

coodex billing

一个可复用的计费模型框架

主要对象

- Chargeable, 可计费对象, 主要跟业务对应, 由其决定计费的领域, 例如: 一次KTV消费, 一顿饭、一次住宿、一次罚款、一次停车等等
- Bill, 一笔账单, 包含计费明细
- Calculator, 费用计算器, 根据可计费对象计算Bill
- Revision, 计费时对计费的调整项目, 例如: 优惠券, 其他消费抵扣, 包月等等
- Adjustment, 对账单的调整项目, 例如: 账单比例优惠, 抹零, 附加费等等

按量计费

太简单, 不多说。

时序计费

coodex-billing为时序计费提供了一个简单框架, 主要对象包括:

- Period, 时间段, 包含开始时间, 结束时间
- TimeBasedChargeable, 时序可计费对象, 包含一个起止时间段, 模型编码, 模型参数id
- TimeBasedBill, 包含时间段信息的账单
- TimeBasedDetail, 包含时间段信息的账单明细
- BillingModel, 时序计费的模型, 也就是对应指定领域中一种具体的、完整的计费规则
- BillingModel.Instance, 计费模型的一个实例, 例如, 某连锁宾馆, 不同店计费模式相同, 但是单价不同, 每个店的计费标准就是一个模型实例
- BillingModel.Algorithm, 某一个时段的计费法则
- BillingModel.Fragment, 计费段。时序计费时, 可能会把一个连续时长按照标准切分成多个时段, 逐一计算。时序段包含一个计费法则和一个时间段
- TimeBasedRevision, 时序的调整项目
 - WholeTimeRevision, 切分前调整
 - FragmentRevision, 基于特定计费段的调整

coodex-billing也提供了一个时序计费模型的参考实现, 方便开发者使用, 参考实现的约定包括:

- AbstractAlgorithm
 - 增加了是否允许不连续时段的属性, 例如, 一个计费段是[11:00 - 23:00], 消费的时段为[12:00 - 20:00], 使用一张[14:00 - 15:00]抵扣的优惠券, 那么此计费段内需要计费的时段变成了[12:00 - 14:00]和[15:00 - 20:00], 该属性为真时, 计费时长为7小时, 否则, 会分两次计算, 可根据计费标准指定
- AbstractBillingModel, 计费模型的参考实现, 开发者只需要指定具体的TimeBasedChargeable记忆模型的编码即可
- ModelProfileFactory, 模型参数仓库, 开发者自行实现, 例如, 不同模型的变量结构是不一样的, 可以为每个模型见一个数据库表, 然后根据id来提取数据

- **FragmentSlicer**，计费段切片器，**coodex-billing**提供了3种开箱即用的切片器，也可以通过 **SlicerProfile/SlicerFactory**自行扩展
 - 按自然日切片(**FxiedDateSlicerProfile**, **FxiedDateSlicerFactory**)，可以指定从每天的何时开始进行切片，默认从0点开始
 - 固定小时数切片(**FixedHoursSlicerProfile**, **FixedHoursSlicerFactory**)，可以指定多少小时一段，不得小于1
 - 时段切片(**FragmentSlicerProfile**, **FragmentSlicerFactory**)，可以指定从何时到何时进行切片，例如，00:00 - 04:00。当开始时间描述大于结束时间描述时，说明结束时间为次日，例如，20:00 - 08:00，表示当日20:00到次日的08:00
- 计费规则体系，使用计费参数(**AlgorithmProfile**)及计费实例工厂(**AlgorithmFactory**)自行实现

实践示例

我们以一个按时计费的KTV为案例实践一遍，由浅入深，一点点扩展，最终达到**月月鸟**在某KTV中包消费一段时长，并额外消费了XX，使用多优惠券、VIP、包时段来出结算单的场景

话说，**月月鸟**童鞋忙完了一个项目，拿到一笔丰厚的奖金，准备犒劳犒劳自己，在大街上走着走着就走进了之前常来的一家KTV，进门以后就看见收费标准如下：

时间段	大包	中包	小包
02:00 - 08:00	228元/小时	128元/小时	88元/小时
08:00 - 18:00	118元/小时	68元/小时	48元/小时
18:00 - 次日02:00	448元/小时	248元/小时	168元/小时

