

题目描述

本周学校举行大型春招宣讲活动，许多企业会在**本周周一至周五的8:00-18:00**在学校进行招聘宣讲会。现在你是一名学校的后台工程师，被要求为毕业生们实现一个在线预约参加宣讲会的系统。学校的要求如下：

- 宣讲会开始和结束的时间均为**整点**，并且**至少持续一小时，至多持续一整天（即10小时）**。
- 毕业生可以预约多场宣讲会，但预约一场宣讲会之后，**无法再预约与其有时间冲突的其他宣讲会**。此处，**时间冲突**的定义为有整小时的冲突时间，例如11:00-13:00和12:00-15:00有冲突，而12:00-15:00和15:00-18:00没有冲突（忽略毕业生赶到下一场宣讲会的时间，假设他会瞬移）。同时，毕业生也可以取消预约。毕业生可以在系统中为某场宣讲会打分，宣讲会的得分取所有人的**平均分**。
- 不同宣讲会有不同的开放预约时间，毕业生只能预约已开放预约的宣讲会。企业可以随时开启或关闭预约通道。

代码实现

经过分析，这些功能需要通过两个类来实现：CareerTalk类表示不同的宣讲会，ReservationTable类表示某个学生的预约信息。具体要求的代码接口与解释如下：

CareerTalk

- 你需要在CareerTalk类里实现如下接口：（为节省时间，可将整段代码复制进.cpp文件）

```
// 构造函数，初始化CareerTalk对象
CareerTalk::CareerTalk(const char *company_name, int day, int start_time,
int end_time) {
    /**
     * 说明：
     * company_name: 举办单位名称，例如："蚂蚁金服"
     * day: 取值范围[1, 5]，表示在星期几举办
     * start_time: 取值范围[8, 17]，表示几点开始（显然，不可能18:00开始）
     * end_time: 取值范围[9, 18]，表示几点结束（显然，不可能8:00结束）
     * 注意：不用考虑start_time >= end_time，取值范围不符等非法输入。后面的接口也一样不用考虑
     */
}

// 成员函数，返回宣讲会企业或机构名称
const char *CareerTalk::GetName() {}

// 成员函数，返回宣讲会举办日期
int CareerTalk::GetDay() {}

// 成员函数，返回宣讲会开始时间
int CareerTalk::GetStartTime() {}

// 成员函数，返回宣讲会结束时间
int CareerTalk::GetEndTime() {}
```

```

// 成员函数，修改宣讲会具体信息
void CareerTalk::AdjustInfo(int day, int start_time, int end_time) {
    // 说明：企业可能会在开放预约之后修改时间信息，就可能会引起冲突，冲突需要在下个类的成员
    函数中进行处理
}

// 成员函数，开放这个宣讲会的预约通道
void CareerTalk::OpenAccess() {}

// 成员函数，关闭这个宣讲会的预约通道
void CareerTalk::CloseAccess() {}

// 成员函数，返回是否已开放预约
bool CareerTalk::IsOpen() {}

// 成员函数，对该宣讲会评分
void CareerTalk::Comment(int score) {
    // 说明：score取值范围为[0, 10]，不考虑输出超过该范围的情况；毕业生随时可以对宣讲会
    评分
}

// 成员函数，返回该宣讲会的平均评分
float CareerTalk::GetScore() {
    /**
     * 说明：
     * 如果该宣讲会还没有人评分，则分数为0
     * 否则，分数为所有人评分的均分
     * 不需要考虑保留几位小数的问题
     */
}

```

- 调用示例：

```

// 初始化宣讲会信息，腾讯校招宣讲会安排在周二的9:00-11:00
CareerTalk tencentTalk("Tencent", 2, 9, 11);

// 查询信息
int day = tencentTalk.GetDay();
int start_time = tencentTalk.GetStartTime();
int end_time = tencentTalk.GetEndTime();

// 修改宣讲会信息，调整到周四11:00-13:00
tencentTalk.AdjustInfo(4, 11, 13);

// 开放预约通道
tencentTalk.OpenAccess();

// 查询预约通道是否开启
bool isOpen = tencentTalk.IsOpen();

// 给宣讲会打8分和10分
tencentTalk.Comment(8);
tencentTalk.Comment(10);

// 获取宣讲会均分（预期为9分）
float avgScore = tencentTalk.GetScore();

