

Explanation to A5

Provider: SA- YeeTone WANG

Lab Teacher: Yueming ZHU

Lecture Teacher: Yepang LIU

[Notice]

1. 请基于Structure框架进行本次作业的完成。本次作业5题相互关联，错综复杂，使用Structure框架可以避免OJ和LJ上的编译错误！
2. OJ上已经部署了Structure框架下的CourseOperator.java和CourseTime.java，因此提交时对这两个类做出的任何修改都将视为无效！

Problems

1. 设计Building类，以及一些基本的枚举类和抽象类(10 points);
2. 设计Classroom类(20 points);
3. 设计Course类(20 points);
4. 设计Teacher类，要求继承于抽象类Person，并实现CourseOperator接口(25 points);
5. 设计Student类，要求继承于抽象类Person，并实现CourseOperator接口(25 points);

Problem1: Building

类对象内部成员：

```
private List<Classroom> rooms;  
private Location location;  
private int id;
```

1. 构造方法

```
public Building(){}  
public Building(Location location , int id){}
```

【要求解读】

1. 任一构造方法都需要初始化rooms的list对象；
2. 参数列表需要用于初始化内部成员对象；

【题解提示】

List接口的实现类有ArrayList, LinkedList等等。

2. getter和setter方法

```
public Location getLocation(){}  
public void setLocation(Location location){}  
public int getId(){}  
public void setId(int number){}
```

```
public List<Classroom> getRooms(){}  
public void setRooms(List<Classroom> rooms){}
```

【要求解读】

无

【题解提示】

无

3. addRoom和deleteRoom方法

```
public boolean addRoom(Classroom room){}  
public boolean deleteRoom(Classroom room){}
```

【要求解读】

操作增加和删除之前，需要检查传入的Classroom对象是否在当前building中。如果是同一Building，那么就执行相应的增加和删除。

哦对了，删除还要检查当前building对象里面是否存在传入的room。

【题解提示】

1. 比较是否是同一building可以用==，也可以用equals方法；
2. ArrayList中add方法和remove方法返回的结果都是boolean类型。其中add方法返回一定是true，remove方法如果找到了就移除并返回true，否则返回false；

4. 重写toString方法

```
public String toString(){} 
```

【要求解读】

无

【题解提示】

按照题目意思来即可。

5. 基本枚举类和抽象类

枚举类型：Location, CourseType, Day

抽象类型：Person

【要求解读】

无

【题解提示】

按照Structure的模板来即可。

Problem2: Classroom

类对象内部成员：

```
int id;//eg:101  
int seatNum;//eg:50  
CourseType type;// Lecture or Lab  
Building building;  
Map<CourseTime, Course> schedule;
```

1. 构造方法与getter,setter方法

```
public Classroom(){}  
public Classroom(int id , int seatNum , Building building, CourseType type){}
```

【要求解读】

任一构造方法都需要初始化schedule。对于有参的构造方法，分别实例化每个所属的成员对象。

【题解提示】

1. Map接口的实现子类有HashMap等等；
2. getter和setter方法最好都写上；

2. 重写toString方法

```
public String toString(){}  

```

【要求解读】

无

【题解提示】

按照题目要求书写即可。

3. addCourse方法

```
public String addCourse(Course course){}
```

【要求解读】

进行一系列检查，询问最终的返回结果字符串。

【题解提示】

1. 时间段有无安排课程：containsKey方法；
2. Type是否相同：拿出Type做equals判断或者==判断；
3. seat是否足够：seat与course的capacity做比较；
4. 添加课程：put方法；

4. deleteCourse方法

```
public boolean deleteCourse(Course course){}
```

【要求解读】

检查当前schedule内有没有指定课程，有则进行移除操作并返回true；否则返回false。

【题解提示】

Map类的remove(key, value)方法中可以判断(value, key)键值对是否存在并进行移除操作。存在则移除并返回true，否则返回false。

5. getCourse方法

```
public Course getCourse(CourseTime courseTime){}
```

【要求解读】

根据传入的CourseTime返回对应的course，如果不存在则返回null。

【题解提示】

Map类的get方法可以返回key值对应的value值，如果不存在key值则返回null。

6. printSchedule方法

```
public String printSchedule(){} 
```

【要求解读】

无

【题解提示】

按照题目通过StringBuilder类要求书写即可。

7. getScheduleCourseNum方法

```
public int getScheduleCourseNum(){} 
```

【要求解读】

无

【题解提示】

Map类的size方法可以返回当前Map当中存放的键值对个数。

Problem3: Course

类对象内部成员：

```
static int idCnt = 0; // number of courses created
int id; // generated automatically from 1
String name; // Introduction to Computer Programming A
String abbrevName; // JavaA
String code; // CS102A
CourseTime time;
Teacher teacher;
Classroom room;
List<Student> students; // who selected this course
int capacity; // maxium number of students
CourseType type; // Lecture, Lab
```

