实现一个简单的电信计费程序：

假设南昌市电信分公司针对市内座机用户采用的计费方式：

月租20元，接电话免费，市内拨打电话0.1元/分钟，省内长途0.3元/分钟，国内长途拨打0.6元/分钟。不足一分钟按一分钟计。

假设本市的区号：0791，江西省内各地市区号包括：0790~0799以及0701。

输入：

输入信息包括两种类型

1、逐行输入南昌市用户开户的信息，每行一个用户，含手机和座机用户

格式：u-号码 计费类型 （计费类型包括：0-座机 1-手机实时计费 2-手机A套餐）

例如：u-079186300001 0

座机号码由区号和电话号码拼接而成，电话号码包含7-8位数字。

本题只考虑计费类型0-座机计费，系列2、3会逐步增加计费类型。

2、逐行输入本月某些用户的通讯信息，通讯信息格式：

座机呼叫座机：t-主叫号码 接听号码 起始时间 结束时间

t-079186330022 058686330022 2022.1.3 10:00:25 2022.1.3 10:05:11

以上四项内容之间以一个英文空格分隔，

时间必须符合"yyyy.MM.dd HH:mm:ss"格式。提示：使用SimpleDateFormat类。

以上两类信息，先输入所有开户信息，再输入所有通讯信息，最后一行以"end"结束。

注意：

本题非法输入只做格式非法的判断，不做内容是否合理的判断（时间除外，否则无法计算），比如：

1、输入的所有通讯信息均认为是同一个月的通讯信息，不做日期是否在同一个月还是多个月的判定，直接将通讯费用累加，因此月租只计算一次。

2、记录中如果同一电话号码的多条通话记录时间出现重合，这种情况也不做判断，直接计算每条记录的费用并累加。

3、用户区号不为南昌市的区号也作为正常用户处理。

输出：

根据输入的详细通讯信息，计算所有<u>已开户的用户</u>的当月费用（精确到小数点后2位，单位元）。假设每个用户初始余额是100元。

每条通讯信息单独计费后累加，不是将所有时间累计后统一计费。

格式：号码+英文空格符+总的话费+英文空格符+余额

每个用户一行，用户之间按号码字符从小到大排序。

错误处理：

输入数据中出现的不符合格式要求的行一律忽略。

建议类图：

参见图1、2、3，可根据理解自行调整：

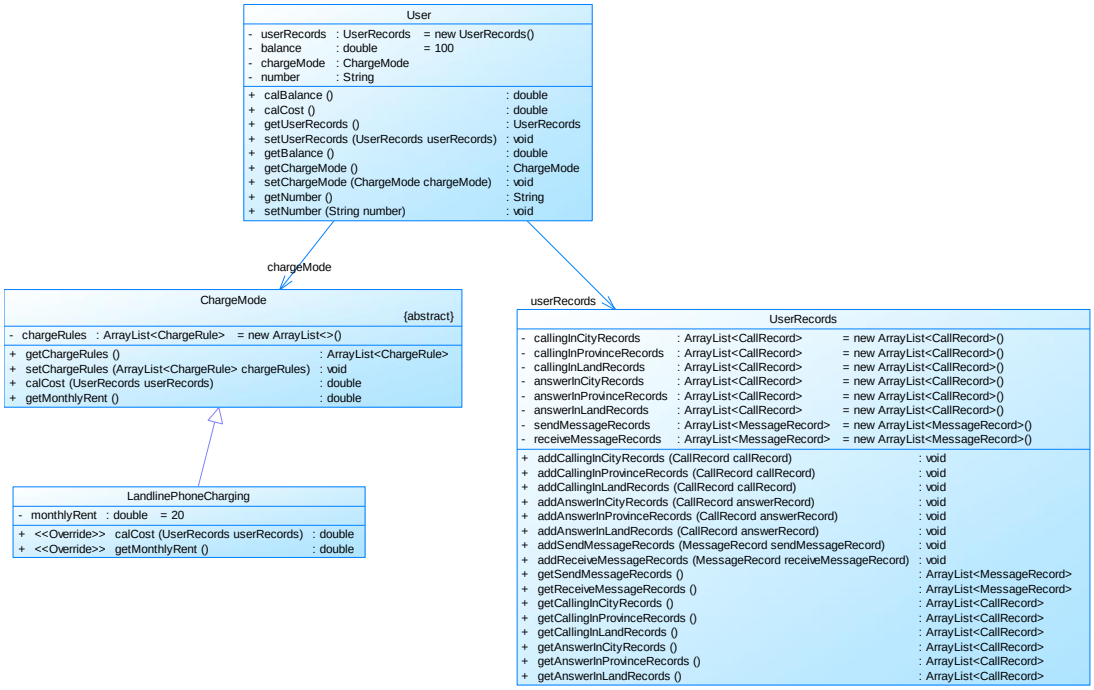

图1

图1中User是用户类，包括属性：

userRecords （用户记录）、balance（余额）、chargeMode（计费方式）、number（号码）。

ChargeMode是计费方式的抽象类：

chargeRules是计费方式所包含的各种计费规则的集合，ChargeRule类的定义见图3。

getMonthlyRent（）方法用于返回月租（monthlyRent）。

