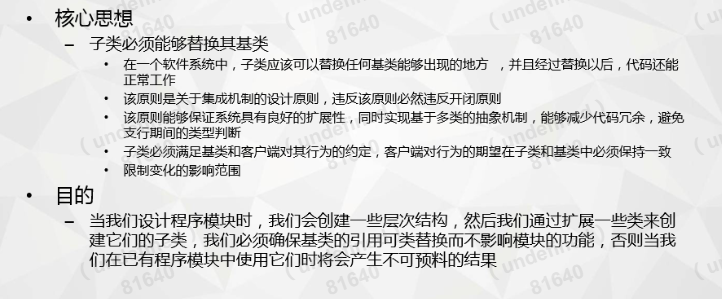
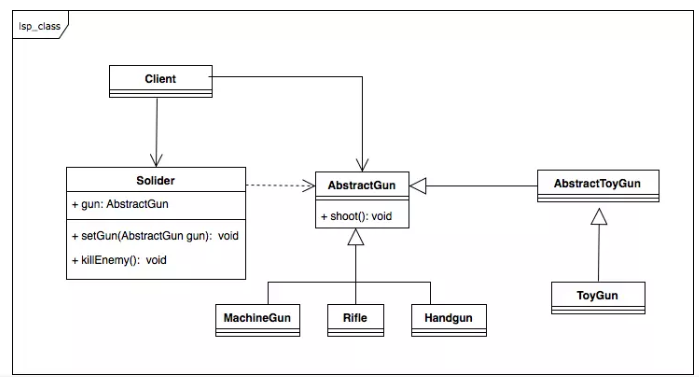
1. **定义**



**里氏替换原则，所有引用基类的地方必须能透明地使用其子类的对象。**

**2、实现**

**（1）定义手枪抽象类**

public abstract class AbstractGun {

abstract void shoot();

}

**（2）定义玩具枪抽象类**

public abstract class AbstractToy {

abstract void shoot();

}

**（3）实现手枪射击**

public class Handgun extends AbstractGun {

@Override

void shoot() {

Log.i("JimmyZhang", "手枪射击...");

}

}

**（4）实现机枪扫射**

public class MachineGun extends AbstractGun {

@Override

void shoot() {

Log.i("JimmyZhang", "机枪扫射...");

}

}

**（5）实现步枪射击**

public class Rifle extends AbstractGun {

@Override

void shoot() {

Log.i("JimmyZhang", "步枪射击...");

}

}

**（6）实现士兵类**

public class Soldier {

AbstractGun gun;

public void setGun(AbstractGun gun) {

this.gun = gun;

}

public void killEnemy() {

gun.shoot();

}

}

**（7）测试实现**

public class Client {

public static void main(String[] args) throws IOException {

Soldier soldier = new Soldier();

AbstractGun gun = new MachineGun();

soldier.setGun(gun);

soldier.killEnemy();

}

}

**3、优点**

**（1）代码共享，减少创建类的工作量，每个子类都拥有父类的方法和属性；**

**（2）提高代码的重用性；**

**（3）子类可以形似父类，但又异于父类；**

**（4）提高代码的可扩展性；**

**（5）提高产品或者项目的开放性。**

**4、缺点**

**（1）继承是入侵性的；**

**（2）降低代码的灵活性；**

**（3）增加耦合性。**

0人点赞

[设计模式](https://www.jianshu.com/nb/3970364)