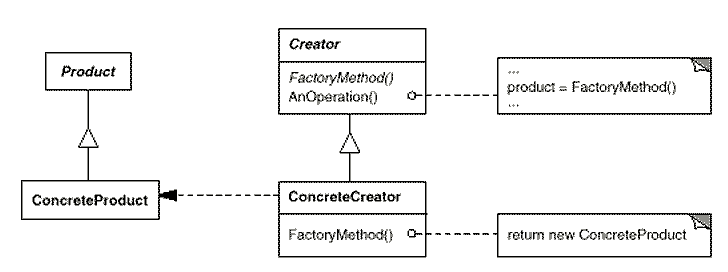
Factory Pattern

GOF : **Define an interface for creating an object, but let subclasses decide which class to instantiate. Factory Method lets a class defer instantiation to subclasses.**

# **Structure Diagram**



Java Code

**class** ConcreteProductA **implements** Product {

@Override

**public** String getName() {

**return** "Product-A";

}

}

**interface** Product {

String getName();

}

**public** **class** Test {

**public** **static** **void** main(String[] args) {

AbstractProductCreator creator = **new** ConcreteCreator();

Product product = creator.getProduct("A");

System.***out***.println(product.getName());

}

}

**class** ConcreteCreator **extends** AbstractProductCreator {

**private** Product product = **null**;

**public** Product getProduct(String type) {

**if** (type.equals("A"))

product = **new** ConcreteProductA();

**else**

product = **new** ConcreteProductA();

**return** product;

}

}

**public** **abstract** **class** AbstractProductCreator {

**public** **abstract** Product getProduct(String type);

}

**class** ConcreteProductB **implements** Product {

@Override

**public** String getName() {

**return** "Product-B";

}

}

Class Diagram

