



Clase Abstracta e interfaces

Facultad de Ciencias

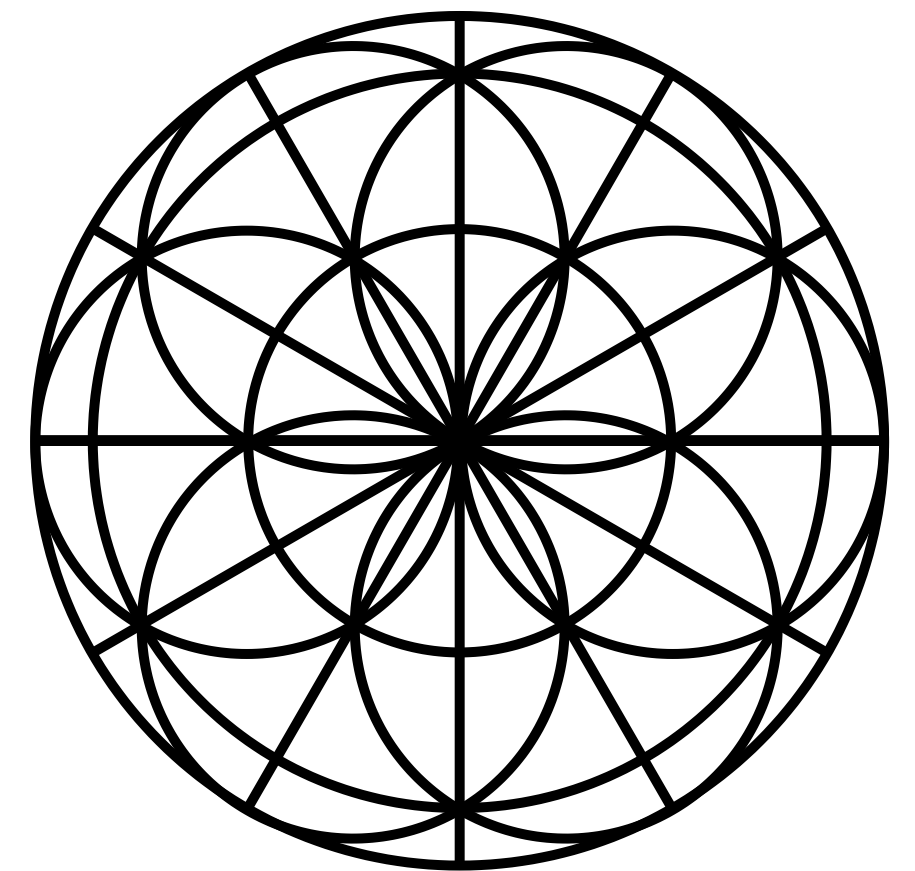
Diana Isabel Ramírez García

Introducción a Ciencias de la Computación 2021-1

¿Qué es una clase abstracta?

Una clase se define como abstracta cuando al menos tiene un método abstracto.

Un método abstracto es aquel que se deja sin implementar porque no es posible definir lo, entonces ese trabajo corresponde a las subclases.



¿Qué es una clase abstracta?

Es necesario añadir la palabra reservada **abstract** en la **firma de la clase** y de **todo método abstracto**.

```
public abstract class Animal{  
...  
public abstract void hacerSonido();  
}
```

