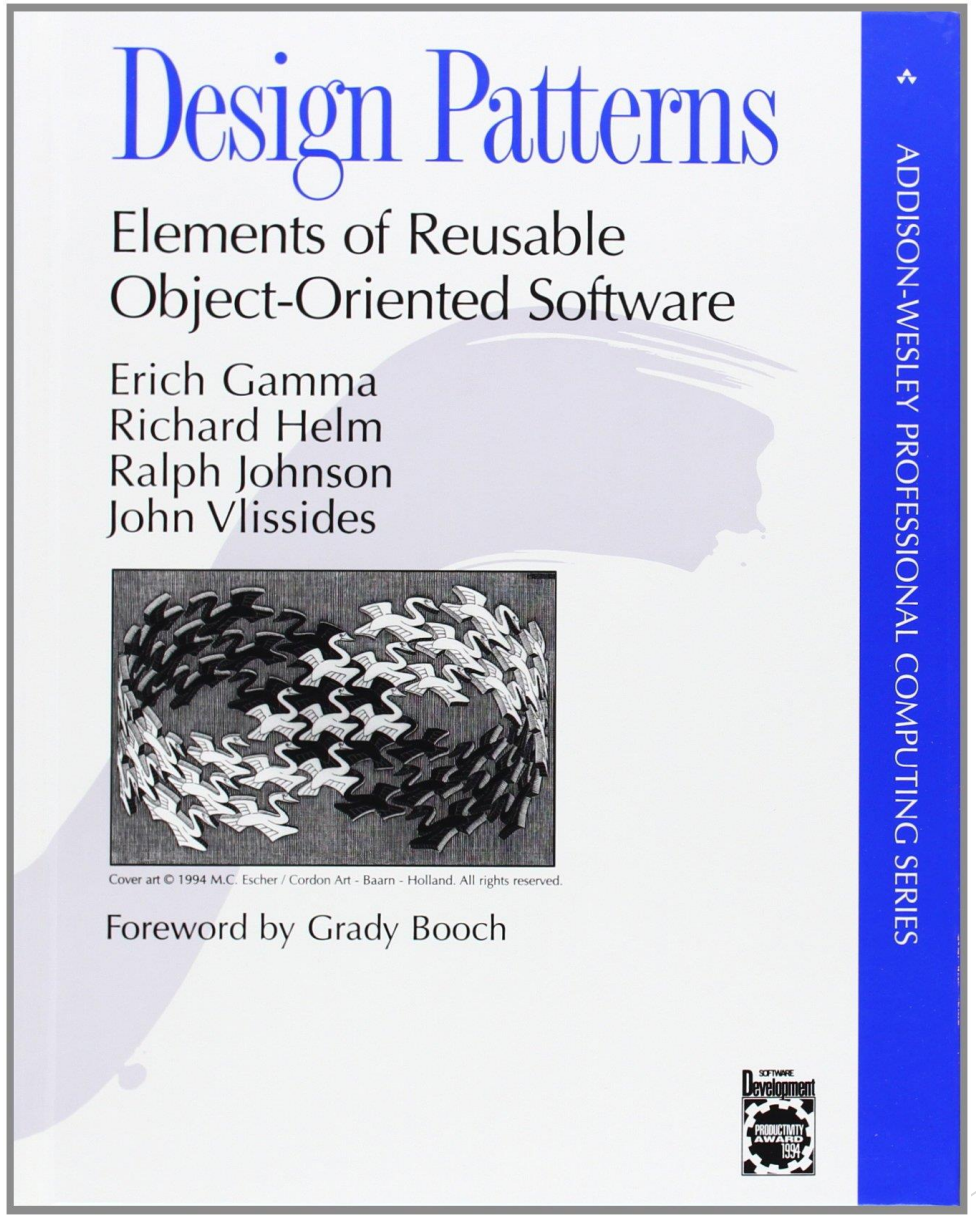
FACTORY METHOD

Triệu Quốc Huy
Lý Kim Long
Đinh Đạt Thành
Vũ Hoàng Quân

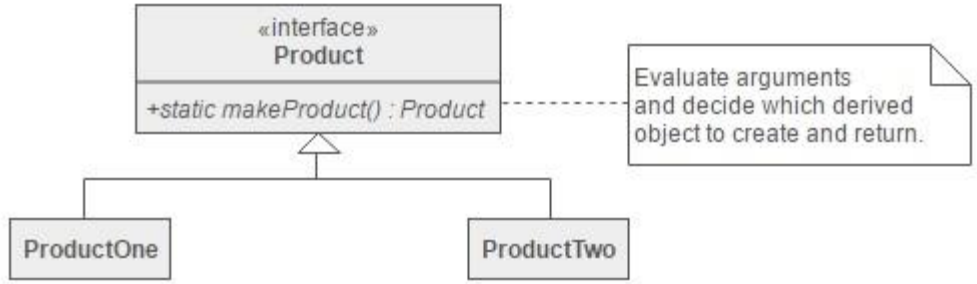
"Define an interface for creating an object, but let subclasses decide which class to instantiate. The Factory method lets a class defer instantiation it uses to subclasses.“



DEFINITION

