Achieving Good Object-Oriented Design

Charlotte Pierce

COS20007 Object Oriented Programming

SWINBURNE UNIVERSITY OF TECHNOLOGY

Good design is often described in terms of design goals

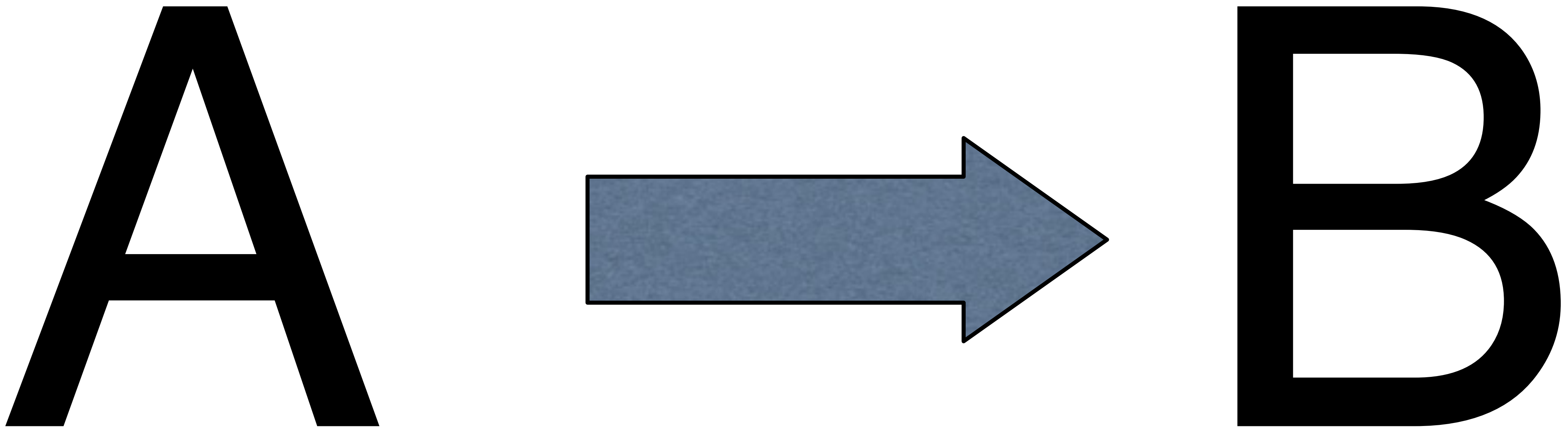
Extensible!

Robust!

Flexible!

Modular!

Developers must learn **how** to achieve good object-oriented design



A

B

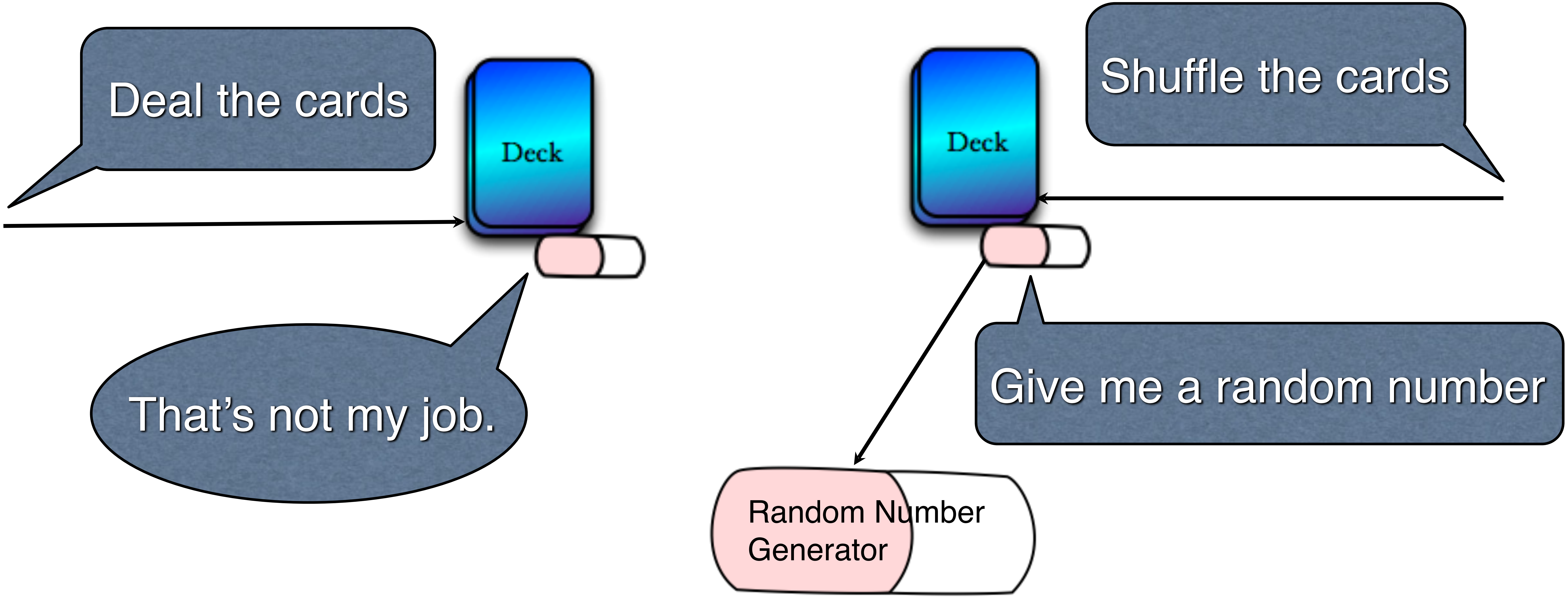
It is not enough to know the desired characteristics of the end product