* 加入VIP/VVIP享受更多优惠，详情请咨询前台

月月鸟童鞋心想，上次的小包腾挪不开，这次有钱了，来个中包吧。

三天以后，**月月鸟**满意的来到前台结算，OK，我们开始。

计费对象

```
package org.coodex.billing.demo;

import org.coodex.billing.Revision;
import org.coodex.billing.timebased.Period;
import org.coodex.billing.timebased.TimeBasedChargeable;
import org.coodex.util.Common;

import java.text.ParseException;
import java.util.List;

import static org.coodex.billing.demo.Constants.MODEL_01;

public class KTVConsumption implements TimeBasedChargeable {

    public enum Room{
        LARGE, MIDDLE, SMALL
```

```

    }

    public static void main(String[] args) {
        KTVConsumption ktvConsumption = new KTVConsumption();
        Bill<KTVConsumption> bill = BillCalculator.calc(ktvConsumption);
        StringBuilder builder = new StringBuilder("`月月鸟`在`KTV`的消费总金额 ")
            .append(bill.getAmount()).append(" 元。明细如下: ");
        for (Bill.Detail detail : bill.getDetails()) {
            builder.append("\n ");
            if (detail instanceof TimeBasedDetail) {
                TimeBasedDetail timeBasedDetail = (TimeBasedDetail) detail;
                builder.append("[")

.append(Common.calendarToStr(timeBasedDetail.getPeriod().getStart()))
                .append(" - ")

.append(Common.calendarToStr(timeBasedDetail.getPeriod().getEnd()))
                .append("]");
            }
            builder.append(detail.item()).append(" 金额: ")
.append(detail.getAmount()).append("元");
            if (detail.usedRevision() != null) {
                builder.append(" ").append(detail.usedRevision().getName());
            }
        }
        System.out.println(builder.toString());
    }

    @Override
    public Period getPeriod() {
        try {
            return Period.BUILDER.create(
                Common.strToCalendar("2019-09-18 16:47:30",
Common.DEFAULT_DATETIME_FORMAT),
                Common.strToCalendar("2019-09-21 20:50:00",
Common.DEFAULT_DATETIME_FORMAT)
            );
        } catch (ParseException e) {
            throw new RuntimeException(e.getLocalizedMessage(), e);
        }
    }

    @Override
    public String getModel() {
        return MODEL_01;
    }

    @Override
    public String getModelParam() {
        return MODEL_01 + "_01";
    }

    @Override
    public List<Revision> getRevisions() {

```

```

        // 暂不使用优惠
        return null;
    }

    public Room getRoomType(){
        return Room.MIDDLE;
    }
}

```

说明一下，此处我们没有通过外部设置业务对象，实际业务场景中，需要由业务逻辑根据数据构建可计费对象

KTVConsumption领域下的账单计算器

```

package org.coodex.billing.demo;

import org.coodex.billing.timebased.AbstractTimeBasedCalculator;

import java.util.concurrent.TimeUnit;

public class KTVBillCalculator extends AbstractTimeBasedCalculator<KTVConsumption>
{
    @Override
    protected TimeUnit getTimeUnit() {
        return TimeUnit.MINUTES;
    }

    @Override
    public boolean accept(KTVConsumption param) {
        return param != null;
    }
}

```

放入到SPI:META-INF/services/org.coodex.billing.Calculator

```
org.coodex.billing.demo.KTVBillCalculator
```

MODEL_01计费模型

```

package org.coodex.billing.demo;

import org.coodex.billing.timebased.reference.AbstractBillingModel;

import static org.coodex.billing.demo.Constants.MODEL_01;

public class Model01 extends AbstractBillingModel<KTVConsumption> {

```

```
@Override
protected String getModelCode() {
    return MODEL_01;
}
}
```

放入到SPI:META-INF/services/org.coodex.billing.timebased.BillingModel

```
org.coodex.billing.demo.Model01
```

模型参数工厂

```
package org.coodex.billing.demo;

import org.coodex.billing.timebased.reference.AlgorithmProfile;
import org.coodex.billing.timebased.reference.FragmentProfile;
import org.coodex.billing.timebased.reference.ModelProfile;
import org.coodex.billing.timebased.reference.ModelProfileFactory;

import java.util.Arrays;
import java.util.List;

import static org.coodex.billing.demo.Constants.MODEL_01;

public class Model01ProfileFactory implements ModelProfileFactory {
    @Override
    public ModelProfile build(String s) {
        return new ModelProfile() {
            @Override
            public AlgorithmProfile getWholeTimeAlgorithmProfile() {
                return null;
            }

            @Override
            public List<FragmentProfile> getFragmentProfiles() {
                // 一会增加
                return Arrays.asList();
            }
        };
    }

    @Override
    public boolean accept(String param) {
        return param != null && param.equals(MODEL_01);
    }
}
```