A static method of a class that returns an object of that class' type.

Defines and interface for creating the object and let subclasses decide which class to instantiate.



WHEN TO USE

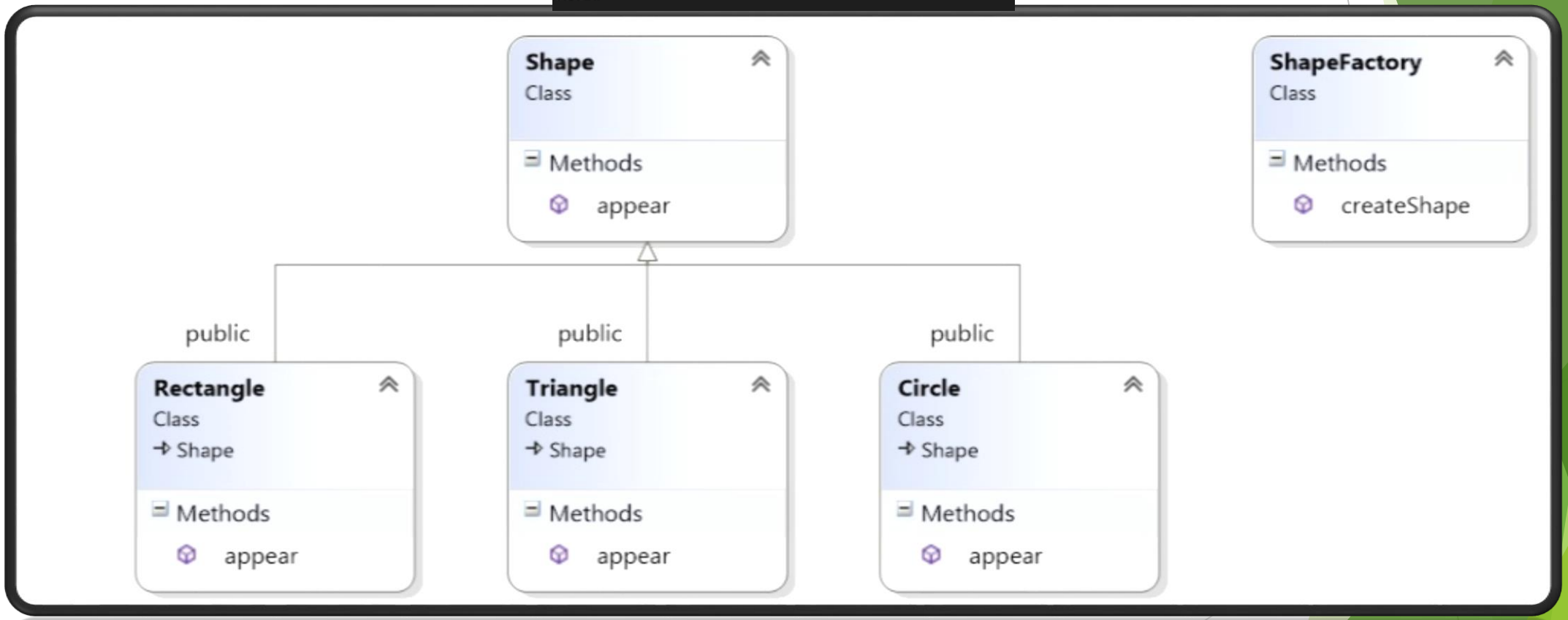
- A class can’t anticipate the class of objects it must create.
- A class wants its subclasses to specify the objects it creates.

