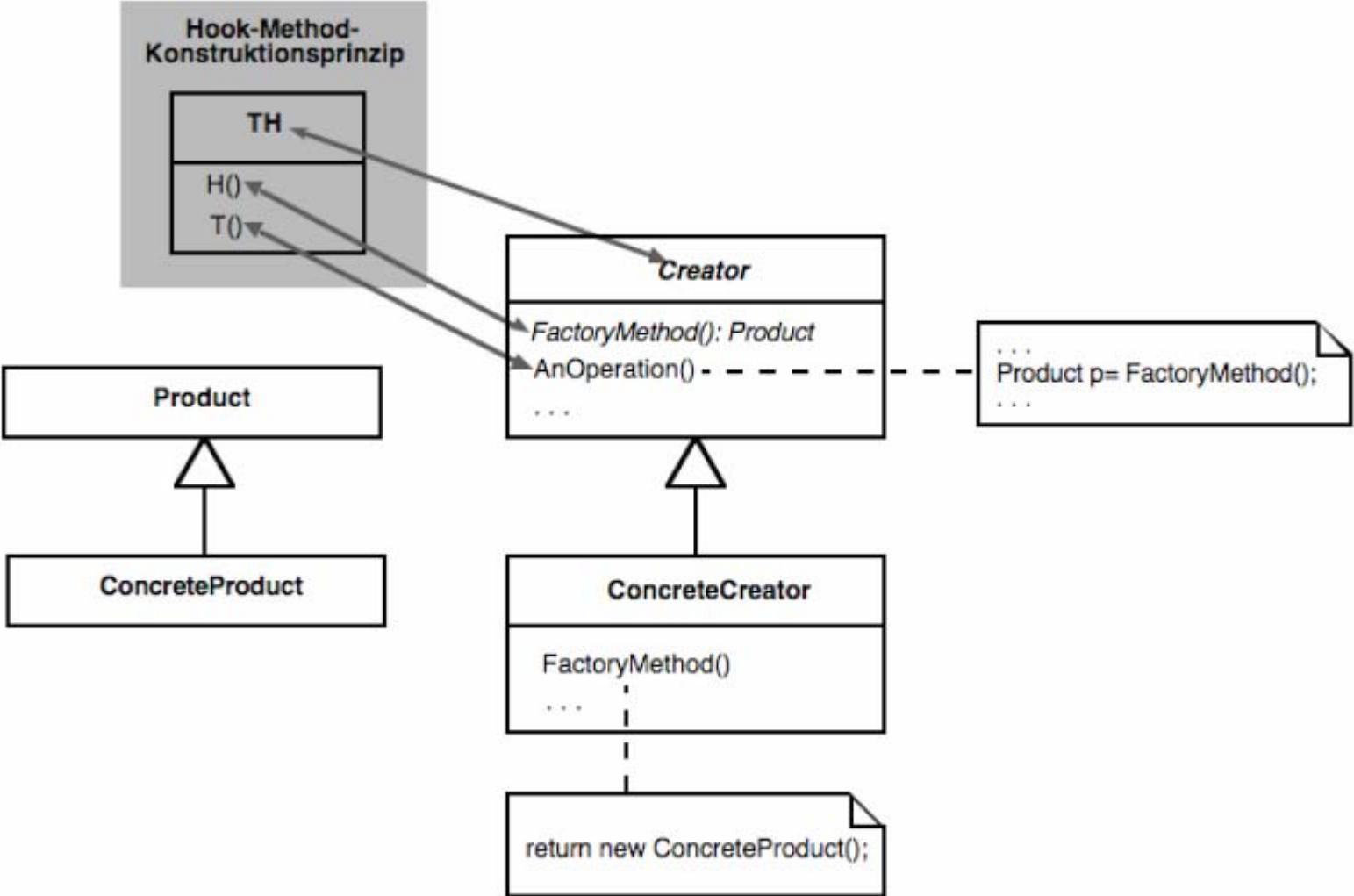


Construction of Flexible Software

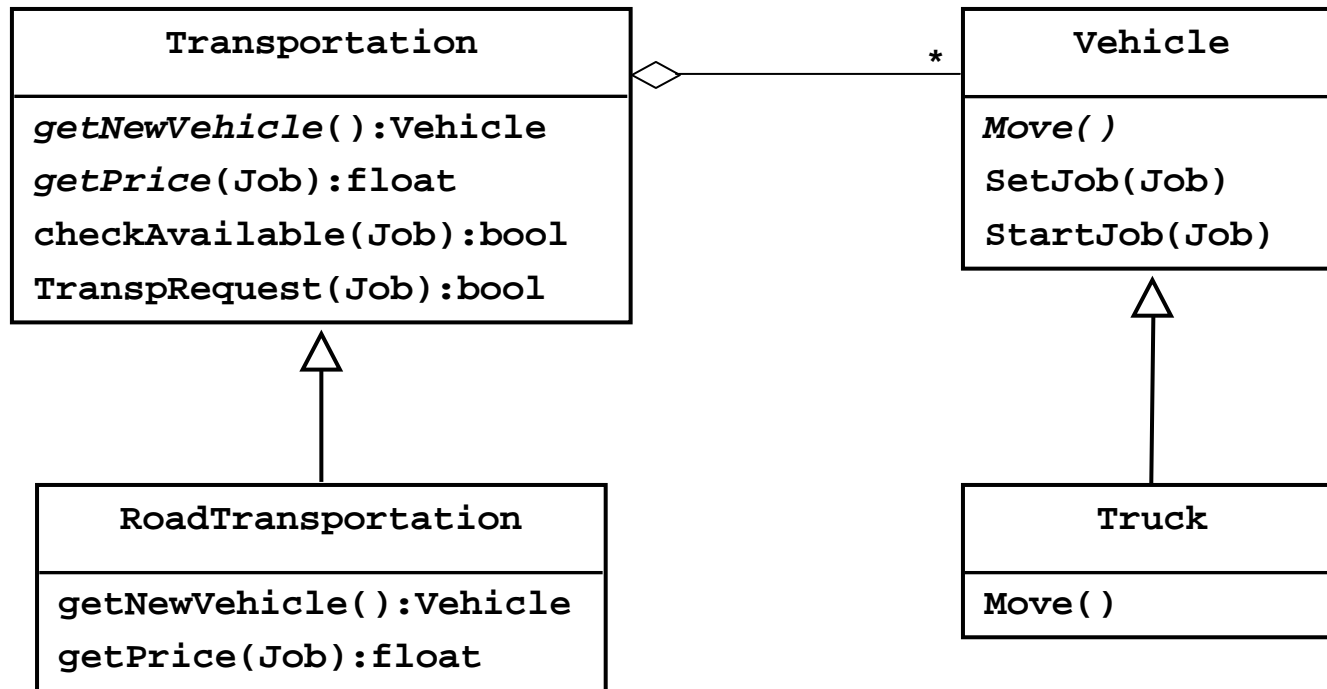
Design Patterns

- Factory Method
