

一、项目内容

1.1 项目主题

通过命令行方式，编写 RPG 游戏两个角色 PK 的简单场景。

游戏里有角色和道具两种对象。游戏角色有三种：战士、法师、妖怪，有共同的属性：生命值、攻击力、防御力和金钱，具体的属性值因角色的不同而变化。道具又分为三种：武器，盔甲和补药。战士武器为巨剑，分为四个等级：白银，黄金，白金和钻石；法师的武器为法杖，同样分四个等级；妖怪的武器为灵刃，分四个等级。战士的盔甲为铠甲，法师的是面纱，妖怪的是鳞甲，同样分为四个等级。这些道具均可以用金钱升级，并且可以用低等级的合成高等级的。补药可以回复一定的生命值，使用后有一定的冷却时间。

游戏采用战棋类玩法，两个角色轮流进行操作，每回合可以做一个下列动作的一种：攻击、买武器、买盔甲、补血。每一回合，角色所持有的金钱数随机的增加（200~300）。每一回合攻击时，被攻击方掉血量等于攻击方的攻击力减去被攻击方的防御力，当有一方生命值变为 0，则游戏结束。

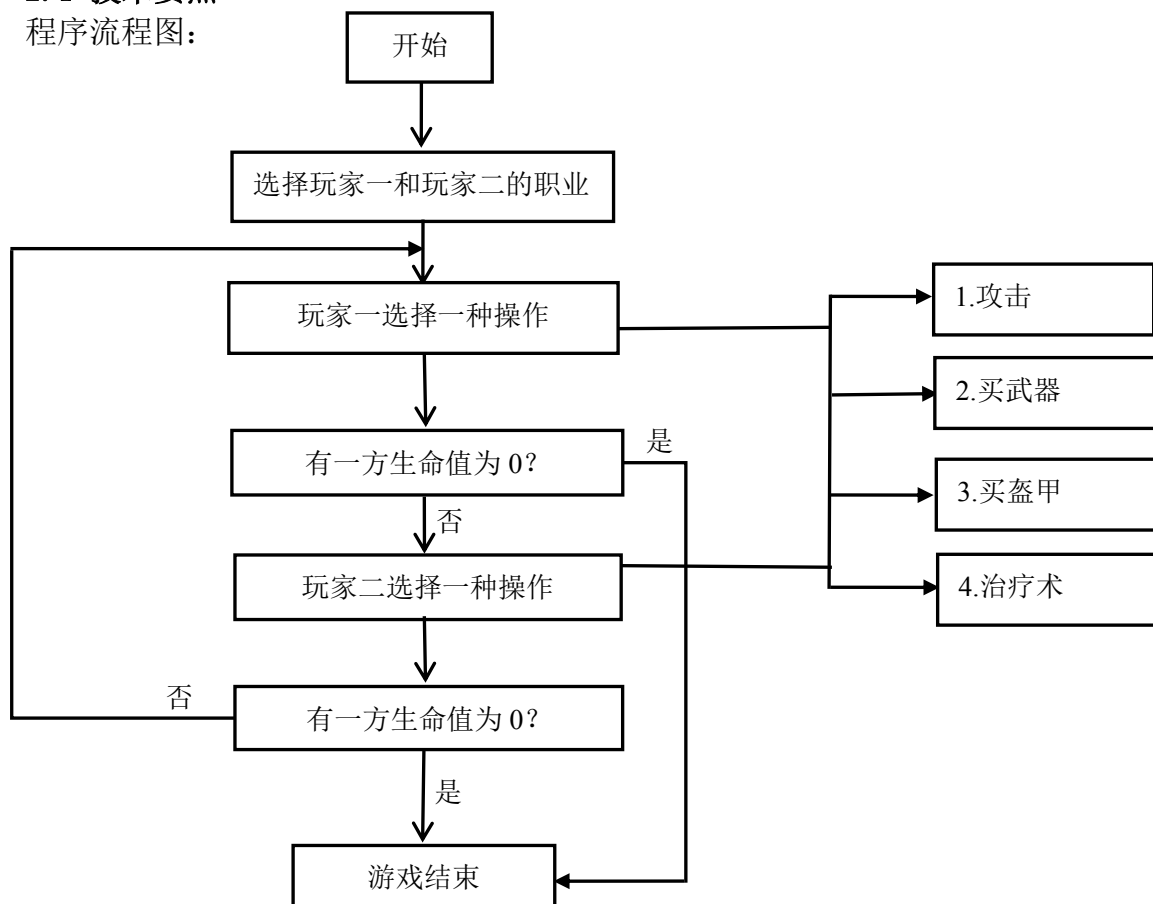
1.2 运行环境

运行在 windows 软件（Visual Studio, 2008）环境下。

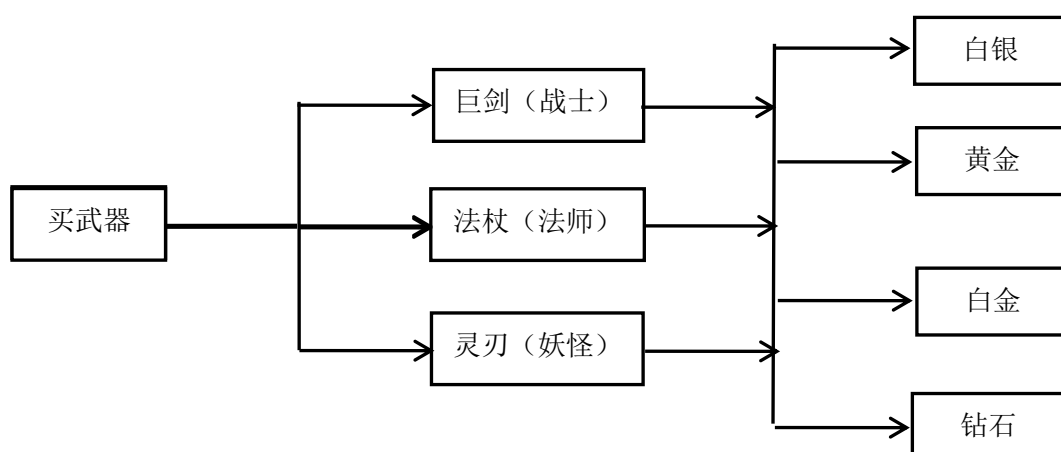
二、程序分析

2.1 技术要点

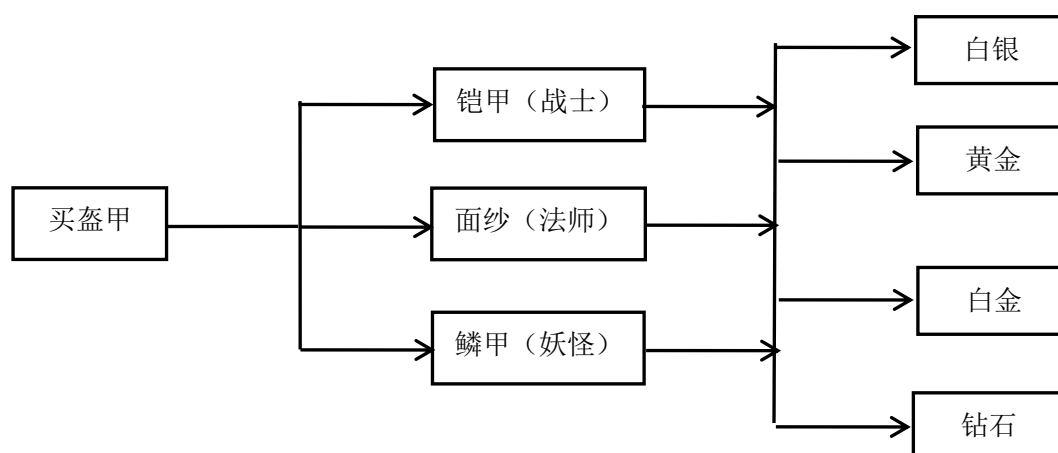
程序流程图：



