الأكاديمية العربية الدولية
Arab International Academy

إسم المادة: مدخل إلى لغة البرمجة - JAVA

إسم المدرب : المهندس عبدالقادر عبدالله

الأكاديمية العربية الدولية – منصة أعد

محتويات المحاضرة :

- تعريف عام بالبرمجة
- تعريف عام بلغة JAVA
- مجالات استخدام لغة JAVA
- اهم المحررات للتعامل مع لغة JAVA
- امر الطباعة في اللغة واهميته
- المتغيرات وانواعها
- العمليات الحسابية
- الجملة الشرطية والاحوال المتعددة
- الحلقات المتكررة
- المصفوفات
- الدوال وانواعها
- التعامل مع الفئات
- مدخل إلى البرمجة كائنية التوجه

تعريف عام بالبرمجة

قبل تعريف البرمجة بشكل مباشر لابد من الإشارة إلى وظيفة الحاسوب الاساسية , والمتصلة بشكل مباشر بتسميته , وذلك أنه يمتلك سرعة عالية جدا بالقيام بالعمليات الحسابية مقارنةً بالإنسان .

للتمكن من استخدام هذه السرعة العالية لابد من توجيه امر او عدة اوامر للحاسوب من شأنها أن تخبره بما يجب عليه أن يفعل وما النتيجة التي يجب الوصول إليها

البرمجة هي عملية ترتيب وتوجيه هذه الاوامر للحاسوب بشكل منظم بهدف بناء برنامج يقوم بأداء مهمة واحدة او عدة مهام .

يهدف الإنسان من خلال البرمجة إلى توظيف سرعة الحاسوب العالية لتنفيذ خوارزميات معينة .

تتم عملية البرمجة باستخدام إحدى لغات البرمجة والتي يختارها المبرمج مسبقا لتنفيذ الخوارزميات وبناء البرامج

تعرف لغات البرمجة على انها لغات التواصل بين الإنسان والآلة وتنقسم من حيث التصنيف إلى لغات عالية المستوى واخرى ضعيفة المستوى

تعريف عام بلغة JAVA

لغة جافا هي لغة برمجة كائنية التوجه عالية المستوى , وهي احد اكثر لغات البرمجة شيوعا.
تم ابتكارها عام 1992 في مختبرات Sun microsystems بواسطة جيمس غوسلينغ

تتميز لغة جافا بعدة ميزات جعلتها احد اكثر اللغات البرمجية إنتشارا ومنها :

- تدعم مفهوم البرمجة الشيئية (البرمجة كائنية التوجه).
- سهولة الحصول عليها.
- امنة.
- قابلة للنقل والتنفيذ.
- إضافة الحركة والصوت إلى صفحات الويب.
- كتابة الألعاب والبرامج المساعدة.
- إنشاء برامج ذات واجهة مستخدم رسومية.

مجالات استخدام لغة JAVA

تستخدم لغة جافا في معظم الصناعات البرمجية ومنها :

- تطبيقات الويب
- تطبيقات سطح المكتب
- الالعاب
- تطبيقات الجوال
- تطبيقات الساعات الذكية والتلفاز الذكي

كما أن البرامج التي تعتمد على لغة جافا تعمل على جميع انظمة التشغيل وذلك لأن لغة جافا مستقلة عن نظام التشغيل وتتمتع بمشغل خاص يدعى : آلة جافا الافتراضية – Java VM

اهم المحررات للتعامل مع لغة JAVA

NetBeans

IntelliJ IDEA

Eclipse

امر الطباعة في اللغة واهميته

```
11   */
12  public class HelloWorld {
13
14      /**
15       * @param args the command line arguments
16       */
17      public static void main(String[] args) {
18          System.out.println("Hello world !");
19      }
20
21  }
22
```

يعد امرالطباعة هو الطريقة التي تساعد المبرمج على عرض النتائج الاولية للعمليات البرمجية والخوارزميات التي قد قام بصياغتها
ويعد احد اكثر الاوامر البرمجية استخداما خصوصا خلال رحلة التعلم

يقل استخدام امر الطباعة مع التقدم في اللغة ولكنه يبقى في مرحلة اختبار الرمز البرمجي واختبار اتصال الفئات والكائنات المختلفة امر لا يمكن الاستغناء عنه