- Abstract Factory
- Singleton

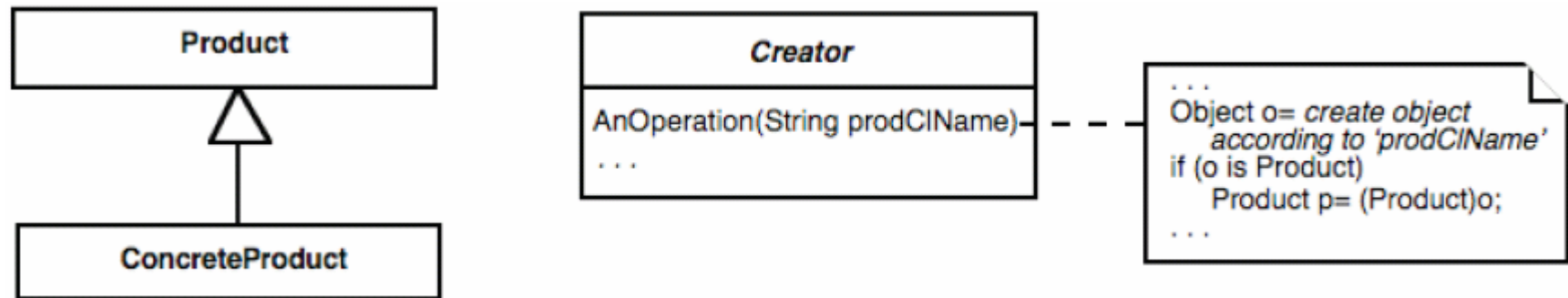
Template and Hook Methods in the Factory Method Design Pattern



