Abstract Factory with LogicAJ

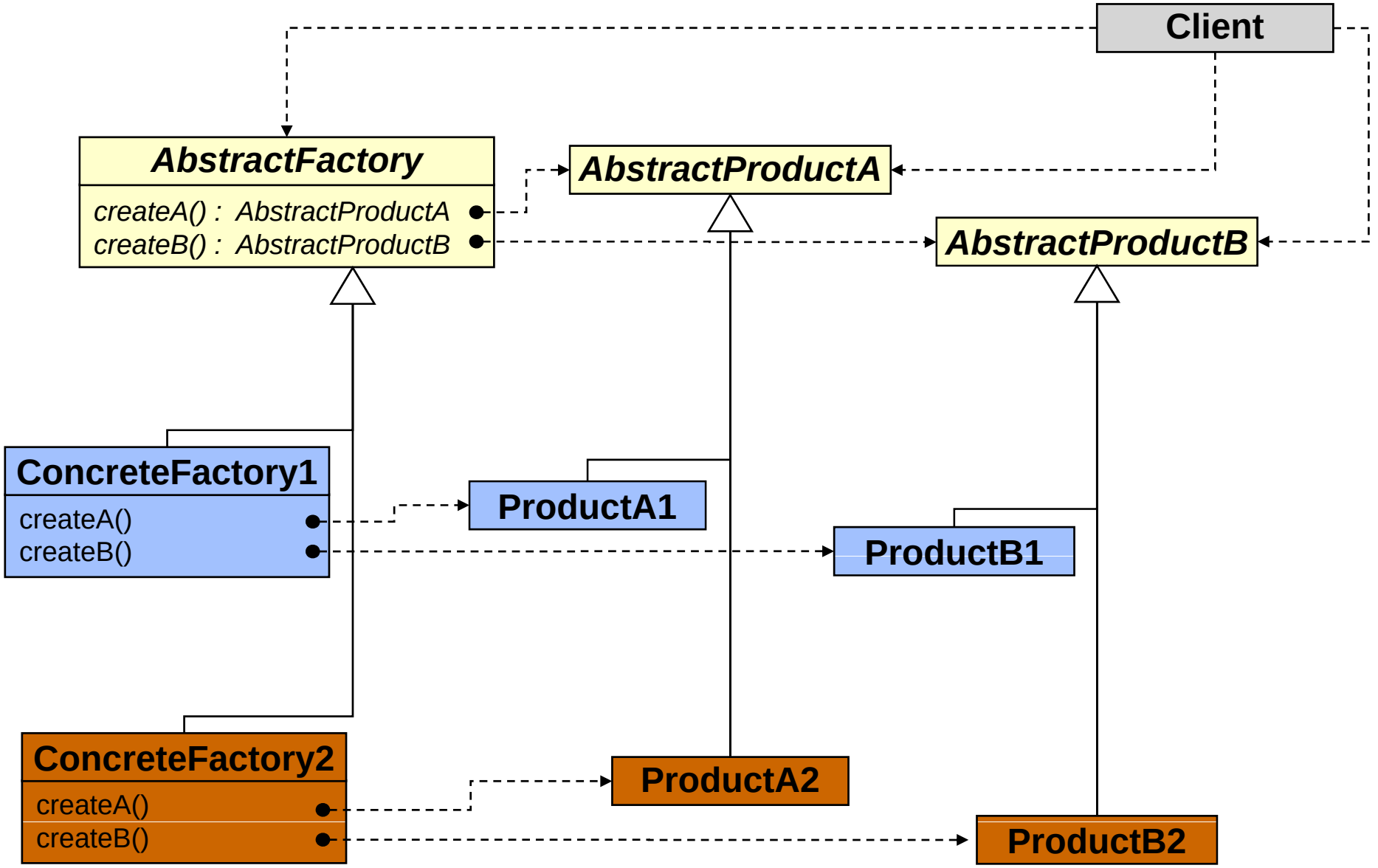
GOF-Structure of Abstract Fatory

Dependencies

