Design Principles

- **Identify the aspects of your application that vary and separate them from what stays the same.**
  - Encapsulate what varies
  - Program to an interface, not to an implementation
  - Favor composition over inheritance
- For our example:
  - Pull the duck *behavior* out of the duck *class*

Liskov Substitution Principle

Let q(x) be a property provable about objects x of type T. Then q(y) should be provable for objects y of type S where S is a subtype of T.