



UNIVERSIDAD TÉCNICA DE COTOPAXI
UNIDAD ACADÉMICA DE CIENCIAS DE LA INGENIERÍA Y APLICADA
CARRERA DE INGENIERÍA EN INFORMÁTICA Y SISTEMAS
COMPUTACIONALES

NOMBRE: Cristian Simba

CARRERA: Ingeniería en Sistemas **ASIGNATURA:** Programación Orientación a objeto

FECHA: 23-10 -2013

TEMA: Concepto, características, sintaxis y ejemplo de clase-objetos

OBJETIVO GENERAL

- Determinar los conceptos, características, sintaxis y ejemplos de clase-objetos mediante una investigación detallada en diferentes fuentes bibliográficas, para así adquirir nuevos conocimientos para resolver de mejor manera los programas propuestos en clase.

MARCO TEÓRICO

Definición de clase

Una clase, por otro lado, no es más que una plantilla genérica a partir de la cuál instanciar los objetos; plantilla que es la que define qué atributos y métodos tendrán los objetos de esa clase.

Es una abstracción que define un tipo de objeto especificando qué propiedades (atributos) y operaciones disponibles va a tener.

Definición de objeto

Un objeto es una entidad que agrupa un estado y una funcionalidad relacionados. El estado del objeto se define a través de variables llamadas atributos, mientras que la funcionalidad se modela a través de funciones a las que se les conoce con el nombre de métodos del objeto.

Los objetos son la clave para entender la tecnología orientada a objetos. Si mira a su alrededor ahora mismo se encontrará con muchos ejemplos de objetos del mundo real: su perro, su escritorio, su televisor, su bicicleta.

Características de clase

- Todas las variables y funciones de Java deben pertenecer a una clase. No hay variables y funciones globales.
- Si una clase deriva de otra (extends), hereda todas sus variables y métodos.
- Java tiene una jerarquía de clases estándar de la que pueden derivar las clases que crean los usuarios.
- Por defecto, las clases de usuario derivan de una clase de Java llamada Object.
- Sólo se puede heredar de una clase (en Java no hay herencia múltiple).
- En un fichero no puede haber más que una clase public. Este fichero se llama como la clase public con extensión *.java. Con algunas excepciones, lo habitual es escribir una sola clase por fichero.
- Si una clase contenida en un fichero no es public, no es necesario que el fichero se llame como la clase.
- Los métodos de una clase pueden referirse de modo global al objeto de esa clase al que se aplican por medio de la referencia this.
- Las clases se pueden agrupar en packages, introduciendo una línea al comienzo del fichero (package packageName;).

Características de objeto

- Los objetos se pueden agrupar en rubros (o tipos) denominados Clases
- El estado de los objetos está determinado por los datos del mismo
- Permite lo que se conoce como Ocultación de datos
- Pueden heredar propiedades de otros objetos
- Por medio de los Mensajes un objeto se puede comunicar con otro
- Los métodos definen el comportamiento de los objetos

Sintaxis de clase

```
classNombre_De_Clase {  
    tipo_de_variable nombre_de_atributo1;  
    tipo_de_variable nombre_de_atributo2;  
    tipo_devuelto nombre_de_método1(lista_de_parámetros ) {
```

```

        cuerpo_del_método1;
    }
    tipo_devuelto nombre_de_método2(lista_de_parámetros ) {
        cuerpo_del_método2;
    }
}

```

Sintaxis de objeto

```

classNombre_De_Clase {
    tipo_de_variable nombre_de_atributo1;
    tipo_de_variable nombre_de_atributo2;
    tipo_devuelto nombre_de_método1(lista_de_parámetros ) {
        cuerpo_del_método1;
    }
    tipo_devuelto nombre_de_método2(lista_de_parámetros ) {
        cuerpo_del_método2;
    }
}

```

EJEMPLO 1

Crear un programa para trabajar con la clase coche

```

public static void main (String args[]) {
    Coche c;
    c = new Coche();
    c.cilindrada= 1800;
    c.color = "plata";
    c.precio = 25000.0;
    System.out.println ("¿Qué coche es este?");
    c.verCoche();
    System.out.println("Vamos a arrancar el coche");
    c.arrancar();
    System.out.println ("¿Qué coche es este?");
    c.verCoche();
    System.out.println("Vamos a arrancar el coche");
    c.arrancar();
}

```

CLASE

```

class Coche {
    Integer cilindrada;
    String color;
    Double precio;
    booleanisArrancado;
    void arrancar() {
        if (isArrancado != true) {
            isArrancado = true;

```

```

System.out.println ("¡¡¡Brrrrrr!!! El coche acaba de arrancar.");
} else {
System.out.println ("¡¡¡Kisch, kishch!!! El coche ya está arrancado.");
}}
void verCoche() {
System.out.println("Este coche es un " + marca + " de color " + color + " y cuesta " + precio + "
euros.");
if (isArrancado == true) {
System.out.println ("y ahora mismo está arrancado.");
} else {
System.out.println ("y ahora mismo está parado.");
}}}

```

EJERCICIO 2

Crear una clase persona

```

public static void main(String[] args) {
    Persona P1=new Persona();
    P1.AsignarDatos("000555", "Pepito Cebollas",12, "Quito");
    Persona P2=new Persona();
    P2.AsignarDatos("", "",0, "");
    Persona P3=new Persona();
    P3.AsignarDatos("111444", "María Cecilia",20, "Mas allá.....");
    P1.ImprimirDatos();
    P2.ImprimirDatos();
    P3.ImprimirDatos();
}
}

```

CLASE

```

public class Persona {
    String cedula;
    String nombre;
    int edad;
    String direccion;
    public void AsignarDatos(String ced, String nom, int ed,String dir){
        cedula=ced;
        nombre=nom;
        edad=ed;
        direccion=dir;
    }
    public void ImprimirDatos(){
        System.out.println();
        System.out.println("Cedula:["+cedula+"]");
    }
}

```

```
System.out.println("Nombre:["+nombre+"]");  
System.out.println("Edad:["+edad+"]");  
System.out.println("Direccion:["+direccion+"]");  
}}
```

CONCLUSIONES:

- He determinado que un objeto que el estado del objeto se define a través de variables llamadas atributos, mientras que la funcionalidad se modela a través de funciones a las que se les conoce con el nombre de métodos del objeto.
- He concluido que una clase, por otro lado, no es más que una plantilla genérica a partir de la cuál instanciar los objetos; plantilla que es la que define qué atributos y métodos tendrán los objetos de esa clase.

RECOMENDACIONES:

- Repasar los programas visto anteriormente para no tener inconvenientes en el momento de programar en el lenguaje java y así adquirir nuevos conocimientos y alcanzar un mejor aprendizaje.

BIBLIOGRAFIA:

- <http://www1.ceit.es/Asignaturas/Informat2/Clases/Clases9899/Clase06/JavaClasesIntro/tsld003.htm>
- <http://es.scribd.com/doc/19789129/Los-Objetos-en-Java>
- <http://tutorial.saviashoft.com/java-basico/html/object.html>
- <http://www.monografias.com/trabajos42/manual-de-java/manual-de-java.shtml#6136#ixzz2iTgQp6tW>