Use principles - or rules of thumb - to guide design decisions

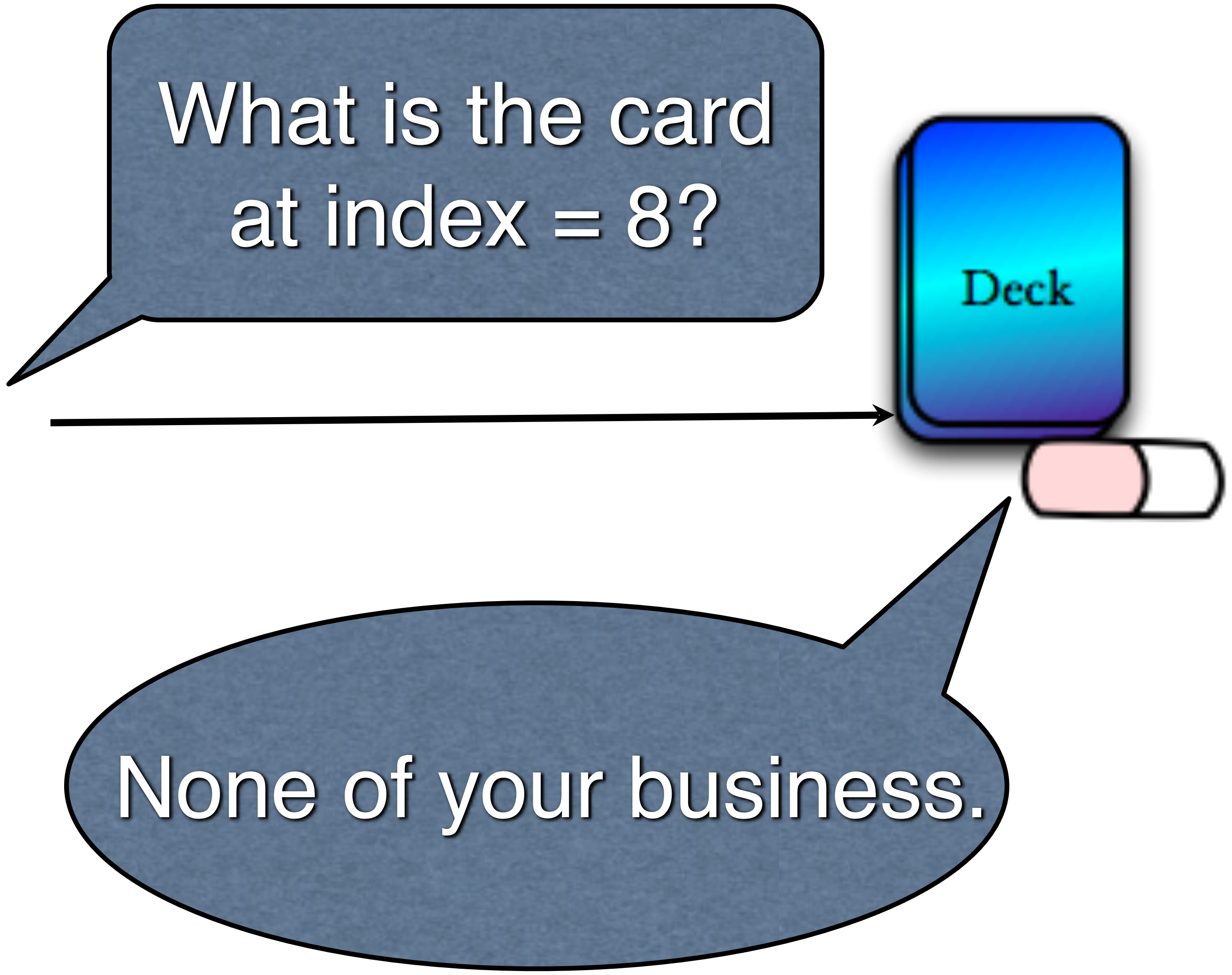
Adopt a small set of simple rules to achieve good object-oriented design

