OC_2

初始化方法

把公共的方法和实例变量写在父类里,子类只需要写自己独有的实例变量和方法即可。

继承

特点

没有父类的类称为根类,OC中的根类是NSObjec(祖宗)。

继承的上层:父类,继承的下层:子子类。

继承的内容:所有实例变量和方方法。

继承是单向的,不能相互继承。

继承具有传递性:A继承于B,B继承于C,A具有B和C的特征和行行为。

如果子子类不满意父父类方方法的实现,可以重写(oeie)父父类的方方法。

super self

super

编译器指令,不是对象

给super发信息,可以执行父类该方法的实现

self

系统关键字

self在方法中指代当前方法的调用者

self 在实例方法中指代调用当前方法的对象

```
- (void)objectMethod
{
}
```

self 在类方法中,指代当前类

```
+ (void)classMethod
{
}
```

初始化方法

初始化

开辟空间

为某些实例变量赋初值

只能使用一次(一个对象的初始化阶段只有一次)

初始化方法

```
- (id)init {
    //给super发送init消息：执行父类中实现的init方法
    self = [super init];
    //判断从父类继承过来的init方法是否初始化成功
    if (self) {
        //初始化设置
    }
    //返回初始化完成的对象
    return self;
}
```

子类定义了除父类中公共实例变量之外的实例变量。

在自身的初始化方法中，优先向super发送init消息，初始化公共变量，初始化成功之后，再初始化自身特有变量，从而完成全部实例变量的初始化。

特征

初始化方法是”-“方法

id或者instancetype类型的返回值

以init开头

可以带0到多个参数

内部实现

先执行super的初始化方法,再初始化自身变量,后return self

过程

1. 自己的初始化方法中,优先调用父类的初始化方方法。
2. 父类的初始化方法中再调用父类的初始化方法,依次往上调用。
3. 处于最上层的初始化完成之后,回到第二层的初始化方法中, 完 成第二层的初始化。
4. 第二层的初始化完成之后,回到第三层的初始化方法中,依次执 行初始化方法,直到本类的初始化方法完成。

指定初始化方法

一个类可以有多个初始化方法

指定初始化方法

- 这些初始化方法都有相同的部分，如何优化代码?

```
- (id)initWithName:(NSString *)name {
    self = [super init];
    if (self) {
        _name = name;
    }
    return self;
}
```

```
- (id)initWithGender:(NSString *)gender {
    self = [super init];
    if (self) {
        _gender = gender;
    }
    return self;
}
```

```
- (id)initWithName:(NSString *)name gender:(NSString *)gender {
    self = [super init];
    if (self) {
        _name = name;
        _gender = gender;
    }
    return self;
}
```

指定初始化方法

遍历构造器

封装了对象创建过程

特点

是”+“方法

返回 本类型的实例

方法名以类名开头

可以有0到多个参数

实现-使用

- 声明

```
+ (id)personWithName:(NSString *)name gender:(NSString *)gender;
```

- 实现

```
+ (id)personWithName:(NSString *)name gender:(NSString *)gender {
    return [[Person alloc] initWithName:name gender:gender];
}
```

- 调用

```
Person *per = [Person personWithName:@"Frank" gender:@"男"];
```

总结

. 继承是面向对象三大特性之一,合理的继承,能减少很多沉余代码,加快开发速度.

. 初始化方法以init开头,在对象的生命周期中只使用一次.

遍历构造器封装了对象的创建过程,进一步简化了对象创建的步骤.