Universidad Autónoma del Estado de Hidalgo

Área Académica de Computación y Electrónica
Licenciatura en Ciencias Computacionales
Programación Orientada a Objetos

Tema:
Estructura básica de una clase en Java

Elaboró:
M.C.C. Iliana Castillo Pérez

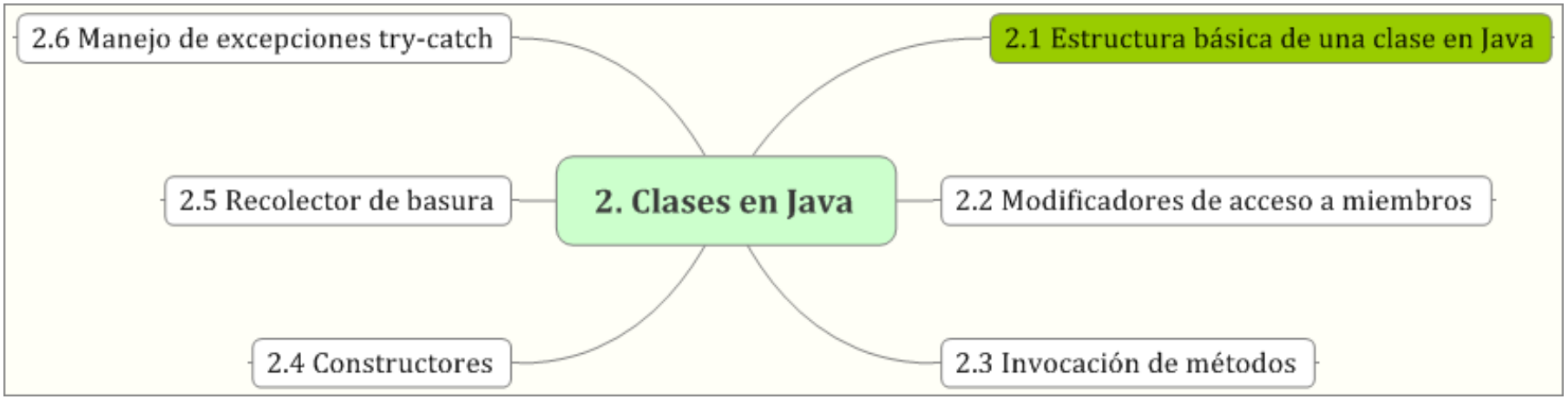
Fecha de elaboración: septiembre/2019

ICBI

Contenido

- *Resumen*
- *Palabras Clave*
- *Desarrollo del tema*
- *Conclusiones*
- *Bibliografía*