放入到SPI:META-INF/services/org.coodex.billing.timebased.reference.ModelProfileFactory

```
org.coodex.billing.demo.Model01ProfileFactory
```

好了，截止到目前位置，针对KTV消费领域的模型框架搭好了，跑一下

```
`月月鸟`在`KTV`的消费总金额 0 元。明细如下：  
[2019-09-18 16:47:30 - 2019-09-21 20:50:00]no algorithm found. 金额: 0元
```

下面开始完成月月鸟看到的计费规则

计费法则

- 计费法则的变量

```
package org.coodex.billing.demo;  
  
import org.coodex.billing.timebased.reference.AlgorithmProfile;  
  
public class PerHourAlgorithmProfile implements AlgorithmProfile {  
    private int priceLargeRoom;  
    private int priceMiddleRoom;  
    private int priceSmallRoom;  
  
    public PerHourAlgorithmProfile(int priceLargeRoom, int priceMiddleRoom, int  
priceSmallRoom) {  
        this.priceLargeRoom = priceLargeRoom;  
        this.priceMiddleRoom = priceMiddleRoom;  
        this.priceSmallRoom = priceSmallRoom;  
    }  
  
    public int getPriceLargeRoom() {  
        return priceLargeRoom;  
    }  
  
    public int getPriceMiddleRoom() {  
        return priceMiddleRoom;  
    }  
  
    public int getPriceSmallRoom() {  
        return priceSmallRoom;  
    }  
}
```

- 计费法则工厂

```

package org.coodex.billing.demo;

import org.coodex.billing.Bill;
import org.coodex.billing.timebased.AbstractTimeBasedCalculator;
import org.coodex.billing.timebased.BillingModel;
import org.coodex.billing.timebased.Period;
import org.coodex.billing.timebased.reference.AbstractAlgorithm;
import org.coodex.billing.timebased.reference.AlgorithmFactory;

public class PerHourAlgorithmFactory implements AlgorithmFactory<KTVConsumption,
PerHourAlgorithmProfile> {
    @Override
    public BillingModel.Algorithm<KTVConsumption> build(final
PerHourAlgorithmProfile perHourAlgorithmProfile) {
        return new AbstractAlgorithm<KTVConsumption>() {

            private int getPricePerHour(KTVConsumption ktvConsumption) {
                switch (ktvConsumption.getRoomType()) {
                    case LARGE:
                        return perHourAlgorithmProfile.getPriceLargeRoom();
                    case MIDDLE:
                        return perHourAlgorithmProfile.getPriceMiddleRoom();
                    default:
                        return perHourAlgorithmProfile.getPriceSmallRoom();
                }
            }

            private long getPrice(KTVConsumption ktvConsumption, long duration) {
                int remainder = (int) (duration % 60);
                int quotient = (int) (duration / 60);
                int pricePerHour = getPricePerHour(ktvConsumption);
                return quotient * pricePerHour + (remainder > 0 ? pricePerHour :
0);
            }

            private String getItemName(KTVConsumption ktvConsumption) {
                switch (ktvConsumption.getRoomType()) {
                    case LARGE:
                        return "大包包间费";
                    case MIDDLE:
                        return "中包包间费";
                    default:
                        return "小包包间费";
                }
            }

            @Override
            protected Bill.Detail calc(Period period, long duration,
KTVConsumption chargeable) {
                return new AbstractTimeBasedCalculator.TimeBasedDetailImpl(
                    period,
                    getPrice(chargeable, duration),
                    getItemName(chargeable) + " 消费 " + duration + " 分钟"
                );
            }
        };
    }
}

```

```

    );
    }
    };
}

@Override
public boolean accept(PerHourAlgorithmProfile param) {
    return param != null;
}
}

