

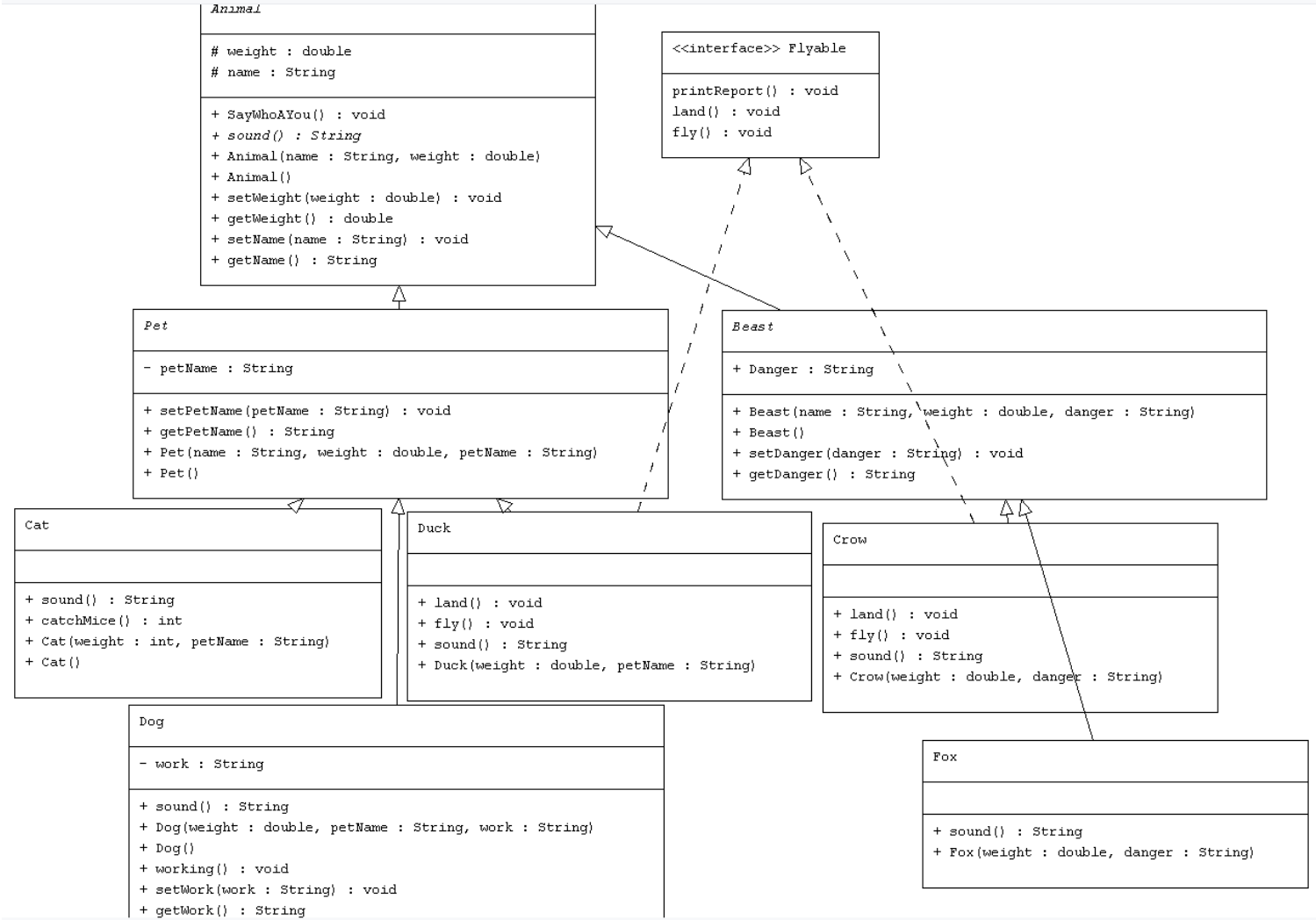
Лабораторная работа 02. Абстрактные классы и интерфейсы

Задание 1. Реализовать систему классов, соответствующую UML-диаграмме:

Класс `Animal` должен быть абстрактным, с абстрактным методом `sound()`. Метод `sayWhoAYou` выводит на экран всю доступную информацию о животном и еще вызывает метод `sound()`.

Классы `Pet` и `Beast` тоже абстрактные.

В каждом конкретном классе животных (`Cat`, `Dog`, `Duck`, `Crow`, `Fox`) метод `sound()` возвращает строку с символами, показывающую какие звуки издает это животное.

