

Objective-C中私有成员变量与私有方法笔录

创建 Student 类,其 .h 中代码如下:

```
#import <Foundation/Foundation.h>
@interface Student : NSObject
{
    NSString * _name;
    NSUInteger _age;
}

//提供getter与setter方法
-(void)setName:(NSString *)name;
-(void)setAge:(NSUInteger)age;
-(NSString *)name;
-(NSUInteger)age;
@end
```

.m 文件中代码:

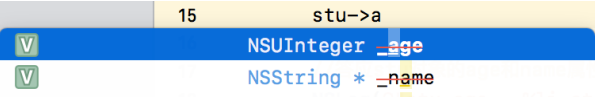
```
#import "Student.h"
@implementation Student
- (void)setName:(NSString *)name{
    _name = name;
}
- (void)setAge:(NSUInteger)age{
    _age = age;
}
- (NSString *)name{
    return _name;
}
- (NSUInteger)age{
    return _age;
}
@end
```

首次测试代码如下:

```
Student * stu = [[Student alloc] init];
stu.age = 10;
stu.name = @"min";

//答应stu对象的age和name属性
NSLog(@"stu.age = %li stu.name = %@",stu.age,stu.name);
```

测试图示:

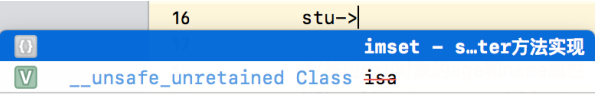


测试结果:

```
2018-06-05 10:49:50.704322+0800 测试属性[2657:68320] stu.age = 10 stu.name = min
Program ended with exit code: 0
```

第一次测试结论: 从测试结果中得出, 在.h文件中只要声明了成员变量, 那么在第测试代码中可以直接通过 -> 查看到有哪些成员变量。这样破坏了对象的封装性。 所以可以通过把变量私有化来封装对象的成员变量。 可以将成员变量定义在 @implementation (即 .m 文件)中并去除 @interface 中声明成员变量的部分, 此时成员变量变为 私有变量

测试图例:



从上图看出, 此时成员变量被隐藏, 在测试方中无法查看且被调用。

第二次测试:

.h 中代码:

```
#import <Foundation/Foundation.h>
@interface Student : NSObject
-(void)setAge:(NSUInteger)age;
-(NSUInteger)age;
@end
```

.m 中代码:

```
#import "Student.h"
@implementation Student
{
    NSString * _name;
    NSUInteger _age;
}
-(void) setAge:(NSUInteger) age{
    _age = age;
}
- (NSUInteger)age{
    return _age;
}
@end
```

测试代码:

```
Student * stu = [[Student alloc] init];
stu.age = 20;
NSLog(@"stu.age = %lu",stu.age);
```

测试结果:

```
2018-06-05 11:38:39.008546+0800 测试属性[3180:91325] stu.age = 20
Program ended with exit code: 0
```

第二次测试结论: 从测试结果中得出, 私有变量 可在本类中被实例方法访问, 通过setter和getter方法也可以直接采用点语法直接访问。

第三次测试:

.h 文件中代码:

```
#import <Foundation/Foundation.h>
@interface Student : NSObject

@end
```

.m 文件中代码:

```
#import "Student.h"
@implementation Student
- (void)method{
    NSLog(@"Student Method");
}
@end
```

测试代码图:

```
Student * stu = [[Student alloc] init];
[stu method];
```

No visible @interface for 'Student' declares the selector 'method'

第三次测试结论: 从测试结果中得出, 当在 .h 文件中未声明方法, 只在 .m 文件中实现,此时无法在测试代码中向该对象方法发送调用消息。那么该方法则变成为对象的 私有方法 。

私有方法 与 私有成员变量 都不能被外界访问且显示, 只能本类自己访问。

注: 在@implementation中的成员变量与在@interface中声明的成员变量加上@private变量修饰符仍旧有地方不同: @interface中声明的成员变量加上@private变量修饰符虽然

重点: 在Objective-C没有真正的私有方法

第四次测试:

.h 与 .m 文件中代码与第三次测试代码相同

测试代码如下:

```
Student * pp = [Student new];  
[pp performSelector:@selector(method)];
```

测试结果:

```
2018-06-05 12:02:27.472052+0800 测试属性[3533:104987] Student Method  
Program ended with exit code: 0
```

备注1: 对象可以通过选择器,调用对应的私有方法。原因: 因为oc中调用方法是消息机制,oc方法是动态绑定,只有在运行时才会去查看对象是否有该方法。