Start with these three simple rules

Classes should be lazy

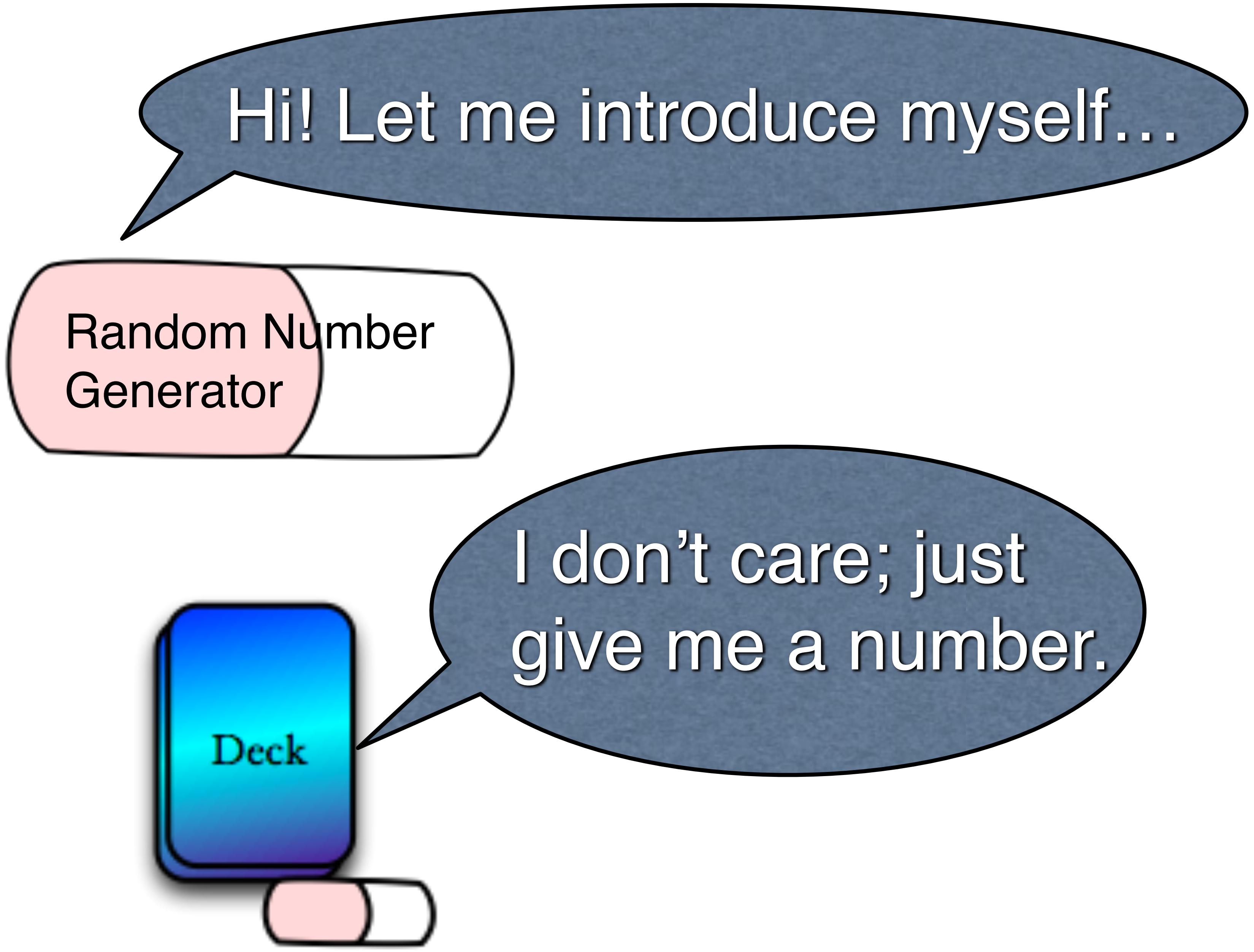


Classes should be antisocial



What is the card at index = 8?

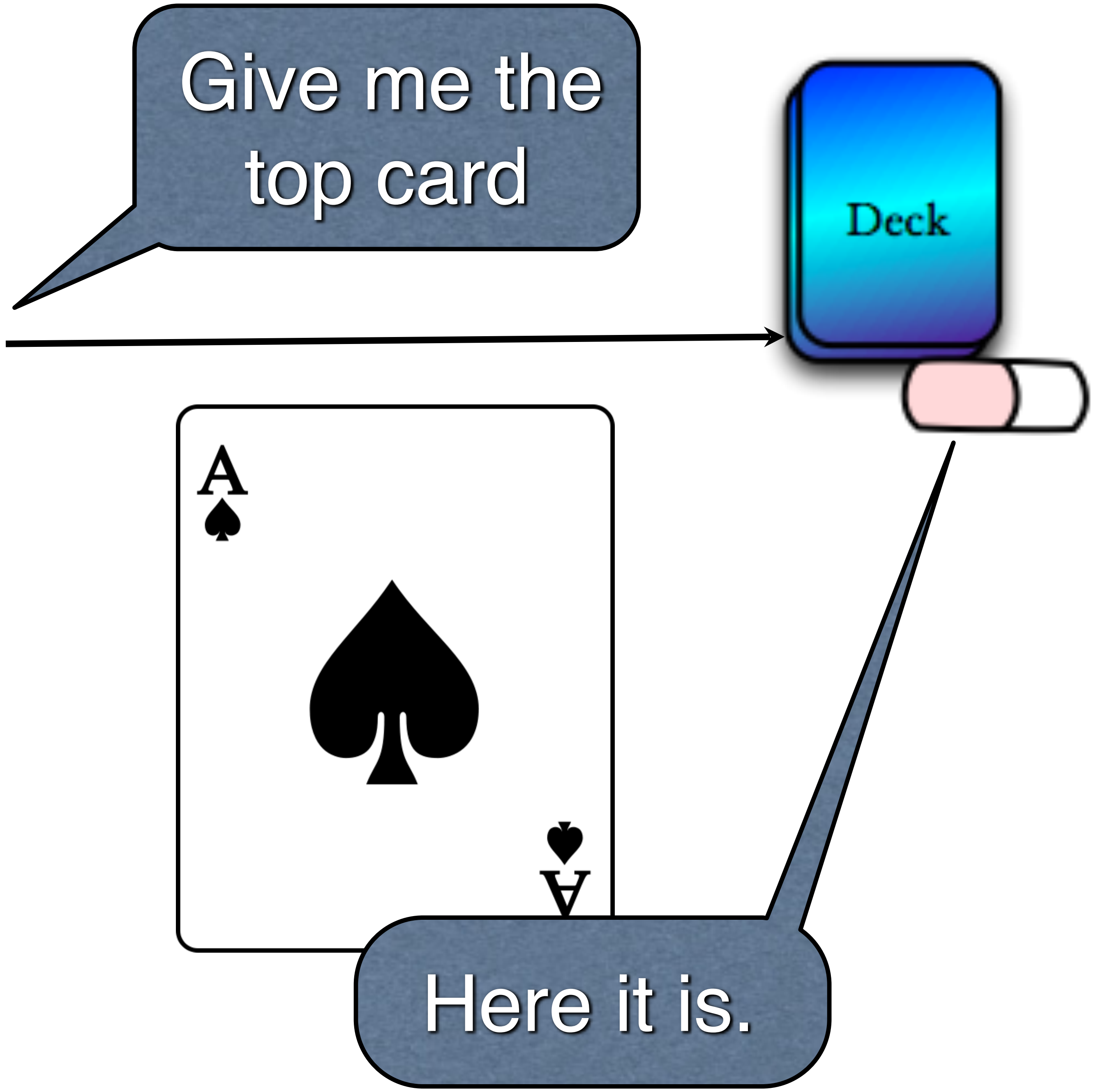
None of your business.