Factory Method Example

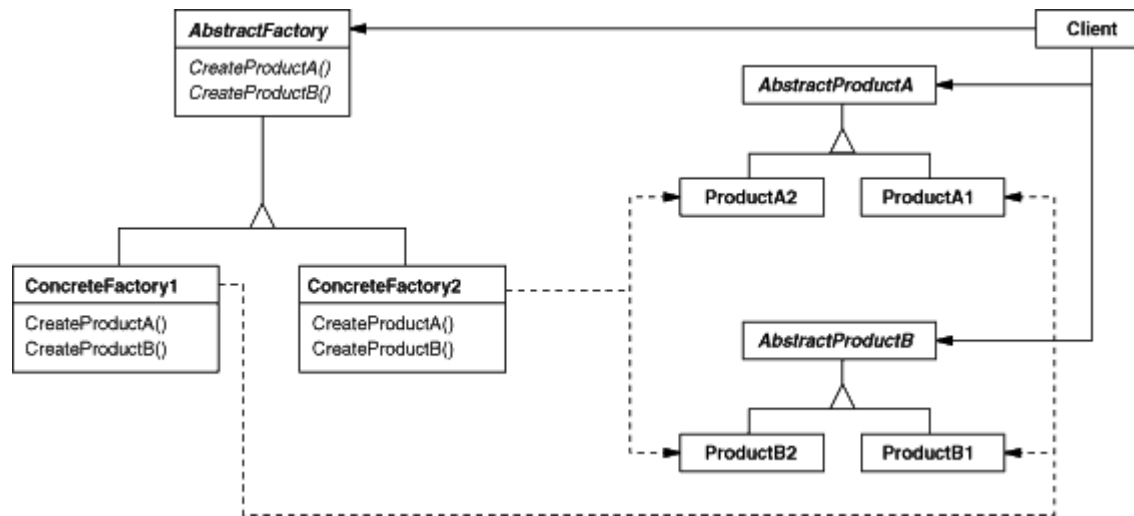


Flexible Object Production Based on Meta-Information (e.g. in Java and C#)



