数据结构
struct kitty
    dna: u128 //DNA
    id: u128 //猫的id
    owner: Address //猫的主人
    selling: bool //是否正在出售
    price: u256 //出售的价格，只在出售时有效
    parent: Address[] //小猫的父母，是个数组

存储
1. kitties: kittens[] //小猫的数组
2. ownership: address => kittens[] //地址对所拥有的小猫的数组的mapping

可调用的函数:
1. create(owner: Address)
2. breed(parentA: Kitty, parentB: Kitty, owner: Address)
3. transfer(kitty: Kitty, from: Address, to: Address)
4. sell(kitty: Kitty, price: u256)
5. buy(kitty: Kitty, buyer: Address)
6. getKittiesByOwner(owner: Address)
7. getAllKitties()

生出来的小猫的DNA生成算法
parent1的DNA的前40位 + parent2的DNA的的前40位 + 随机的48位

解释如何在链上实现（伪）随机数
1. 使用之前的出块的hash值当做随机数 - 矿工有可能作弊
2. 引入随机数预言机来作为随机数 - 更复杂
3. 如果场景里面有对手方，可以多方都出一部分，组成一个随机数 - 场景受限