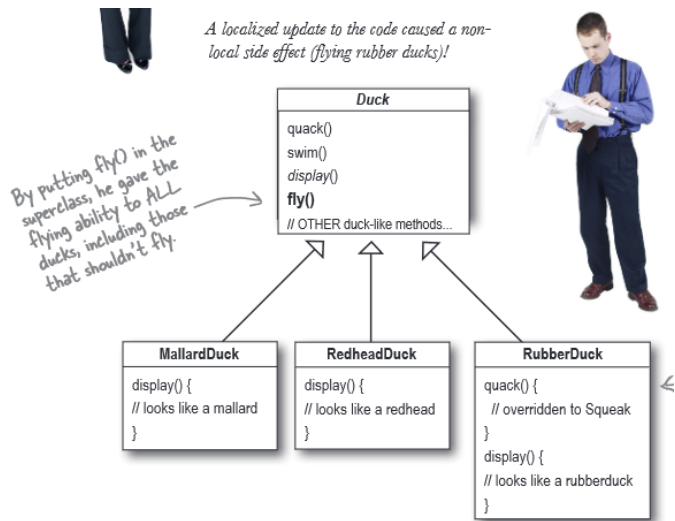


В программе

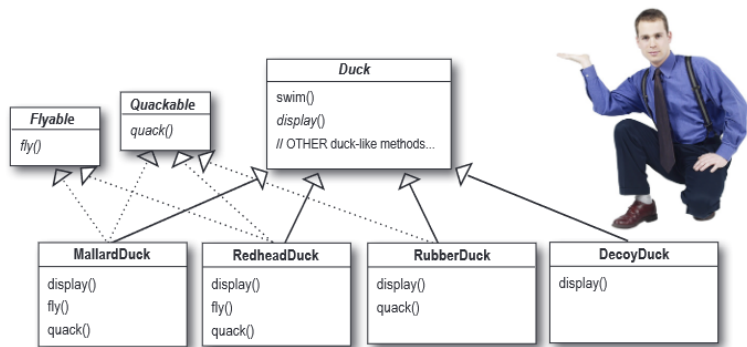
- а) Создать массив из 7 разных животных, на этом массиве
 - 1) Вызвать для каждого животного метод `sayWhoAYou`
 - 2) Вывести животного, у которого самый длинный звук `sound()`, (сделать для этого отдельный метод)
 - 3) Получить общий вес всех животных
 - 4) Получить из массива только диких животных `Beast`, для каждого из них вывести его класс опасности `Danger`
- б) Из первого массива сформировать `ArrayList<Flyable>`, в который включить только тех животных, кто умеет летать; у них реализован интерфейс `Flyable`.
Для каждого такого объекта вызвать методы `fly` и `land`, которые выводят на консоль строку с особенностями поведения животного в полете и при посадке на землю.

Задание 2. Паттерн «Стратегия»

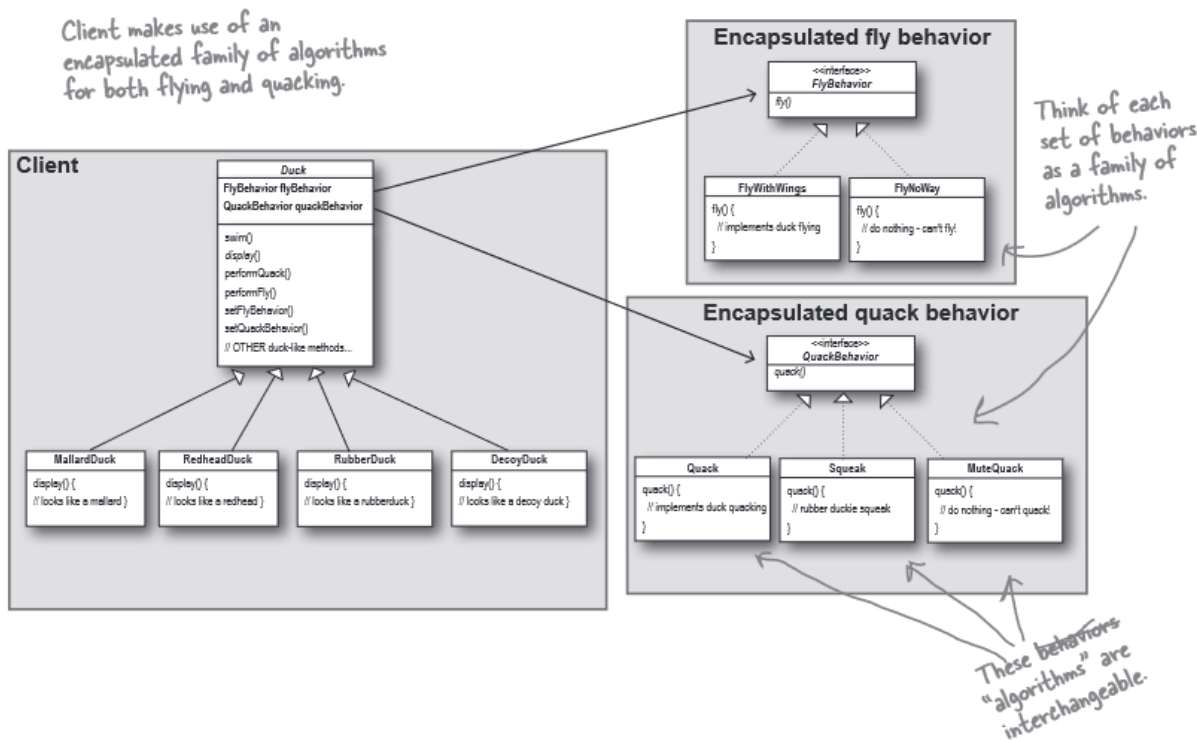
В книге **Kathy Sierra, Bert Bates - Head First Java (Second Edition)** прочитать Chapter 8.
В книге **Eric Freeman, Elisabeth Robson - Head First Design Patterns Building Extensible and Maintainable Object-Oriented Software** прочитать Chapter 1.
Примеры из этой главы книги реализовать в одном java-проекте, в разных пакетах.
В каждом пакете создать классы и интерфейсы из диаграммы и метод `main`, в котором они тестируются.
Пакет 1. Пример классов по диаграмме



Пакет 2. Пример классов по диаграмме



Пакет 3. Пример классов по диаграмме



Пакет 4. Самостоятельно реализовать паттерн Strategy для моделирования поведения рыбака с разными стратегиями рыбалки.