Unidad Temática
2. Clases en Java

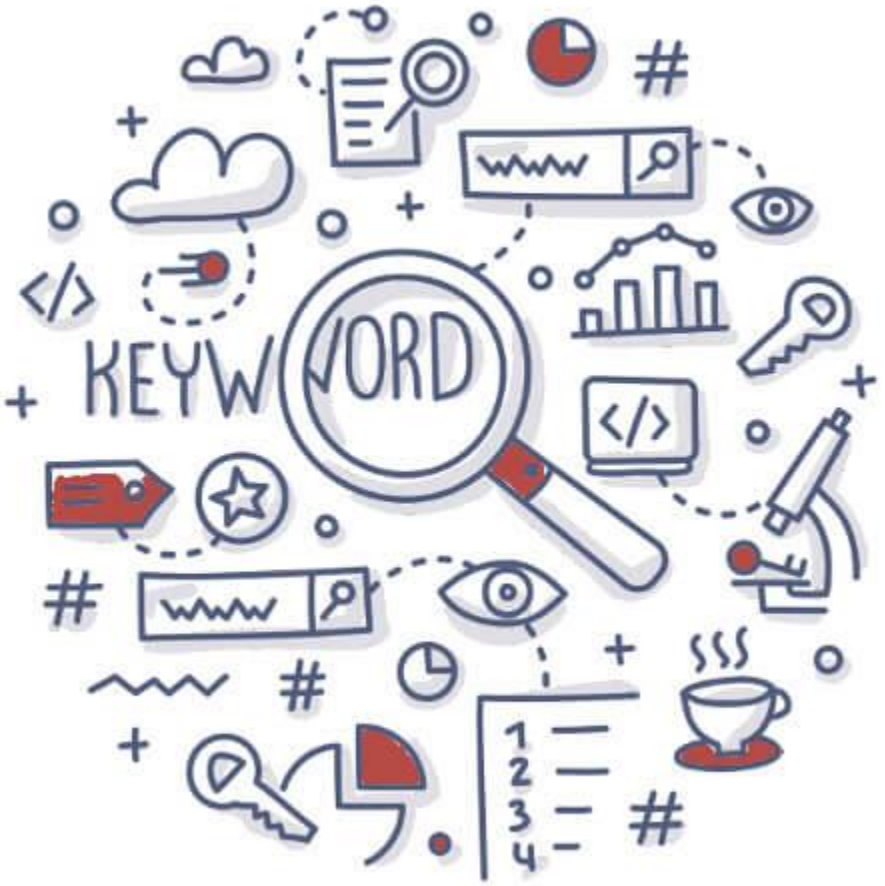


Resumen

“La Programación Orientada a Objetos (POO) es un paradigma de programación enfocado en la construcción de clases y objetos o instancias. Las clases definen tipos de estructuras de datos y las funciones que operan sobre esas estructuras de datos. Las instancias de estos tipos de datos son conocidas como objetos y pueden contener datos miembro y funciones miembro definidos por el programador. El lenguaje de programación Java permite construirlas y en estas notas los estudiantes encontrarán los conceptos principales para poder implementarlas”.

Palabras Clave

- *Clases*
- *Objetos*
- *Instancias*
- *Datos miembro*
- *Funciones miembro*
- *Miembros de la clase*



Objetivos

Objetivo Asignatura

- Desarrollar aplicaciones que apoyen la automatización de procesos, haciendo uso de lenguajes de programación, entornos y herramientas bajo el paradigma orientado a objetos.

Objetivo Unidad

- Identificar los elementos principales de una clase y programar clases de objetos que implementen los conceptos de programación orientada a objetos con el fin de modelar y resolver situaciones del entorno.

Objetivo Temático

- Identificar los elementos que conforman una clase en Java a partir del análisis de los mismos para su posterior implementación.

Objetivo de Aprendizaje

- Reconocer la estructura de una clase en el lenguaje Java para desarrollarla e implementarla en la solución de problemáticas definidas.

Resultados de aprendizaje

1. Reconocer la estructura de una clase en el lenguaje Java.
2. Codificar una clase en el lenguaje Java.
3. Instanciar la clase creada en una aplicación.
4. Compilar y ejecutar la clase y la aplicación desde la línea de comandos.

Introducción:

El paradigma de la Programación Orientada a Objetos se basa en la creación, uso y destrucción de objetos o instancias. Un **objeto** es un conjunto de atributos y métodos, los atributos describen al objeto de manera única y los métodos conforman el comportamiento que se le puede dar al objeto. La unión de este conjunto en un solo paquete es llamado **clase**.

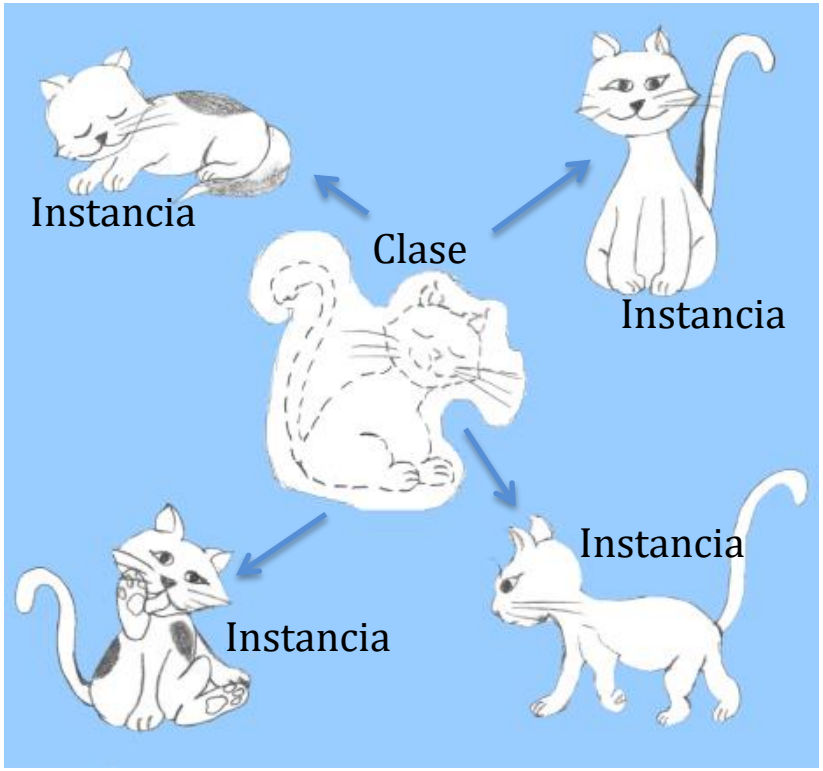
Definición:

Una **clase** se puede considerar como una plantilla para crear objetos de esa clase o tipo.

Una clase describe los métodos y atributos que definen las características comunes a todos los objetos de esa clase.