Hi! Let me introduce myself…

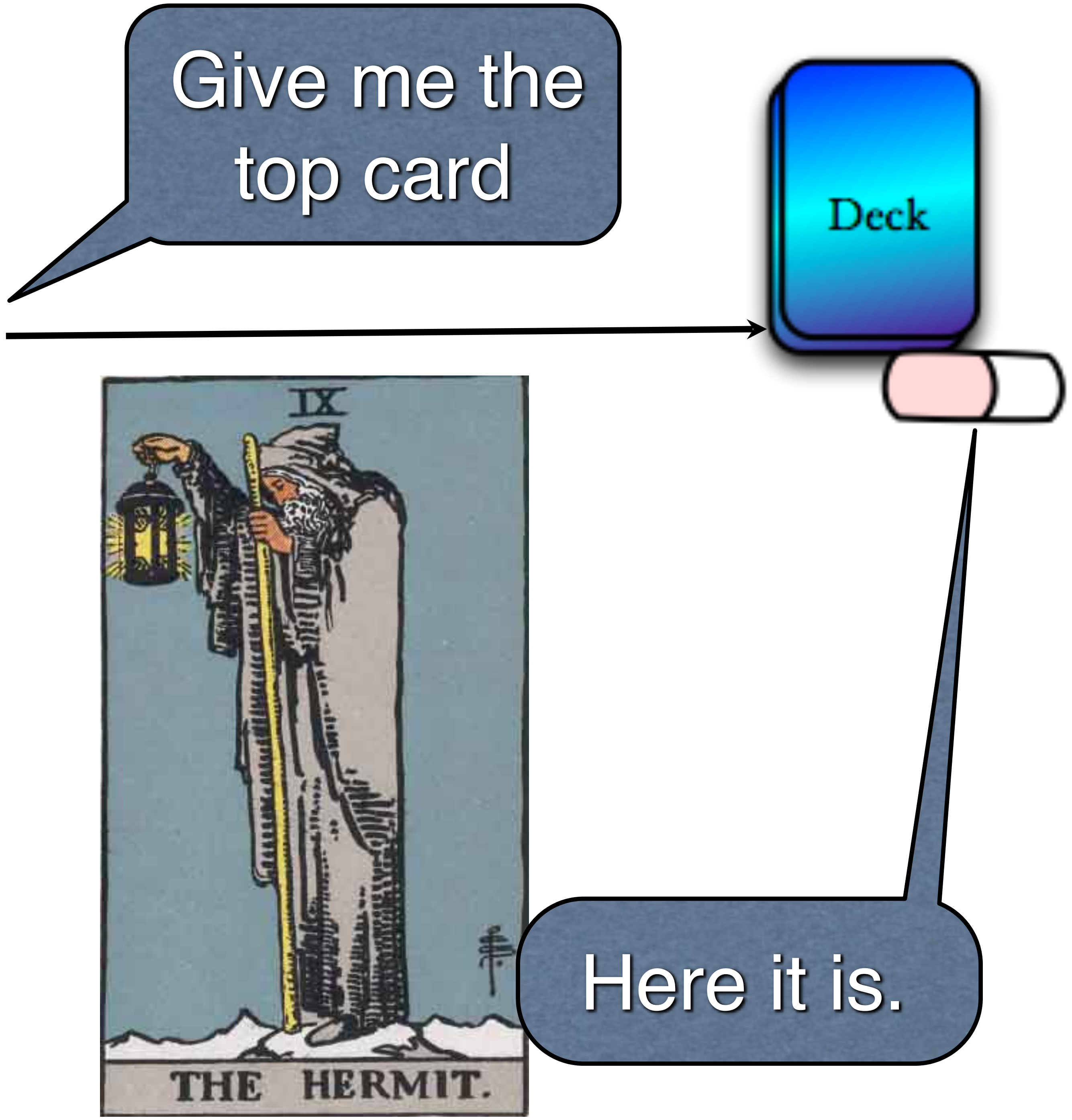
I don’t care; just give me a number.