武器又分为三种：巨剑、法杖、灵刃，分别对应战士、法师、妖怪，且武器分成四个等级。



盔甲分为三种：铠甲、面纱、鳞甲，分别对应战士、法师、妖怪，且盔甲分成四个等级。



- 1、通过不同的选择，从而获得不同的道具，可以提升攻击力或者防御力。
- 2、不同的角色有着属于自己的武器和盔甲，用金钱购买，低等级的道具可合成高等级的道具。
- 3、攻击时，减少的血量等于攻击方的攻击力减去被攻击方的防御力。
- 4、使用治疗术后，会有四个回合的冷却时间，在冷却时间内不能再次使用。

2.2 项目任务书

杨郭镱负责编写各种基类和派生类，编写主函数。

苏伟负责调试程序，完善系统，撰写报告。

黄子彤负责优化程序，完善系统。

三、技术路线

3.1 总体方案

3.1.1 编译软件及开发思想

采用微软公司的 Visual Studio 2008 版本作为编译软件。利用面向对象的

思想设计出通过命令行方式、简单的 RPG 游戏。使用类的封装性、继承性和多态性，实现‘开闭原则’。

3.1.2 总体方案

1. 设置一个基类 Character（角色），再由这个基类派生出三个派生类，分别为 Warrior（战士），Wizard（法师），Monster（妖怪）。再设置一个类 CharacterFactory（角色产生工厂），根据提示进行输入，可以得到想要的角色，比如输入 1 可以 return new Warrior（）。

