**An entire restaurant**

Department: Electronic and Electrical Engineering

Title:Sheffield restaurant

Student Name: your name

Registration Number: --------

**I declare that this work is my own and I acknowledge the contribution of**

**others where appropriate.**

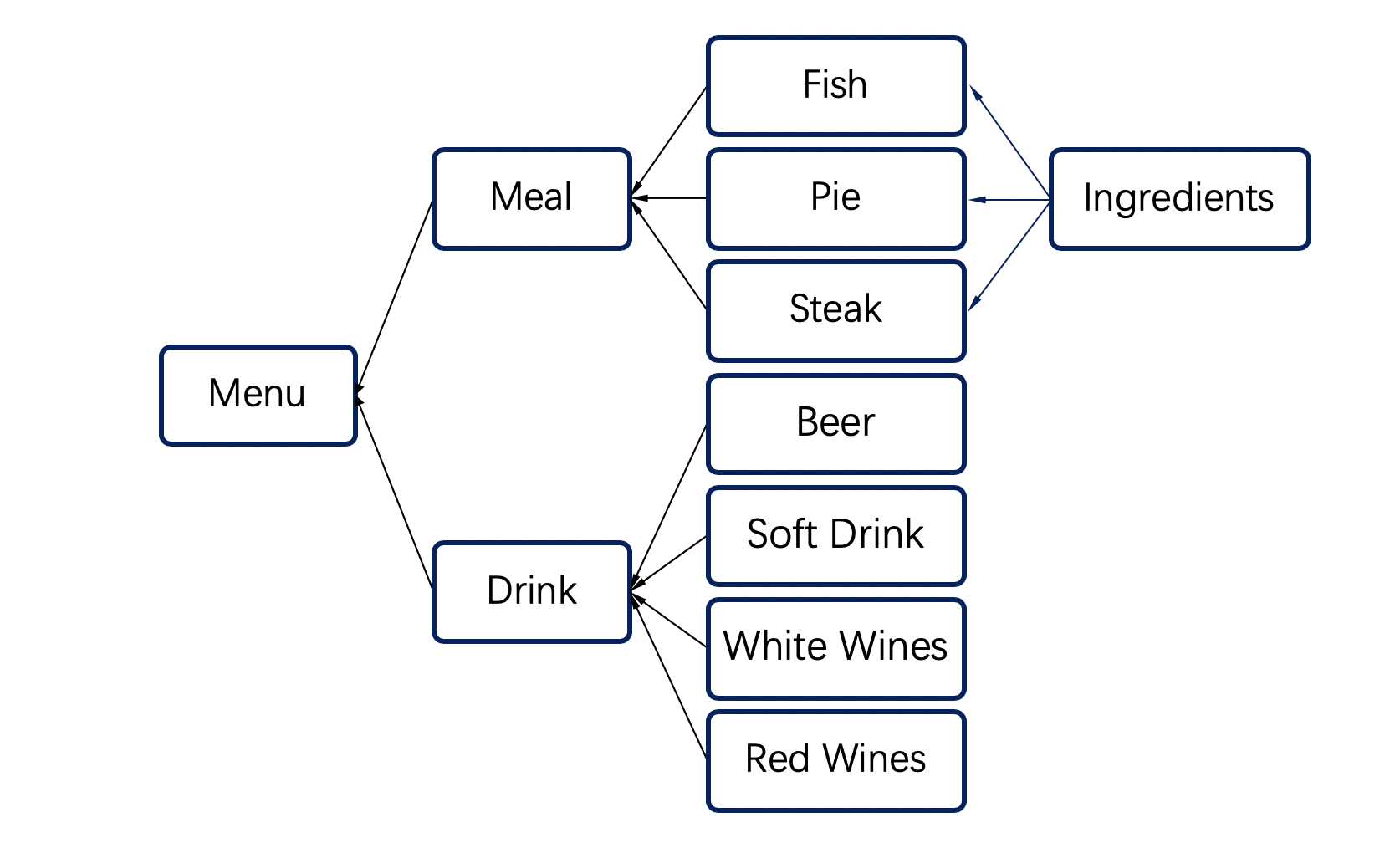
**Overview**

The whole project consists of three packages, namely, *drink*, *meal* and *Sheffield*. In the drink package, multiple drinks are implemented, such as *beers* and *redwines*. In the *meal* package, three main foods and ingredients are implemented. Finally, the *Menu* class is used to instantiate and display the above mentioned classes, that is, to integrate them into a fixed menu style.

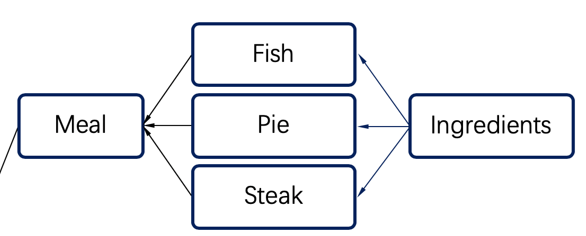
The UML of the project is as follows:

Figure 1 A Full Menu

In order to accomplish an entire menu, different kinds of java codes are employed to combine each components of the menu. In fact, the whole program contains a main program which is named Menu and three subprograms which are named Meal, Price and Drink, respectively.



