

327: Object-oriented programming

Lecture 20

11/15/2021

Professor Barron

Finished chapter 10...

- ... on to chapter 11
 - adapter
 - facade
 - command
 - abstract factory
 - flyweight
 - composite
- bonus patterns!
 - memento
 - ...

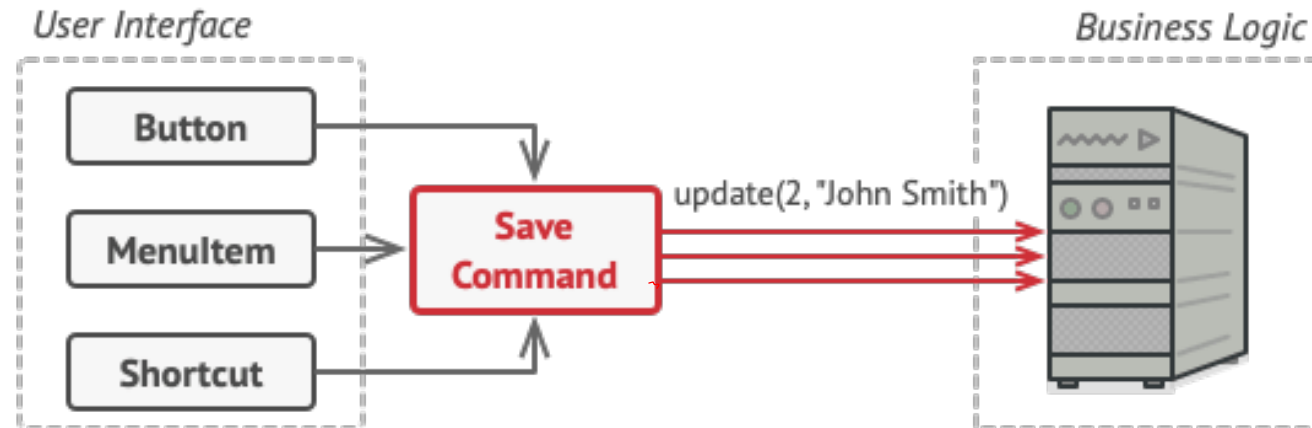
Command pattern

- Behavioral pattern
- Represent an action as an object
- Object-oriented callback
- Useful for deferring execution
- Easier to add or extend actions available
- Can help with undo/redo since the action can have a method to reverse itself

