

Java&Android第二轮作业

1. 考察的知识点

- 封装、继承、抽象、多态（面向对象四大特性）、接口
- 异常处理
- 泛型
- 集合
- 了解git的使用
- 多线程（19级专属）

本次考核20级只需完成前两道题，19级需完成所有题目，教程同第一轮

2. 本轮题目

- 本轮涉及的知识点比较多，如果有不懂的可以在群里问或者私聊工具人（活跃一点加分）
- 第一题题目较长，阅读时请耐心

西二炸鸡店：

工作室给了你一笔钱去创业，本着造福大家的目的（绝不是因为自己爱吃）你决定开一家炸鸡店！目前只出售套餐（一份炸鸡+一份饮料），饮料有各种啤酒和果汁。作为被工作室寄予厚望的创业者，你决定自己开发炸鸡店管理系统为经营做准备。

- 你需要完成以下几个类

- 饮料类Drinks（抽象类）
 - 变量：自己的名字，成本（double），生产日期（LocalDate类），保质期（单位：天）
 - 函数&方法：全参构造函数，判断当前是否过期方法（使用LocalDate），，抽象的toString方法
- 啤酒类Beer（继承自饮料类）
 - 变量：酒精度数（float）
 - 保质期为30天
- 果汁类Juice（继承自饮料类）
 - 保质期2天
- 套餐类SetMeal
 - 变量：套餐名（如“套餐1”，“套餐2”等），套餐价格（double），炸鸡名，饮料类成员变量（具体是哪种饮料（啤酒还是果汁）取决于实际情况（多态））
 - 方法：覆写toString方法
- 炸鸡店接口FriedChickenRestaurant：
 - 抽象方法：出售套餐，批量进货
- 自定义异常类：
 - 定义两个异常类IngredientSortOutException（果汁或啤酒售完），OverdraftBalanceException（进货费用超出拥有余额），继承自RuntimeException
 - 在啤酒或者果汁售完时应该告知顾客，进货是如果货款大于店铺余额需要告知店主，异常中均需要携带一定的信息，方便捕获异常后进行处理（比如哪种饮料售空，或者进货差多少钱）

- 西二炸鸡店类West2FriedChickenRestauran（实现炸鸡店接口）
 - 变量：餐厅账目余额（判断是否买的起原料），啤酒列表、果汁列表，套餐列表（注意这三个变量不是普通数组）用于保存店里所有的啤酒、果汁和套餐
 - 私有方法：重载use函数（即两个use方法传入的参数不同），分别接受Beer或者Juice参数，在出售套餐是同时移除对应的饮料（通过饮料名字在列表中判断是否有对应的饮料，如果存在选择出售第一个符合条件的，否则应该抛出异常）
 - 实现炸鸡店接口，能够进行售卖和批量进货（用instanceof来判断是啤酒还是果汁）
 - 在创建该类时对套餐列表进行初始化，套餐初始化后不可变，要求使用静态代码块

Tips:

- 如果遇到与现实逻辑不符的问题以考核内容为主
- 抽象类的成员变量均为protected，注意要实现的类变量类型
- 保存啤酒、果汁和套餐的列表使用ArrayList、LinkedList或者其他List，加分项：为什么你选择这个List保存啤酒和果汁
- 注意各个类之间的关系
- ! ! ! ! ! 注意命名规范，第一轮考核给了类名还有奇奇怪怪的命名，注意不要直接更改类名，导致项目无法运行后果自负
- Android方向的小伙伴如果想用kotlin写也可以（不做强制要求），但是不要用编译器自带的Java转kotlin功能

git初步:

- 请注册一个github账号，熟悉github和常用的git命令行（实在不会用可以下载GitHub Desktop），将考核作业整个项目（注意是整个项目！！）提交在github上并且提交仓库链接（后续以收集文档方式提交链接）

git的教程：菜鸟教程<https://www.runoob.com/git/git-basic-operations.html>

多线程（20选做）：

小明是一个非常喜欢数学的孩子，有一天他突然想知道1~n中含有某个数字的所有数字之和（比如x是9，在1~n内所有含有9的数字之和，9，19，29.....91，92这样），于是他写下了

```
public static void main(String[] args) {
    long ans = 0;
    Scanner scanner = new Scanner(System.in);
    int x = scanner.nextInt();
    for (long i = 1; i < 1000000000; i++) {
        if (contain(i, x)) ans += i;
    }
    System.out.println(ans);
}

private static boolean contain(long num, int x) {
    return String.valueOf(num).contains(String.valueOf(x));
}
```

但是跑了很久很久才出来答案，于是小明想到用多线程来快速得到答案，现在让你帮他实现这一个功能（提示：用多线程将循环分块处理）

注意：抄袭是严格禁止的，如果是组员之间相互交流，请理解后自己写一遍！！！！代码雷同率太高会影响双方进入下一轮！！

截止时间：2020年12月12日23点59分59秒
