

Praktikum

Class Abstract dan Interface

Topik

- Abstract Class
- Interface

Bekerja dengan Interface dan Abstract Class



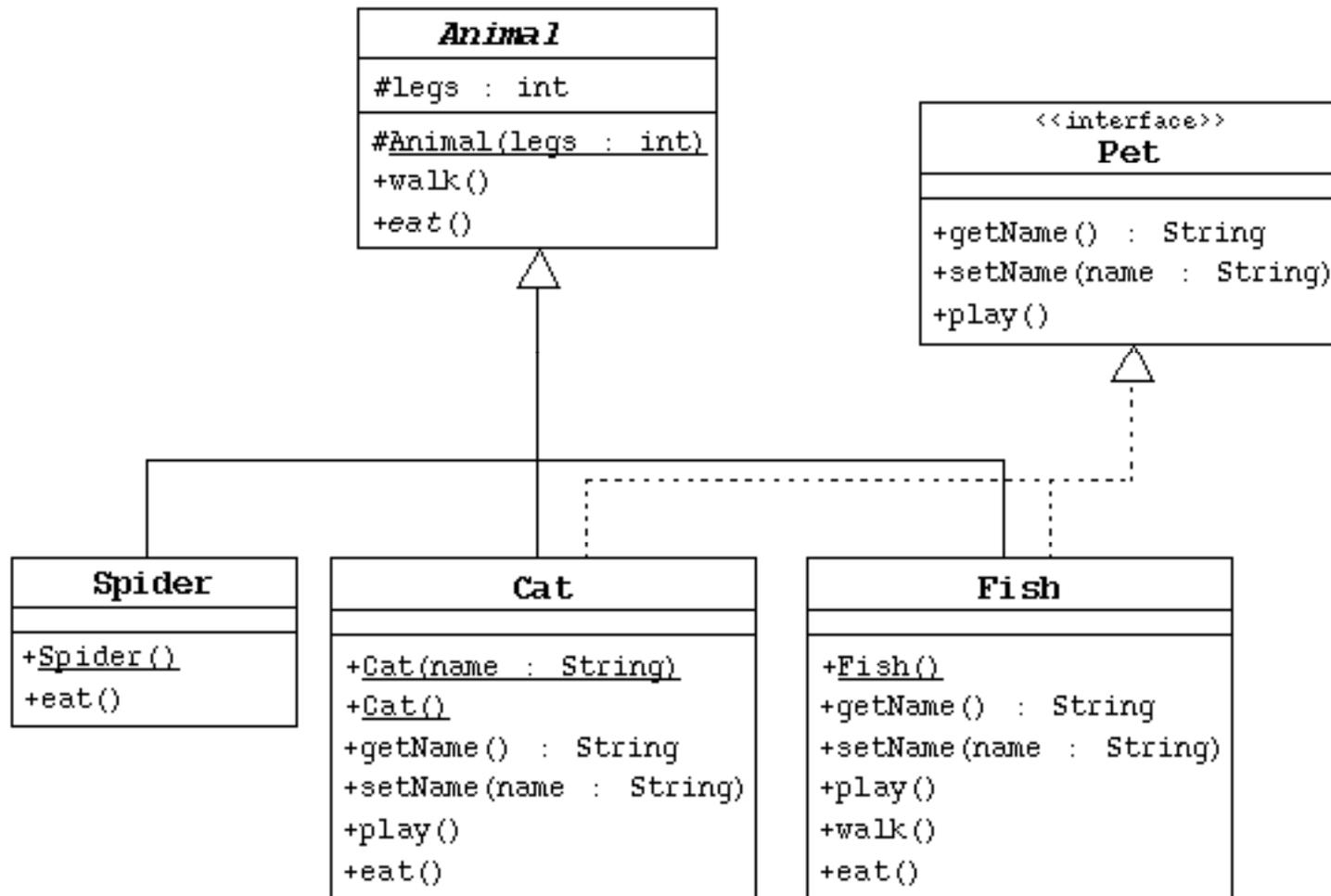
Bekerja dengan Interface dan Abstract Class

Pada latihan ini akan dibuat hirarki dari hewan dengan abstract class `Animal` sebagai root.

Beberapa class hewan akan dibuat dengan mengimplementasikan suatu interface `Pet`.

Hal yang harus dilakukan adalah mendefinisikan method-method dan melakukan polymorphism.

UML Diagram



Membuat Class Animal

- Animal adalah abstract superclass dari semua hewan.
- Buatlah protected integer attribute dengan nama `legs`, atribut ini digunakan untuk menyimpan informasi jumlah kaki hewan.
- Buatlah protected constructor yang digunakan untuk menginisialisasi variabel `legs`.
- Buatlah abstract method `eat`.
- Buatlah concrete method `walk` yang digunakan untuk menampilkan tulisan tentang bagaimana hewan berjalan dan jumlah kaki hewan tersebut (misal: hewan ini berjalan dengan 4 kaki).

Membuat Class Spider

- Class `Spider` merupakan anak dari class `Animal`.
- Buatlah constructor yang digunakan untuk memanggil superclass constructor, constructor ini juga digunakan untuk menginisialisasi jumlah kaki spider (kita tahu bahwa semua spider pasti mempunyai kaki sebanyak `8` buah).
- Implementasikan method `eat`.

Membuat Interface Pet

Buatlah interface `Pet` sesuai dengan diagram UML.

Membuat Class Cat

- Class `cat` adalah anak dari class `Animal` dan mengimplementasikan interface `Pet`.
- Buatlah variabel `name` yang bertipe `String` yang digunakan untuk menyimpan nama `Cat`. (variabel ini tidak digambarkan pada UML diagram).
- Buatlah constructor dengan satu argumen bertipe `String` yang digunakan untuk mengeset nama `Cat`. Constructor ini juga harus memanggil superclass constructor untuk mendefinisikan bahwa `Cat` mempunyai kaki sebanyak `4` buah.
- Buatlah constructor lain yang tidak mempunyai argumen. Buat constructor ini supaya memanggil constructor pada poin sebelumnya (dengan menggunakan kata kunci `this`) dan passing empty string sebagai argumen (empty string \rightarrow `""`).
- Implementasikan method-method yang ada pada interface `Pet`.
- Implementasikan method `eat` (diwarisi dari class `Animal`).

Membuat Class Fish

- Class Fish adalah anak dari class Animal.
- Lakukan override pada semua method Animal dan definisikan bahwa ikan tidak berjalan tetapi berenang.
- Class Fish mengimplementasikan interface Pet
- Jangan lupa untuk mendefinisikan method kepunyaan interface.

TestAnimal

Buat class TestAnimal yang berisi main method dengan definisi sebagai berikut:

```
public class TestAnimals {
    public static void main(String[] args) {
        Fish f = new Fish();
        Cat c = new Cat("Fluffy");
        Animal a = new Fish();
        Animal e = new Spider();
        Pet p = new Cat();

        // Demonstrate different implementations of an interface
        f.play();
        c.play();

        // Demonstrate virtual method invocation
        e.eat();
        e.walk();

        // Demonstrate calling super methods
        a.walk();
    }
}
```