Example

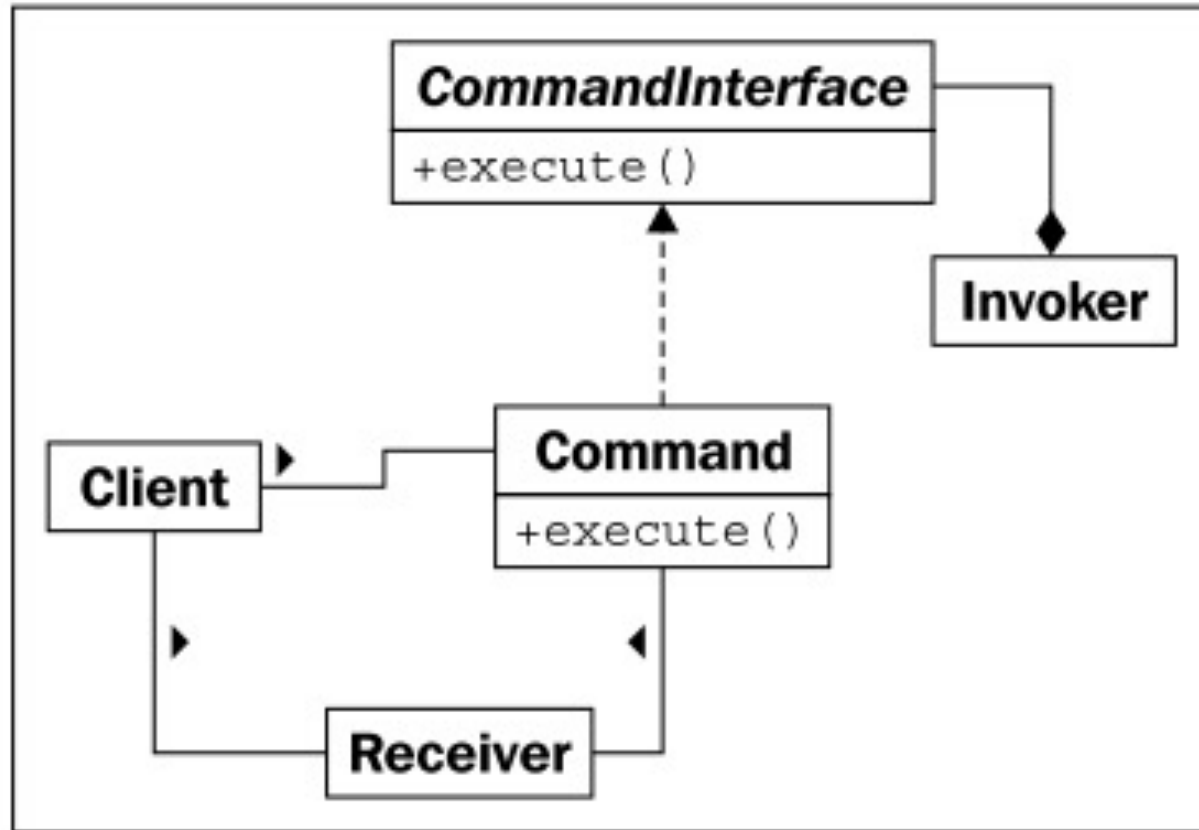


Before

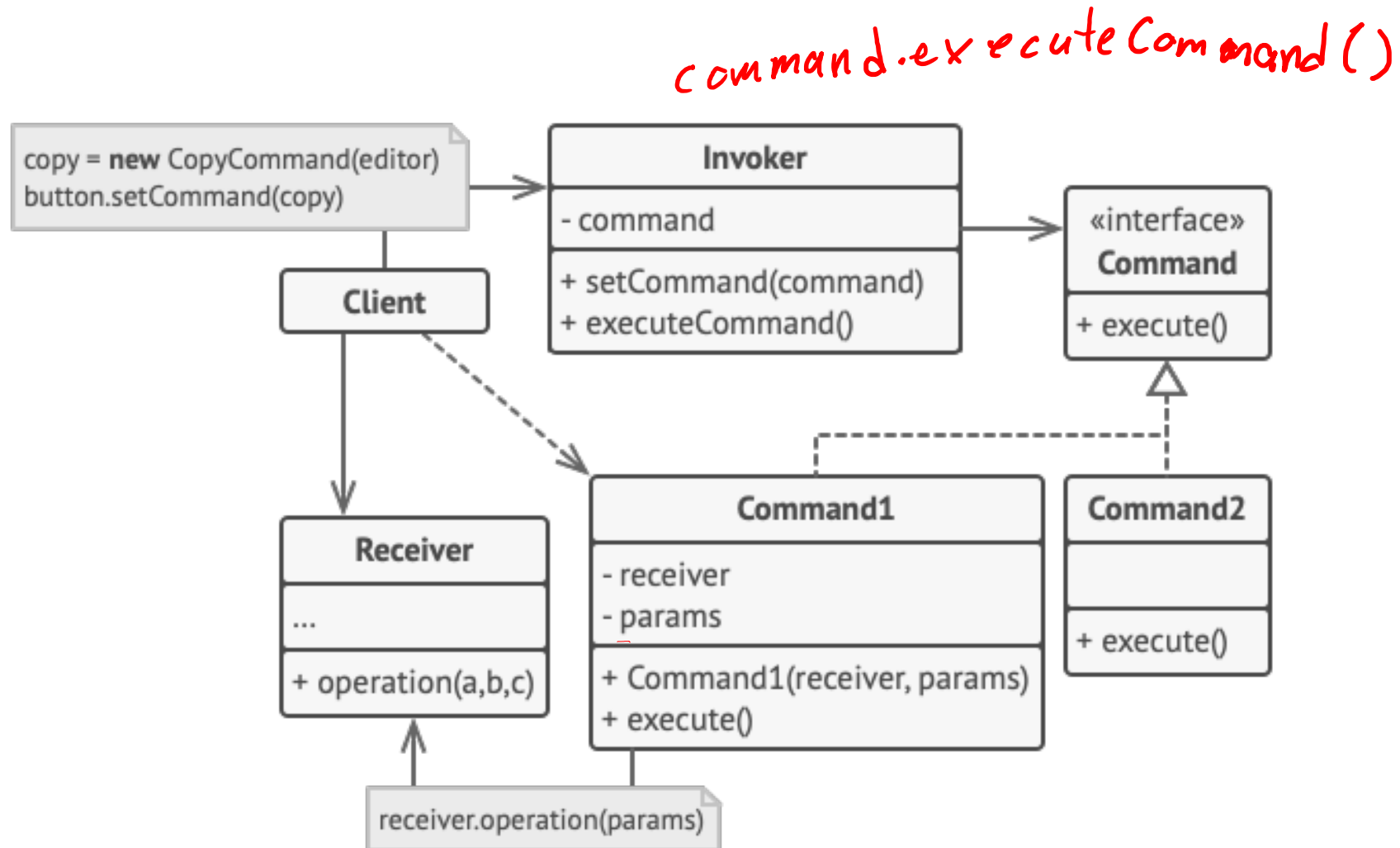


After

Class diagram



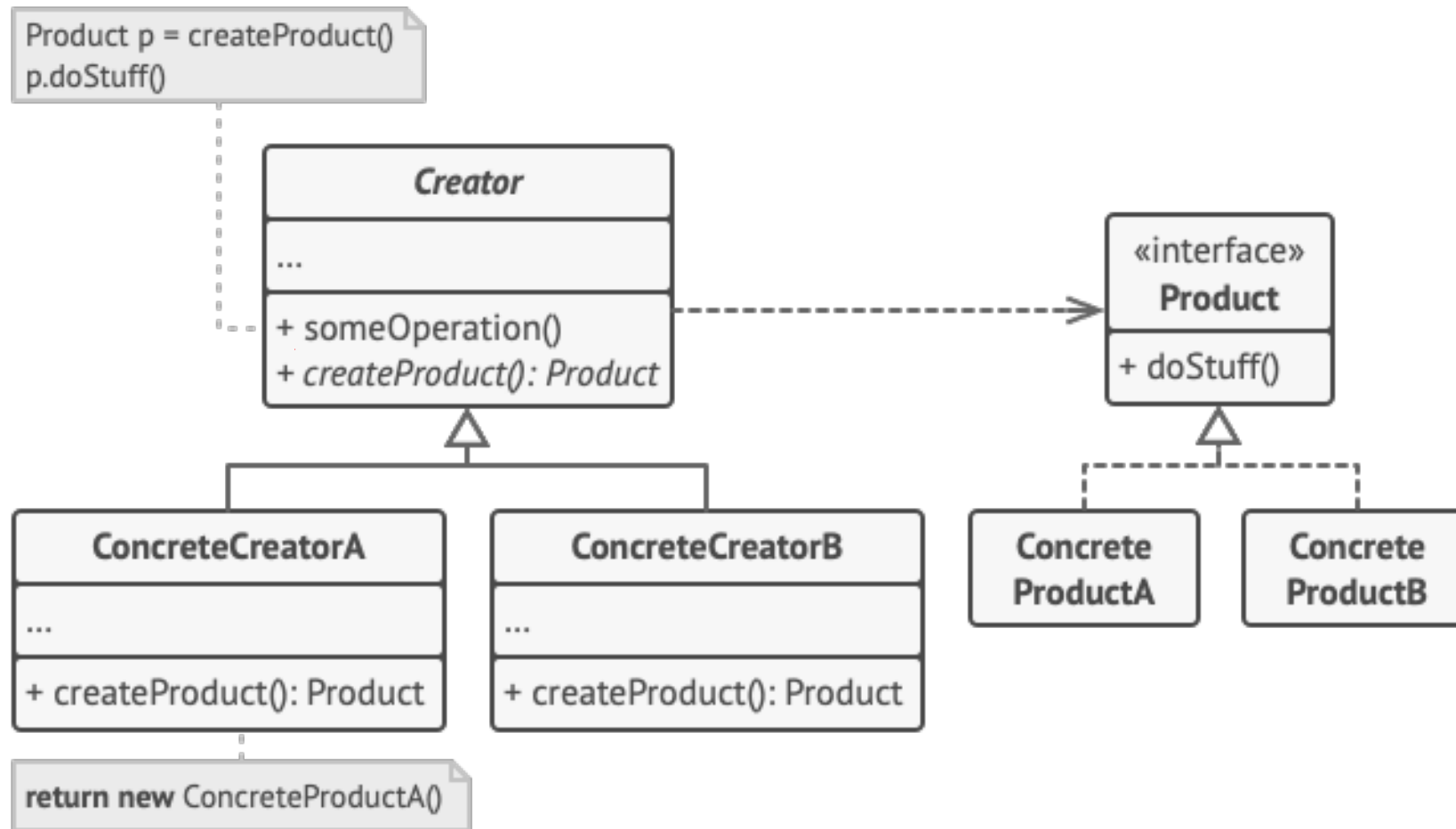