1. 构造方法

```
public Course( String code , String name , String abbrevName , Teacher teacher ,
int capacity , CourseType type){}
public Course( String code , String name , String abbrevName , Teacher teacher ,
int capacity , CourseType type, CourseTime time , Classroom room){}
```

【要求解读】

根据参数列表初始化成员对象。其中如果参数不带CourseTime和Classroom, 则无需进行初始化。

【题解提示】

同文档上的notice.

2. getter和setter方法

```
public String getAbbrevName() { }
public void setAbbrevName(String abbrevName) { }
public void setAbbrevName() { }
public CourseTime getTime() { }
public void setTime(CourseTime time) { }
public Teacher getTeacher() { }
public void setTeacher(Teacher teacher) { }
public Classroom getRoom() { }
public void setRoomTime(Classroom room, CourseTime time) { }
public int getCapacity() { }
public void setCapacity(int capacity) { }
public void setType(CourseType type) { }
public CourseType getType() { }
public List<Student> getStudents() { }
```

【要求解读】

getAbbrevName()方法中，如果当前对象的abbrevname是null或者是空字符串，那么就通过操作name来获取abbrevname。

【题解提示】

1. 判定字符串是否为空有isEmpty方法；
2. 操作方法：先将abbrevname以空格split开，然后逐一遍历检查首字母，如果首字母是大写则加入StringBuilder。最终结果即为StringBuilder中的内容；

3. setRoomTime方法

```
public void setRoomTime(Classroom room, CourseTime time){ }
```

【要求解读】

无

【题解提示】

无

4. addStudent方法

```
public boolean addStudent (Student student){}
```

【要求解读】

如果当前list中已有传入的student对象，则添加失败返回false；否则直接添加然后返回true。

【题解提示】

List类中的contains方法可以快速判断有无当前对象。

5. deleteStudent方法

```
public boolean deleteStudent (Student student){}
```

【要求解读】

如果当前list中已有传入的student对象，则移除失败返回false；否则直接移除然后返回true。

【题解提示】

List类中的remove方法可以判断是否存在对象；如果存在则移除元素并返回true，否则返回false。

Problem4: Teacher

要求：继承Person抽象类，实现CourseOperator接口。
类内额外成员：

```
private Location preferLocation;
```

并书写对应的getter和setter方法：

```
public Location getPreferLocation() {}  
public void setPreferLocation(Location preferLocation){}
```

【题目解读】

这个属性反映了老师喜欢上课的地方。如果老师在两个地方都可以上课的话，那么就优先考虑preferLocation；除非老师在preferLocation没有地方上课，那么才考虑其他Location。

1. 构造方法

```
public Teacher(){ }  
public Teacher(String id, String name){ }
```

【要求解读】

根据参数列表初始化成员对象，如果参数列表不足则不进行初始化。

【题解提示】

可以通过super关键字调用Person类的构造方法。

2. getFreeClassroom方法

```
public List<Classroom> getFreeClassroom(CourseTime time, int capacity, CourseType type){}
```

【要求解读】

从Db.buildings中搜索可用教室房间，优先考虑preferLocation。

【题解提示】

两次两重循环，分别遍历Db.buildings和building.getRooms。

检查条件如下：

1. type相同；
2. seatNum不小于Capacity；
3. 教室的schedule不能存在key值为time的键值对；
4. Location要相同；
第一次二重循环遍历检查preferLocation；结束后如果结果仍然为empty，那么就进行第二次二重循环。

3. createCourse方法

```
public boolean createCourse(Course course ) {}  
public boolean createCourse(String code , String name , String abbrevName,  
CourseTime time , Classroom room , int capacity , CourseType type ) {}
```

