

PROGRAMACIÓN ORIENTADA A OBJETOS

PRÁCTICA #7

Clases abstractas

Clases concretas

Polimorfismo

Uso de downcast



CLASES ABSTRACTAS Y POLIMORFISMO

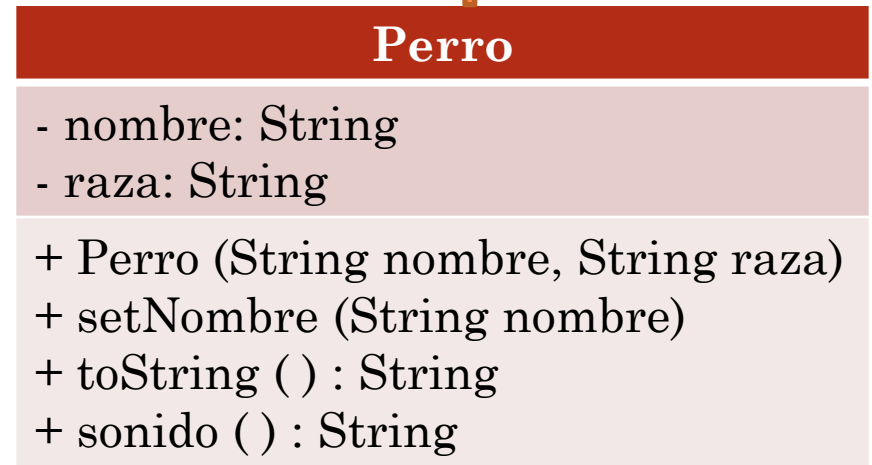
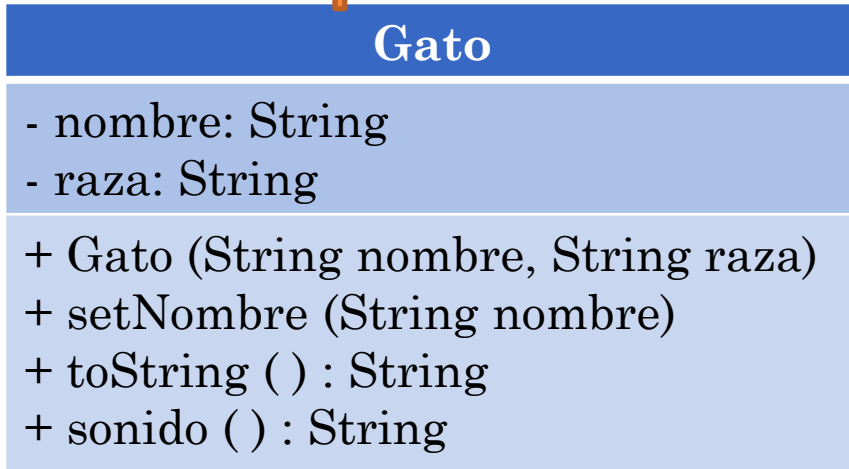
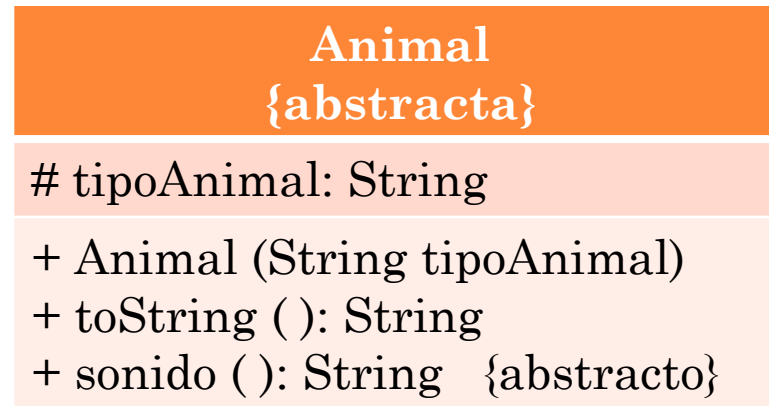
- De la jerarquía de clases de la diapositiva 5 de esta presentación, implementa lo siguiente:
 - La superclase **Animal**.
 - El método **toString()** describe al animal de la siguiente forma: “Un <tipo>”. Donde <tipo> es el tipo de animal, por ejemplo: *gato* o *perro*.
 - El método **sonido()** es abstracto dado que no se sabe aún el tipo de sonido que hará el animal creado.
 - Implementa las subclases **Gato** y **Perro**.
 - En cada caso, el método **toString()** describe al animal de la siguiente forma: “Un <tipo>, <nombre>, el <raza>”. Donde <nombre> es el nombre del animal, por ejemplo: *minni* o *lassie*; y <raza> es la raza del animal, por ejemplo: *siamese* o *poodle*.
 - El método **sonido()** implementa “meow meow” para el Gato y “woof woof” para el Perro.

CLASES ABSTRACTAS Y POLIMORFISMO

- No olvides implementar los respectivos constructores y mutadores (set's) de cada clase, como se te indica en la jerarquía de la diapositiva 5.
- Posteriormente prueba tu jerarquía de clases con la clase de prueba que te proporciona tu profesora: **PruebaAnimal.java**
 - Deberás codificar los métodos estáticos:
 - `public static String anuncia (Animal a)`
 - Imprime la descripción del animal junto con la palabra “dice” y su sonido.
 - `public static void cambiaNombre (Animal a, String name)`
 - Cambia el nombre del animal
 - Verifica manejar **CORRECTAMENTE** el polimorfismo (es decir, si es necesario o no realizar un downcast)

SALIDA ESPERADA

```
Un perro, Murphy el Poodle  
Un gato, Debbie el Siamese  
Un perro, Murphy el Poodle dice woof woof  
Un gato, Debbie el Siamese dice meow meow  
Un perro, Lassie el Poodle  
Un gato, Garfield el Siamese
```



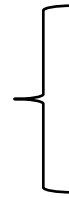
POSTERIORMENTE

- Agrega otros dos métodos abstractos a la clase Animal
- Agrega al menos dos clases, concretas, que representen animales.
- En la clase de prueba crea al menos dos animales de cada tipo (dos perros, dos gatos, dos....) y realiza pruebas polimórficas con los dos nuevos métodos abstractos que definiste.
- Agrega, al menos, un método particular en cada una de las clases concretas que definiste (este método NO se define desde la clase Animal, es un método propio de la clase concreta que definiste)
- En la clase de prueba realiza *pruebas polimórficas* invocando a este método particular. Por ejemplo, crea un método estático que reciba de parámetro un Animal y que, como parte de la implementación de este método, se tenga que invocar a alguno de los métodos particulares de las clases concretas. (Ojo: DEBES hacer un downcast)

¿QUÉ DEBES ENTREGAR?

- Enviar un correo electrónico con los siguientes documentos:

- *Animal.java*
- *Gato.java*
- *Perro.java*
- *PruebaAnimal.java*
- Tus DOS nuevas clases concretas que definiste



Nombre del archivo .zip:
Practica07.zip

- Cada documento .java que generes o modifiques DEBE incluir, en la parte superior la siguiente documentación (con la información correcta):

```
/* Clase <nombre clase>.java
   Autor 1: <nombre alumno 1> <matrícula alumno 1>
   Autor 2: <nombre alumno 2> <matrícula alumno 2>
   Fecha: <fecha de entrega>
   Práctica # 7 – Clases abstractas y polimorfismo
*/
```

- *Subject* del correo:
 - POO – Práctica #7 – Clases abstractas y polimorfismo
- En el cuerpo del correo los nombres de los integrantes del equipo