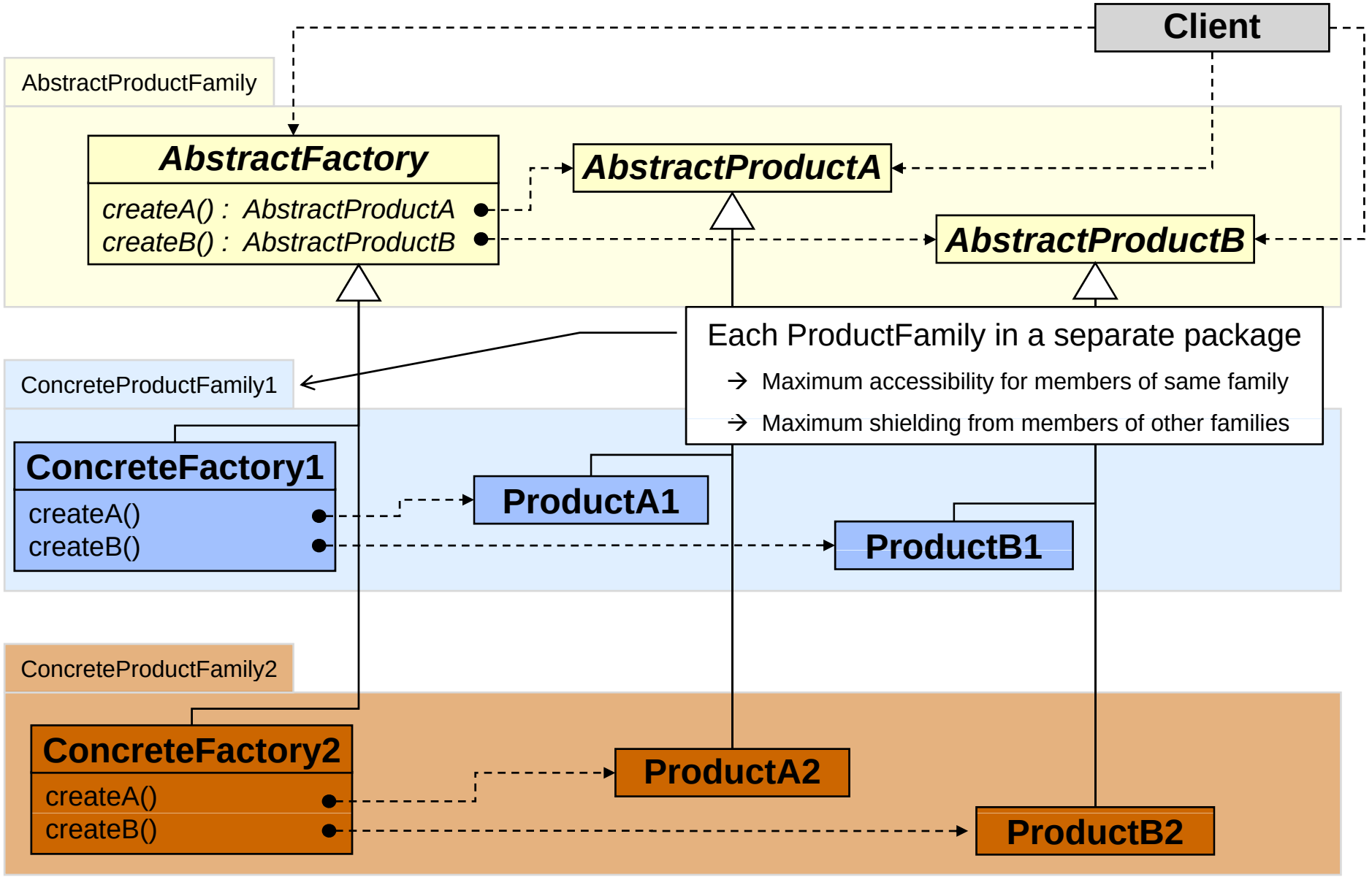
Challenges

Homework: Generating Concrete Factories for pre-existing Abstract Factories and Product Hierarchies

