



ESCUELA DE
INGENIERÍA EN CIENCIAS Y SISTEMAS
FACULTAD DE INGENIERÍA
UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



Día, Fecha:

28/02/2023

Hora de inicio:

09:00

Introducción a la Programación y Computación 1 [Sección B]

Diego Fernando Cortez Lopez

Agenda

- Clase 5
- Dudas
- Ejemplo Práctico

Programación Orientada a Objetos (POO)

Clase 5

Clases abstractas

Es una clase que no se puede instanciar directamente, sino que se utiliza como una plantilla para crear subclases. Una clase abstracta se define con la palabra clave “abstract” .

Los métodos abstractos son aquellos que no tienen implementación en la clase abstracta, sino que se declaran sin cuerpo y se definen en las subclases

```
public abstract class Animal {  
    public abstract void hacerSonido();  
    public void dormir() {  
        System.out.println("ZZZ");  
    }  
}
```

```
public class Perro extends Animal {  
    public void hacerSonido() {  
        System.out.println("Guau");  
    }  
}
```

Interfaces

Una interfaz es una colección de métodos abstractos y constantes que definen un conjunto de operaciones que una clase debe implementar. Una interfaz se define utilizando la palabra clave “interface”, y se pueden implementar varias interfaces en una sola clase.

```
public interface MiInterfaz {  
    public void metodoUno();  
    public int metodoDos();  
    public static final int CONSTANTE = 5;  
}
```

```
public class MiClase implements MiInterfaz {  
    public void metodoUno() {  
        System.out.println("Implementación de método uno");  
    }  
    public int metodoDos() {  
        return 10;  
    }  
}
```

Miembros Estáticos y Miembros de Instancia

Los miembros de una clase se dividen en dos tipos: miembros estáticos y miembros de instancia.

Los miembros estáticos son aquellos que pertenecen a la clase en sí misma, mientras que los miembros de instancia pertenecen a cada instancia de la clase.



Miembros Estáticos

Estos miembros existen solo una vez en la memoria y pueden ser accedidos a través del nombre de la clase. Pueden incluir variables estáticas y métodos estáticos.

```
public class MiClase {  
    public static int miVariable = 0;  
}
```

```
public class MiClase {  
    public static void miMetodo() {  
        // código del método  
    }  
}
```

La variable “miVariable” puede ser accedida a través del nombre de la clase, “MiClase.miVariable”.

Para el método se puede acceder como: “MiClase.miMetodo()”

Miembros de Instancia

Los miembros de instancia se definen sin la palabra clave “static” y existen una vez por cada instancia de la clase.

```
public class MiClase {  
    public int miVariable = 0;  
}
```

```
public class MiClase {  
    public void miMetodo() {  
        // código del método  
    }  
}
```

La variable “miVariable” se debe de acceder a través de una instancia de la clase:

“MiClase instancia = new MiClase();

instancia.miVariable = 5;”

Para el método es: “instancia.miMetodo();”



Clases paramétricas

Son las clases que permiten definir una clase que puede trabajar con diferentes tipos de datos sin necesidad de definir una clase para cada tipo.

“T” es un parámetro de tipo y se utiliza para indicar que se puede trabajar con cualquier tipo de objeto.

El nombre puede ser cualquier identificador válido en Java, pero por lo general se utiliza una sola letra en mayúscula, como “T”, “E”, “K”, etc.

```
public class MiClase<T> {  
    private T variable;  
  
    public T getVariable() {  
        return variable;  
    }  
  
    public void setVariable(T variable) {  
        this.variable = variable;  
    }  
}
```

Principios Básicos de UML

¿Qué es UML?

El lenguaje Unificado de Modelado (UML) es un lenguaje de modelado gráfico utilizado para visualizar, especificar, construir y documentar sistemas de software.

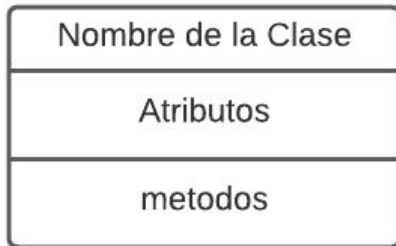
Los diagramas más comunes del UML son:

- Diagrama de Clases
- Diagrama de Objetos
- Diagrama de Casos de Uso
- Diagrama de Estados
- Diagrama de Secuencias
- Diagrama de Actividades
- Diagrama de Componentes



Diagrama de Clases

Este diagrama representa la estructura de clases y las relaciones entre ellas en un sistema de software.



A los atributos y métodos se indica si es público(+), privado(-) o protegido(#).

Multiplicidad

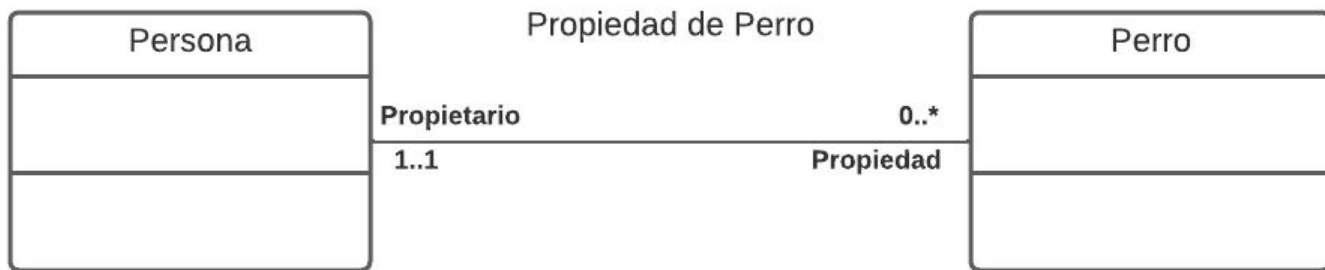
Se utiliza para especificar la cantidad de objetos que pueden estar involucrados en una relacion entre dos o más clases.