```

```
// 关闭预约通道
tencentTalk.CloseAccess();
```

ReservationTable

- 你需要在ReservationTable类里实现如下接口：

```
// 构造函数，初始化ReservationTable对象
ReservationTable::ReservationTable() {}

// 成员函数，预约某个CareerTalk并更新table
bool ReservationTable::AddTalk(CareerTalk *career_talk) {
    /**
     * 说明：
     * 若宣讲会开放预约并且与已预约的其他宣讲会没有时间冲突，则添加进table，并返回true。
     时间冲突的定义见PDF顶部说明
     * 否则，不添加进table，返回false，预约失败。若已预约该宣讲会，则也预约失败
     */
}

// 成员函数，取消预约
bool ReservationTable::CancelTalk(CareerTalk *career_talk) {
    /**
     * 说明：
     * 若已经预约了该宣讲会，则从table中移除，并返回true
     * 否则，返回false，取消失败
     * 与上次OJ类似，两个CareerTalk*变量的相等与否可以直接用指针相等判断
     */
}

// 成员函数，返回某天某时的具体宣讲会
CareerTalk* ReservationTable::QueryTalk(int day, int time) {
    /**
     * 说明：
     * day取值范围为[1, 5]，表示查询星期几的宣讲会
     * time取值范围为[8, 17]，表示查询[time, time + 1)时间范围内正在进行的宣讲会。例
     如，某宣讲会时间为11:00-13:00，则((time == 11) || (time == 12))情况下能查询到
     * 若该时间点没有预约的宣讲会，返回NULL
     */
}

// 成员函数，返回某天有多少预约了的宣讲会
int ReservationTable::TalksInADay(int day) {}

// 成员函数，根据某个修改了的宣讲会信息更新table
bool ReservationTable::AdjustTalk(CareerTalk *career_talk) {
    /**
     * 说明：
     * 若调整后的宣讲会时间与其他预约的宣讲会时间冲突，则自动取消预约，从table里删除，并返
     回false。时间冲突的定义见PDF顶部说明
     * 若调整后无冲突，则调整成功，返回true，此时table内部对应宣讲会的信息应更新到最新
     * 若没有预约这场宣讲会，返回false
     */
}
```

- 调用示例:

```
CareerTalk talk1 = CareerTalk("Tencent QQ", 1, 8, 10);
CareerTalk talk2 = CareerTalk("Alibaba", 2, 14, 16);
CareerTalk talk3 = CareerTalk("Baidu", 3, 12, 15);
talk1.OpenAccess();
talk2.OpenAccess();

// 初始化对象
ReservationTable table;

// 预约宣讲会，根据开放预约与否会有不同的结果
bool addState1 = table.AddTalk(&talk1);
bool addState2 = table.AddTalk(&talk2);
bool addState3 = table.AddTalk(&talk3);

// 查询周一9:00-10:00的宣讲会信息
CareerTalk *queryTalk = table.QueryTalk(1, 9);

// 查询周一预约了几场宣讲会
int talkNum = table.TalksInADay(1);

// 调整宣讲会时间，如果冲突则自动取消预约被改动的宣讲会，不冲突则将table中的宣讲会信息更新
talk1.AdjustInfo(4, 15, 17);
bool adjustState = table.AdjustTalk(&talk1);
talk1.AdjustInfo(2, 15, 17);
adjustState = table.AdjustTalk(&talk1);

// 取消预约
bool cancelState = table.CancelTalk(&talk2);
```

注意

- 请正确处理头文件和实现文件之间的关系，文件、函数的命名严格按照给定要求，注意大小写。
- 将4个文件(CareerTalk.cpp, CareerTalk.h, ReservationTable.cpp, ReservationTable.h)打包成ZIP压缩包上传（ZIP包中不要包含文件夹或者其他文件）。
- 请不要在你提交的代码中包含main函数。