Derived classes should be conformist



Give me the top card

Here it is.

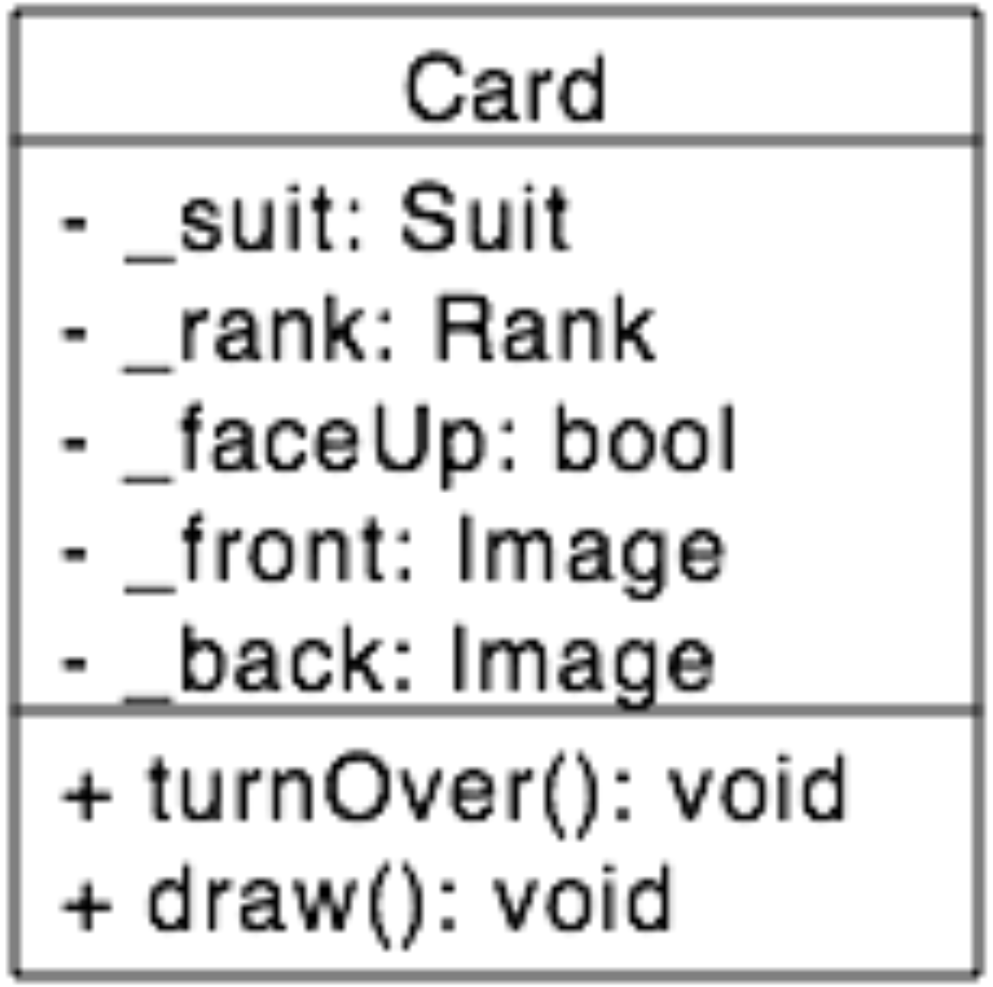


Give me the top card

Here it is.

Apply three simple rules to help evaluate object-oriented designs

Is this class sufficiently lazy?

