OC基础三

属性、点语法

今日作业

- 为 上节课作业的15个类添加 属性

课上任务

- 为 Student类添加属性

课程目标

- 区别属性和实例变量
- 熟练使用属性和点语法
- 了解封装（面向对象三大特性之一）

实例变量的访问

- @public 直接使用“->”
- @private @protected 都需要分别给出setter和getter。
-

实例变量的访问

- @public 直接使用“->”
- @private @protected 都需要分别给出setter和getter。
- 如果类有100个实例变量呢?

属性

- 一个类100个实例变量，我们的解决方式：写100个设置器和100个访问器。
- OC2.0中定义了属性（property）的概念，属性是一组设置器和访问器，它简化了上述操作。
- 与类相似，属性需要声明和实现。

属性的声明

- 属性的声明写在类的.h文件中
  ```
  @property int age;
  ```

- 相当于声明了两个方法:

- ```
  - (void)setAge:(int)age;
  ```

- ```
  - (int)age;
  ```

属性的实现

- 属性的实现写在类的.m文件中
- `@synthesize age = _age;`
- 相当于实现了两个方法:
- `- (void)setAge:(int)age`
- `- (int)age`

属性的实现

- 属性的实现写在类的.m文件中
- `@synthesize age = _age;`
- 相当于实现了两个方法:
- `- (void)setAge:(int)age`
- `- (int)age`

代码演示

练习

- 为 Teacher类添加属性

PROPERTY的ATTRIBUTE

- 属性也可以设置特性（attribute）主要包括三个方面：
- 读写特性
- 原子性特性
- setter语义特性

读写属性

- 读写属性可以通过下面关键字进行设置:
- readwrite 可读写（既有设置器也有访问器）默认
- readonly 只读（只有访问器没有设置器）
- getter = 方法名 指定访问器的方法名
- setter = 方法名 指定设置器的方法名

原子性属性

- 原子性属性可以通过下面关键字设置：
- nonatomic 非原子性，不保证多线程安全
- atomic 原子性，多线程访问时较安全

语义属性

- 语义属性可以通过一下关键字设置:
- assign 直接赋值 适用于基本数据类型
- retain 赋值时做内存优化 适用于对象类型
- copy 复制一个副本 适用于特殊的对象类型

语义属性

- 语义属性可以通过一下关键字设置:
- assign 直接赋值 适用于基本数据类型
- retain 赋值时做内存优化 适用于对象类型
- copy 复制一个副本 适用于特殊的对象类型

代码演示

点语法

- 一旦我们有了实例变量的设置器和访问器，就可以使用OC的点语法。

- 比如： p.age = 20;等价于
- [p setAge:20];
- int age = p.age;等价于
- int age = [p age];

小结

- 属性和实例变量是不同的，前者是一组方法，后者是一个变量

- 定义属性 @property 和@synthesize

- 基本数据类型使用assign，对象类型使用retain，实现了NSCopying协议的对象使用copy。

- 点语法和[receiver message]是等价的。

封装

- 封装：隐藏内部实现、提供接口调用。
- 属性 封装了实例变量
- 方法 封装了具体实现代码
- 类 封装了属性和方法

封装

- 封装：隐藏内部实现、提供接口调用。
- 属性 封装了实例变量
- 方法 封装了具体实现代码
- 类 封装了属性和方法

```
NSLog(@“%@”,@”Hello 蓝鸥!”);
//系统提供了控制台输出函数，我们不知道怎样实现
//但可以轻松使用
```

封装的具体实现

```
@interface Person : NSObject
{
    int _age;
    NSString *_name;
}
@property int age;
@end
@implementation Person
- (void)setAge:(int)age
{
    if(age<0){age = 18;}
    _age = age;
}
..
@end
```

```
 Person *p = [[Person alloc]
init];
 [p setAge:-10];
 NSLog(@"%d",[p age]);
```

输出结果为18

封装的具体实现

封装的具体实现

- 封装的好处:
- 使用更加简单
- 变量更加安全
- 可以隐藏内部实现细节
- 开发速度加快

总结

- 如果只有访问器，能使用点语法吗？只有设置器呢?

术语和技巧

- @synthesize age = _age; 作用是setAge:以及age方法都是在操作实例变量_age，没有_age的话，编译器会帮你生成_age

- @synthesize age;作用为setAge:以及Age方法都是在操作实例变量age，没有age的话，编译器会帮你生成age

说明

- 本节重点：掌握属性和点语法的使用
- 区别实例变量和属性
- 下节课讲：iOS内存管理（very important）
- 知识就像一张网，不可能一天掌握全部东西，每天积累一点点，不久，网就清晰了，不要纠结细节

今日作业

- 为 上节课作业的15个类添加 属性