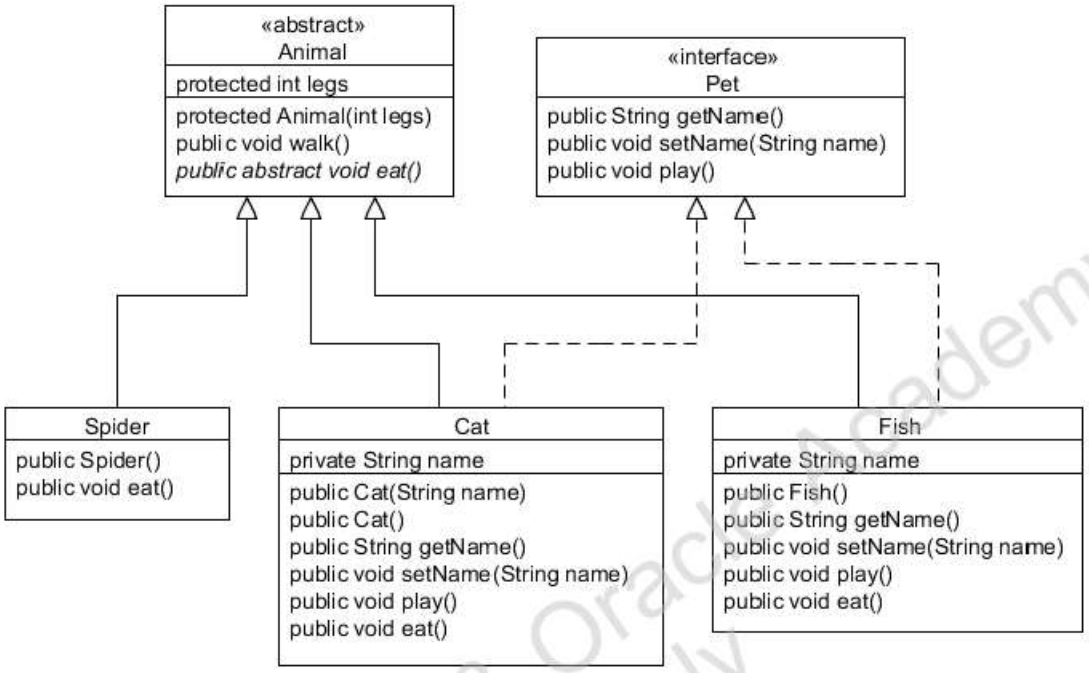
Ejercicio 04: Pet

El objetivo de este ejercicio es aplicar todos los conceptos sobre la POO: herencia, clases abstractas, interfaces, y polimorfirsmo.

Para resolver este ejercicio se recomienda haber consolidado los conocimientos expuestos en ICB0003-S09-C01-V01.. ICB0003-S09-C01-V08.

En los materiales del curso, dispones del código fuente de la aplicación Java Pet. El objetivo de esta actividad es completar la aplicación de acuerdo al siguiente diagrama de clases.



Sigue los siguientes pasos:

a. Abre el proyecto en tu IDE IntelliJ y tomate un tiempo para revisar su contenido. Al ejecutar la aplicación debería mostrarse texto en la consola de salida.
b. Define la interface Pet en el package com.company de acuerdo a los requisitos especificados en el diagrama de clases
c. Define la clase Fish en el package com.company, de acuerdo a los requisitos especificados en el diagrama de clases teniendo en cuenta:
    - Un Fish tiene 0 legs, por tanto en su método constructor se pasará un 0 al constructor de su superclase
    - La implementación del método play() consistirá en mostrar por consola "Just keep swimming."
    - La implementación del método eat() consistirá en mostrar por consola "Fish eat pond scum"
    - Sobre-escribirá el método walk() de la superclase Animal, llamando primero al método walk de la superclase y luego mostrará por consola "Fish, of course, can`t walk, they swim"
d. Define la clase Cat en el package com.company, de acuerdo a los requisitos especificados en el diagrama de clases teniendo en cuenta

- Un Cat tiene 4 legs, por tanto en su método constructor que recibe como argumento una string, se pasará un 4 al constructor de su superclase
- En el método constructor que no tiene argumentos, se llamará al otro método constructor de la clase pasándole como parámetro "Fluffy"
- La implementación del método play() consistirá en mostrar por consola el valor del atributo name seguido de " likes to play with string."
- La implementación del método eat() consistirá en mostrar por consola "Cats like to eat spiders and fish"

e. Añade en Main.java, el código necesario para testear las clases Cat y Fish que has codificado: usa cada método constructor y cada método (play(), eat(), walk()).

¿Qué ocurre si se testea un objeto de la clase Fish con una referencia a Animal? ¿Y si se testea un objeto de la clase Cat con una referencia a Pet?

Animal a = new Fish();
a.play();
a.eat();
a.walk();
	Pet p = new Cat();
	p.play();
	p.eat();
	p.walk();

f. En Main.java, implementa el método public void playWithAnimal(Animal a), de forma que:
   a. Si el objeto recibido como argumento implementa la interface Pet, llama al método play(). Comprueba que como el método play() está declarado en la interface Pet, antes de llamar al método deberás realizar el casting del argumento a Pet.
   b. Si el objeto recibido como argumento NO implementa la interface Pet, muestra por consola "Danger! Wild animal."

g. En Main.java, usa el método playWithAnimal(Animal a), utilizando como argumento objetos de las clases Spider, Cat, Fish