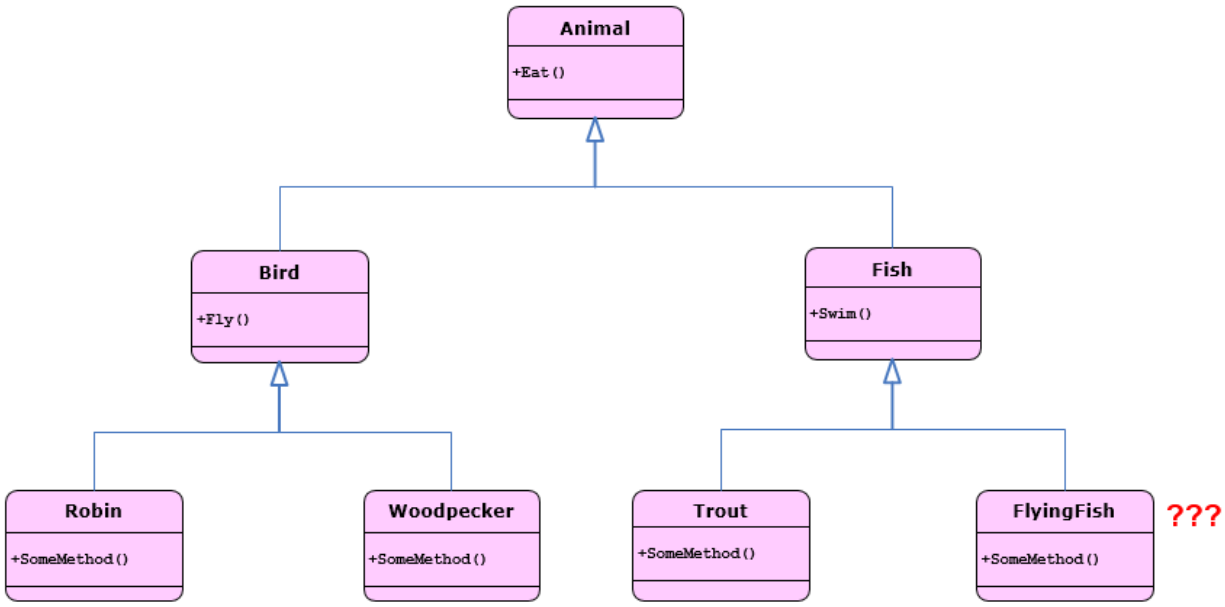
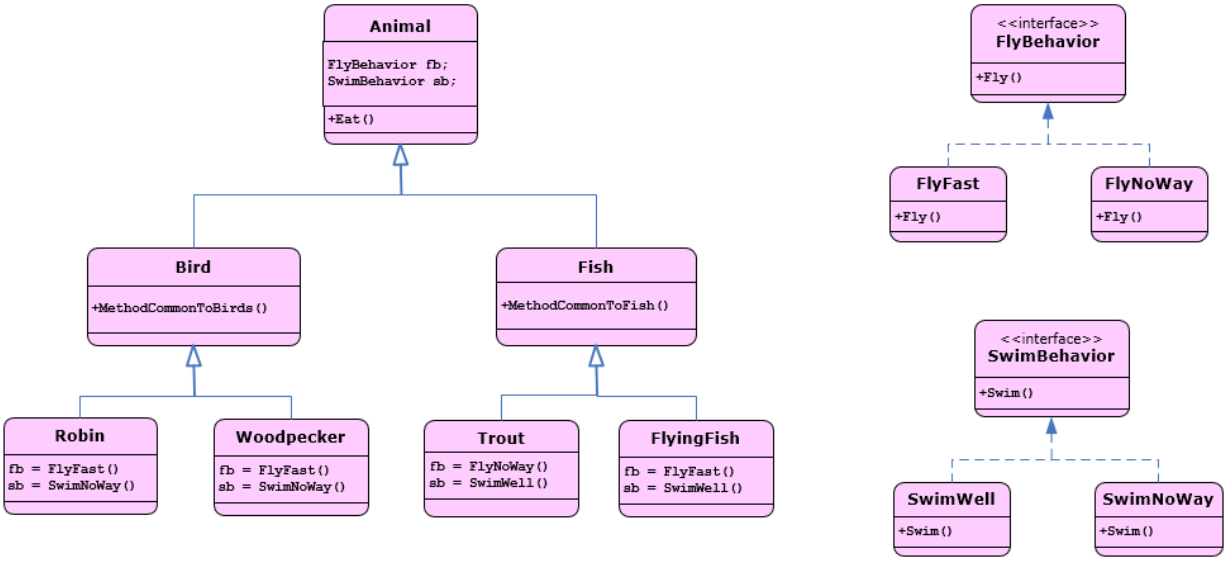
The Flying Fish Problem



Q: How can the FlyingFish fly AND swim?

A: One answer is to move the Fly() method up to the base class. But then all fish will be able to fly. A more elegant solution is to use the strategy pattern (see next slide).

The Flying Fish Solution
Using the Strategy Pattern



Benefits
- Favor object composition over inheritance
- Forces developer to make a conscious decision about behavior when coding it (as opposed to creating default implementations in the base class that the developer must remember to override).
- By abstracting the varying behavior, it is now available for use by other clients.
- Being more loosely coupled means it's easier to adhere to the open/closed principle, where code should be open for extension but closed for modification.

The varying behavior (flying and swimming) is now encapsulated and can vary independently from other parts of the system.