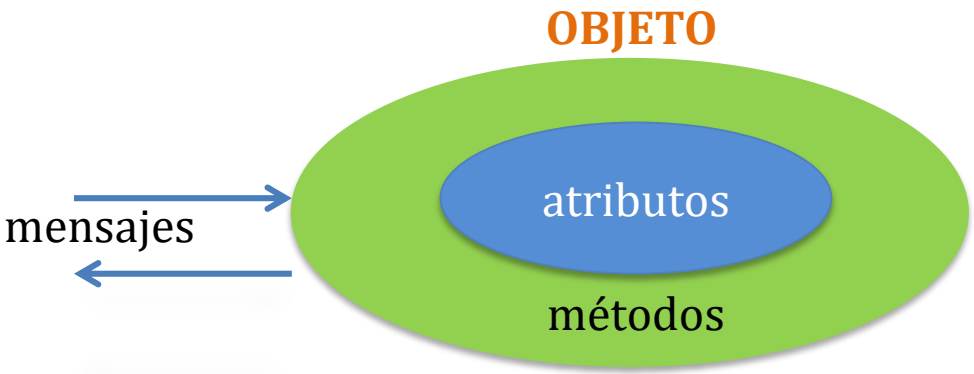
La clave de la Programación Orientada a Objetos está en abstraer los métodos y atributos comunes a un conjunto de objetos y *encapsularlos* en una clase.

Una *clase* es un tipo de objeto **definido por el usuario**. En otras palabras, una clase equivale a la generalización de un tipo específico de objeto. Un **objeto** es la concreción de una clase (*instancia*).

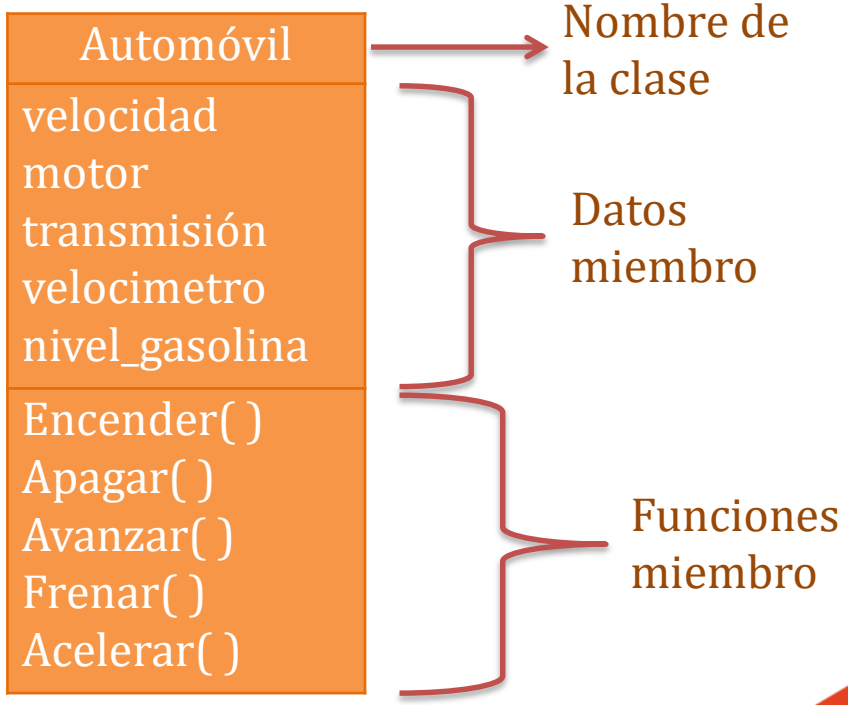


Clases de objetos:

Las **clases** tienen la propiedad de ocultar información, esto significa que aunque los objetos de una clase pueden saber como comunicarse entre sí, por lo general a las clases no se les permite saber como se implementan otras clases, los detalles de implementación están ocultos dentro de las mismas clases.



Las **clases** contienen datos así como un conjunto de funciones que manipulan esos datos. A los datos que componen una clase se les llama *datos miembro*. A las funciones que componen una clase se les llama *funciones miembro* (métodos).



Declaración de clases:

Por cada instancia en el dominio del problema se debe declarar una clase. La sintaxis para declarar una clase es:

[modificador] **class** Nombre_de_la_clase

Donde:

- El modificador determina el nivel de acceso que otras clases tienen hacia esta clase. El modificador es opcional.
- La palabra reservada ***class*** indica al compilador que el bloque de código contiene la declaración de una clase.
- El Nombre_de_la_clase es el nombre asignado a la clase.

