

# ASN\_03 工程笔记

2020年2月22日 下午 2:54

作业前置要求:

## 1. 完成junit连接到项目

☒ a. 问题: idea无法找到org.junit.platform.runner

i. 解决尝试:

- 1) 安装eclipse并使用教授讲的方法连接junit (无效, 无法解决问题)
- 2) 在连接库junit-jupiter(即JUnit 5.4)之前, 首先完成对org.junit.platform.runner的连接 (有效, 问题已解决)

ii. 原因分析: idea(Version 2019.3)无法自动识别\*.platform.runner因而无法自动下载库

☒ b. 问题: idea无法找到org.junit.platform.suite

i. 解决尝试:

- 1) 安装eclipse并使用教授讲的方法连接junit (无效, 无法解决问题)
- 2) 在连接库junit-jupiter(即JUnit 5.4)之前, 首先完成对org.junit.platform.suite的连接 (有效, 问题已解决)

ii. 原因分析: idea(Version 2019.3)无法自动识别\*.platform.suite因而无法自动下载库

待完成部分:

## 1. Pizza.java

a. 要求

☒ i. 对象Pizza需要包含属性:

- 1) 大小, size, "Small", "Medium", "Large"
- 2) 全素, vegetarian, "true", "false"(如果该属性为真, 则ham为"None")
- 3) 奶酪, cheese, "single", "double", "triple"
- 4) 菠萝, pineapple, "None", "Single"
- 5) 青椒, greenpepper, "None", "Single"
- 6) 火腿, ham, "None", "Single"

☒ ii. 第一个构造方法需要接受以上全部六个属性

☒ iii. 第二个构造方法接受五个属性, 全素"vegetarian"默认为否

☒ iv. 第三个构造方法参数为空, 默认创建一个size="Small", vegetarian="false", cheese="Single", ham="Single"的结果

☒ v. 使用一个getCost访问器以计算价格

☒ vi. 创建一个toString的方法, 用于返回一个描述披萨详情的字符串:

1) E.g.

Medium vegetarian pizza, Double cheese. Cost: \$10.50 each.

Small pizza, Single cheese, pineapple, green pepper, ham. Cost: \$11.50 each.

Large pizza, Triple cheese, pineapple, ham. Cost: \$17.00 each.

vii. 对象Pizza应当有java.io.Serializable接口以便保存到LineItem对象的文件中

viii. 对象Pizza不可更改

ix. 披萨价格算法: 小奶酪披萨 (仅含单个奶酪) 为\$ 7.00, 中型奶酪 (仅含奶酪) 为\$ 9.00, 大

奶酪（仅含奶酪）为\$ 11.00。 每增加一个浇头成本为\$ 1.50。

- x. 包含一个equals方法，用于判断是否所有的属性都identical (该方法将会在junit中得到测试)

## 2. IllegalPizza.java

### a. 要求

- i. 这是一个Exception类对象，写一个参数为String的构造方法。

## 3. LineItem.java

### a. 要求

- i. 包含一个指向Pizza对象的指针，以及订购的数量（在1到100之间）
- ii. 第二个构造函数默认使用一个披萨作为订单
- iii. 订单的数量可变，但是属性不变
- iv. 需要包含能获取上述两个属性的访问器，同时还要包含一个能返回订单总价的访问器，如果订单数量在10到20之间，则有9折优惠，如果在20个以上，则有85折优惠。
- v. 需要包含toString方法，用于显示完整的订单。如果订单项小于10，则要在字符串前加入空格，使得字符串对其
- vi. 需要一个基于订单项总费用的compareTo方法，令ArrayList排序。当订单显示时，将会按照价格从高到低排序，若价格差异在1元以内，则返回0。
- vii. 拥有java.io.Serializable接口，令LineItem示例可以直接保存到文件

待补充与测试部分（资源由教授提供）

## 1. AssnUnitTestSuite.java

## 2. LineItemTest.java

## 3. PizzaTest.java

## 4. PizzaOrderSystem.java

完成总结

- 本次作业涉及的知识点繁多，且大多数知识点并未在课堂上提及
- 关于junit
  - junit是java运行时用于测试程序的一个库，本次程序教授提供了junit-jupiter版本。由于教授仅仅提供了eclipse的junit安装方法。intellij idea的junit安装方法中需要使用在project structure-module-dependency中maven添加需要的库。要确保添加的版本一致，且所有库都由maven添加，否则容易出现无法运行测试的方法
- 关于方法重载
  - 在java中@Override提供了重写父类方法的手段。当需要使可克隆时，我们需要继承(implements) cloneable类，然后重载其中的clone()方法
  - 方法重写还可以用于重写equals。一般equals会对比数值和他们的存储位置，所以在对比object的时候，需要重写equals以确保存储位置不会被对比
- 关于可序列化
  - 如果希望对象可序列化以写入文件，需要在声明class时，继承(implements) Serializable。以实现Serializable接口
- 关于自定义错误对象
  - 自定义错误对象需要继承(extend) 一个已有的错误类，本次作业中继承的是RunTimeException