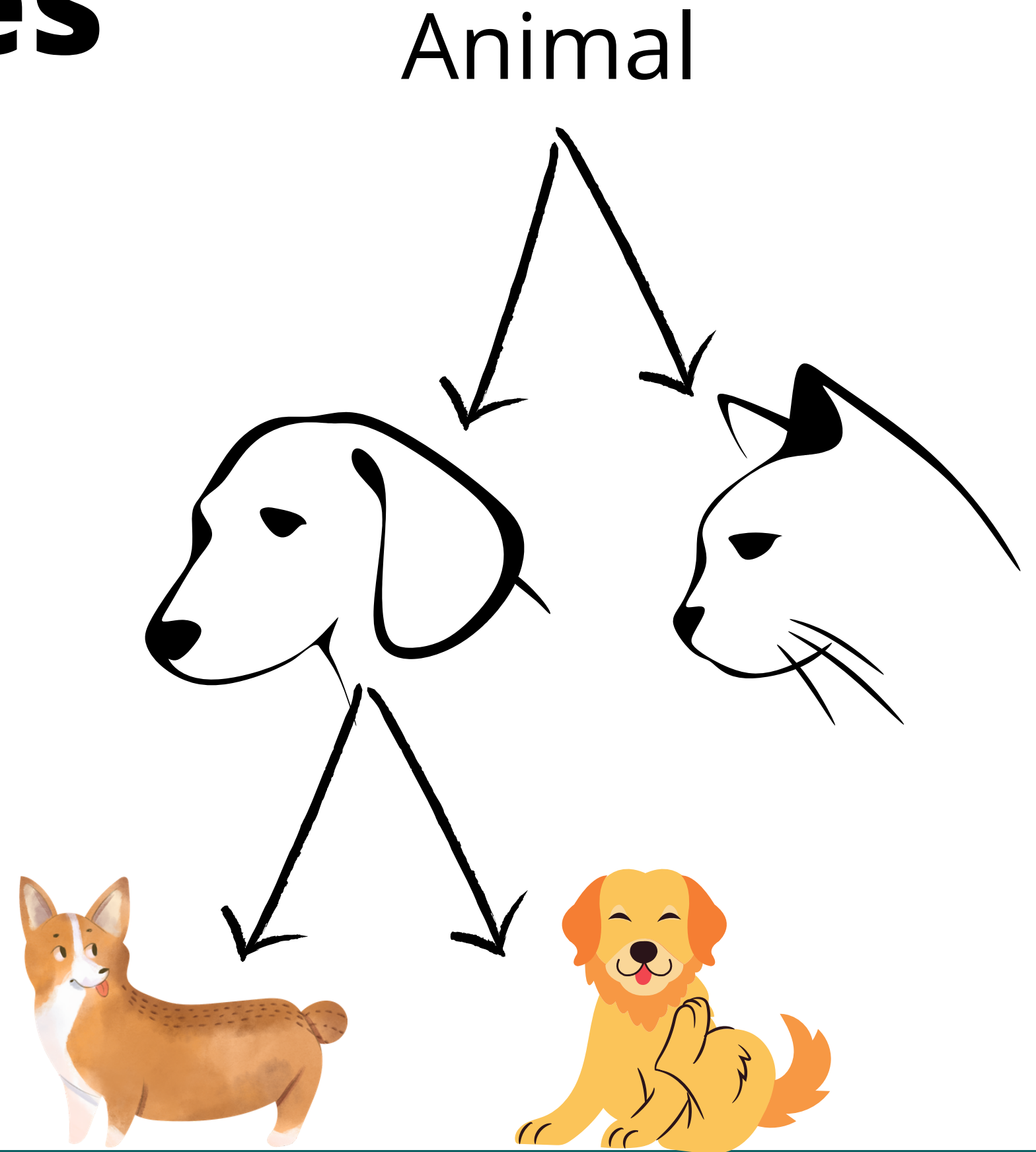
¿Qué es una clase abstracta?

- Una clase abstracta nos asegura que sus clases hijas deberán implementar todos los métodos abstractos definidos en la superclase y heredará el comportamiento que si está definido.
- No es posible crear instancias de una clase abstracta porque no todo el comportamiento está definido, pero si es posible crear una referencia.

Jerarquía de clases

abstractas

En este caso el polimorfismo vuelve a tener un papel importante debido a que podemos tener diferentes definiciones para un método, pero estas lo pueden heredar a otras subclases



Interfaces

- Es aquella en donde todos los métodos son abstractos, es decir la implementación de todos los métodos se deja como trabajo para las clases que lo implementen.
- Una interfaz nos sirve como una plantilla para asegurar un comportamiento básico.



Interfaces

- Una interfaz no es una clase, por ello no podemos hablar de una jerarquía de clases.
- Clases sin relación alguna pueden implementar la misma interfaz.
- Una interfaz nos sirve como una plantilla para asegurar un comportamiento básico.

Interfaces

- Para crear una interfaz es necesario utilizar la palabra reservada interface.

```
public interface Inflable {
```

```
...
```

```
}
```


Interfaces

- En caso de implementar una interfaz en la firma de la clase utilizamos implements.

```
public class Rectangulo implements  
Inflable {
```

```
...
```

```
}
```

Interfaces

- Si una clase hereda de otra e implementa una interfaz por convención extends va antes de implements.

```
public class Rectangulo extends  
FiguraGeometrica implements Inflable {  
  
...  
}
```

Referencia a Interfaces

- De igual manera que en las clases abstractas no podemos crear instancias pero si referencias de estas.
- Una referencia a una interfaz almacena una referencia a un objeto de la clase que la implemente.

Inflable pelota = new Circulo();

Implementación de más de una interfaz

Una clase puede implementar más de una interfaz, para indicarlo es necesario separarlas por comas en la firma de la clase.

```
class Circulo extends FiguraGeometrica  
implements Inflable, Desplazable {  
...}
```

Definición de grupos de constantes

- Todo atributo en una interfaz está declarado y asignado. Estos atributos se manejan como constantes para las clases que implementen la interfaz.

C. abstractas vs. interfaces

- Tiene métodos implementados.
- Puede tener elementos public, private o protected.
- Existe en una jerarquía de clases

- No tiene ningún método implementado
- No puede tener elementos privados.
- Clases sin relación pueden implementarla.

C. abstractas vs. interfaces

- Puede tener constructores.
- Puede extender de otra clase.
- Es posible posponer métodos aunque se extienda una clase abstracta.

- No tiene ningún constructor.
- No puede extender de ninguna clase.
- Si se implementa una interfaz deben programarse todos los métodos.

C. abstractas vs. interfaces

- Una clase solo puede heredar de una superclase.
- Una clase puede implementar más de una interfaz.

Bibliografía

- López, A.. (2014). Introducción al desarrollo de programas con Java. México: Las prensas de Ciencias.