Cómo nombrar una clase:

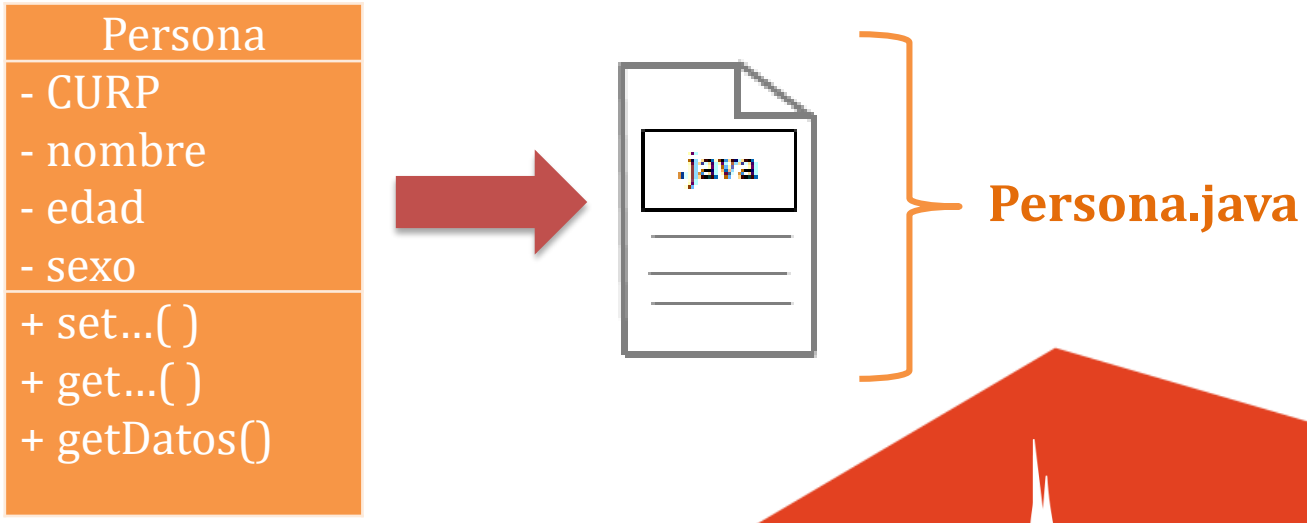
La nomenclatura utilizada para dar el nombre a una clase, sigue las siguientes normas:

1) Los nombres de clase deben ser sustantivos escritos en letras mayúsculas y minúsculas, con la primera letra de cada palabra en mayúscula, por ejemplo MiClase.

2) Los nombres de clase deben contener palabras completas. Se debe evitar el uso de siglas y abreviaturas, a menos que la sigla sea más conocida y utilizada que el nombre completo, como es el caso de JVM o UML.

Requisitos del archivo fuente:

El código para cada clase que se diseñe, deberá estar en su propio archivo de texto o archivo de código fuente. En el lenguaje Java, cada archivo de código fuente debe ser idéntico al nombre de la clase public que contiene el archivo y debe ir seguido de la extensión .java. Por ejemplo, la clase **Persona** debe guardarse en un archivo llamado *Persona.java*.



Ejemplo de una clase

```
public class Punto {
    private int x;
    private int y;

    public void setX(int valX) {
      x = valX;
    }
    public void setY(int valY) {
      y = valY;
    }
    public int getX() {
       return x;
    }
    public int getY() {
       return y;
    }
}
```

Datos miembro de la clase

Funciones miembro de la clase

Implementación de la clase

```
// programa que utiliza la clase punto
import java.io.*;

public class PruebaPunto {
 public static void main(String[] args) {
   Punto A = new Punto();
   Punto B = new Punto(); //creación de dos objetos de tipo Punto

   A.setX(15); //asignación de la coord. x al objeto A
   A.setY(8); //asignación de la coord. y al objeto A
   B.setX(-25);  //asignación de la coord. x al objeto B
   B.setY(-13);  //asignación de la coord. y al objeto B

System.out.println("Los puntos tienen las siguientes coordenadas: ");
System.out.println("Punto A(" + A.getX() + ", " + A.getY() + ")");
System.out.println("Punto B(" + B.getX() + ", " + B.getY() + ")");
 }
}
```

Compilar y ejecutar desde la línea de comandos:

Conclusiones

Indicar lo que se concluye del material didáctico que se elaboró.

Bibliografía

Acedo, J. (2012). *Apuntes de programación*. Recuperado de: Modificadores en Java: http://programacion.jias.es/2012/07/modificadores-en-java/

Ceballos, F. J. (1997). *Programación Orientada a Objetos con C++*. 2da ed., Alfaomega Ra-Ma.

Deitel, P., & Deitel, H. (2016). *Como programar en Java*. 9na ed., Pearson Prentice Hall.

Sun microsystems. (2008). *Lenguaje de programación Java*. Canada.

Datos de contacto

M.C.C. Iliana Castillo Pérez
Profesora-Investigadora
Área Académica de Computación y Electrónica
Instituto de Ciencias Básicas e Ingeniería
Universidad Autónoma del Estado de Hidalgo
Correo-e: ilianac@uaeh.edu.mx
Teléfono: 771 7172000 ext. 6734