pdf
Lcd:

/ /

المحاضرة الأولى

البرمجة غرضية التوجه بلغة Java

object oriented pRograming (OOP)

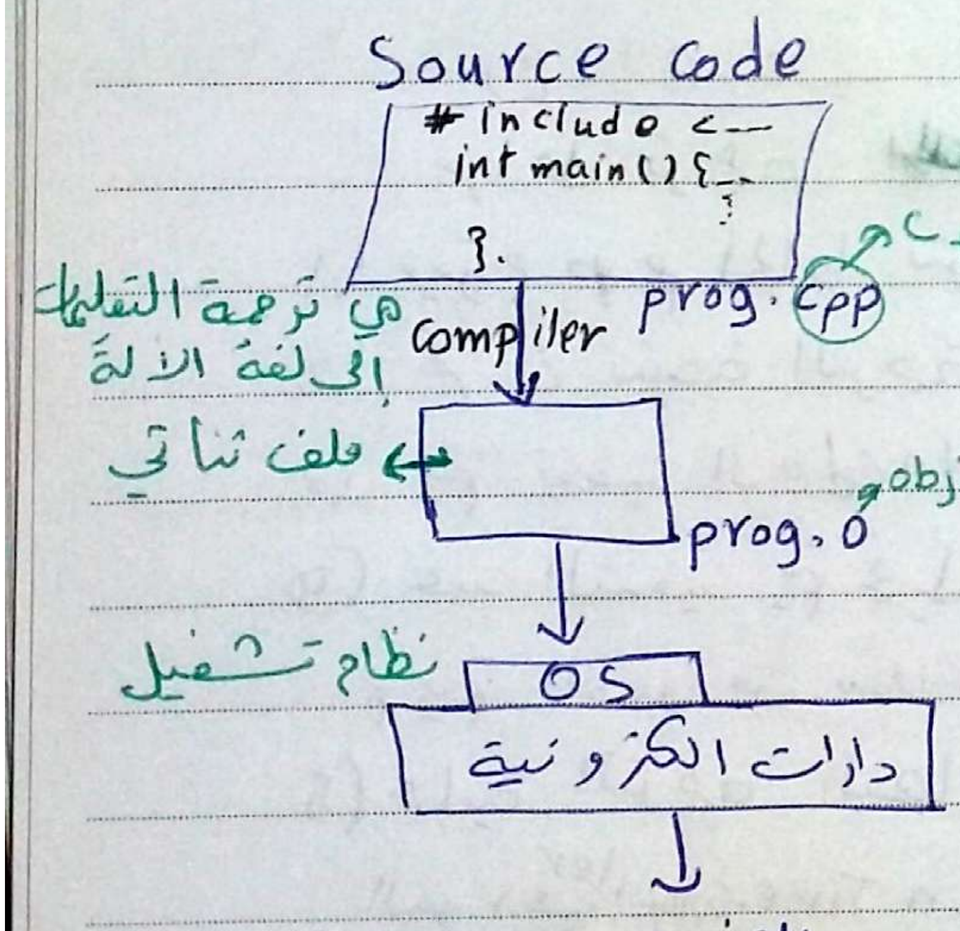
مقدمة - ظهرت لغة Java في عام 1990، نظراً للتطور الكبير في تقنية صناعة الحواسيب وانتشارها في جميع مجالات الحياة المختلفة، واستخداماتها المتعددة في شتى المجالات، فأصبح علياً لزاماً معرفة هذه الحاسبات وكيفية التعامل معها والاستفادة منها لأنها توفر الجهد والوقت وتنجز الكثير من الأعمال بدقة كبيرة بالإضافة لقدرتها على الاحتفاظ بالبيانات، ومن الطرق الشائعة للاستفادة من القدرات للحواسيب هو بناء البرامج التي تقوم بحل الكثير من المشكلات.

لغة Java: وهي لغة برمجية غرضية التوجه ونقية 100٪ وهي لغة محمولة portable. أي صممت كلغة غرضية التوجه.

معنى غرضية التوجه: أي تساعدنا في تنظيم البيانات وترتيبها بشكل أقرب إلى الواقع والعنصر الرئيسي للبرمجة غرضية التوجه هو الغرض object وتساهم في تطوير وصيانة البرامج بشكل أسرع وأرخص وتسهل إجراء النمذجة.