Class diagram



Abstract factory and factory method

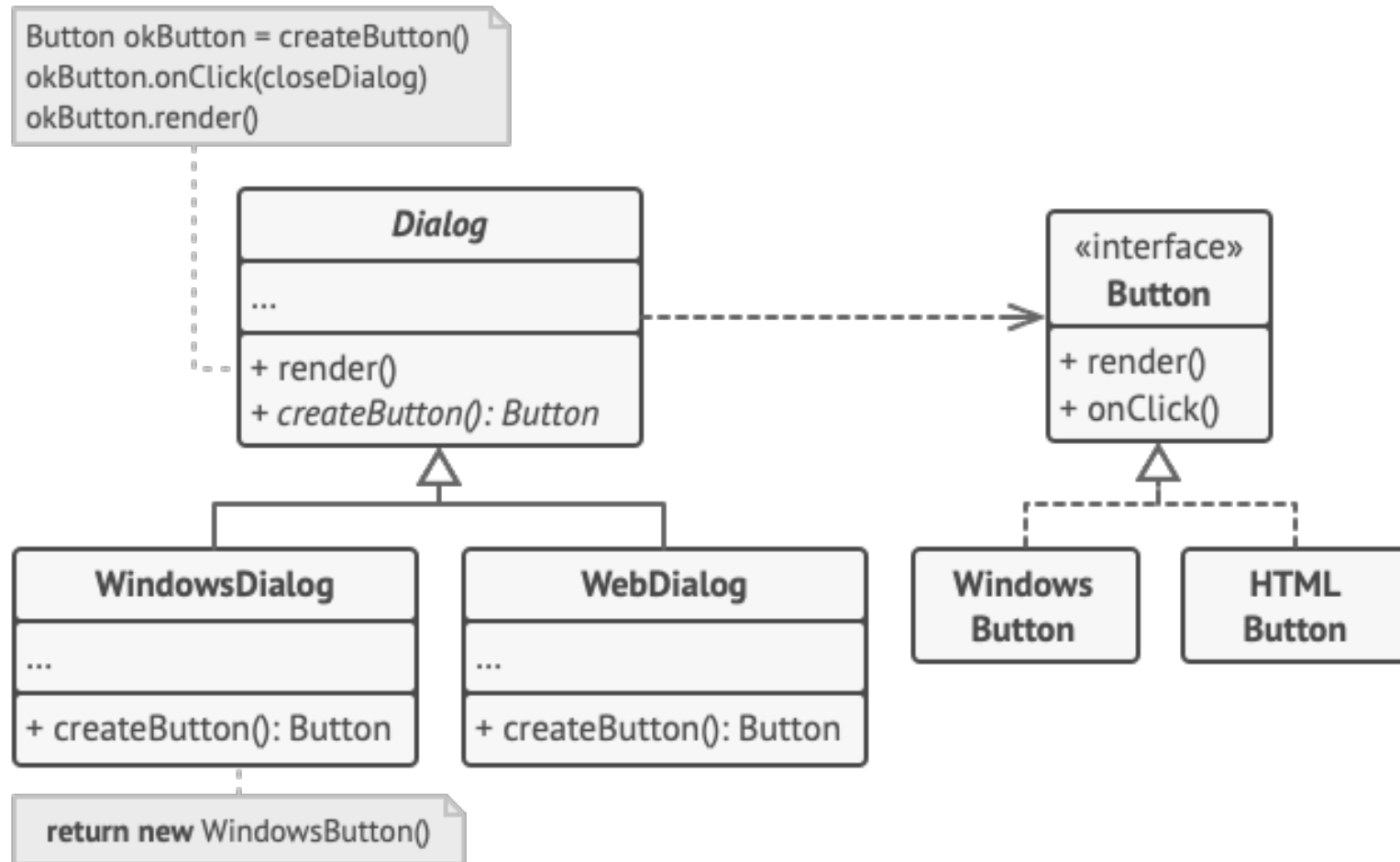
- Creational pattern
- Create related objects without specifying the concrete classes
- Which concrete type to create can be decided based on some input, configuration, or platform specific information
- Use factory methods that are like constructors, but may re-use objects, create objects of a subclass, have more descriptive names for what they create