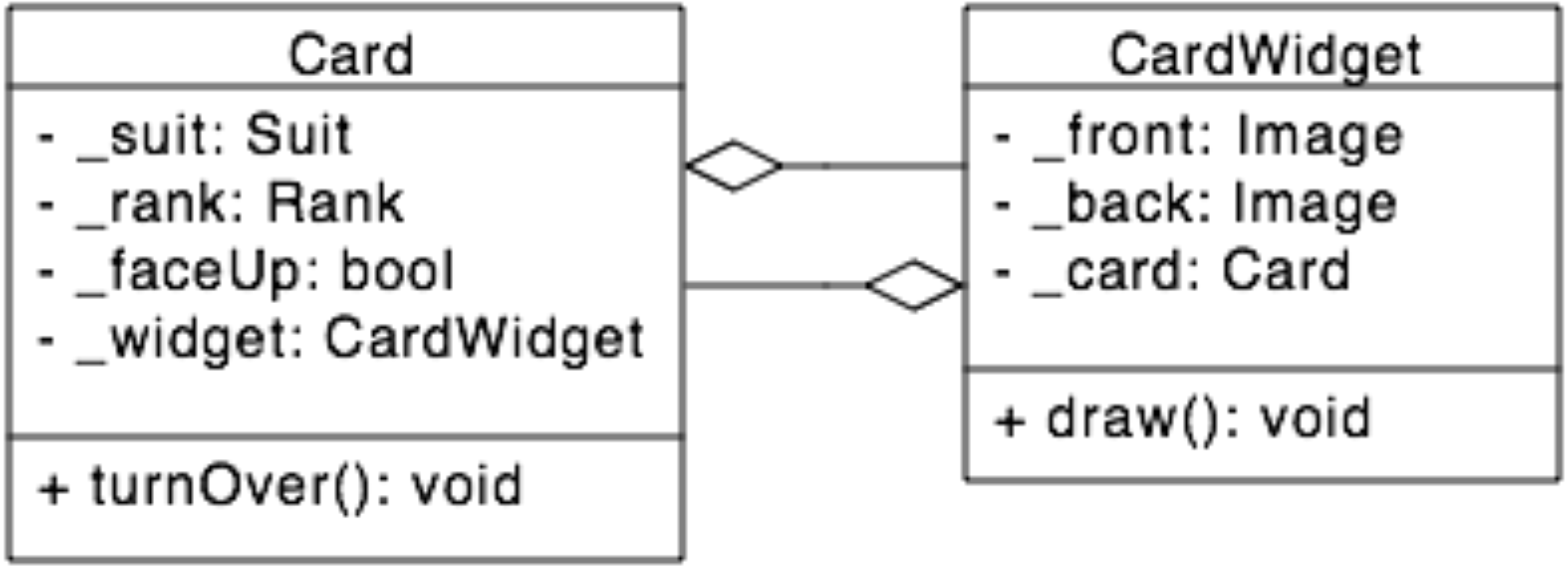


Card

- _suit: Suit
- _rank: Rank
- _faceUp: bool
- _front: Image
- _back: Image

+ turnOver(): void
+ draw(): void

Are these classes sufficiently antisocial?



