

Java语言基础

【Day18】

常用的设计原则

常用的设计原则

- 开闭原则 (Open Close Principle)
对扩展开放对修改关闭，为了使程序的扩展性好，易于维护和升级。
- 里氏代换原则 (Liskov Substitution Principle)
任何基类可以出现的地方，子类一定可以出现，多使用多态的方式。
- 依赖倒转原则 (Dependence Inversion Principle)
尽量多依赖于抽象类或接口而不是具体实现类，对子类具有强制性和规范性

常用的设计原则

- 接口隔离原则 (Interface Segregation Principle)
尽量多使用小接口而不是大接口，避免接口的污染，降低类之间的耦合度。
- 迪米特法则 (最少知道原则) (Demeter Principle)
一个实体应当尽量少与其他实体之间发生相互作用，使系统功能模块相对独立。
- 合成复用原则 (Composite Reuse Principle)
尽量多使用合成/聚合的方式，而不是继承的方式。

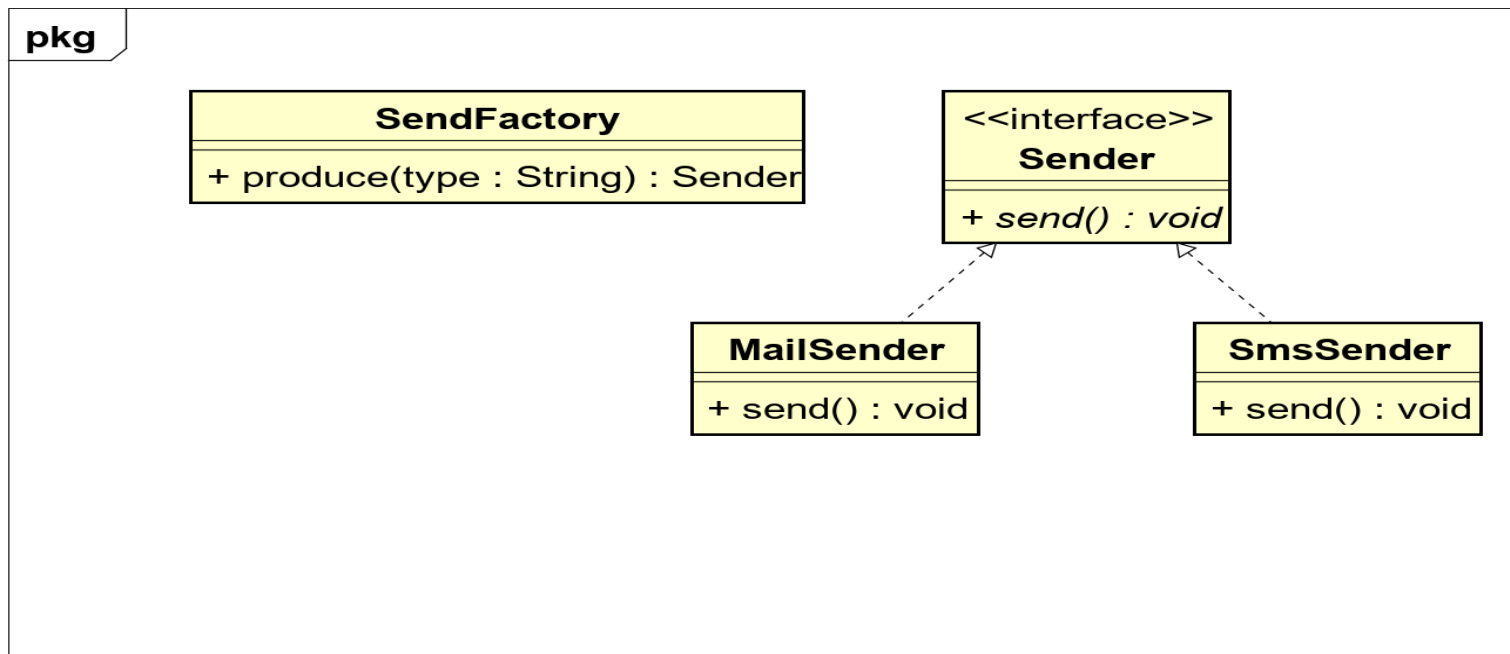
常用的设计模式

设计模式的基本概念

- 设计模式 (Design pattern) 是一套被反复使用、多数人知晓的、经过分类编目的、代码设计经验的总结。
- 设计模式就是一种用于固定场合的固定套路。