Factory method class diagram

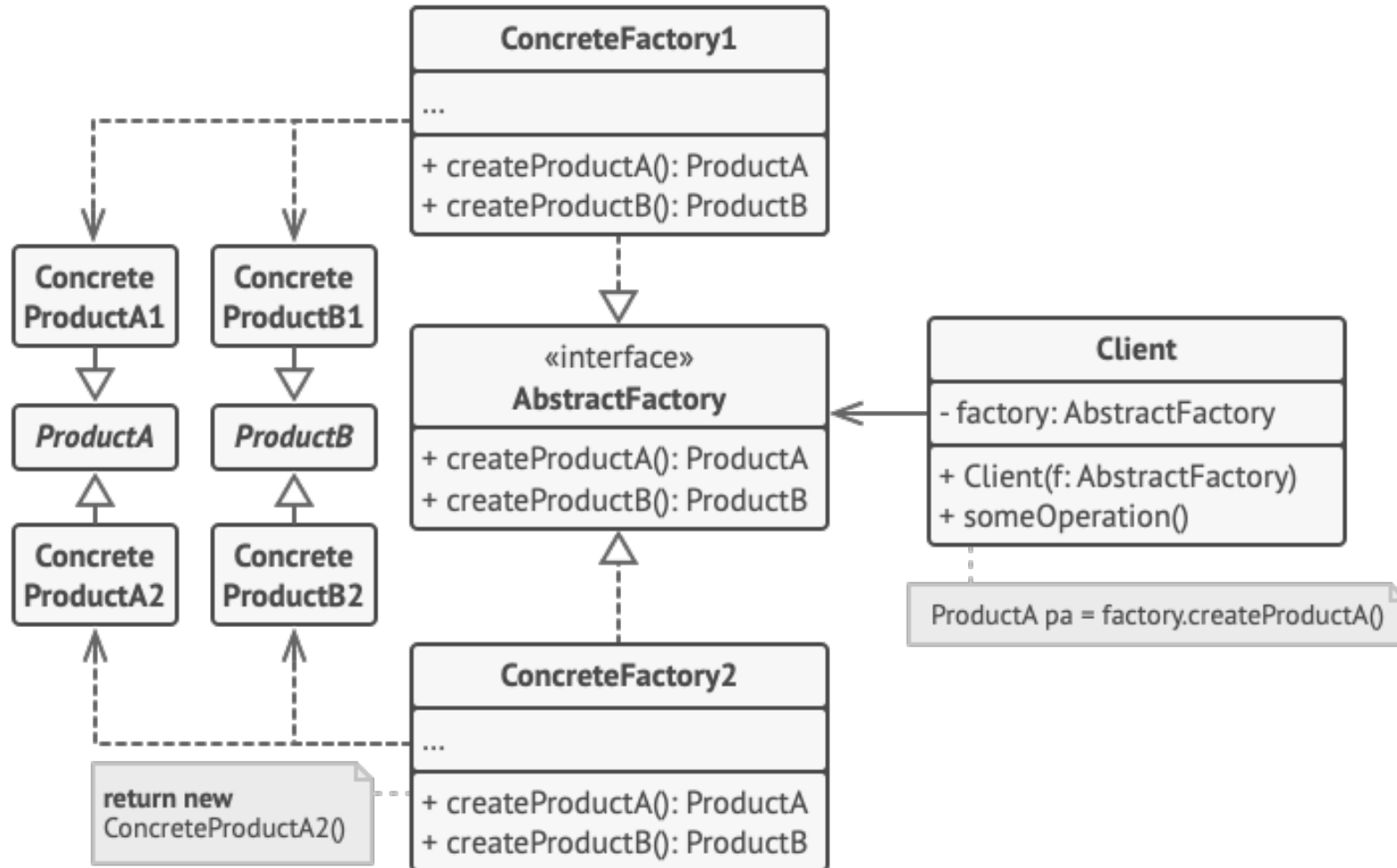


Doesn't always have to be a new object

Factory method example



Abstract factory class diagram

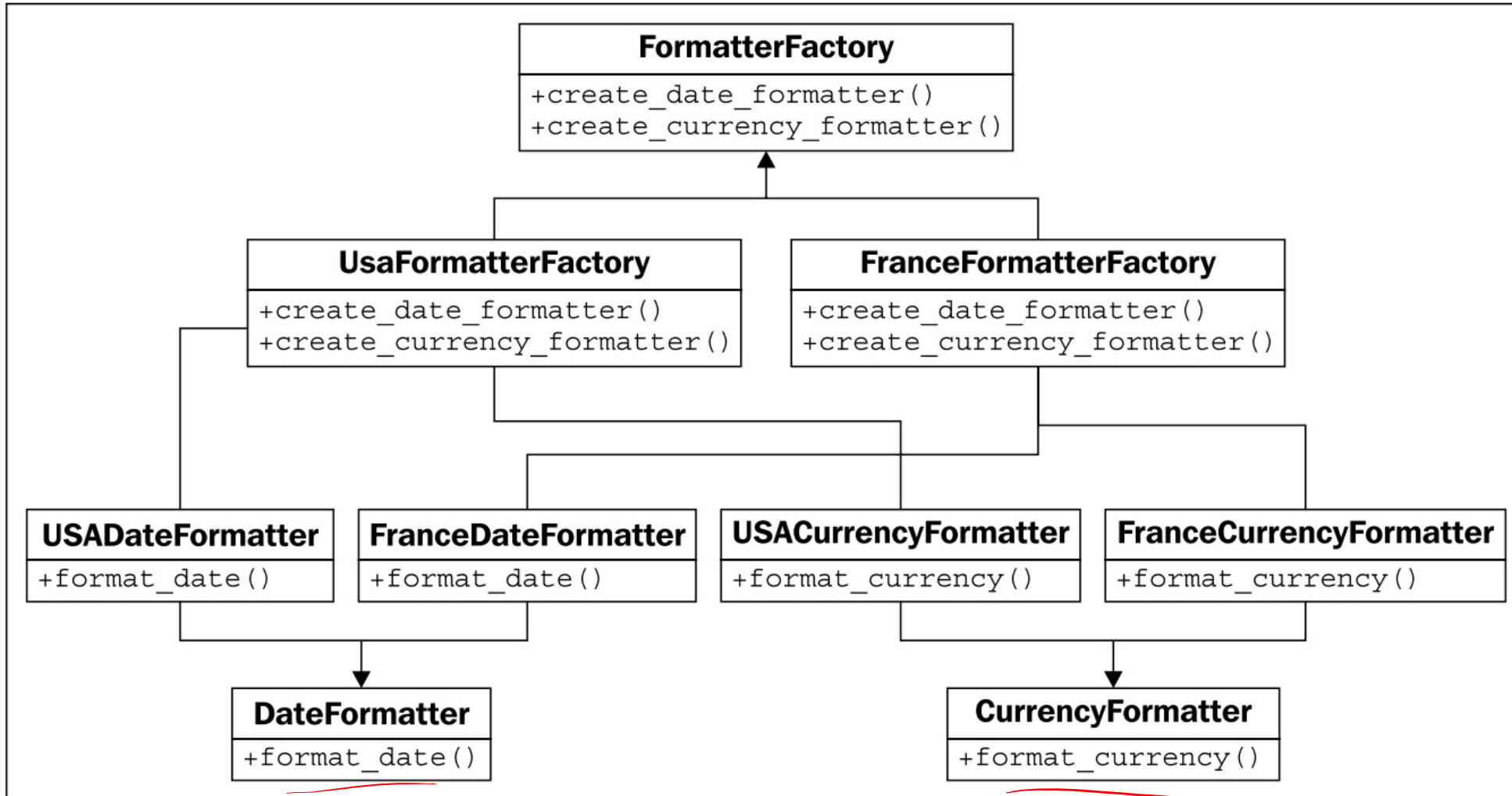


Example

- Region specific formatting

	USA	France
