

# 面向对象程序设计Lab5实验报告

姓名：吴优 学号：22307130158

## 1.实验目标

在本次 Lab 中，需要模拟全家零售店售卖系统，每天早上进货，进行销售，晚上清理过期商品，不断重复。

## 2.结构设计

在认真阅读要求文档后，设计的类与方法如下：

FamilyMart类	超市类	物品的买入、卖出及清理
Product类	产品类	保存物品的价格、生产日期等基本信息
Main类	主函数类	负责读入、写出文件

## 3.具体功能实现

### 3.1 Product类

- 属性:
  - name：商品名
  - price：商品价格
  - life:保质期时间
  - produceYear、producemonth、produceDay：生产年月日
  - produceDate：以开业时间为零点计算的生产日期。
- 方法:
  - compareTo:通过重写 compareTo 方法，定义了 Product 对象的自然排序方式，即根据它们的生产日期进行比较。该方法将当前对象（this）的 produceDate 与另一个 Product 对象（other）的 produceDate 进行比较

### 3.2 FamilyMart类

- 属性
  - allProducts：HashMap类型，以每一种商品的名称为键，以PriorityQueue为值
  - date：记录当天的开业日期
  - turnover：开业至当天的累计收入
- 方法
  - purchase:用于每一天的购买商品
  - sell：每一天的出售商品
  - clearOutOfDate：每一天的清理过期商品
  - WriteToFile：将每日的turnover写入文件
  - addProduct：将指定商品加入到allProduct中
  - isOutOfDate:判断指定商品是否过期

## 4.核心代码分析

### 4.1 purchase方法

```
public void purchase(String filePath)
{
    try(BufferedReader br=new BufferedReader(new FileReader(filePath)){
        String firstLine =br.readLine();
        String line;
        while((line=br.readLine())!=null)
        {
            String []data=line.split("\\s+");//以若干个空白符分割
            Product a=new Product();
            if(data.length==4)
            {
                a.name=data[0];
                a.price=Double.parseDouble(data[1]);
                a.life=Integer.parseInt(data[2]);

                String []dateData=data[3].split("/");
                a.produceYear=Integer.parseInt(dateData[0]);
                a.produceMonth=Integer.parseInt(dateData[1]);
                a.produceDay=Integer.parseInt(dateData[2]);
                LocalDate endDate =
LocalDate.of(a.produceYear,a.produceMonth,a.produceDay);
                LocalDate startDate = LocalDate.of(2022, 5, 2);
                // 计算日期之间的天数差异
                a.produceDate = (int)ChronoUnit.DAYS.between(startDate, endDate);
            }
            if(!isOutOfDate(a)) {
                addProduct(a);
            }
        }
    } catch (IOException e) {
        e.printStackTrace();
    }
}
```

- 由于所给文件中数据的分隔格式不统一，" "或者"/"分隔都存在于文件，所以采用正则表达式"\s+"来进行对于整行数据的拆分
- 计算生产日期与开业时间的间隔，使用java的库函数直接进行计算，并将数据存入produceDate中
- 最后判断购入时是否过期，如果没有过期就将物品添加到allProducts中，防止购入过期商品

### 4.2 sell方法

```
public void sell(String filePath)
{
    try (BufferedReader br=new BufferedReader(new FileReader(filePath)))
    {
        String temp=br.readLine();
        String line;
        while((line=br.readLine())!=null) {
            String[] data = line.split("\\s+");
            if (allProducts.get(data[0]) != null) {
                Product soldProduct = allProducts.get(data[0]).poll();
                if (soldProduct != null && data.length == 1) {

                    turnover += soldProduct.price;

                }
                if (soldProduct != null && data.length == 2) {

                    turnover += soldProduct.price * Double.parseDouble(data[1]);

                }
            }
        }
    }

    }catch(IOException e)
    {
        {
            e.printStackTrace();
        }
    }
}
```

- 由于没有折扣的商品的discount值为空，所以需要判断拆分后数组的长度来进行分别处理，防止数组访问越界。

### 4.3 clearOutOfDate方法

```
public void clearOutOfDate() {
    List<String> keysToRemove = new ArrayList<>();
    for (Map.Entry<String, PriorityQueue<Product>> entry : allProducts.entrySet()) {
        PriorityQueue<Product> queue = entry.getValue();
        Iterator<Product> iterator = queue.iterator();
        while (iterator.hasNext()) {
            Product product = iterator.next();
            if (isOutOfDate(product)) {
                iterator.remove(); // 删除过期商品
            }
            else
            {
                break;
            }
            // 如果队列为空, 记录需要删除的键
            if (queue.isEmpty()) {
                keysToRemove.add(entry.getKey());
            }
        }
        // 删除空队列对应的键值对
    }
    for (String key : keysToRemove) {
        allProducts.remove(key);
    }
}
```

- 遍历哈希表, 对于每一个优先队列使用迭代器遍历, 判断每个商品是否过期, 过期则删除。遍历结束后再检测队列是否为空, 删除队列为空的键值对。

## 5.附加部分

### 5.1

在本次lab中, 关于商品的存储方式采取了HashMap<String,PriorityQueue<Product>>的方式存储, 主要在两个方面提高了代码性能:

- 采取哈希表存储每一类的商品, 在卖出指定名称的商品是查询耗时缩小到O~ (1) ~。
- 采用优先队列存储每一类中的具体所有商品, 以produceDate升序排列, 当卖出物品时只需要取队首元素即可。

### 5.2

实现了跨年的商品日期处理, 在purchase方法中, 使用java提供的库函数LocalDate, temporal.ChronoUnit等用来计算生产日期和开业日期2022.5.2之间的天数。

## 实验收获和感想

本次实验实现了全家售卖系统，主要锻炼了对于类与方法的设计与实现能力，并且在实现过程中发现问题(商品买入时可能已经过期)，并解决问题。同时思考提高性能的方法，选择使用哈希表和优先队列，锻炼对于封装结构和函数的使用，收益颇多。