```

放到 `SPI:META-INF/services/org.coodex.billing.timebased.reference.AlgorithmFactory`

```
org.coodex.billing.demo.PerHourAlgorithmFactory
```

ok, 跑一把看看

‘月月鸟’在‘KTV’的消费总金额 11176 元。明细如下：

[2019-09-18 16:47:30 - 2019-09-18 18:00:00]	中包包间费	消费 72 分钟	金额: 136元
[2019-09-18 18:00:00 - 2019-09-19 02:00:00]	中包包间费	消费 480 分钟	金额: 1984元
[2019-09-19 02:00:00 - 2019-09-19 08:00:00]	中包包间费	消费 360 分钟	金额: 768元
[2019-09-19 08:00:00 - 2019-09-19 18:00:00]	中包包间费	消费 600 分钟	金额: 680元
[2019-09-19 18:00:00 - 2019-09-20 02:00:00]	中包包间费	消费 480 分钟	金额: 1984元
[2019-09-20 02:00:00 - 2019-09-20 08:00:00]	中包包间费	消费 360 分钟	金额: 768元
[2019-09-20 08:00:00 - 2019-09-20 18:00:00]	中包包间费	消费 600 分钟	金额: 680元
[2019-09-20 18:00:00 - 2019-09-21 02:00:00]	中包包间费	消费 480 分钟	金额: 1984元
[2019-09-21 02:00:00 - 2019-09-21 08:00:00]	中包包间费	消费 360 分钟	金额: 768元
[2019-09-21 08:00:00 - 2019-09-21 18:00:00]	中包包间费	消费 600 分钟	金额: 680元
[2019-09-21 18:00:00 - 2019-09-21 20:50:00]	中包包间费	消费 170 分钟	金额: 744元

月月鸟一看，嚯，1万多，完了完了完了，回家没法交待了，还好，我有张6折卡，还有上次消费返还的2小时的抵扣时长和1500的代金券，再算算。

优惠

可计费对象里，增加调整项：

```

@Override
public List<Revision> getRevisions() {
    return Arrays.asList(
        new DurationRevision("2小时抵扣", 120),
        new TimeBasedOffAdjustment<KTVConsumption>(0.4f, "6折房间费"),
        new AmountAdjustment(1500, "1500元代金券")
    );
}

```


好的，再看看

月月鸟在KTV的消费总金额 5124 元。明细如下：

```
[2019-09-18 16:47:30 - 2019-09-18 18:47:30]2小时抵扣 金额：0元 2小时抵扣
[2019-09-18 18:47:30 - 2019-09-19 02:00:00]中包包间费 消费 432 分钟 金额：1984元
[2019-09-19 02:00:00 - 2019-09-19 08:00:00]中包包间费 消费 360 分钟 金额：768元
[2019-09-19 08:00:00 - 2019-09-19 18:00:00]中包包间费 消费 600 分钟 金额：680元
[2019-09-19 18:00:00 - 2019-09-20 02:00:00]中包包间费 消费 480 分钟 金额：1984元
[2019-09-20 02:00:00 - 2019-09-20 08:00:00]中包包间费 消费 360 分钟 金额：768元
[2019-09-20 08:00:00 - 2019-09-20 18:00:00]中包包间费 消费 600 分钟 金额：680元
[2019-09-20 18:00:00 - 2019-09-21 02:00:00]中包包间费 消费 480 分钟 金额：1984元
[2019-09-21 02:00:00 - 2019-09-21 08:00:00]中包包间费 消费 360 分钟 金额：768元
[2019-09-21 08:00:00 - 2019-09-21 18:00:00]中包包间费 消费 600 分钟 金额：680元
[2019-09-21 18:00:00 - 2019-09-21 20:50:00]中包包间费 消费 170 分钟 金额：744元
6折房间费 金额：-4416元 6折房间费
1500元代金券 金额：-1500元 1500元代金券
```

月月鸟说：你把时长从后面减掉我看看，好像从后面减合适点

店员：成吧，看在你是熟客，一般人我可真不这样

```
@Override
public List<Revision> getRevisions() {
    return Arrays.asList(
        new DurationRevision("5小时抵扣", 120, false),
        new TimeBasedOffAdjustment<KTVConsumption>(0.4f, "6折房间费"),
        new AmountAdjustment(1500, "1500元代金券")
    );
}
```