VARIATION

- The Creator class is an abstract class and does NOT provide an implementation for the factory method it declares.

- The Creator is a concrete class and provides a default implementation for the factory method.

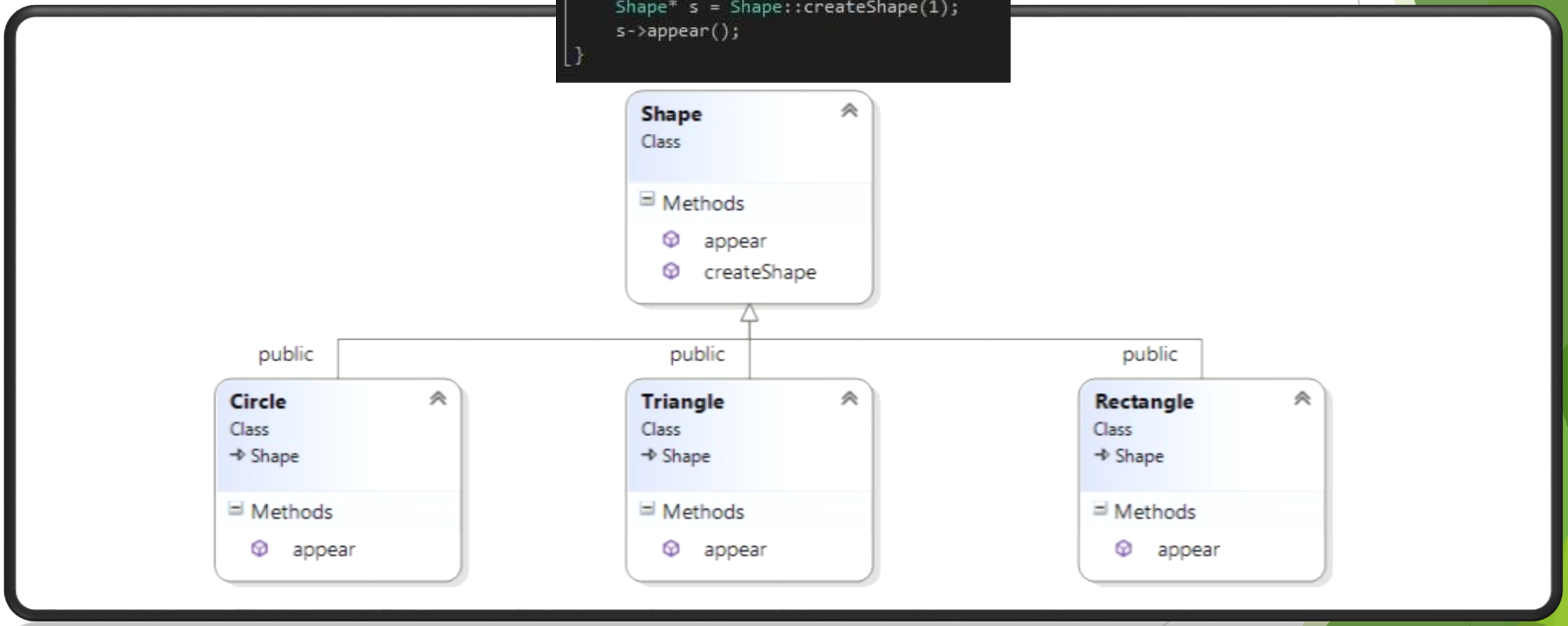
```
class ShapeFactory
{
public:static Shape* createShape(int type)
    {
    if (type == 1)
        return new Triangle;
    else if (type == 2)
        return new Circle;
    else return new Rectangle;
    }
};
```



```
Shape* Shape::createShape(int type)
{
    if (type == 1)
        return new Triangle;
    else if (type == 2)
        return new Circle;
    else return new Rectangle;

}

void main()
{
    Shape* s = Shape::createShape(1);
    s->appear();
}
```



PROPERTY

- Unlike a constructor, the actual object it returns might be an instance of a subclass.
- Unlike a constructor, an existing object might be reused, instead of a new object created.
- Unlike a constructor, factory methods can have different and more descriptive names, for example :
    - Color.make_RGB_color(float red, float green, float blue)
    - Color.make_HSB_color(float hue, float saturation, float brightness)

COMPARISION

FACTORY METHOD	ABSTRACT FACTORY
• A method which can be overridden by subclasses. • Creates object through interfaces. • Contains method to produce the products of its type.	• An object which contains multiple factory method. • Creates object through composition. • Contains family of types and method to produce.

