设计模式面试题

一、编程中的六大设计原则?

1.单一职责原则

通俗地讲就是一个类只做一件事

- `CALayer`：动画和视图的显示。
- `UIView`：只负责事件传递、事件响应。

2.开闭原则

对修改关闭，对扩展开放。
要考虑到后续的扩展性，而不是在原有的基础上来回修改

3.接口隔离原则

使用多个专门的协议、而不是一个庞大臃肿的协议

- `UITableviewDelegate`
- `UITableViewDataSource`

4.依赖倒置原则

抽象不应该依赖于具体实现、具体实现可以依赖于抽象。
调用接口感觉不到内部是如何操作的

5.里氏替换原则

父类可以被子类无缝替换，且原有的功能不受任何影响

例如 KVO

6.迪米特法则

一个对象应当对其他对象尽可能少的了解，实现高聚合、低耦合

二、如何设计一个图片缓存框架？

可以模仿 SDWebImage 来实现。

构成

- Manager
- 内存缓存
- 磁盘缓存
- 网络下载
- Code Manager
    - 图片解码
    - 图片解压缩

图片的存储是以图片的单向 hash 值为 Key

内存设计需要考虑的问题

存储的 Size

因为内存的空间有限，我们针对不同尺寸的图片，给出不同的方案

- 10K 以下的 50 个
- 100Kb 以下的 20 个
- 100kb 以上的 10 个

淘汰的策略

内存的淘汰策略 采取 LRU（最近最少使用算法）

触发淘汰策略的时机有三种

- 1.定期检查（不建议，耗性能）
- 2.提高检查触发频率（一定要注意开销）
    - 1.前后台切换的时候
    - 2.每次读写的时候

磁盘设计需要考虑的问题

- 存储方式
- 大小限制（有固定的大小）
- 移除策略（可以设置为 7 天或者 15 天）

网络设计需要考虑的问题

- 图片请求的最大并发量
- 请求超时策略
- 请求优先级

图片解码

应用 策略模式，针对 jpg、png、gif 等不同的图片格式进行解码

图片解码的时机

- 在 子线程 图片刚下载完时
- 在 子线程 刚从磁盘读取完时

避免在主线程解压缩、解码，避免卡顿

三、如何设计一个时长统计框架？

记录器

- 页面式记录器
- 流式记录器
- 自定义式

记录管理者

- 内存记录缓存
- 磁盘存储
- 上传器

如何降低数据的丢失率？

- 定期写入磁盘
- 每当达到某个值的时候，就写入磁盘

记录上传的时机

- 前后台切换的时候可以上传
- 从无网到有网切换的时候可以上传

上传时机的选择

- 立即上传
- 定时上传
- 延时上传