月月鸟在KTV的消费总金额 4908 元。明细如下：

```
[2019-09-18 16:47:30 - 2019-09-18 18:00:00]中包包间费 消费 72 分钟 金额：136元
[2019-09-18 18:00:00 - 2019-09-19 02:00:00]中包包间费 消费 480 分钟 金额：1984元
[2019-09-19 02:00:00 - 2019-09-19 08:00:00]中包包间费 消费 360 分钟 金额：768元
[2019-09-19 08:00:00 - 2019-09-19 18:00:00]中包包间费 消费 600 分钟 金额：680元
[2019-09-19 18:00:00 - 2019-09-20 02:00:00]中包包间费 消费 480 分钟 金额：1984元
[2019-09-20 02:00:00 - 2019-09-20 08:00:00]中包包间费 消费 360 分钟 金额：768元
[2019-09-20 08:00:00 - 2019-09-20 18:00:00]中包包间费 消费 600 分钟 金额：680元
[2019-09-20 18:00:00 - 2019-09-21 02:00:00]中包包间费 消费 480 分钟 金额：1984元
[2019-09-21 02:00:00 - 2019-09-21 08:00:00]中包包间费 消费 360 分钟 金额：768元
[2019-09-21 08:00:00 - 2019-09-21 18:00:00]中包包间费 消费 600 分钟 金额：680元
[2019-09-21 18:00:00 - 2019-09-21 18:50:00]中包包间费 消费 50 分钟 金额：248元
[2019-09-21 18:50:00 - 2019-09-21 20:50:00]2小时抵扣 金额：0元 2小时抵扣
6折房间费 金额：-4272元 6折房间费
1500元代金券 金额：-1500元 1500元代金券
```

月月鸟：果然，从后面算划算，来吧，结账吧

店员：稍等，查房马上给结果

额外消费

店员：不好意思，您还消费了方便面10盒，XXX七个。服务员的消费您是单独结还是一起结？

```
public List<Revision> getRevisions() {
    return Arrays.asList(
        new DurationRevision("2小时抵扣", 120),
        new TimeBasedOffAdjustment<KTVConsumption>(0.4f, "6折房间费"),
        new AmountAdjustment(1500, "1500元代金券"),
        new ConsumerGoods(10,10,"方便面"),
        new ConsumerGoods(15,7,"XXX"),
        new ConsumerGoods(300,5,"XMA"),
        new ConsumerGoods(400,2,"XMB")
    );
}
```

月月鸟`在`KTV`的消费总金额 7629 元。明细如下：

```
[2019-09-18 16:47:30 - 2019-09-18 18:47:30]2小时抵扣 金额：0元 2小时抵扣
[2019-09-18 18:47:30 - 2019-09-19 02:00:00]中包包间费 消费 432 分钟 金额：1984元
[2019-09-19 02:00:00 - 2019-09-19 08:00:00]中包包间费 消费 360 分钟 金额：768元
[2019-09-19 08:00:00 - 2019-09-19 18:00:00]中包包间费 消费 600 分钟 金额：680元
[2019-09-19 18:00:00 - 2019-09-20 02:00:00]中包包间费 消费 480 分钟 金额：1984元
[2019-09-20 02:00:00 - 2019-09-20 08:00:00]中包包间费 消费 360 分钟 金额：768元
[2019-09-20 08:00:00 - 2019-09-20 18:00:00]中包包间费 消费 600 分钟 金额：680元
[2019-09-20 18:00:00 - 2019-09-21 02:00:00]中包包间费 消费 480 分钟 金额：1984元
[2019-09-21 02:00:00 - 2019-09-21 08:00:00]中包包间费 消费 360 分钟 金额：768元
[2019-09-21 08:00:00 - 2019-09-21 18:00:00]中包包间费 消费 600 分钟 金额：680元
[2019-09-21 18:00:00 - 2019-09-21 20:50:00]中包包间费 消费 170 分钟 金额：744元
6折房间费 金额：-4416元 6折房间费
1500元代金券 金额：-1500元 1500元代金券
方便面[10 x 10.0] 金额：100元 方便面[10 x 10.0]
XXX[15 x 7.0] 金额：105元 XXX[15 x 7.0]
XMA[300 x 5.0] 金额：1500元 XMA[300 x 5.0]
XMB[400 x 2.0] 金额：800元 XMB[400 x 2.0]
```

月月鸟：服务员结过了、结过了，一次一结的

店员：好的

```
@Override
public List<Revision> getRevisions() {
    return Arrays.asList(
        new DurationRevision("2小时抵扣", 120),
        new TimeBasedOffAdjustment<KTVConsumption>(0.4f, "6折房间费"),
        new AmountAdjustment(1500, "1500元代金券"),
        new ConsumerGoods(10,10,"方便面"),
    );
}
```

```

        new ConsumerGoods(15,7,"XXX")
    );
}

