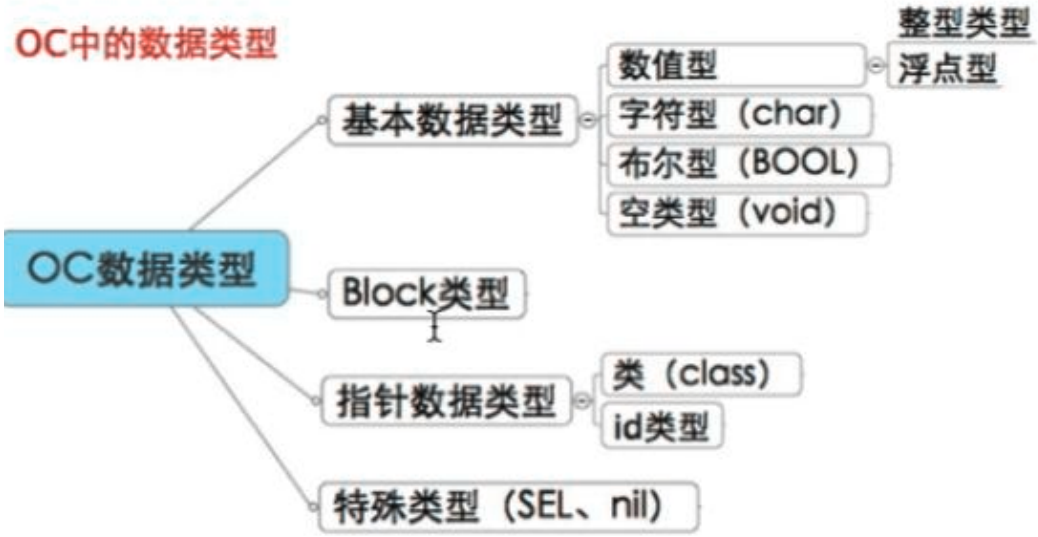
1.#import 可以防止文件重复包含
#include 必须跟预处理指令搭配使用才可防止　条件编译 宏命令
@class 仅告诉文件有这个类　具体类的内容并不知道
2.OC 中基本数据类型



3.面向对象跟面向过程
面向过程强调的时功能行为,关注的时解决问题需要哪些步骤.
面向对象则对功能进行封装,关注的时解决问题需要哪些对象,至于对象怎么做用什么方法实现,都不关注.
4.类和对象
一些事物具有相同的特征和行为,那么这些事物可以称为一类.
对象呢 则是这些类的具体的个体.
5. new
new 在内存中做了三件事
1) 在堆区开辟一块新的空间
2) 初始化实例变量的值
3) 返回一个指针地址
6.#pragma mark
对代码进行分组,方便我们查找.
7.函数跟对象方法的区别
对象只能写在 interface end 中间,实现只能写在 implementation end 中间,并且以-开头,只能有对象调用.

函数都是平行的,它不存在隶属关系,可以写在文件任意位置,使用之前声明就可以,不能访问对象中成员变量.

8.对象和方法之间的关系

对象做为方法的参数，作为方法的返回值.

9.多文件开发

把声明和实现分为两个文件,有更好的可读性,大大提高了开发效率,提高代码的维护性.

10.类方法

以+开头的方法 用在我们不需要使用成员变量的时候,不需要开辟新的空间

优点:节省内存,提高效率,代码更简洁. 缺点:不能访问成员变量

11.alloc init

分开写吧内存的分配跟初始化分开,更加灵活,我们可以人已初始化成员变量的值,而不像 new 每次初始化为 0.

12.封装

上面是:封装就是把对象的属性是实现细节隐藏起来,只提供操作数据或方法的接口.

如何封装:定义相关的方法,读取或写入到成员变量中, 并且成员变量的操作必须通过方法来完成.

好处是:隐藏数据,隔离变化,保护数据安全,提高重用性

不封装缺点:对象失去了对自己成员变量的管理权.

13.对象与对象之间的关系

组合模式 若干个类型相同的对象组合到一起

依赖关系 A 对象作为 B 对象的方法形参 B 依赖于 A

关联关系 一个对象拥有另外一个对象 也就是 B 作为 A 的实例变量时候

14.继承和派生

继承:子类获得父类特性的概念就叫做继承

派生:父类向下产生自雷的过程叫派生,

派生类有基类的属性和方法 还拥有自己新增的方法和属性.

方法重写:把父类的方法在子类中实现了.

注意:基类中的私有变量方法可以被继承,但不能被使用.

15.多态

概念:不同的对象以自己的方式响应父类的同名方法,也就是对同一种行为,不同对象有不同的表现形式.　　代码体现　父类的指针指向了子类的对象.

使用条件:继承　　同名方法　方法的重写
实用注意;有多态　父类可以访问子类中的特有方法.
优点:简化变成接口,容许在类和类之间用一些相同的命名.

16.类对象

类的本质也是一个对象.(类对象)
类对象在程序运行时一直存在.
类对象是一种数据结构存储累的基本信息.
每一个对象都包含一个指向类对象的 isa 指针
类对象只能使用类方法不能使用实例方法.

获取类对象:通过实例对象获取,通过类名获取.

