OC 委托

委托是协议的沿用，委托简单的理解就是某人委托某人去做某事，这个和java中的接口回调机制比较相似。委托在IOS开发中比较常用，比如我们不知道一个列表中的数据有多少，我们可以用委托的方式，将数据委托给其他类，让其他类去填充数据。委托的常用功能主要是传值和事件监听。

我们下面使用委托来实现2个类的委托，即老师委托学生去买本《OC开发》。

用来实现委托的协议(BuyBookDelegate.h)

@protocol BuyBookDelegate

//定义一个委托协议，协议中只有一个方法，用来买书

-(void)buyBook:(NSString\*)name;

@end

老师类

#import <Foundation/Foundation.h>

#import "BuyBookDelegate.h"

@interface Teacher : NSObject

//定义一个委托协议

@property (nonatomic,assign)id<BuyBookDelegate> buyBookDele;

//定义一个老师想让学生买书的方法

-(void)willBuyBook:(NSString\*)bookName;

@end

#import "Teacher.h"

@implementation Teacher

-(void)willBuyBook:(NSString\*)bookName{

NSLog(@"老师想买本《%@》",bookName);

[\_buyBookDele buyBook:bookName];

}

@end

学生类

#import <Foundation/Foundation.h>

#import "Teacher.h"

@interface Student : NSObject <BuyBookDelegate>

@end

#import "Student.h"

@implementation Student

-(void)buyBook:(NSString \*)name{

NSLog(@"学生去买《%@》",name);

}

@end

测试：

#import <Foundation/Foundation.h>

#import "Teacher.h"

#import "Student.h"

int main(int argc, const char \* argv[]) {

@autoreleasepool {

Teacher\* teacher = [[Teacher alloc]init];

Student\* student = [[Student alloc]init];

//为老师设置委托的对象

[teacher setBuyBookDele:student];

//老师要买书了

[teacher willBuyBook:@"OC开发"];

}

return 0;

}

结果：

 老师想买本《OC开发》

 学生去买《OC开发》