Card

- _suit: Suit
- _rank: Rank
- _faceUp: bool
- _widget: CardWidget

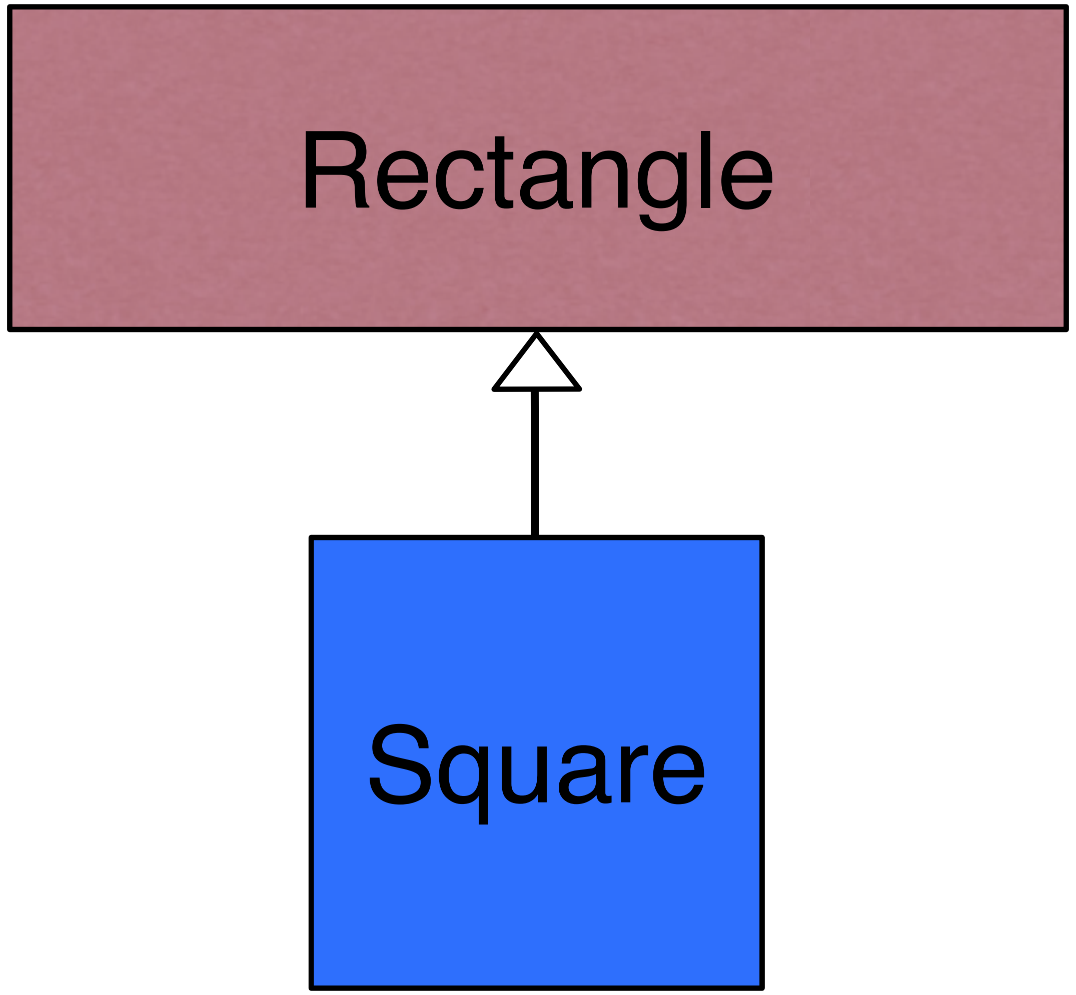
+ turnOver(): void

CardWidget

- _front: Image
- _back: Image
- _card: Card

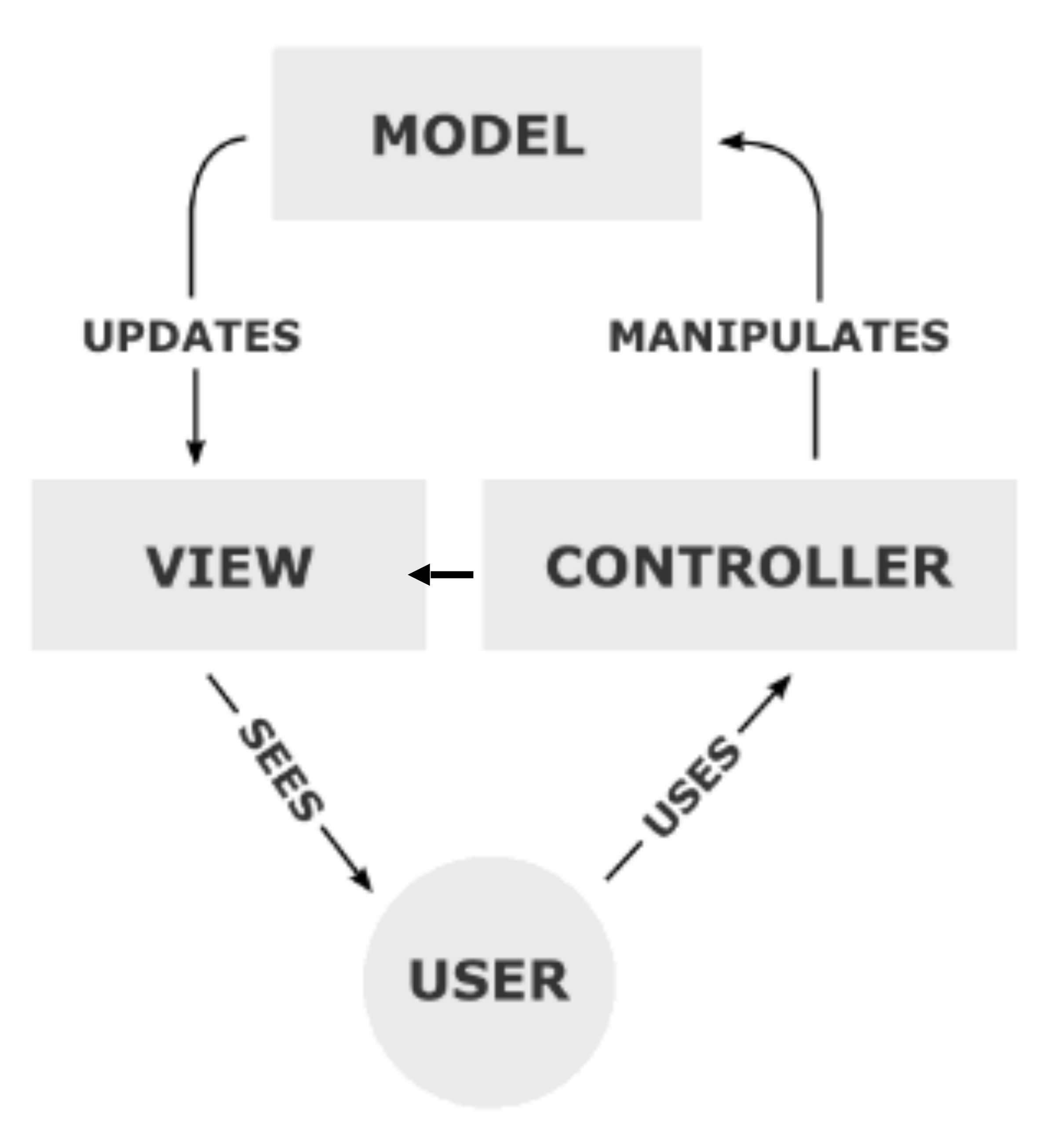
+ draw(): void

Is this derived class sufficiently conformist?