**Meals（Package）**



总共有五个类在这个Package中，分别是Meal，Fish，Steak和Ingredinte

*第一个类Meal*

*Meal* is an abstract class. The member variables include the following:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Name | Price | Type | Ingredintes | Isvegetarian | Isvgan | count |

The name and type (*pie, fish or steak)* are determined by the restaurant manager's input of the dish name through the method public meal (string name, string type).

I*ngredients[]* is the ingredients list price used to store this dish. The price is calculated by price(). It is obtained by multiplying the price of each ingredient by 1.3 (20% labor cost, 10% profit)

Variables of *Isvegetarian* and *isvgan* are obtained by checking whether the ingredients are all (V) or (VV) by the functions *isvegetarian()* and *isvgan().*

*Contains()* can determine whether the ingredient is in the main food by entering the name of the ingredient. *Addingredients()* can add an ingredient to the current main food object.

Finally, the abstract method, *drinksrecommendation(),* must be implemented by its subclass inheritance.

以上是meal类中比较重要的函数，但不是全部

第二个类是Pie类

They inherited the meal class and overridden the *drinksrecommendation()* method. The following class returns a *beers* object, which recommends beers as the drink when the main

food is pie.返回的Drink类对象具体是怎么样的会在对package Drink中进行描述，你可以用*public Pie(String name,String type) {super(name,type);}*

输入pie的名字和种类来实例化一个pie对象

第三个类是Fish类

和上个类一样，继承了meal类并且重写了drinksRreconmmendation()method ,他将返回一个WhiteWine对像，类里的基本信息可以参考meal，

第四个类是Steak类

和Pie和Fish类相似，重写了drinksRreconmmendation()method，具体就不再阐述了

第五个类是Ingredintes

*Ingredients* is an public class. The member variables include the following:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | cost | isVegetarian | isVegan | consumption | unitPrice |

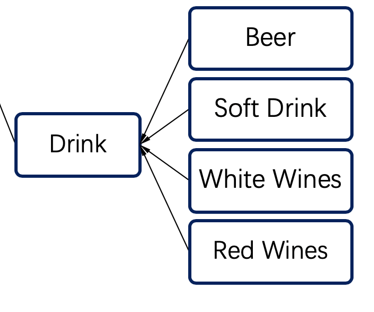
The quantity and unit price of the food material class need to be entered when instantiating its object.

*Isvegetarian()* and *isvegan()* are used to judge whether the ingredients are vegetarian. *Addcost()* method to calculate the cost of this ingredient.

**Prices**

最终meal的售价取决于每个组成这个meal的食材的成本价，即上面计算的每个食材的cost总和并且加上20%的员工费用，10%的收益，也就是这个meal的售价为食材费用总和的1.3倍

**Drinks（Package）**

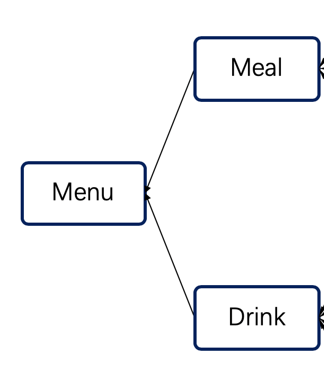


Package Drinks 中包含了五个类，一个Drink类，以及，Beer ，Soft Drink White Wines和Red Wines

The main attributes of class *drink* are *name* and *price*, it is the parent of all different drinks. And it can be inherited by other specific drink classes.

这样子设计是为了将代码简化，对每个不同类型的饮品有不同的价格，提供了setPrice和setName的类方法，如 当初始化一个Beer对象时，会自动调用setPrice和setName方法，为这个Beer设置我们提前设计好的价格，这个价格是可以修改的

**Menu（Main）**



*Menu* is the public main class. The member variables include the following:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Title | Pies | Drinks | Fishes | Steaks | count |

在这个类中，提供了addPie() addFish() addDrink() addSteak()这四个类方法来向实例化后的menu对象添加商品

同时提供了display方法来显示整个菜单的信息

The main process of Menu, that is, the core process of the whole project. First, read in the data. Create an *ingredient* object and add it to the *ingredients* list. Then create the corresponding *meal* object, repeat the process until all the *ingredients* are added, and then add a specific drink to the menu by instantiating a *drink* object. Finally, call the *display()* function to display the menu.

**Conclusion**

The main function of the whole project is to input information in a certain format and then output a menu in a fixed format. Information of the same kind will be output to the adjacent area and the appropriate price will be calculated automatically.

**并且会自动的推荐对应合适的饮品，每次有新的商品需要上架的时候，只需要在输入的txt的末尾加上这个商品的信息，程序就能自动得将整理好后的菜单打印出来**

*Java* is an object-oriented programming language, which has three characteristics: inheritance, polymorphism and abstraction. And Class is the carrier to encapsulate the properties and behaviors of objects

The whole project makes good use of the above three characteristics to make the project simple, efficient, easy to update and maintain.