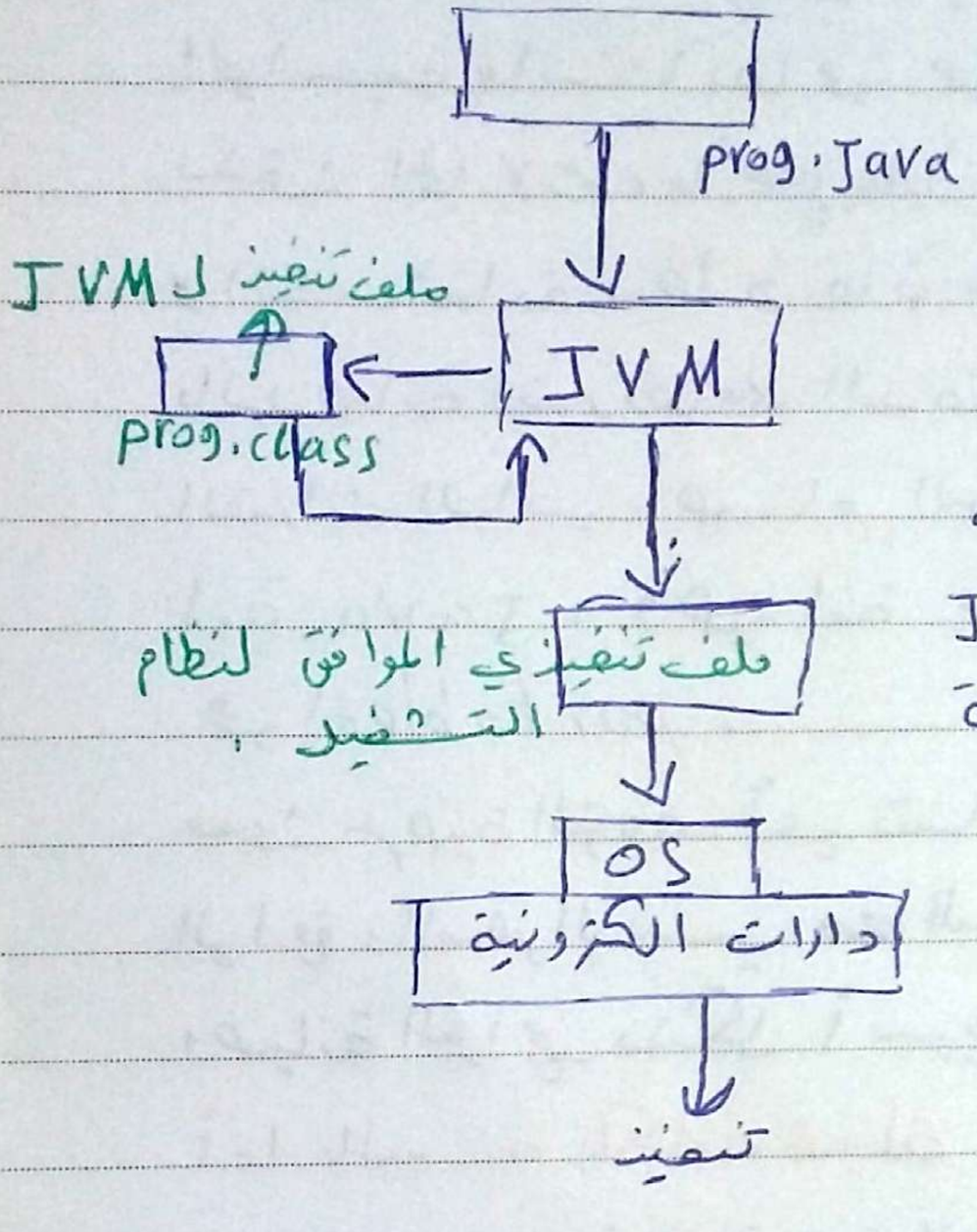
أما بالنسبة لكلمة محمولة

لغة C++ نقوم ترجمة البرنامج إلى شكل يناسب بنية الحاسوب ونظام التشغيل الذي نجري عليه الترجمة مما يجعل البرامج قابلة للتنفيذ على هذا الحاسوب والحواسيب التي تماثله من حيث البنية ونظام التشغيل.