ADVANTAGES

- Factory Method makes a design more customizable and only a little more complicated. Other design patterns require new classes, whereas Factory Method only requires a new operation.
- It can return the same instance multiple times, or can return a subclass rather than an object of that exact type.

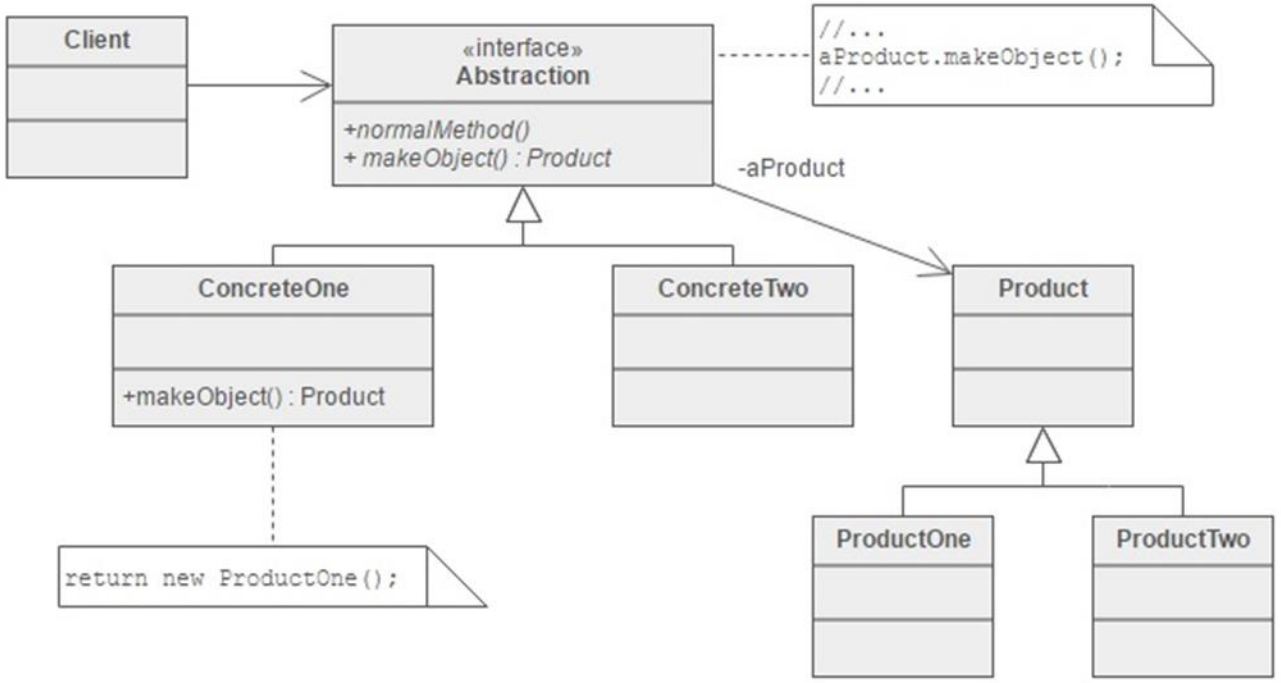
DISADVANTAGES

Additional code, difficulty in understand the general flow and a confusing inducing class diagram.

APPLICATION

Factory Methods are routinely specified by an architectural framework, and then implemented by the user of the framework.

People often use Factory Method as the standard way to create objects


Client
«interface»
Abstraction
+normalMethod()
+ makeObject() : Product
//...
aProduct.makeObject();
//...
-aProduct
ConcreteOne
+makeObject() : Product
ConcreteTwo
Product
ProductOne
ProductTwo
return new ProductOne();

THE END

Thanks for listening!