Creational Design Pattern

Hung Tran

Fpt software

August 11, 2021

Outline

Creational Pattern Overview

Construction process of an object.

- **Singleton**: Ensure only one instance.
- **Factory Method**: Create instance without depending on its concrete type.
- **Object pool**: Reuse existing instances.
- **Abstract factory**: Create instances from a specific family.
- **Prototype**: Clone existing objects from a prototype.
- **Builder**: Construct a complex object step by step.

"new" operator problem

```
1  #include <iostream>
2
3  using namespace std;
4
5  class Box {
6  private:
7    double length;
8    double breadth;
9    double height;
10 };
11
12 int main(void) {
13   Box *pBox = new Box();
14   delete pBox;
15   return 0;
16 }
```

- Need name of class
- Tightly coupled with the name
- Add new class, modify the existing code
- Compiler does not know which instance created at compile time or an instance has to be created at runtime?

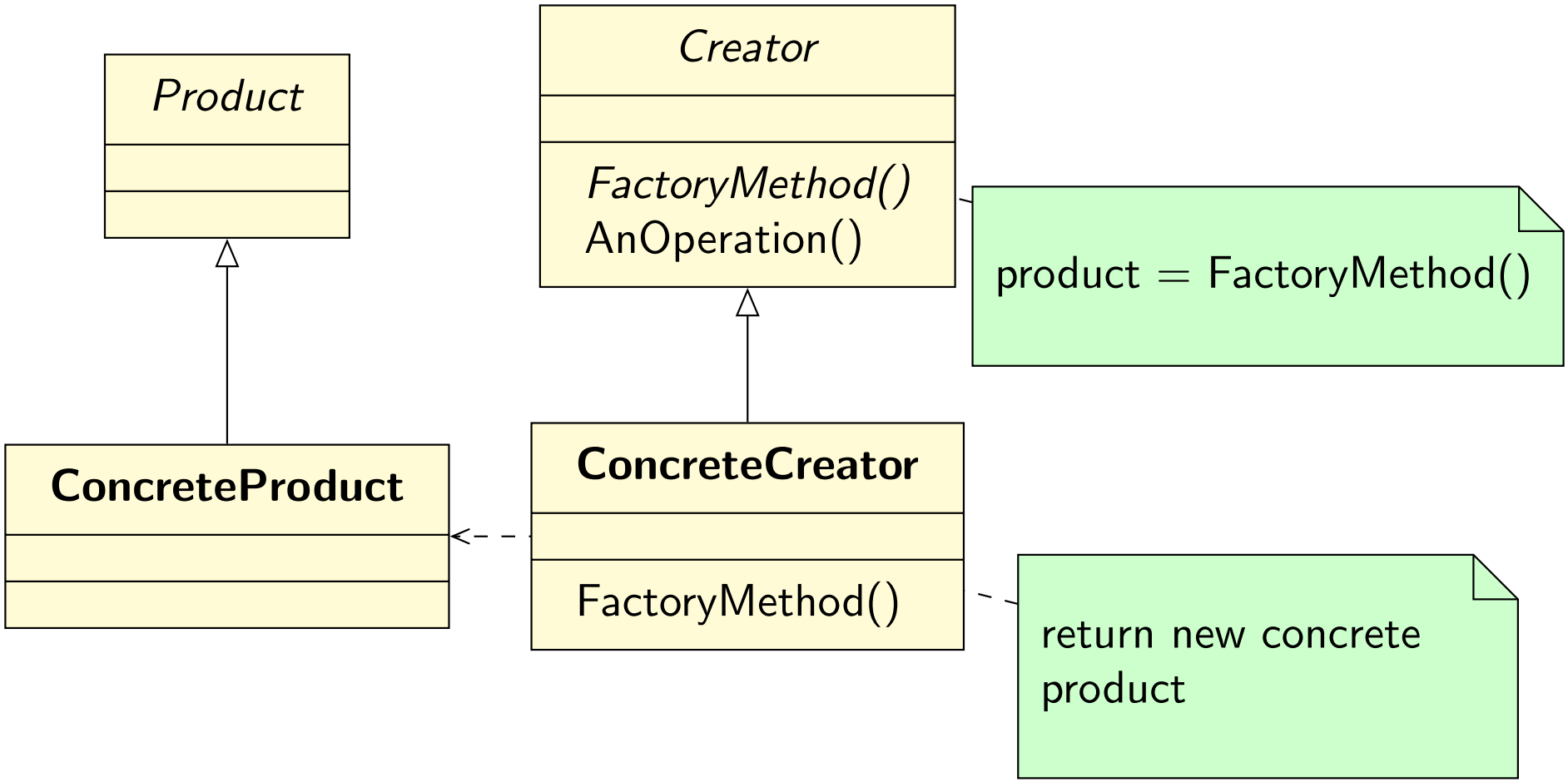
The Intent of Factory Method Design Pattern

Define an interface for creating an object, but let subclasses which class to instantitate. Factory method lets class defer instantiation to subclasses.