وهنا إذا أخذ البرنامج التنفيذي وحاولنا أن ننفذه على حاسب آخر فإنه لن يعمل إلا في حال تطابق (بنيته والهندسة ونظام التشغيل) بين الحاسبين وهنا تكون اللغة C++ هي لغة غير محمولة على عكس لغة Java

إما لغة Java : تجعل برامجها قابلة للتنفيذ على جميع الحواسيب دون النظر إلى نمية الحاسوب و نظام التشغيل.



Java virtual Machine (JVM) :
هي عبارة عن معالج افتراضي يحول مجموعة تعليمات مكتوبة بـ Bytecode إلى لغة الآلة التي يفهمها معالج الآلة التي تشغل برنامج Java وهذه التعليمات ناتجة عن ترجمة الملفات ذات اللاحقة Java و ينتج عن هذه الترجمة ملفات ذات اللاحقة class والتي تحوي Bytecode وتكون نتيجة تنفيذ هذه التعليمات عبر JVM هي ناتج تنفيذ برنامج Java

شرح إضافي :

مراحل ترجمة وتنفيذ البرامج :

1) يقوم مترجم الجافا بترجمة البرامج إلى مرحلة byte code
2) تخزن نتيجة الترجمة تمهيداً لتنفيذها على حاسوب معين
3) يتم تنفيذ الـ byte code في JVM.
4) عند التنفيذ يتم تحميل صفوف جافا المترجمة مرحلياً بواسطة class loader ثم يتم التحقق من سلامة الصف وعدم مخالفته لأي من القيود على أمنية المعلومات
5) عملية الترجمة النهائية إلى شكل قابل للتنفيذ تتم من خلال مترجم جافا الفوري Just-In-Time-compiler الموجود على الحاسوب الذي تريد تنفيذ البرامج عليه.

JVM نكتب مرة البرنامج و شغلوين فايدك على أي جهاز

المحاضرة الثانية. / /

الشكل العام للصف في Java.

```
Class  class name {
        [ معطيات اعضاء ]
        [ طرق اعضاء ]
}
```

لكتابة الدالة Main هي دالة رئيسية ويجب أن تكون ضمن صف خاص بها

```
class My main {
public static void main (String [ ] args)
{
} // end of main
} // end of My main class
```

الصف يبدأ بحرف كبير وهي اتفاقية مبرمجين لا تؤثر على البرنامج إن كتبناها بحرف صغير

public: تعني أن التعريف الحدد متاح لأي كائن

البدخال والإخراج:

- تعليمة الطباعة. هي: `System.out.print( );` أو `System.out.println( );`

تبدأ بحرف كبير لأنها صنف

ملاحظة: الوسطاء هم عبارة عن String وهي عبارة عن محارف.

+ هي عملية جمع أعداد وإذا استخدمت مع سلاسل محرفية هي دمج لها.

توضيحات بالمحاضرات القادمة

- تعليمة البدخال، لا توجد تعليمة دخل واضحة لأن ذلك يعطي الحرية أكبر لكل من المبرمجين والمستخدمين.

stdin = standard input

للمبرمج سنفرض له لدينا الصنف stdin ويحتوي الطرق التالية:

```
int readInt( );
```

لقراءة عدد صحيح x فقط نكتب

```
int x = Stdin.readInt( );
```

ملاحظة للمحاضرات القادمة:

لا يوجد فرق إن كتبنا (String[] args) أو (String args[])

لدينا نعتبر args هو متحول String .

مثلاً: لا يؤثر [] إن كتبت قبل أو بعد المتحول A

```
int [ ] A ;
int A [ ] ;
```

تمرين:

اكتب برنامج يقوم بإدخال عددين صحيحين ومجموعهما؟

```
Class My main {
public static void main (string org [ ]){
    int x, y, z ;
    x = stdin . read Int( ) ;
    y = stdin . read Int ( ) ;
    z = stdin . read Int( ) ;
    z = x + y ;
        System . out . print ("z =" + z ) ;
    } // end of main
} // end of Mymain
```

System . out . print ("z=" (+x+y)) يمكننا أن نكتب بتعليمة الطباعة

نضع +x+y بين قوسين لتغيير أولوية العمليات فيقوم بإجراء العملية التي ضمن القواس ثم يطبعها .