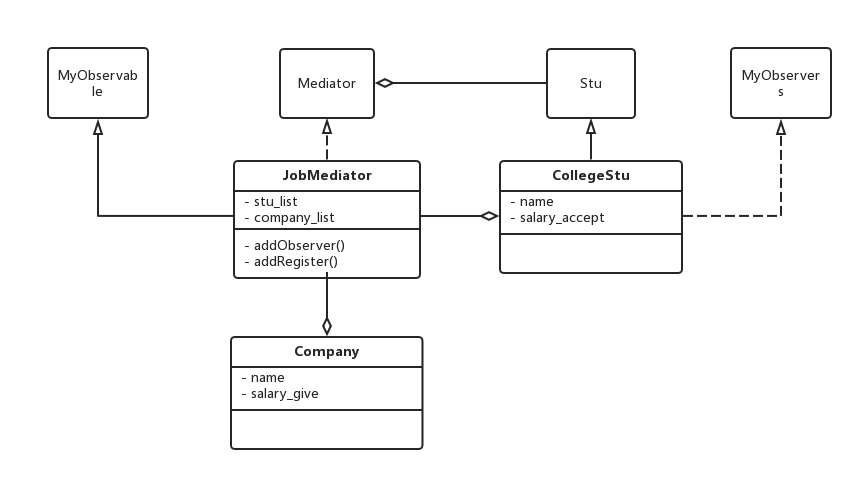
系统要求

3. 有一家JobMediator（工作信息发布网站），公司可以往上面发布招聘信息。而求职者可以根据平台上发布的信息来选择自己的有意向的公司发布信息。我们可以通过JobMediator 来实现求职者与招聘者间的通信而避免了对象间复杂的通信。另外当有新的公司注册并且达到了某些求职者的要求的时候（例如薪水，职位，工作地点），我们可以给这些求职者（相当于观察着）发送有新公司符合他求职意愿的消息。请综合考虑上述场景，帮助该公司实现一个工作信息发布系统。

UML类图



类图使用PrcessOn（https://www.processon.com）在线生成工具完成。

程序采用观察者模式进行设计。

主要包含三个类，JobMediator、CollegeStu、Company。各类之间关系见上图，上图中Mediator、Stu表示接口，MyObservers、MyObserverable表示简单类。

JobMediator类用于信息发布系统的抽象，包含两个方法，两个属性：

两个属性为：

1. stu\_list：记录在该系统发布信息的学生；
2. company\_list：记录在该系统发布信息的公司；

两个方法为：

1. addObserver()：在相应的记录中添加学生信息，学生依据设计原则充当观察者；
2. addRegister(): 在相应的记录中添加公司信息，公司依据设计原则充当被观察者；

CollegeStu类为求职学生的抽象，包含两个属性。

两个属性为：

（1） name：记录学生姓名；

（2） salary\_accept：记录学生期望的薪资；

Company类为公司的抽象，包含两个属性。

两个属性为：

（1） name：记录公司姓名；

（2） salary\_give：记录公司给出的薪资；

系统实现

该版本的系统使用C++实现。工程包含七个文件（main.cpp jobmediator.cpp jobmediator.h company.cpp company.h collegestu.cpp collegestu.h）。

其中：

1. main.cpp: 简单模拟学生和公司在系统上发布信息的过程；
2. company.cpp & company.h: 包含对公司类的实现，对公司给出薪资的设定；
3. collegestu.cpp & collegestu.h：包含对学生类的实现，对学生期望薪资的设定；
4. jobmediator.cpp & jobmediator.h：包含发布信息系统类的实现，添加在系统上发布信息的学生或公司记录，对匹配的公司或学生进行消息提醒；

工作信息发布系统实现过程：首先初始化学生，包含学生姓名与期待工资，后将该学生信息发布到发布系统；发布系统记录学生信息并进行查重操作，并将学生记录默认为观察者；随后初始化公司，包含公司姓名与给予工资，后将公司信息发布到发布系统；发布系统对公司信息进行查重后记录公司信息，随后遍历学生记录，找到薪资要求匹配的学生，对相应的学生和公司进行消息提醒。

由于时间有限，需要注意的实现细节如下：

1. 我们简化了求职者的工作要求，只考虑薪资的高低来判断意向；
2. 本版本实现过程中，我们对上述的类图关系进行了简化，没有抽象封装传统的观察者模式中的observer角色；
3. 本版本功能实现上，消息下发的触发机制在于新公司的添加；功能更加完备的系统的触发机制还应满足新学生的添加，我们期待在下一个版本中完善这一功能；
4. 模拟发布系统过程中，我们采取了简单的顺序模拟，即，顺序添加学生和公司信息；更完备的模拟测试学生和公司应满足随机到达，然而由于时间有限，我们期待在下一版本中完善这一流程；

系统运行结果