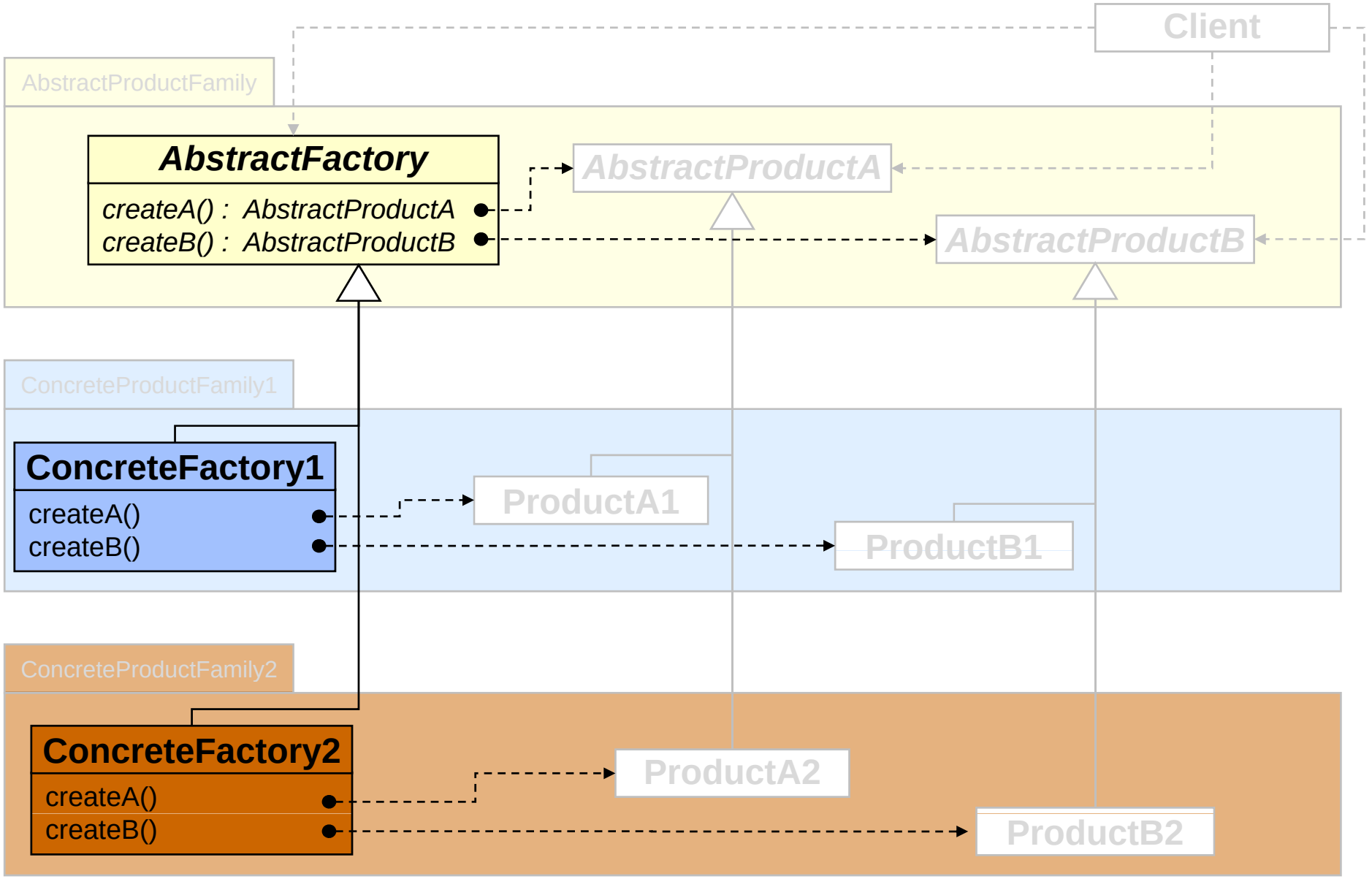
Abstract Factory: Schema



Abstract Factory: Extended Schema

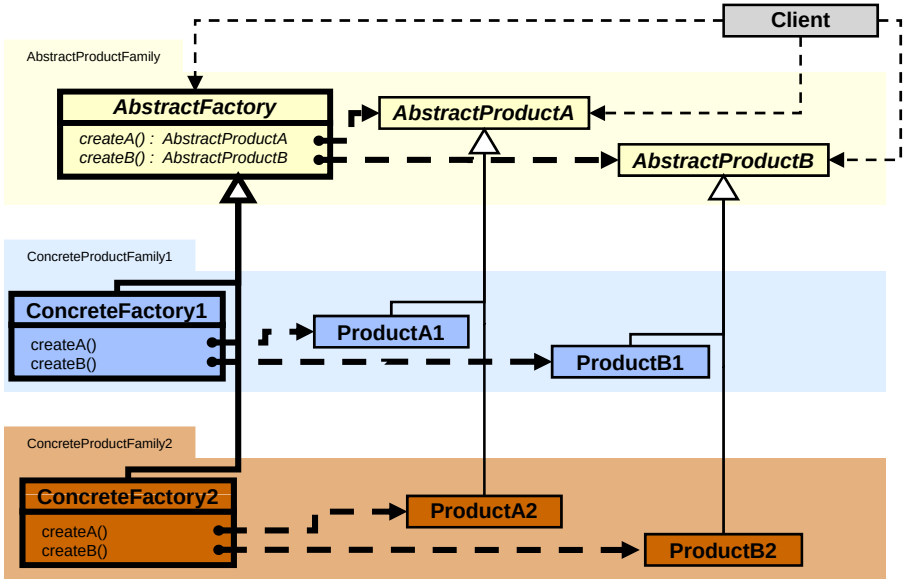


The Code to be Generated (Black)



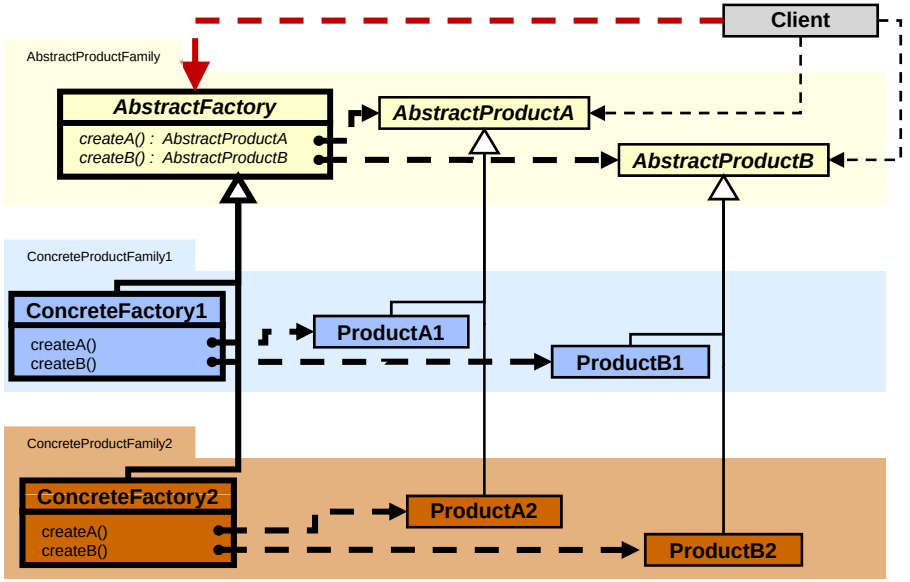
Dependencies

- AbstractFactory
  - Contains one abstract factory method for each constructor of an abstract product in the same abstract family
  - The abstract factory method for the constructor *C* of abstract product *AP* has
    - the name “create*AP*”
    - the parameters of *C*
    - *AP* as return type
- ConcreteFactory
  - If products of the concrete family CF inherit from products of the abstract family AF then there is a concrete factory CF that inherits from factory AF
  - The factory CF contains one method for each method in AF
  - A concrete factory method with return type AP returns a new instance of the ConcreteProduct that inherits from AP in the family CF



Implementation Challenges

- Expressing Dependencies
  - Compute all dependencies from the previous slide
  - Generate corresponding code
- Accessing generated code
  - Client must “know” about the AbstractFactory
  - Otherwise the base program does not compile




```
// Typical client code:
AbstractFactory factory = AbstractFactory.getFactory(family);
```

- Solution: Only generate ConcreteFactories

Homework: Generic Implementation of Concrete Factories with LogicAJ