设计模式

概念

单设计模式可以说是大多数开发人员实际中使用的最多的，常见的Spring默认创建的bean就是单例模式的

单例设计模式有很多好处比如可以节约系统内存空间 控制资源的使用

其中单例设计最重要的就是确保对象只有一个

简单来说保证一个类在内存中的对象就一个

RunTime就是最典型的单例设计，我们通过对RunTime类进行分析一窥究竟

基础模式

委托模式

接口模式

代理模式

[抽象工厂模式](https://baike.baidu.com/item/%E6%8A%BD%E8%B1%A1%E5%B7%A5%E5%8E%82%E6%A8%A1%E5%BC%8F)(Abstract Factory) ,提供一个创建一系列相关或相互依赖对象的接口，而无需指定它们具体的类。

生成器模式 (Builder)，将一个复杂对象的构建与它的表示分离，使得同样的构建过程可以创建不同的表示。

[工厂方法模式](https://baike.baidu.com/item/%E5%B7%A5%E5%8E%82%E6%96%B9%E6%B3%95%E6%A8%A1%E5%BC%8F)(Factory Method) ，定义一个用于创建对象的接口，让子类决定将哪一个类实例化。Factory Method使一个类的实例化延迟到其子类。

原型模式 (Prototype) ，用原型实例指定创建对象的种类，并且通过拷贝这个原型来创建新的对象。

[单例模式](https://baike.baidu.com/item/%E5%8D%95%E4%BE%8B%E6%A8%A1%E5%BC%8F)(Singleton)，保证一个类仅有一个实例，并提供一个访问它的全局访问点。

适配器模式 (Adapter) ，将一个类的接口转换成客户希望的另外一个接口。Adapter模式使得原本由于接口不兼容而不能一起工作的那些类可以一起工作。

[桥接模式](https://baike.baidu.com/item/%E6%A1%A5%E6%8E%A5%E6%A8%A1%E5%BC%8F)(Bridge) ，将抽象部分与它的实现部分分离，使它们都可以独立地变化。

[组合模式](https://baike.baidu.com/item/%E7%BB%84%E5%90%88%E6%A8%A1%E5%BC%8F)(Composite) ，将对象组合成树形结构以表示“部分-整体”的层次结构。它使得客户对单个对象和复合对象的使用具有一致性。

容器模式

修饰模式 (Decorator) ，动态地给一个对象添加一些额外的职责。就扩展功能而言， 它比生成子类方式更为灵活。

扩展性模式

[外观模式](https://baike.baidu.com/item/%E5%A4%96%E8%A7%82%E6%A8%A1%E5%BC%8F)

享元模式

管道与过滤器模式

[代理模式](https://baike.baidu.com/item/%E4%BB%A3%E7%90%86%E6%A8%A1%E5%BC%8F)(Proxy) ，为其他对象提供一个代理以控制对这个对象的访问。

行为模式

[责任链模式](https://baike.baidu.com/item/%E8%B4%A3%E4%BB%BB%E9%93%BE%E6%A8%A1%E5%BC%8F/5295498) (Chain of Responsibility) ，为解除请求的发送者和接收者之间耦合，而使多个对象都有机会处理这个请求。将这些对象连成一条链，并沿着这条链传递该请求，直到有一个对象处理它。

命令模式 (Command) ，将一个请求封装为一个对象，从而使你可用不同的请求对客户进行参数化；对请求排队或记录请求日志，以及支持可取消的操作。

柯里化模式

事件监听器模式

[解释器](https://baike.baidu.com/item/%E8%A7%A3%E9%87%8A%E5%99%A8)模式

迭代器模式

中介者模式

备忘录模式 (Memento) ，在不破坏封装性的前提下，捕获一个对象的内部状态，并在该对象之外保存这个状态。这样以后就可将该对象恢复到保存的状态。

[观察者模式](https://baike.baidu.com/item/%E8%A7%82%E5%AF%9F%E8%80%85%E6%A8%A1%E5%BC%8F)(Observer) ，定义对象间的一种一对多的依赖关系,以便当一个对象的状态发生改变时,所有依赖于它的对象都得到通知并自动刷新。

状态模式 (State) ，允许一个对象在其内部状态改变时改变它的行为。对象看起来似乎修改了它所属的类。

策略模式 (Strategy) ，定义一系列的算法,把它们一个个封装起来, 并且使它们可相互替换。本模式使得算法的变化可独立于使用它的客户。

模板方法模式

访问者模式 (Visitor)，表示一个作用于某对象结构中的各元素的操作。它使你可以在不改变各元素的类的前提下定义作用于这些元素的新操作。

层次访问者模式

并发模式

模式 Action at a distance

模式 Balking

模式 Guarded suspension

模式 Scheduler

模式 Read write lock

模式 Double checked locking

模式 Disable job requests while running job

实时模式

模式 Scheduled task

模式 User interface

模式 Disable job requests while running job

其他

模型—视图—控制器模式

最常见的就是单例模式

源码刨析