Factory Pattern

Guowei Lv

April 12, 2015

Contents

1 Definition

The Factory Method Pattern defines an interface for creating an object, but lets subclasses decide which class to instantiate. Factory Method lets a class defer instantiation to subclasses.

2 Example

2.1 A framework for the pizza store

This is the abstract PizzaStore class:

```
package com.factory;
public abstract class PizzaStore {
    public Pizza orderPizza(String type) {
Pizza pizza;
pizza = createPizza(type);
pizza.prepare();
pizza.bake();
pizza.cut();
pizza.box();
return pizza;
    }
```

```
    abstract Pizza createPizza(String type);
}
```

Here are some concrete subclasses:

```
package com.factory;
public class NYPizzaStore extends PizzaStore {
  Pizza createPizza(String item) {
    if (item.equals("cheese")) {
      return new NYStyleCheesePizza();
    } else {
      return null;
    }
  }
}

package com.factory;
public class ChicagoPizzaStore extends PizzaStore {
  Pizza createPizza(String item) {
    if (item.equals("cheese")) {
      return new ChicagoStyleCheesePizza();
    } else {
      return null;
    }
  }
}
```

Here is the Pizza class:

```
package com.factory;

import java.util.ArrayList;

public abstract class Pizza {
  String name;
  String dough;
  String sauce;
  ArrayList toppings = new ArrayList();

  void prepare() {
    System.out.println("Preparing " + name);
    System.out.println("Tossing dough...");
    System.out.println("Adding sauce...");
    System.out.println("Adding toppings: ");
    for (int i = 0; i < toppings.size(); i++) {
      System.out.println("    " + toppings.get(i));
```

```
    }
  }

  void bake() {
    System.out.println("Bake for 25 minutes at 350");
  }

  void cut() {
    System.out.println("Cutting the pizza into diagnal slices");
  }

  void box() {
    System.out.println("Place pizza in official PizzaStore box");
  }

  public String getName() {
    return name;
  }
}
```

Now we just need some concrete Pizza class:

```
package com.factory;
public class NYStyleCheesePizza extends Pizza {
  public NYStyleCheesePizza() {
    name = "NY Style Sauce and Cheese Pizza";
    dough = "Thin Crust Dough";
    sauce = "Marinara Sauce";

    toppings.add("Grated Reggiano Cheese");
  }
}
```

```
package com.factory;
public class ChicagoStyleCheesePizza extends Pizza {
  public ChicagoStyleCheesePizza() {
    name = "Chicago Style Deep Dish Cheese Pizza";
    dough = "Extra Thick Crust Dough";
    sauce = "Plum Tomato Sauce";

    toppings.add("Shredded Mozzarella Cheese");
  }

  void cut() {
    System.out.println("Cutting the pizza into square slices");
  }
}
```

The main class:

```
package com.factory;

public class Main {
  public static void main(String[] args) {
    PizzaStore nyStore = new NYPizzaStore();
    PizzaStore chicagoStore = new ChicagoPizzaStore();

    Pizza pizza = nyStore.orderPizza("cheese");
    System.out.println("Ethan ordered a " + pizza.getName() + "\n");

    pizza = chicagoStore.orderPizza("cheese");
    System.out.println("Joel ordered a " + pizza.getName() + "\n");
  }
}
```

3 The Gradle Build Script

4 How to run

Press `C-c C-c` on the code block below.