17.SEL

全称 selector　是一种用来表示方法名称的数据类型.吧方法包装称 SEL 类型.
作用:可以定义变量　　作为方法的形参　　作为方法的实参

18.点语法

本质上展开就相当于　set 和 get 方法.实质是方法调用而不是访问成员变量.

19. @property　age

1)在.h 文件中生成 age 的　set 和 get 方法的声明
2)在.m 文件中生成_age 实例变量(前提是在.h 没有这个实例变量)
3)在.mz 中生成 age 的 set 和 get 方法的实现.

20.动态类型

1)动态类型　就是程序在运行时才能确定的类型　　比如　多态, id
2)id　类型是万能指针,可以指向任何继承自　NSObject　对象　　只能指向对象类型,不能指向非对象类型.

21.NSobject 和 id 区别

都可以指向任何对象
NSObject　编译需要强制转换,id 不需要强制转换可以直接使用

22.instancetype 和 id

都可以作为方法的返回值
instancetype 可以返回方法所在类相同类型的对象.id 只能返回位置类型对象
instancetype　只能做为返回值. id 可以作为参数.

23.动态类型检测

1)对象和类之间:isKindOfClass　　判断对象是不是这个类或者这个子类的实例对象
　　isMemberOfClass　　判断对象是否是这个类的实例
2)类和类之间　isSubclassOfClass　　判断 A 类是否是 B 类的子类

24.构造方法

用来初始化对象实例变量值的方法.是一个对象方法.

重写构造方法目的是 让对象创建之后就有一个默认值
注意:1)先对从父类继承而来的成员变量进行初始化
2)原则 先初始化父类的,在初始化子类的
自定义构造方法:在对象被创建的时候,对里面的某个成员变量进行初始化设置..

24.内存管理
1)移动设备内存有限,APP 内存占有也有限,我们需要回收一些不再使用的内存空间,防止系统崩溃.
主要是对内存中的堆区进行管理. 管理的时继承 NSObject 的对象.基本数据类型不需要管理.
管理原则:引入计数器 retainCount retain new copy alloc 计数器+1,release,autorelease 计数器-1.当计数器为 0 对象被回收.
当对象被回收时,会执行 dealloc 方法
2)内存管理原则
谁创建 谁 release
谁 tetain 谁 release
4) 研究内容
野指针 指针变量指向空间没有初始化 指向空间已经被释放
内存泄露 栈区内存已经释放,堆区的内存没有释放,就被泄露

25. @property 参数(MRC)
1)原子性 atomic noatomic 加锁 不加锁
2)读写性 readwrite rearonly 读写 只读
3)set 方法 assign retain copy
在一个类中有其他对象时候 用 tetain ,非对象 用 assign

26. @class
告诉编译器 这是一个类
作用:提高编译效率,防止重复包含

27. autorelease
自动释放池.在池子被释放时候,自动对池子中每个对象发出一次 release 消息

28.ARC
property 参数 strong __weak
原理 代码在编译时候在合适位置插入 release 和 retain
判断准则:有一个强指针指向,对象就存在. 没有强指针指向,对象被释放(赋值为 nil)
解决循环引入 一端用 strong 一端用 weak
总结:不允许是用 retain release 允许重写 dealloc 但不许使用[super dealloc]

29.分类
对现有的类进行扩展. 在不修改原有类的基础上增加新的方法(不能添加成员变量)

30.非正式协议
是一个类别,NSObject 或者 Foudation 框架中子类类别 都是非正式协议

31.延展
扩展,是一个匿名的分类,为这个类扩展私有方法和私有变量.他没有自己的实现文件
特点 可以在延展中定义实例变量.

32.协议
就是一些方法的声明. 定义一个规范,双方必须遵守这个规范
使用: 1)定义一个协议
2) 让一个类遵守这个协议
3) 在类的 m 文件中实现这个协议的方法

当一个类遵守这个协议的时候,就拥有这个协议里所有的方法

注意:在协议中不能声明成员变量

只要父类遵守这个协议,那么其子类页遵守这个协议

OC 中一个类可以遵守多个协议,一个协议也可以遵守多个协议

协议中关键字: @require 修饰必须遵守,默认

@optional 可以遵守,也可以不遵守

33.代理模式

传入的对象代替当前类完成某些功能,就成为代理模式

思路:1)定义一个协议,里面声明代理类需要实现的方法列表

2)创建一个代理类,遵守上面协议

3)在需要代理的类中,定义一个对象类型 id 且遵守协议的成员变量_delegate

4)在需要代理类中调用类的方法

5).m 文件中为需要代理类赋值

34.Foundation 框架

是由许多的类,方法,函数,文档按一定的逻辑组合起来的集合,方便我们使用

35.字符串中常用

比较	compare
判断是否相等	isEqualToString
检测后缀	hasprefix
前缀	hassuffix
查找	rangeOfString

36.NSRange

是 Foundation 框架中常用的结构体,用来表示一个事物的范围

有两个成员 location length

37.NSURL

表示统一资源定位符, 也表示这个资源的地址.路径

38.可变和不可变

不可变:指的是字符串在内存中占用空间内容,长度是固定的.不能被修改

可变: 空间不固定,可以被改变

39.NSArry

OC 中的数组类,下标有序.只能存放任意 OC 对象类型,不能存放非对象类型.并且是不可变的