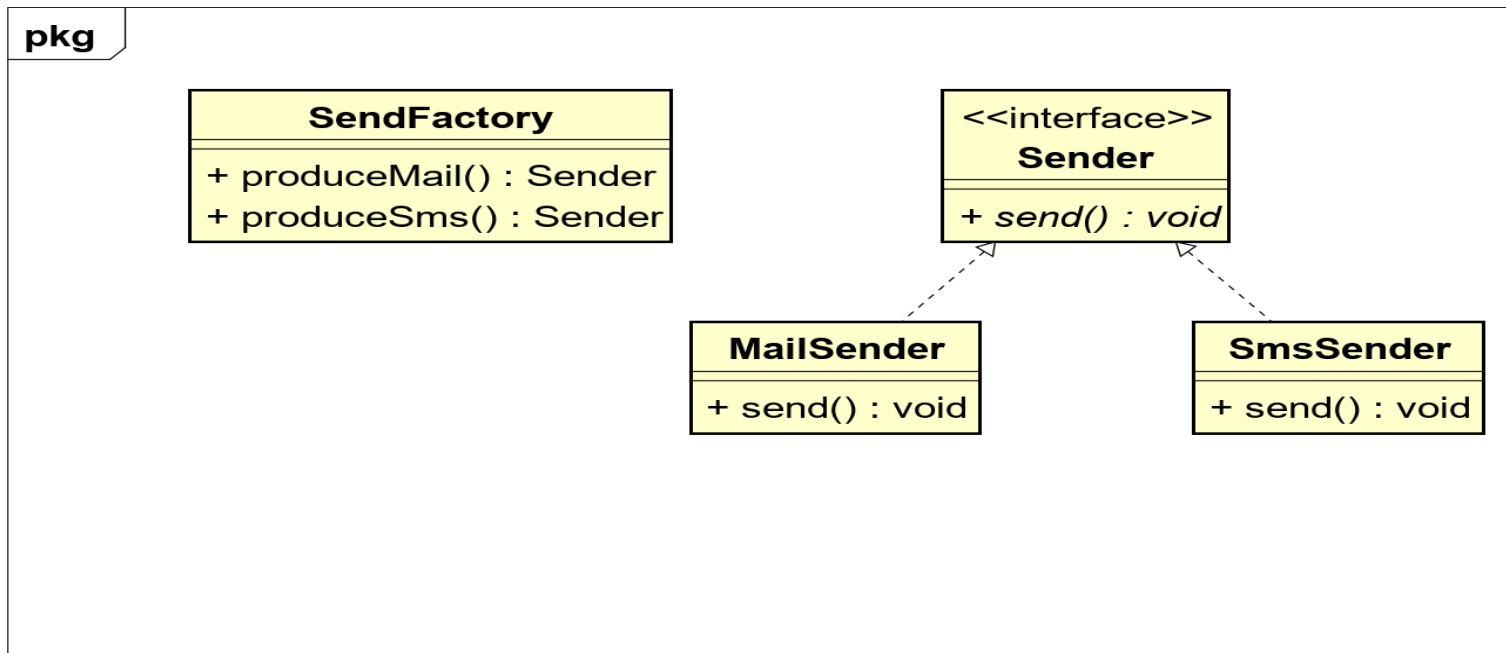
普通工厂方法模式

- 普通工厂方法模式的类图如下：



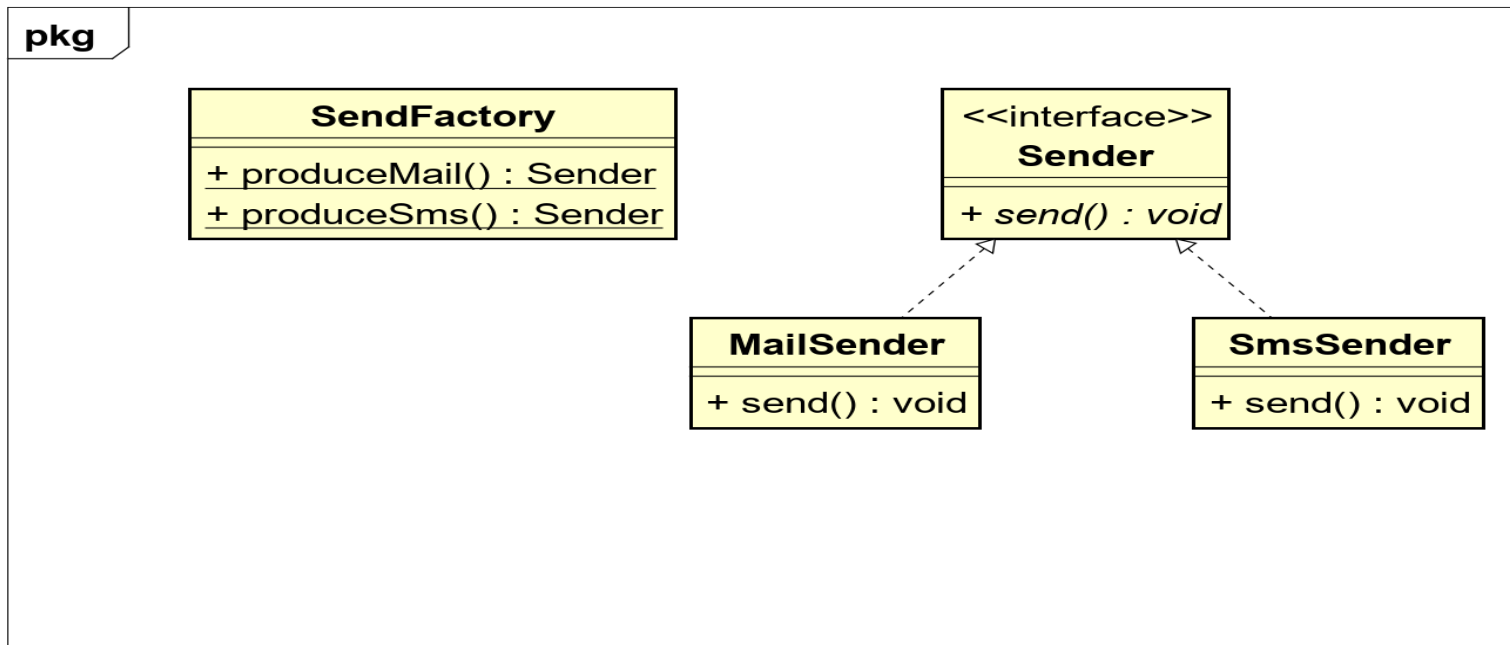
多个工厂方法模式

- 多个工厂方法模式的类图如下：



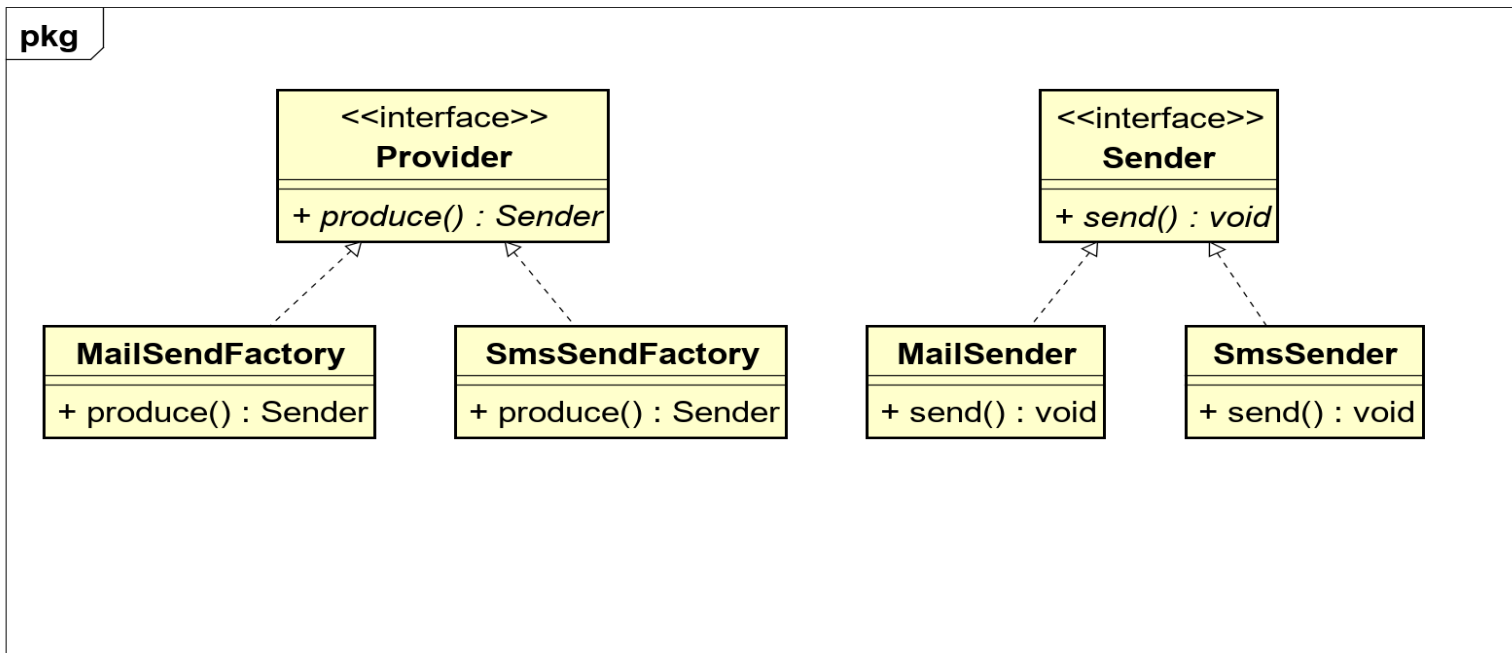
静态工厂方法模式

- 静态工厂方法模式的类图如下：



抽象工厂方法模式

- 抽象工厂方法模式的类图如下：



常用的查找算法

线性查找算法

- 使用目标元素与样本数列中的第一个元素起依次比较；
- 若找到与目标元素相等的元素，则表示查找成功；
- 若目标元素与所有样本元素比较完毕也不相等，则表示查找失败；

折半查找算法

- 假定样本数列中的元素是从小到大依次排序的；
- 使用目标元素与样本数列中的中间元素进行比较；
- 若目标元素与中间元素相等，则表示查找成功；
- 若目标元素小于中间元素，则去中间元素的左边进行查找；
- 若目标元素大于中间元素，则去中间元素的右边进行查找；
- 直到目标元素与所有该比较的元素比较完毕后也不相等，则表示查找失败；

常用的排序算法

冒泡排序算法

- 比较相邻位置的两个元素，若第一个元素比第二个元素大则交换；
- 从开始的第一对元素一直到结尾的最后一对元素，经过这一轮找到了最大值并放在了最后；
- 持续对越来越少的元素进行两两比较，直到所有元素不再发生交换为止；

总结与答疑

技术资料





IT兄弟连
ITXDL.CN

变态严管 让学习成为一种习惯