```

月月鸟在KTV的消费总金额 5329 元。明细如下：

```

[2019-09-18 16:47:30 - 2019-09-18 18:47:30]2小时抵扣 金额：0元 2小时抵扣
[2019-09-18 18:47:30 - 2019-09-19 02:00:00]中包包间费 消费 432 分钟 金额：1984元
[2019-09-19 02:00:00 - 2019-09-19 08:00:00]中包包间费 消费 360 分钟 金额：768元
[2019-09-19 08:00:00 - 2019-09-19 18:00:00]中包包间费 消费 600 分钟 金额：680元
[2019-09-19 18:00:00 - 2019-09-20 02:00:00]中包包间费 消费 480 分钟 金额：1984元
[2019-09-20 02:00:00 - 2019-09-20 08:00:00]中包包间费 消费 360 分钟 金额：768元
[2019-09-20 08:00:00 - 2019-09-20 18:00:00]中包包间费 消费 600 分钟 金额：680元
[2019-09-20 18:00:00 - 2019-09-21 02:00:00]中包包间费 消费 480 分钟 金额：1984元
[2019-09-21 02:00:00 - 2019-09-21 08:00:00]中包包间费 消费 360 分钟 金额：768元
[2019-09-21 08:00:00 - 2019-09-21 18:00:00]中包包间费 消费 600 分钟 金额：680元
[2019-09-21 18:00:00 - 2019-09-21 20:50:00]中包包间费 消费 170 分钟 金额：744元
6折房间费 金额：-4416元 6折房间费
1500元代金券 金额：-1500元 1500元代金券
方便面[10 x 10.0] 金额：100元 方便面[10 x 10.0]
XXX[15 x 7.0] 金额：105元 XXX[15 x 7.0]

```

抹零

月月鸟：你把零头抹了吧

店员：好的

```

public List<Revision> getRevisions() {
    return Arrays.asList(
        new DurationRevision("2小时抵扣", 120),
        new TimeBasedOffAdjustment<KTVConsumption>(0.4f, "6折房间费"),
        new AmountAdjustment(1500, "1500元代金券"),
        new ConsumerGoods(10,10,"方便面"),
        new ConsumerGoods(15,7,"XXX"),
        new ZeroRemainderAdjustment("抹零",100)
    );
}

```

月月鸟在KTV的消费总金额 5300 元。明细如下：

```

[2019-09-18 16:47:30 - 2019-09-18 18:47:30]2小时抵扣 金额：0元 2小时抵扣
[2019-09-18 18:47:30 - 2019-09-19 02:00:00]中包包间费 消费 432 分钟 金额：1984元
[2019-09-19 02:00:00 - 2019-09-19 08:00:00]中包包间费 消费 360 分钟 金额：768元
[2019-09-19 08:00:00 - 2019-09-19 18:00:00]中包包间费 消费 600 分钟 金额：680元
[2019-09-19 18:00:00 - 2019-09-20 02:00:00]中包包间费 消费 480 分钟 金额：1984元
[2019-09-20 02:00:00 - 2019-09-20 08:00:00]中包包间费 消费 360 分钟 金额：768元
[2019-09-20 08:00:00 - 2019-09-20 18:00:00]中包包间费 消费 600 分钟 金额：680元
[2019-09-20 18:00:00 - 2019-09-21 02:00:00]中包包间费 消费 480 分钟 金额：1984元

```

[2019-09-21 02:00:00 - 2019-09-21 08:00:00]中包包间费 消费 360 分钟 金额: 768元
[2019-09-21 08:00:00 - 2019-09-21 18:00:00]中包包间费 消费 600 分钟 金额: 680元
[2019-09-21 18:00:00 - 2019-09-21 20:50:00]中包包间费 消费 170 分钟 金额: 744元
6折房间费 金额: -4416元 6折房间费
1500元代金券 金额: -1500元 1500元代金券
方便面[10 x 10.0] 金额: 100元 方便面[10 x 10.0]
XXX[15 x 7.0] 金额: 105元 XXX[15 x 7.0]
抹零 金额: -29元 抹零

店员：一共5300，老板是现金、刷卡还是电子支付？