【要求解读】

教师创建课程。注意考虑各种可能的非法情况，以及各处可能要进行的更新操作。

【题解提示】

1. 第二个方法可以调用第一个方法从而使得代码简洁，反过来调用则不可以，会导致course的idCnt错误；
2. 第一个方法的判定规则如下：
 - 2.1 course对应的courseTime或者classroom不能是null；
 - 2.2 teacher的schedule里面不能有course对应的courseTime的key值；
 - 2.3 course的type和course里的room的type要保持一致；
 - 2.4 course对应的classroom的schedule里面不能有course对应的courseTime的key值；
 - 2.5 course的capacity不能超过course对应的classroom的seatNum；如果以上规则都满足，那么就需要在schedule和classroom中都添加对应的课程，然后return true。

4. dropCourse方法

```
public boolean dropCourse( Course course ){} 
```

【要求解读】

教师移除课程。注意考虑各种可能的非法情况，以及各处可能要进行的更新操作。

【题解提示】

1. 如果教师schedule的key值中不包含course对应的courseTime，直接return false；
2. 根据course对应的courseTime找到对应的course，然后需要在course对应的classroom和teacher的schedule中移除对应的time键值对，最后return true即可。

5. changeCourse方法

```
public boolean changeCourse(Course oldCourse1 , Course newCourse2){ }
```

【要求解读】

老师要换课。

【题解提示】

简单。只需要先dropCourse，如果drop成功了就再createCourse。如果都能成功就return true，其他情况都为false。

6. printSchedule方法

```
public String printSchedule(){} 
```

【要求解读】

无

【题解提示】

1. 无论有没有课，每行开头的数字后面都需要增加1个空格；
2. 其他的按照题目意思来书写即可。

7. getScheduleCourseNum方法

```
public int getScheduleCourseNum(){} 
```

【要求解读】

无

【题解提示】

Map类的size方法可以返回当前Map当中存放的键值对个数。

Problem5: Student

要求：继承Person抽象类，实现CourseOperator接口。

无额外类成员对象。

1. 构造方法

```
public Student(){}  
public Student( String id , String name ) {}
```

【要求解读】

根据参数列表初始化成员对象，如果参数列表不足则不进行初始化。

【题解提示】

可以通过super关键字调用Person类的构造方法。

2. courseExist方法

```
public boolean courseExist(String code , String name , CourseType type){}  
public boolean courseExist(Course course){}
```

【要求解读】

查找schedule中是否存在符合要求的course。

【题解提示】

1. 第二个方法可以调用第一个方法从而使得代码简洁，反过来调用则不可以，会导致course的idCnt错误；
2. 第一个方法先通过values方法获取所有值构成的集合，然后循环遍历进行逐一比较检查即可。

3. chooseCourse方法

```
public boolean chooseCourse( Course course){}
```

【要求解读】

学生要选课。注意各种可能的非法情况。

【题解提示】

方法判定规则如下：

1. 课程不能选满了；
 2. 学生的schedule不能有course对应的courseTime的键值对；
 3. 学生的schedule不能有name, type和code与传入的schedule一致的course；
- 以上规则判定完成如果都符合的话，那么就在学生的schedule中增加键值对，以及course中也添加对应的学生即可。

4. dropCourse方法


```
public boolean dropCourse( Course course){}
```

【要求解读】

学生要退课。注意各种可能的非法情况。

【题解提示】

1. 首先检查schedule的values集中有无对应的course与之相同；
2. 如果有相同的，那么根据courseTime移除对应的键值对，然后再course的students列表中移除当前对象(this)，最后返回true即可；
3. 其他情况都返回false。

5. changeCourse方法

```
public boolean changeCourse(Course oldCourse1 , Course newCourse2){}
```

【要求解读】

学生要换课。注意各种可能的非法情况。

【题解提示】 该方法书写不同于teacher，需要做如下检查：

1. 传入的newCourse2不能是null；
2. newCourse2的学生数量不能超过newCourse2的capacity；
如果上述两个条件满足，那么只需要做以下操作，然后return true即可；
3. 在schedule中移除oldCourse1的courseTime所在键值对；
4. 在schedule中增加newCourse2的courseTime的对应键值对；
5. oldCourse1的学生列表中移除当前对象(this)；
6. newCourse2的学生列表中增加当前对象(this)；

6. printSchedule方法

```
public String printSchedule(){}
```

【要求解读】

无

【题解提示】

1. 题目中描述出现问题，后续的Classroom.toString需要删除；
2. 无论有没有课，每行开头的数字后面都需要增加1个空格；
3. 其他的按照题目意思来书写即可。

7. getScheduleCourseNum方法

```
public int getScheduleCourseNum(){}
```

【要求解读】

无

【题解提示】

Map类的size方法可以返回当前Map当中存放的键值对个数。