المتغيرات وانواعها

المتغيرات هي عناوين للتخزين في الذاكرة

للمتغيرات عدة انواع فمنها ما يحفظ العدد الصحيح ومنها ما يحفظ العدد العشري ومنها ما يحفظ الرموز او الاحرف وسلاسلها لتشكيل النصوص ومنها ما يحفظ قيمة منطقية

يتم تخزين هذه القيم في الذاكرة بهدف استدعائها عدة مرات في البرنامج عند الحاجة

في لغة جافا يتم تعريف المتغير بعد تحديد نوعه واسمه ثم إعطاء قيمة إبتدائية مع العلم أن القيمة ممكن أن يتم تحديدها لاحقا.

```
public static void main(String[] args) {
    int intVar = 10  ;
    float floatNumber = 1.2f;
    double doubleNumber = 1.254;
    String stringVar ="String";
    boolean bool = true;
}
```

العمليات الحسابية

مثل العديد من اللغات البرمجية تتمتع لغة جافا بسهولة القيام بالعمليات الحسابية كونها قريبة من لغة البشر (عالية المستوى).

وتجب الإشارة إلى أن العمليات الحسابية تخضع للقوانين الرياضية العامة
حيث أن الاولوية للاقواس ثم للضرب والقسمة واخيرا للجمع والطرح

في البرمجة يمكن إسناد قيمة للمتغير بناءً على قيمته القديمة مما يعني أنه من الممكن أن تجد امر برمجي بالصورة التالية :

```
X = X + 1 ;
```

حيث تصبح هنا قيمة X الجديدة تساوي قيمتها القديمة + 1

الجملة الشرطية والاحوال المتعددة

تعد الجملة الشرطية واحدة من اهم المفاهيم البرمجية بشكل عام والتي يمكن تنفيذها باستخدام لغة جافا بسهولة

يعتمد الشرط بشكل اساسي على قيمة منطقية (حقيقي – غير حقيقي) ويتم الحصول على هذه القيمة اما بشكل مباشر (متغير من نوع boolean)

او عبر اجراء مقارنة بين متغيرين او اكثر

```
    */
    public static void main(String[] args) {
        int intVar = 10  ;
        if (intVar == 10) {// تحقق من قيمة المتغير في هذه الحالة يتم ارجاع قيمة حقيقي
            // نفذ امر برمجي بعد التحقق من الشرط
        }

}
```

ممكن اضافة عدة شروط متصلة ببعضها باستخدام عبارات شرطية مثل

else if

else

للتحقق من اكثر من حالة في ذات الوقت

الجملة الشرطية والاحوال المتعددة

الأحوال المتعددة (Switch Case) في لغة جافا هي عبارة تحكم تُستخدم لتنفيذ كود بناءً على قيمة متغيرة محددة. تُعتبر بديلاً أكثر تنظيماً من استخدام جمل الشرط المتعددة (if-else) في حالة التعامل مع خيارات متعددة. يتم تحديد قيمة المتغير داخل جملة switch، وتُستخدم جملة case لتحديد الإجراءات المطلوبة لكل قيمة. يتم استخدام جملة breakللخروج من الكتلة بعد تنفيذ الكود المطلوب، بينما يمكن استخدام (default) لتنفيذ كود إذا لم تتطابق أي من القيم مع الحالات المعطاة.

```
int day = 3;
switch (day) {
    case 1:
        System.out.println("الأحد");
        break;
    case 2:
        System.out.println("الاثنين");
        break;
    default:
        System.out.println("يوم غير معروف");
        break;
}
```

في المثال على اليسار تعتمد عبارة التحكم على المتغير الذي يعبر عن رقم اليوم وبناءً عليه سيتم تحديد هذا اليوم
في حالة أن المتغير يحمل قيمة لا تعبر عن يوم محدد مثل 15 تقوم عبارة التحكم بالتوجه إلى الحالة الغير معرفة والتي سيتم تنفيذها عند عدم توافق المتغير المدخل مع الخيارات المتاحة

الحلقات المتكررة

الحلقات المتكررة في لغة Javaتُستخدم لتنفيذ مجموعة من التعليمات بشكل متكرر طالما أن شرطًا معينًا محقق.
هناك ثلاثة أنواع رئيسية من الحلقات:
حلقة (for) : تُستخدم عندما تعرف عدد التكرارات مسبقًا، وتتيح التحكم في المتغيرات المستخدمة في التكرار.

