**package** cn.simplefactory;

**public** **interface** Computer {

**void** run();

}

**package** cn.simplefactory;

**public** **class** Apple **implements** Computer {

@Override

**public** **void** run() {

// **TODO** Auto-generated method stub

System.*out*.println("苹果电脑运行了");

}

}

**package** cn.simplefactory;

**public** **class** Asus **implements** Computer {

**public** **void** run() {

// **TODO** Auto-generated method stub

System.*out*.println("华硕电脑运行了");

}

}

**package** cn.simplefactory;

/\*\*

\* 简单工厂模式（工厂模式是一种生产实例的模型，实现了对象的产生和调用的分离）

\* 当生产另外生产一个新类型的对象的是，需在factory类中增加一个产生对象的方法。

\* **@author** Administrator

\*

\*/

**public** **class** Factory {

**public** **static** Computer createAsusComputer(){

**return** **new** Asus();

}

**public** **static** Computer createAppleComputer(){

**return** **new** Apple();

}

}

**package** cn.simplefactory;

**public** **class** Client {

/\*\*

\* **@param** args

\*/

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

// **TODO** Auto-generated method stub

Computer asus = Factory.*createAsusComputer*();

Computer apple = Factory.*createAppleComputer*();

asus.run();

apple.run();

}

}

方法工厂模式：

**package** cn.factorymethod;

/\*\*

\* 简单工厂模式（工厂模式是一种生产实例的模型，实现了对象的产生和调用的分离）

\* 当生产另外生产一个新类型的对象的是，需在factory类中增加一个产生对象的方法。

\* 工厂方法模式当生产另外生产一个新类型的对象的是，增添一个factory子类。

\* **@author** Administrator

\*

\*/

**public** **interface** Factory {

Computer createComputer();

}

**package** cn.factorymethod;

**public** **class** AppleFactory **implements** Factory {

@Override

**public** Computer createComputer() {

// **TODO** Auto-generated method stub

**return** **new** Apple();

}

}

**package** cn.factorymethod;

**public** **class** AsusFactory **implements** Factory {

@Override

**public** Computer createComputer() {

// **TODO** Auto-generated method stub

**return** **new** Asus();

}

}

**package** cn.factorymethod;

**public** **class** Client {

/\*\*

\* **@param** args

\*/

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

// **TODO** Auto-generated method stub

Factory asusFactory = **new** AsusFactory();

Computer asus = asusFactory.createComputer();

asus.run();

Computer apple = **new** AppleFactory().createComputer();

apple.run();

}

}

抽象工厂模式：

**package** cn.abstractfactory;

**public** **interface** HardCard {

**void** function();

}

**class** LuxuryHardCard **implements** HardCard{

@Override

**public** **void** function() {

// **TODO** Auto-generated method stub

System.*out*.println("空间大");

}

}

**class** LowHardCard **implements** HardCard{

@Override

**public** **void** function() {

// **TODO** Auto-generated method stub

System.*out*.println("空间小");

}

}

**package** cn.abstractfactory;

**public** **interface** VideoCard {

**void** function();

}

**class** LuxuryVideoCard **implements** VideoCard{

@Override

**public** **void** function() {

// **TODO** Auto-generated method stub

System.*out*.println("好图像");

}

}

**class** LowVideoCard **implements** VideoCard{

@Override

**public** **void** function() {

// **TODO** Auto-generated method stub

System.*out*.println("差图像");

}

}

**package** cn.abstractfactory;

/\*\*

\* 抽象工厂模式用来生产产品链

\* **@author** Administrator

\*

\*/

**public** **interface** ComputerFactory {

HardCard createHardCard();

VideoCard createVideoCard();

}

package cn.abstractfactory;

import cn.factorymethod.Computer;

import cn.factorymethod.Factory;

public class LuxuryFactory implements ComputerFactory {

@Override

public HardCard createHardCard() {

// TODO Auto-generated method stub

return new LuxuryHardCard();

}

@Override

public VideoCard createVideoCard() {

return new LuxuryVideoCard();

// TODO Auto-generated method stub

}

}

**package** cn.abstractfactory;

**public** **class** LowFactory **implements** ComputerFactory {

@Override

**public** HardCard createHardCard() {

// **TODO** Auto-generated method stub

**return** **new** LowHardCard();

}

@Override

**public** VideoCard createVideoCard() {

// **TODO** Auto-generated method stub

**return** **new** LowVideoCard();

}

}

**package** cn.abstractfactory;

**public** **class** Client {

/\*\*

\* **@param** args

\*/

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

// **TODO** Auto-generated method stub

ComputerFactory luxuryFacoty = **new** LuxuryFactory();

luxuryFacoty.createHardCard().function();

luxuryFacoty.createVideoCard().function();

ComputerFactory lowFacoty = **new** LowFactory();

lowFacoty.createHardCard().function();

lowFacoty.createVideoCard().function();

}

}