UserRecords是用户记录类，保存用户各种通话、短信的记录，

各种计费规则将使用其中的部分或者全部记录。

其属性从上到下依次是：

市内拨打电话、省内（不含市内）拨打电话、省外拨打电话、

市内接听电话、省内（不含市内）接听电话、省外接听电话的记录

以及发送短信、接收短信的记录。

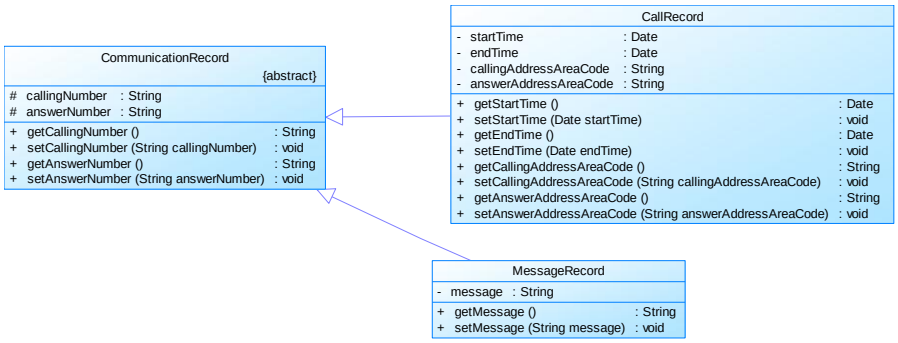

图2

图2中CommunicationRecord是抽象的通讯记录类：

包含callingNumber拨打号码、answerNumber接听号码两个属性。

CallRecord（通话记录）、MessageRecord（短信记录）是它的子类。

CallRecord（通话记录类）包含属性：

通话的起始、结束时间以及　拨号地点的区号（callingAddressAreaCode）、接听地点的区号（answerAddressAreaCode）。

区号用于记录在哪个地点拨打和接听的电话。对于座机，就是本机区号。如果是手机号，则接打地点的区号和本机的开户地区会有差异。

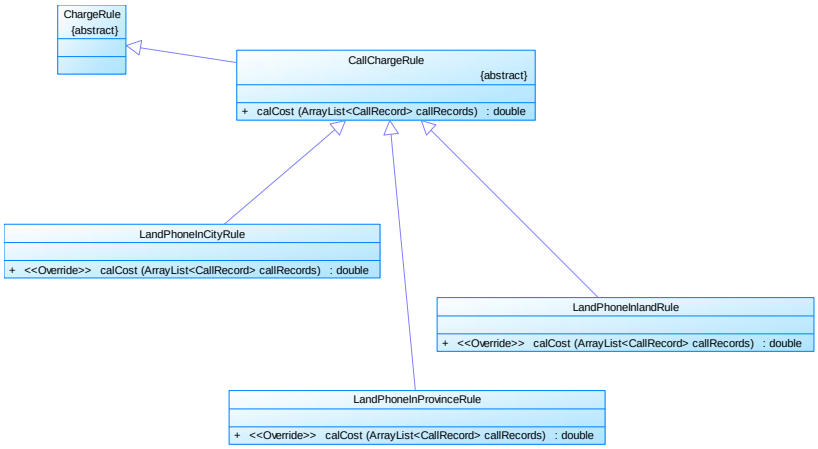

图3

图3是计费规则的相关类，这些类的核心方法是：

calCost（ArrayList<CallRecord> callRecords）。

该方法针根据输入参数callRecords中的所有记录计算某用户的某一项费用；如市话费。

输入参数callRecords的约束条件：必须是某一个用户的符合计费规则要求的所有记录。

LandPhoneInCityRule、LandPhoneInProvinceRule、LandPhoneInLandRule三个类分别是座机拨打市内、省内、省外电话的计费规则类，用于实现这三种情况的费用计算。

（提示：可以从UserRecords类中获取各种类型的callRecords）。

后续扩展说明：

后续题目集将增加手机用户，手机用户的计费方式中除了与座机计费类似的主叫通话费之外，还包含市外接听电话的漫游费以及发短信的费用。在本题的设计时可统一考虑。

部分参考测试用例：

以下内容绿色部分为输入，紧接其后的黑色部分为输出。

```
u-079186300001 0

t-079186300001 058686330022 2022.1.3 10:00:25 2022.1.3 10:05:25

end

079186300001 3.0 77.0
```

```
u-079186300001 0

t-079186300001 058686330022 2022.1.3 10:00:25 2022.1.3 10:05:26

end

079186300001 3.6 76.4
```

```
u-079186300001 0
u-079186300002 0