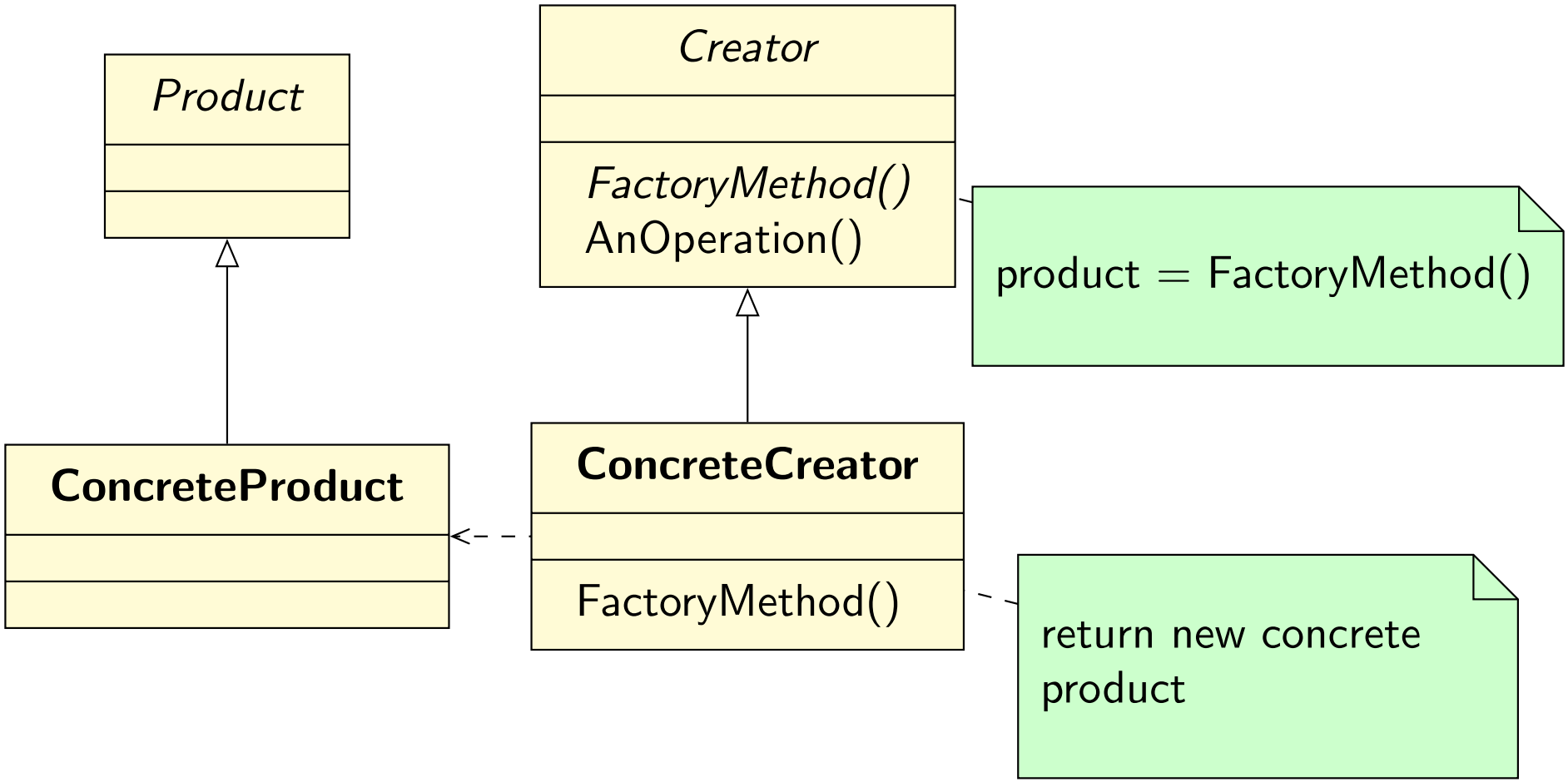
How to Implement of Factory Method Design Pattern?

- Different ways to implement
- An overridable method is provide that returns an instance of a class
- This method can be overridden to return instance of a subclass
- Behave likes **constructor**
- However, the constructor always returns the same instance
- The factory method can returns any sub-type
- The factory method also called **virtual constructor**
- C++ language does not allow virtual constructor

Structure of Factory Method Design Pattern



Basic implementation



Pros and Cons

Pros

- Class itself control the instantiation process.
- Can allow multiple instances.
- Better than global variable.
- Can be subclassed.

Cons

- Testing is difficult
- DCLP is defective
- Lazy destruction is complex

Where to use?

When only one instance should be use because:

- multiple instances cause data corruption.
- managing global state or shared state.
- multiple instances are not required.