# 定义

用原型实例指定创建对象的种类，并且通过拷贝这些原型创建新的对象。

注：这与C++的深拷贝，浅拷贝息息相关，即通过对象的克隆去创建新的对象。

原型对象只是用来作为克隆去使用的，不能直接使用原型对象，使用的时候要十分小心！

# 适用

当要实例化的类是在运行时刻指定时；或者需要创建多个对象并且这些对象内部状态相差不大。

# 代码

class Car implements Cloneable{

private int id;

public int getId() {return id;}

public void setId(int id) {this.id = id;}

public Car clone(){

try {

return (Car)super.clone();

}catch (CloneNotSupportedException e) {

e.printStackTrace();

return null;

}

}

}

class Prototype implements Cloneable{

private int id;

private Car car;

public Car getCar() {return car;}

public void setCar(Car car) {this.car = car;}

public int getId() {return id;}

public void setId(int id) {this.id = id;}

public Object clone(){

try {

boolean deep = true;

if (deep){

/\*\*

\* 深复制，复制出了两辆车

\* \*/

Prototype prototype = (Prototype)super.clone();

prototype.setCar((Car)this.car.clone());

// 继续复制其他引用对象

return prototype;

}else{

/\*\*

\* 浅复制 ，是同一辆车

\* \*/

return super.clone();

}

} catch (CloneNotSupportedException e) {

e.printStackTrace();

return null;

}

}

}

public class TestPrototype {

public void test(){

Prototype p1 = new Prototype();

p1.setCar(new Car());

p1.setId(1);

// 复制

Prototype p2 = (Prototype)p1.clone();

p2.setId(2);

}

}