- + No subclassing necessary
- Static type checking is bypassed

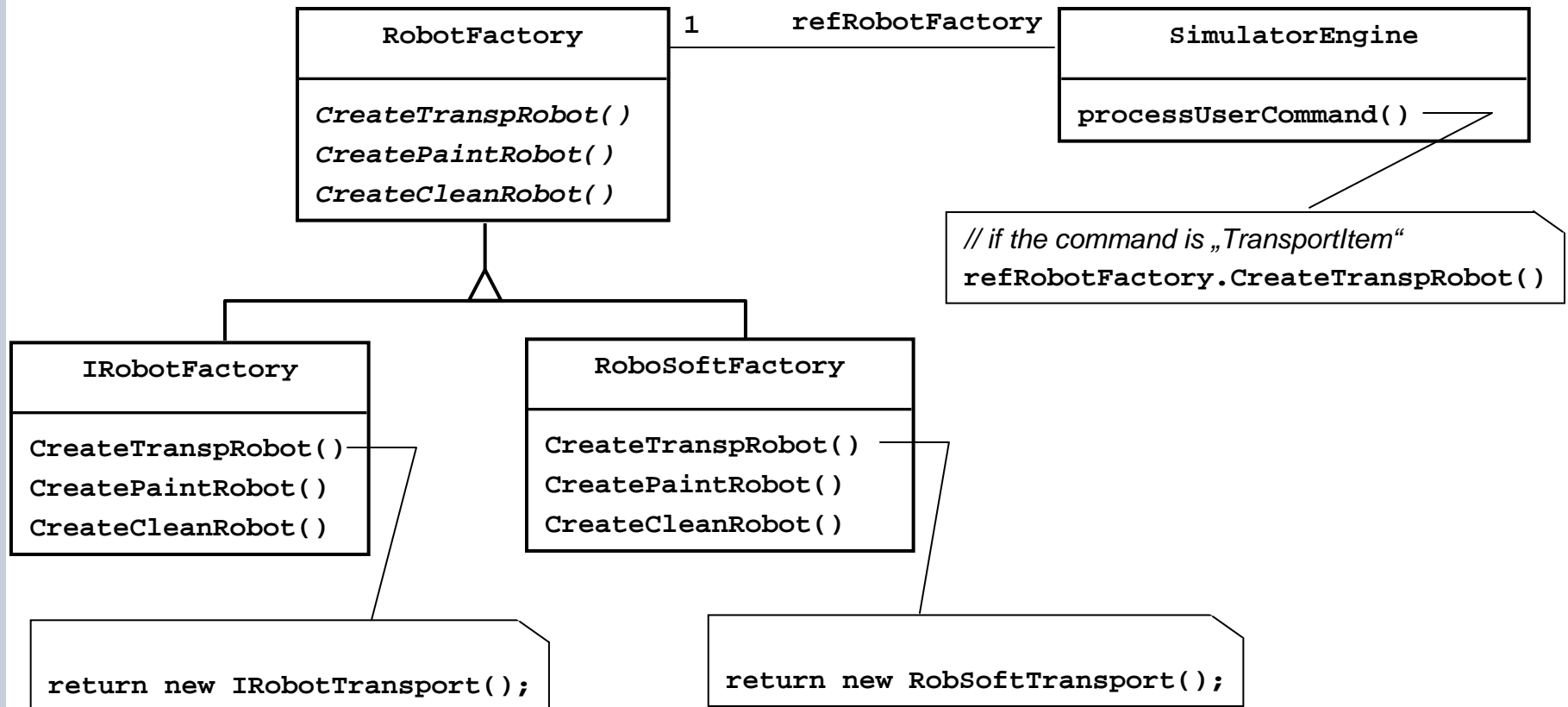
Abstract Factory



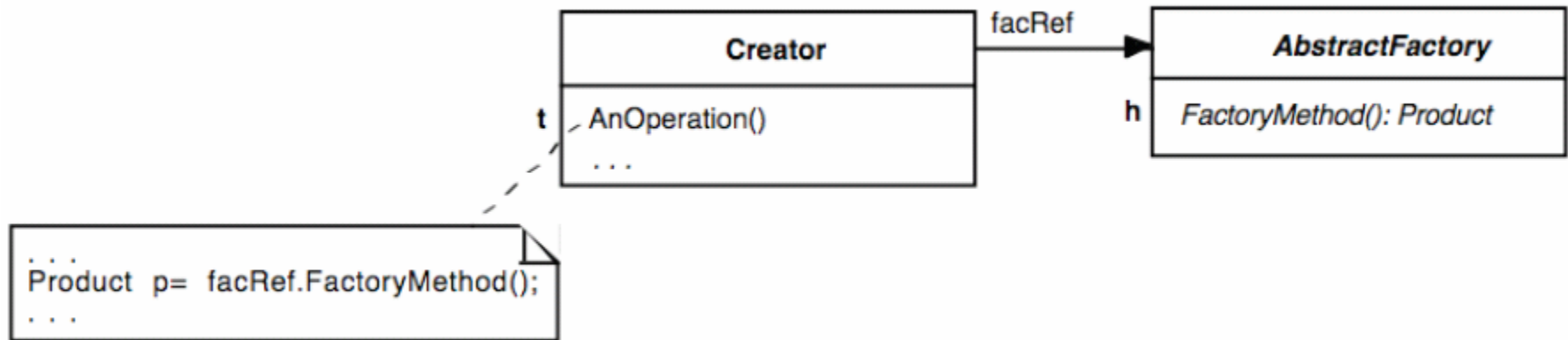
Use when:

- a system should be independent of how its products are created, composed, and represented.
- a system should be configured with one of multiple families of products.
- a family of related product objects is designed to be used together, and you need to enforce this constraint.
- you want to provide a class library of products, and you want to reveal just their interfaces, not their implementations.

Abstract Factory Example

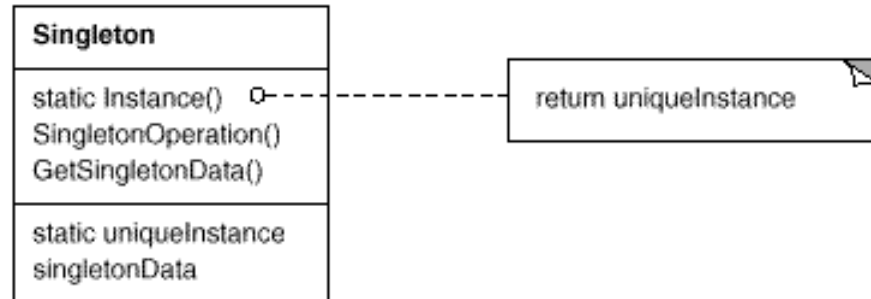


Factory Method (Hook Method) → Abstract Factory (Hook Object)



- The Hook method `FactoryMethod ()` is simply shifted in a separate class or interface

Singleton



Use when:

- there must be exactly one instance of a class, and it must be accessible to clients from a well-known access point.
- when the sole instance should be extensible by subclassing, and clients should be able to use an extended instance without modifying their code.

Singleton Examples

Factory floor

- Maintains the layout of the factory floor
- User configurable

Machine job controller

- Manages machine operations

Robot job controller

- Receives a manufacturing order
- Determines the sequence of robot and machine operations necessary to achieve the order