلمات الآتية تعتبر من الكلمات المحجوزة في لغة الجافا (يمكنك اختيار اكثرأكثر من اجابة) ؟

- ☐ A. public
- ☐ B. static
- ☐ C. void
- ☐ D. class

8- كل عبارات لغة الجافا يجب أن تكتب بحروف صغيرة :

- ○ A. true
- ○ B. false

9- أي اسماء المتغيرات الآتية صحيحًاصحيحًا. يمكن أن تختار أكثر من اجابة :

- ☐ A. radius
- ☐ B. Radius
- ☐ C. RADIUS
- ☐ D. findArea
- ☐ E. FindArea

10- أي الطرق الآتية تستخدم في الأعلانالإعلان عن المتغيرات (يمكن اختيار اكثرأكثر من اجابة) ؟

- ☐ A. int length; int width;
- ☐ B. int length, width;
- ☐ C. int length; width;
- ☐ D. int length, int width;

11- بفرض ان x=1 ما هي قيمة x بعد تنفيذ x+=2 :

- A. 0
- B. 1
- C. 2
- D. 3
- E. 4

12- ماهي قيمة X بعد تنفيذ العملية الآتية ؟ إذا كانت :

```
int x = 1;
x *= x + 1;
```

- A. x is 1;
- B. x is 2;
- C. x is 3;
- D. x is 4;

13- ماهى نتيجة تنفيذ البرنامج التالى ؟

```
public class Test1 {
   public static void main(String[] args) {
     int x = 1;
     int y = x = x + 1;
     System.out.println("y is " + y);
   }
}
```

- y is 0
- y is 1 لأن x ساوت y أولا .
- y is 2 لأن x +1 ساوت x أولا ثم بعد ذلك تم مساواة x=y.
- المترجم سوف يعطي خطأ عند الترجمة لأن x تم اعادة تخصيصها في العبارة y=x=x+1 .

14- ماهي النتيجة التي سوف يتم طبعها على الشاشة ؟

```
public class Test {
   public static void main(String[] args) {
     int x = 1;
     int y = x + x++;
     System.out.println("y is " + y);
   }
}
```

- y is 1.
- y is 2.
- y is 3.
- y is 4.

15- أي العبارات الآتية تقوم بطباعة الآتي (Ahmed\exam1\test.txt) :):

- System.out.println("Ahmed\exam1\test.txt");
- System.out.println("Ahmed\\exam1\\test.txt");
- System.out.println("Ahmed\"exam1\"test.txt");
- System.out.println("Ahmed"\exam1"\test.txt");

16- بفرض اننا أننا نريد ادخال الإدخال قيمة عدد صحيح من لوحة المفاتيح عن طريق استخدام العبارة الآتية

Scanner input = new Scanner(System.in);

ماهي الطريقة المستخدمة فيما يلي لقراءة العدد الصحيح :

- input.nextInt();
- input.nextInteger();
- input.int();
- input.integer();

17- أكتب برنامج يقوم بطباعة الشكل

```
*****
****
***
**
*
*
**
***
****
*****
```

18 – ماهوما هو ناتج تنفيذ البرنامج التالي ؟

```
char ch = 'a';

switch (ch) {
  case 'a':
  case 'A':
    System.out.print(ch); break;
  case 'b':
  case 'B':
    System.out.print(ch); break;
  case 'c':
  case 'C':
    System.out.print(ch); break;
  case 'd':
  case 'D':
    System.out.print(ch);
}
```

- abcd
- a
- aa
- ab
- abc

19- أكتب برنامج يقوم بطباعة الأعداد الفردية على الشاشة في صف واحد بدءاً بدءًا من (1 إلى 50) (50).

20- أاكتب برنامج يقوم بعمل مقارنة بين مصفوفتين من النوع char ,إذاchar, إذا كانت كلًا من المصفوفتين تحتويالمصفوفتين تحتوي على القيم الآتية :الآتية:

{'d','h','r','f'}.

21 – قم بحساب قيمة المضروب لعدد صحيح يتم ادخاله إدخاله من لوحة المفاتيح.

22- قم بحساب مجموع القيم التالية بإستخدامياستخدام مصفوفة من النوع int :

((10 ,90 ,57 ,34 ,55