**上机一：（类实例之间的循环强引用）**

**题目：**

按照以下步骤实现解决类实例之间的循环强引用，具体如下：

1. 创建一个playground文件。
2. 定义一个Person类，并定义它的一个属性apartment，apartment的类型是Apartment。
3. 在Person类中定义一个构造函数，一个析构函数。
4. 定义一个Apartment类，并定义它的一个属性tenant，tenant的类型是Person。在tenant属性定义的最前面加上weak。
5. 在Apartment类中定义一个构造函数，一个析构函数。
6. 分别定义Person和Apartment的实例xiaoMing和unit106并初始化。
7. xiaoMing调用apartment，unit106调用tenant。（引起循环强引用）
8. 把xiaoMing和unit106设置为nil。（Person对象和Apartment对象都没有被释放）
9. 观察程序结果。

class Person {

let name: String

var apartment: Apartment?

init(name: String) {

self.name = name

print("Person构造完成：\(name)")

}

deinit {

print("Person析构完成：\(name)")

}

}

class Apartment {

let unit: String

var tenant: Person?

init(unit: String) {

self.unit = unit

print("公寓构造完成：\(unit)")

}

deinit {

print("公寓析构完成：\(unit)")

}

}

var xiaoMing: Person?

var unit106: Apartment?

xiaoMing = Person(name: "xiaoMing")

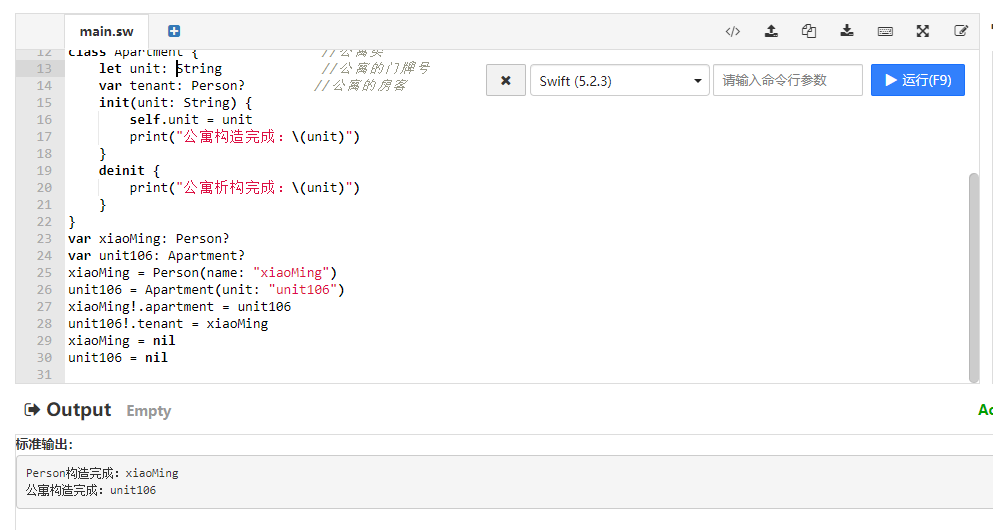
unit106 = Apartment(unit: "unit106")

xiaoMing!.apartment = unit106

unit106!.tenant = xiaoMing

xiaoMing = nil

unit106 = nil



**上机二：（使用弱引用或无主引用解决类实例之间的循环强引用）**

**题目：**

按照以下步骤实现解决类实例之间的循环强引用，具体如下：

1. 创建一个playground文件。
2. 定义一个Person类，并定义它的一个属性apartment，apartment的类型是Apartment。
3. 在Person类中定义一个构造函数，一个析构函数。
4. 定义一个Apartment类，并定义它的一个属性tenant，tenant的类型是Person。在tenant属性定义的最前面加上weak或unowned。
5. 在Apartment类中定义一个构造函数，一个析构函数。
6. 分别定义Person的实例xiaoMing。
7. xiaoMing调用Apartment的一个实例。
8. 把xiaoMing设置为nil。
9. 观察程序结果。

class Person {

let name: String

var apartment: Apartment?

init(name: String) {

self.name = name

print("Person构造完成：\(name)")

}

deinit {

print("Person析构完成：\(name)")

}

}

class Apartment {

let unit: String

weak var tenant: Person?

init(unit: String) {

self.unit = unit

print("公寓构造完成：\(unit)")

}

deinit {

print("公寓析构完成：\(unit)")

}

}

var xiaoMing: Person?

var unit106: Apartment?

xiaoMing = Person(name: "xiaoMing")

unit106 = Apartment(unit: "unit106")

xiaoMing!.apartment = unit106

unit106!.tenant = xiaoMing

xiaoMing = nil

unit106 = nil

