CLASE 1

Requisitos: Programación, conocimiento basico java
Es un curso para el certificado internacional en Oracle Java
Ing. Luis Bertel mail: lbertel@gmail.com

Oracle Java #1 (9 de Marzo de 2019)

Objetivos

Conocer el lenguaje de programación
Demostrar el conocimiento adquirido en JAVA
Implementar nivel intermedio de JAVA y conceptos orientados a objetos (OO)

Programación

Día 1

- Introducción: Qué es JAVA?
- Ejercicio carro de compras: Crear main class, data, arreglos.

Día 2

- Orientación a objetos y clases
- Manejar gestión de datos de un programa
- Uso de métodos

Día 3

- Encapsulación
- Condicionales

Día 4

- Arrays, Loops, Dates
- Herencia

Día 5

- Excepciones
- Interfaces: concepto de funcionalidad
- Como ejecutar los Labs (equipo de fútbol)

Al final del curso se necesita el aplicativo hecho!

Course Environment

Se trabaja en NeatBeans para las prácticas. La guía de estudiantes se usa con la consola.

Programación JAVA

Lo bueno es que funciona en cualquier sistema operativo. Además, es un lenguaje fuertemente orientado a objetos (de forma pura ver *SmallTalk*). Java toma el código fuente, lo compila y genera un **bytecode** (código interpretado y compilado); este bytecode lo ejecuta el **JRE** (Java Runtime Environment), ejecutándose primero en la Máquina Virtual (JVM Java Virtual Machine).

Paradigmas de programación:

Procedimiento o secuencial (programación imperativa)

- C es un lenguaje netamente imperativo, se le tiene que ordenar todo al programa. Ejemplos: Cobol, Fortran, Basic, C, Pascal
- Contras : Cuando es muy largo, mantenerlo es un problema. Programación spaghetti: muchos llamados. Difícil mejorarlo.
  Programación OO
- Small Talk es completamente OO. Otros -ejemplos: Smalltalk, Java, C#, Python, Typescript.
- No necesita secuencia prescrita.
- Pros: Modelar a partir de pequeños conceptos, es fácilmente mantenible. Se puede reusar código. Información oculta (encapsulamiento). Permite Modularidad (migrar fácilmente)
  Programación Funcional
- Inteligencia Artificial.
- Ejemplos: F#, Python.

Ambiente de Trabajo JAVA

Para *desarrollo*, usar JSE (Java Standard Edition) esto trae un JRE + un kit de desarrollo ó JDK (Java Development Kit).
Para *enterprise*, usar JEE (Java Enterprise Edition) esto trae un JSE + Herramientas de: web, persistencia (bases de datos), Rest/soa, RMI, sesiones, contenedor Aplicaciones, etc. Esto es para realizar desarrollos empresariales.

Instalador JDK

Path : Las carpetas donde se encuentran programas ejecutables (desde cmd), para ubuntu es cd

JAVA_HOME : Indica donde está la carpeta de java (JDK)

Comandos utiles windows = *dir* (archivos), *cd* (cambiar directorio), *cls* (clear)

RESUMEN

- Se describió la diferencia entre un lenguaje de bajo y alto nivel
- Se describió lo que significa independencia de plataformas
- Se describió cómo se compila y el formato que tira (Bytecode)
- Se explicó porque está orientado a objetos
- Se determinó la versión de JAVA
- Se usó javac para compilar y java ejecuta

Java CLASSES

Las clases son conceptos descritos a través de características o atributos, acciones o métodos.

The `main` Method

- JVM recnoce esto como punto de partida para cada programa de JAVA.
- La sintaxis es siempre la misma.

```
public static void main (String args[]) {
      // code goes here in the code block
}
```

Un ejemplo de la clase `main`

```
public class Hello{
  public static void main (String[] args){
    // Entry point to the program
    // args se puede poner otro nombre
    // Write code here
    System.out.println("Hello World")
  }
}
```

- **Syntax**: `System.out.println(<some string value>)`.
- **Fixing Sintax Errors**: No reconocer palabra, punto y coma, falta llave, etc.

RESUMEN

- Crear clases en Java
- Definir el metodo `main`

Variables

- Almacenar números, cadenas, certificado
- Concatenar Variables
- Asignación de múltiples Variables
- Declarar e inicializar variables `int` y `double`
- Modificar valores de variables usando operadores numéricos
- Cambiar el orden de las operaciones usando `()`

Definición

Una variable refiere algo que puede ser modificado. Pueden ser inicializadas con un valor que puede ser modificado. Retiene un específico tipo de datos

```
String firstName = "Mary";
// = is optional
```

Tipos de Variables

Algunos tipos de valores para las variables son:

- `String` tipo cadena
- `int`
- `double` es un float en Python
- `boolean` True or False.

Si no están inicializadas (con una asignación =)

- `String` null
- `int` 0
- `double` 0.0
- `boolean` False

Nombrando una variable

- Nombres son sensitivos a mayusculas.
- Nombres **no** pueden tener espacios.
- Escoger nombres que represente lo que van a almacenar.
  - `outOfStock` a boolean
  - `itemDescription` a string

Variable Tipo `String`

- Syntax `type identifier [=value];`
- Son inmutables.

Ejemplos:

```
String customer;
String name, city;
String address = "123 oak"
```

Concatenación `String`

- `String` pueden combinarse usando el operador "+".

Variable Tipo Numérico

- `int` comprende `-2.147.483.648 hasta 2.147.483.648`. Ej: `2,000, 2_000`
- `double` variable alcanza valores conteniendo decimales hasta `2.1E12`.
- Se puede convertir de uno a Otros

Operaciones Numéricas

Similar a Python.

- Operadores Unarios `++` y `--`
- Reglas de "precedence":
  1. Parentesis
  2. Operadores Unarios.
  3. Multiplicacion y division evaluados de izquierda a derecha.
  4. Adición y substracción evaluados de izquierda a derecha.

Tarea

Ejercicio de operaciones

Arreglos

- Expresiones de tipo boolean
- Condicionales `if /else`
- Describir el propósito de un arreglo
- Acceder a elementos de un arreglo.

Toma de Decisiones

```
if (<some condition is true>){
  // do something: Only True or False
}
else {
  // do something different
}
```

Expresiones Booleanas

- Tipos de dato `boolean` True - False
- Es una combinación de variables, valores y operadores que expresan una respuesta `true-false`.

Ej:

```
public class Condicionales{
  public static void main(String[] args) {
    boolean outOfStock = false;
    int quantity = 3;
    String message = "Zapato";

    if (quantity > 1) {
      message = message + "s";
    }
    if (outOfStock) {
      System.out.println("Artículo No Disponible");
    }else{
      System.out.println(quantity + " " + message + "precio:" + 345252);
    }
  }
}
```

Arreglos

Cuando queremos almacenar topicos del mismo tipo. Los arreglos en JAVA se representa por medio de `[]`.

Syntax `type[] arrayIdentifier = {comma-separated list of values}`;

Ej:

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

```
    //String[] names = {"Pedro", "Maria", "Juan", "Carlos"};
    String[] names = new String[4];
    names[0] = "Pedro";
    names[1] = "Maria";
    names[2] = "Juan";
    names[3] = "Carlos";

    //System.out.println("Hay" + names.length + " clientes")
    System.out.printf("Hay %d clientes", names.length);
    // %d data
    // %s strings
    // length nos dice cuantos elementos hay
    System.out.printf("En la posicion 1 esta %s", names[1])
  }
}
```

Iteración

Loops: Son usados para repetir bloques de estamentos.

Ej:

```
for (String name: names) {
  System.out.printf("Cliente: %s \n", name);
```