

1. 一个T类型的值可以调用\*T类型声明的方法，当且仅当T是**可寻址的**。(怎么理解可寻址)

2. 三色标记法若不被STW保护可能会导致对象丢失

- 白色对象被黑色对象引用
- 灰色对象对白色对象的引用丢失（为什么需要这个条件）
- 导致对象丢失

3. 逃逸分析

- **堆有时需要加锁**：堆上的内存，有时需要加锁防止多线程冲突

延伸知识点：为什么堆上的内存有时需要加锁？而不是一直需要加锁呢？

(堆具体时如何分配的，由谁持有)

- `mcache`：线程缓存
- `mcentral`：中央缓存
- `mheap`：堆内存
- 线程缓存 `mcache` 被逻辑处理器 `p` 持有

4. 编译器通过逃逸分析技术去选择堆或者栈

- 检查变量的生命周期是否是完全可知的（生命周期不可知的情况有哪些？指针逃逸？）
- 如果可知，则在栈上分配
- 如果不可知，则认为内存逃逸，必须在堆上分配

5. 原子性

`mutex`和原子锁混用导致`mutex`失效

6. 互斥锁

解锁后会发出信号量通知阻塞的协程（若有多个协程阻塞，如何保证只有一个协程被唤醒；若存在饥饿模式如何保证处于饥饿模式的协程优先获得锁）