t-079186300001 058686330022 2022.1.3 10:00:25 2022.1.3 10:05:25
t-079186300002 058686330022 2022.1.3 11:00:25 2022.1.3 11:05:25
t-079186300001 079186300001 2022.1.3 12:00:25 2022.1.3 12:05:25
end
079186300001 3.5 76.5
079186300002 3.0 77.0
```

```
u-079186300002 0
u-079186300001 0
t-079186300001 058686330022 2022.1.3 10:00:25 2022.1.3 10:05:25
t-079186300002 058686330022 2022.1.3 11:00:25 2022.1.3 11:05:25
t-079186300001 079186300001 2022.1.3 12:00:25 2022.1.3 12:05:25
end
079186300001 3.5 76.5
079186300002 3.0 77.0
```

```
u-079186300002 0
u-079186300001 0
t-079186300001 079286300002 2022.1.3 10:00:25 2022.1.3 10:05:25
end
079186300001 1.5 78.5
079186300002 0.0 80.0
```

```
u-079186300002 0
u-079186300001 0
t-079186300001 079286300002 2022.1.3 10:00:25 2022.1.3 10:05:25
t-079186300001 079186300003 2022.1.3 10:00:25 2022.1.3 10:05:25
end
079186300001 2.0 78.0
079186300002 0.0 80.0
```

```
u-079186300002 0
u-079186300001 0
```

```
t-079186300001 079186300002 2022.1.3 22:00:25 2022.1.3 22:05:32
t-079186300001 079186300003 2022.1.3 10:00:25 2022.1.3 10:05:25
end
079186300001 1.1 78.9
079186300002 0.0 80.0


u-079186300001 0
t-079186300001 079286330022 2022.1.3 12:50:00 2022.1.3 13:05:00
22
rr

end
079186300001 4.5 75.5
```