حلقة (while) : تستمر في التكرار طالما أن الشرط المحدد صحيح، وتستخدم غالبًا عندما لا يكون عدد التكرارات معروفًا.

حلقة (do- while) : تشبه حلقة while، ولكنها تضمن تنفيذ التعليمات مرة واحدة على الأقل حتى لو كان الشرط خاطئًا.

تُستخدم الحلقات لتقليل التكرار اليدوي للكود وللتعامل مع البيانات بشكل أكثر كفاءة.

المصفوفات

المصفوفات في لغة جافا هي هيكل بيانات يُستخدم لتخزين مجموعة من العناصر المتشابهة في النوع (مثل أرقام أو نصوص) في مكان واحد.

يتم تعريف المصفوفة بحجم ثابت عند إنشائها، ولا يمكن تغييره بعد ذلك.

يتم الوصول إلى العناصر عبر الفهارس التي تبدأ من الصفر.

المصفوفات مفيدة لتخزين بيانات متكررة مثل قائمة درجات أو أسماء، ويمكن معالجتها بسهولة باستخدام الحلقات.

مثال:

```
int[] numbers = {1, 2, 3, 4};
```

يمكن تعريف المصفوفة بطرق مختلفة وممكن ان تكون من انواع مختلفة وهي من اهم المفاهيم التي يجب على المبرمج أن يتقنها.

الدوال وانواعها

الدوال في لغة جافا هي وحدات برمجية قابلة لإعادة الاستخدام تؤدي مهمة معينة.
تُعرَّف باستخدام الكلمة المفتاحية (void) إذا كانت لا تُرجع قيمة، أو باستخدام نوع البيانات إذا كانت تُرجع نتيجة.

هناك نوعان من الدوال:

دوال مُعرفة مسبقًا (Built-in Methods): مثل Math.sqrt() و System.out.println()، وهي جاهزة للاستخدام.
دوال مُعرفة من قبل المستخدم (User-defined Methods): يقوم المبرمج بإنشائها لأداء مهام محددة حسب الحاجة.

الدوال تساعد في تنظيم الكود وتسهيل إعادة الاستخدام والصيانة.

التعامل مع الفئات

الفئات في لغة جافا (Classes)هي القوالب التي تُستخدم لإنشاء الكائنات (Objects) ، وهي تُمثل أساس البرمجة الكائنية (OOP)

تحتوي الفئة على خصائص (Attributes) وسلوكيات (Methods) تمثل الكائن.

يتم إنشاء كائنات باستخدام الكلمة المفتاحية new، وتُعرَّف الفئات باستخدام الكلمة المفتاحية class

الفئات تُستخدم لتنظيم الكود وتحديد العلاقات بين الكائنات.

مثال على تعريف فئة:

```
32  class Car {
33      String model;
34      void drive() {
35          System.out.println("The car is driving.");
36      }
37  }
```

مدخل إلى البرمجة كائنية التوجه

البرمجة كائنية التوجه (Object-Oriented Programming - OOP) هي نموذج برمجي يركز على تنظيم الكود حول الكائنات، التي تمثل كيانات من العالم الحقيقي أو المجرد.

تعتمد (OOP) على أربعة مبادئ أساسية:

التغليف (Encapsulation) :جمع البيانات والطرق التي تتعامل معها في كيان واحد يسمى الكائن، مما يحمي البيانات من التعديل المباشر.
الوراثة (Inheritance) : تتيح لفئة ما أن ترث خصائص وسلوكيات من فئة أخرى، مما يُعيد استخدام الكود ويوفر هيكلية تنظيمية.
التعددية (Polymorphism) : السماح للكائنات بتنفيذ نفس الطريقة بطرق مختلفة، اعتمادًا على الفئة التي تنتمي إليها.
التجريد (Abstraction) : إخفاء التفاصيل المعقدة للكائن والتركيز فقط على الجوانب المهمة.

تساعد (OOP) في كتابة كود منظم، قابل للتوسع، وسهل الصيانة.