Date	mm-dd-yyyy	dd/mm/yyyy
Currency	\$14,500.50	14 500€50

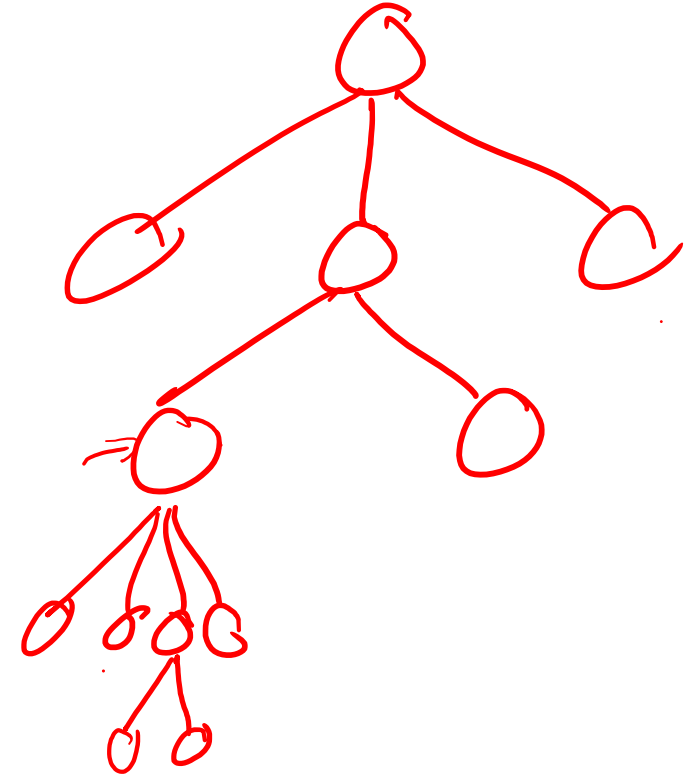
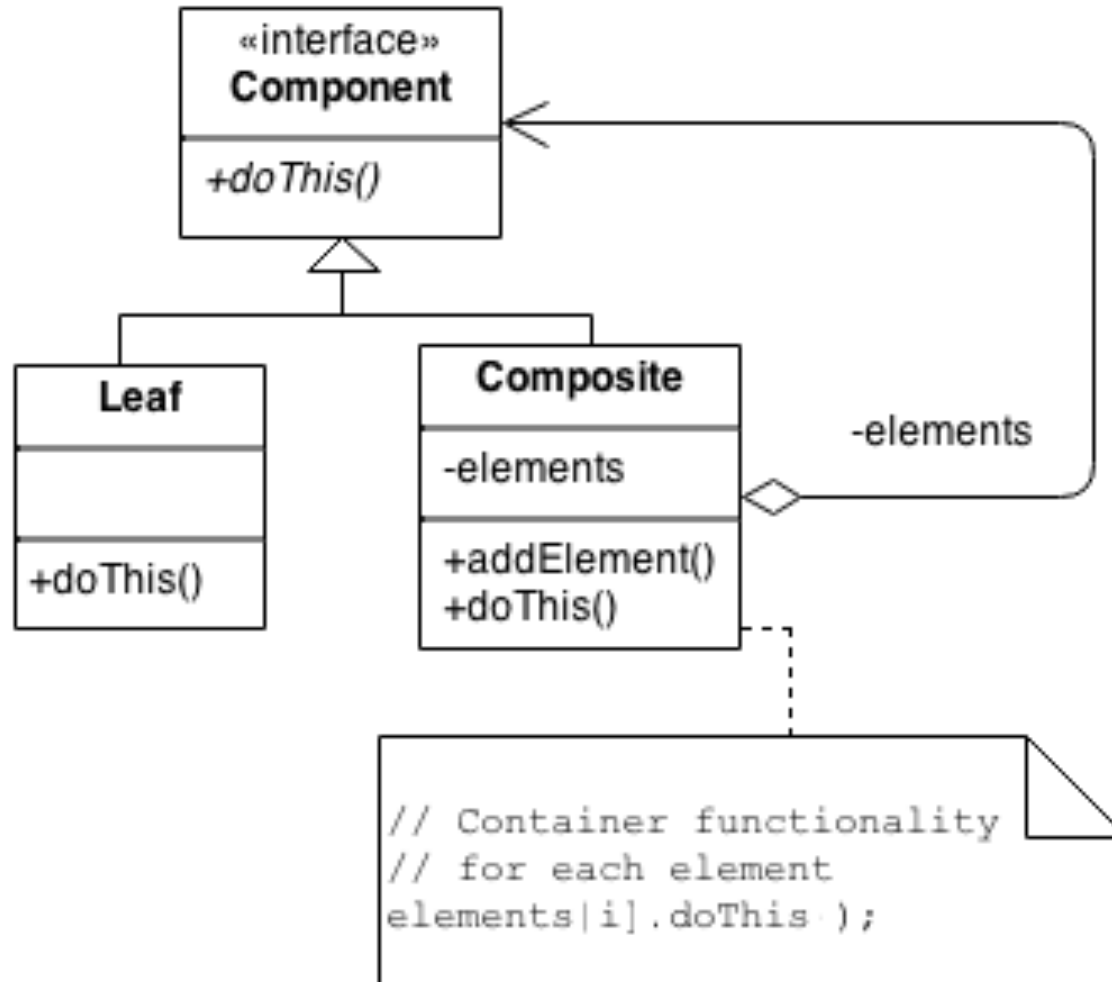
Example class diagram



Composite pattern

- Structural pattern
- Building trees through object references
 - also potentially linked lists and graphs (anything node/pointer based)
- Polymorphism allows individual objects (leaves) and collection objects (internal nodes) to be treated as one type
 - more important for statically typed languages

Class diagram



Component [] elements;