1	No más de uno
0..1	Cero o uno
0..*	Cero o muchos
1..*	Uno o muchos
*	Muchos



Asociación

Representan las relaciones generales entre clases.



Composición

Implica que los componentes de un objeto solo pueden pertenecer a un solo objeto agregado, de forma que cuando el objeto agregado es destruido todas sus partes son destruidas también.



Agregación

Es la relación parte-de, que presenta a un objeto como agregado de otros objetos.



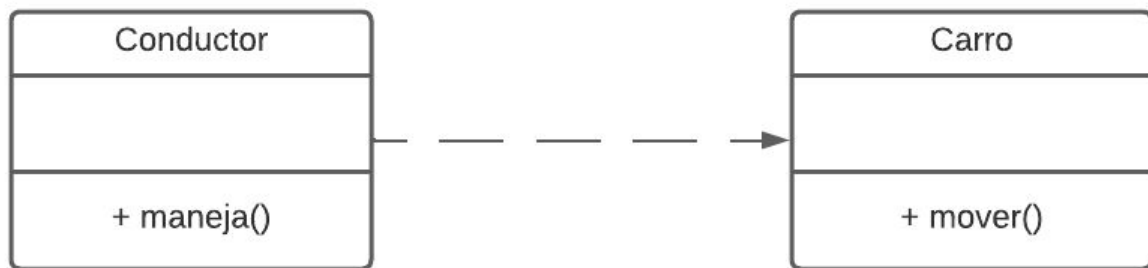
Herencia

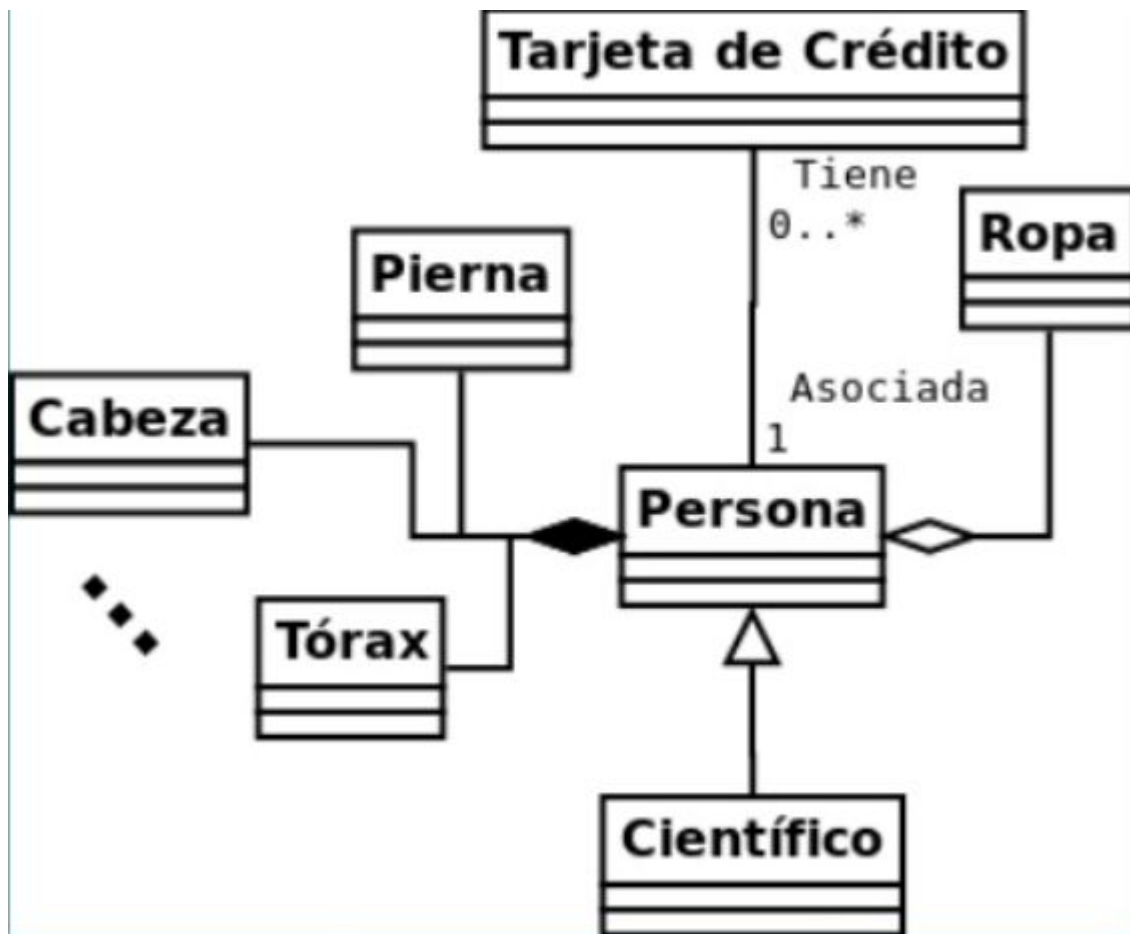
Representa que una o varias clases derivan de una superclase.



Dependencia

Representa cuando una clase depende o usa otra clase.







¿Dudas?