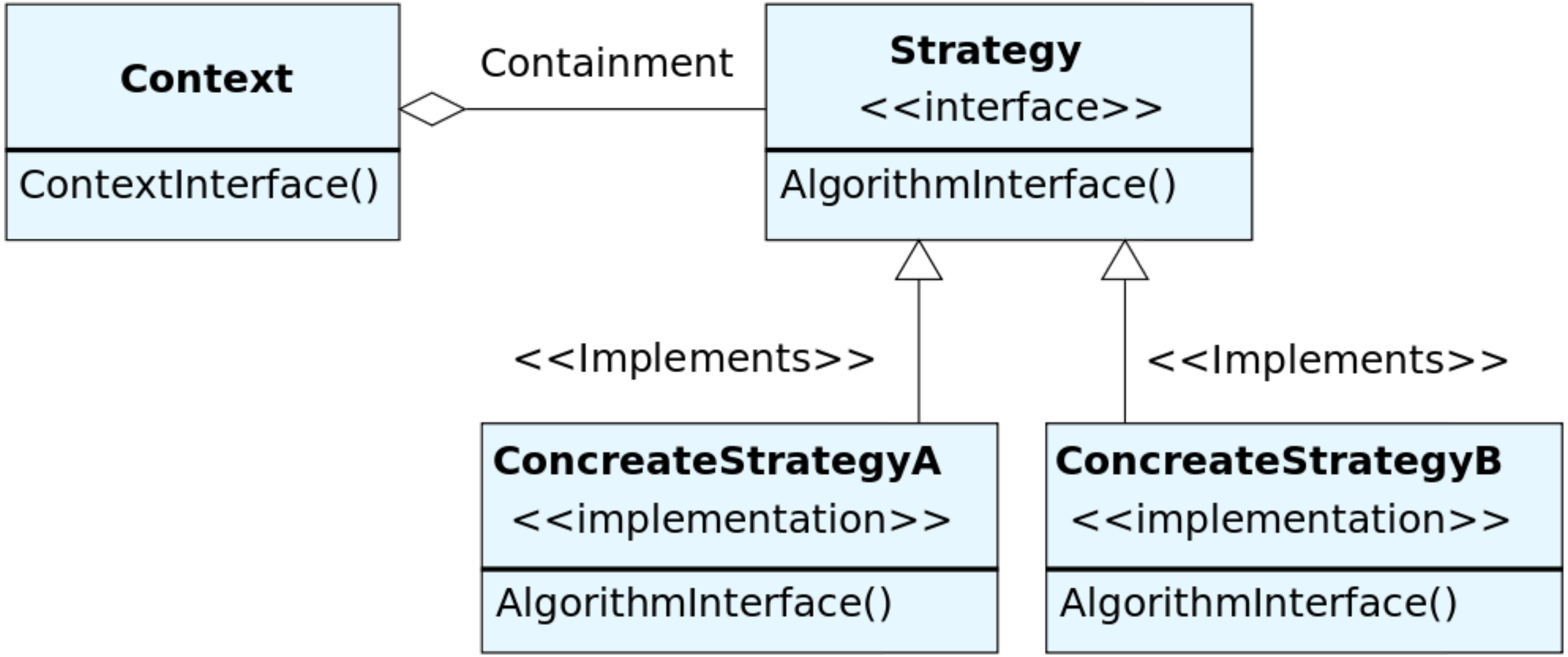


Three simple rules guide the application of OOP principles and paradigms

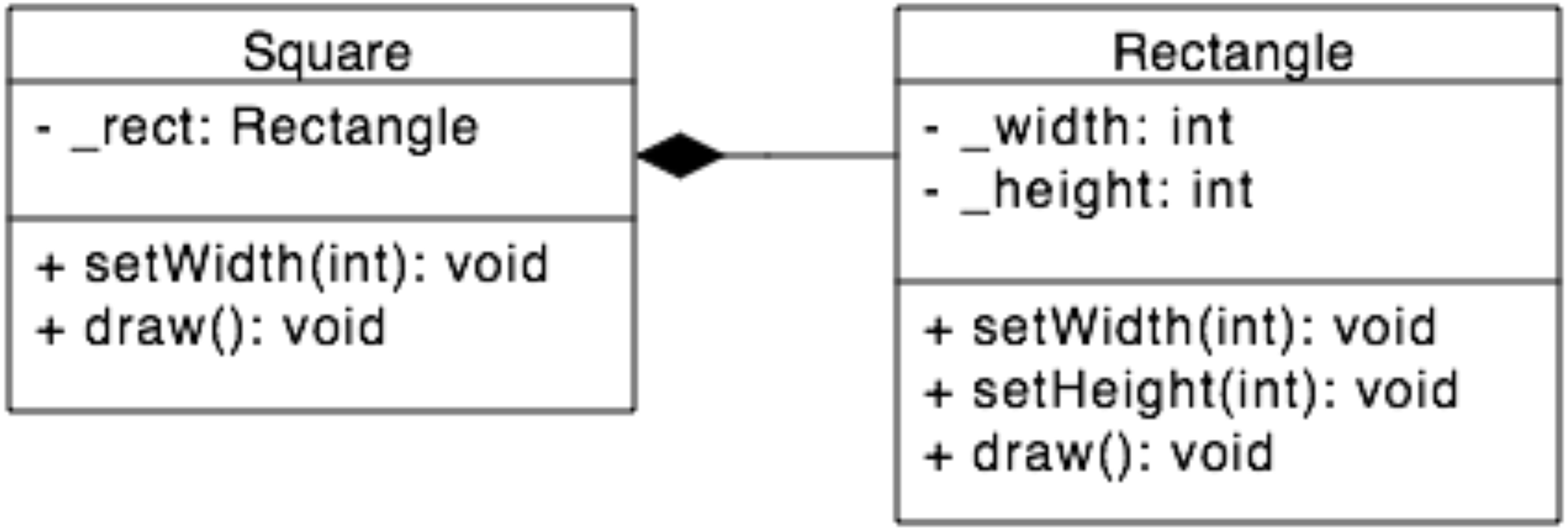
Laziness motivates separation of concerns



Unsociability promotes the use of generalised abstractions



Conformity guides the use of inheritance and polymorphism



There are other questions to consider as well…

Does this class differ in any ***functional*** *way from others?*

I’ve written this code before…could I avoid this with a better design?

Could this switch (if/else) statement be avoided using polymorphism?

This inheritance hierarchy is very deep…what happens if my parent class changes?

It is difficult to write good software without some practical guidelines

Adopt a small set of simple rules to achieve good object-oriented design

Classes should be lazy, antisocial, and conformist

Good design leads to less work