2. 设置一个管理商店类 StoreManager，根据选择的角色，返回相应的武器商店类和盔甲商店类。

3. 设置一个武器商店虚基类，派生出三个相应角色的武器商店类，在商店类中，根据不同的选择，返回相应等级的武器类。对盔甲商店类也进行同样的设置。

4. 设置一个基类 Weapon（武器），再由这个基类派生出三个派生类，分别为：WarriorWeapon（战士武器类），WizardWeapon（法师武器类），MonsterWeapon（妖怪武器类）。每个派生类作为基类再派生出四个派生类，比如 WarriorWeapon 派生出 WarriorWeapon1（白银等级类），WarriorWeapon2（黄金等级类），WarriorWeapon3（白金等级类）和 WarriorWeapon4（钻石等级类），代表四个攻击力逐渐提高的巨剑武器。

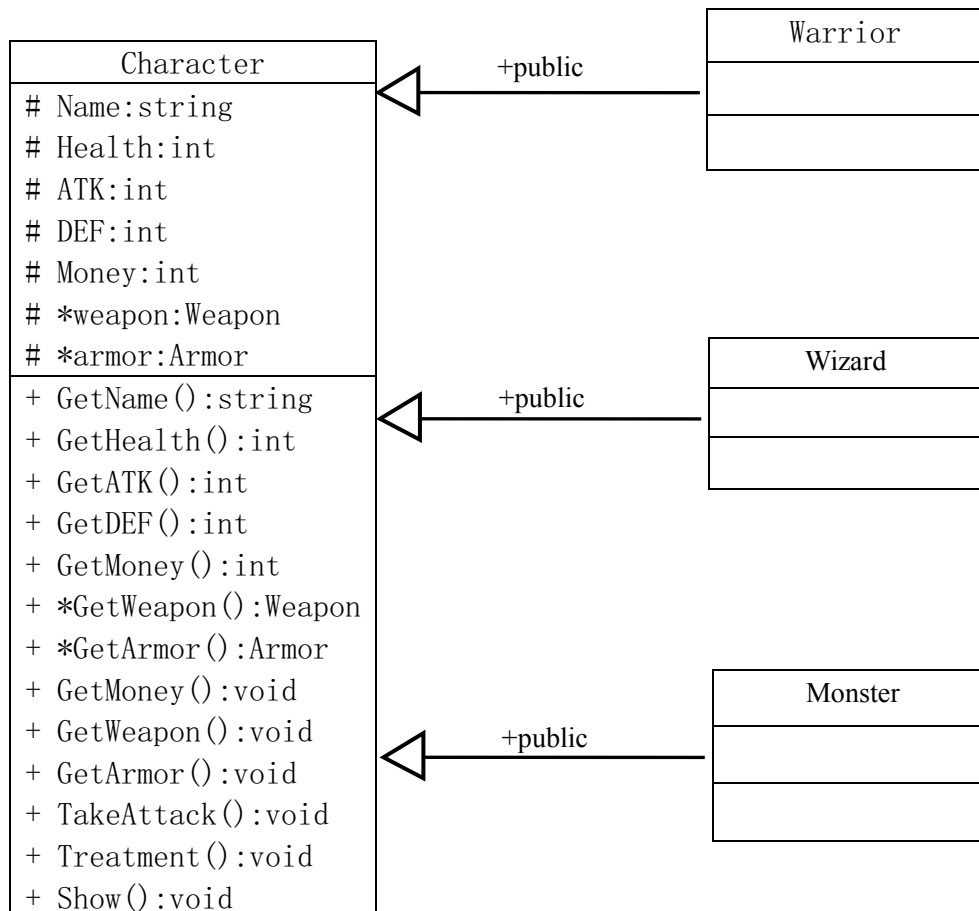
5. 设置一个基类 Armor（盔甲），再由这个基类派生出三个派生类，分别为：WarriorArmor（战士盔甲），WizardArmor（法师盔甲），MonsterArmor（妖怪盔甲）。每个派生类作为基类再派生出四个派生类，比如 WarriorArmor 派生出 WarriorArmor1（白银等级），WarriorArmor2（黄金等级），WarriorArmor3（白金等级）和 WarriorArmor4（钻石等级），代表四个防御力逐渐提高的铠甲。

具体的讲，角色基类 Character 中的保护成员有：职业名称，生命值，攻击力，防御力，指向武器类的指针变量和指向盔甲类的指针变量。还有一些普通的成员函数 GetName()、GetHealth()、GetATK()、GetDEF()、GetMoney()、GetWeapon()、GetArmor() 用来返回职业名称、生命值、攻击力、防御力、金钱、指向武器类的指针变量和指向盔甲类的指针变量。还有使用治疗术的函数 Treatment()，显示函数 Show()，专门用来显示各种属性信息。

对于武器基类 Weapon 中，有武器名称、武器价钱、武器攻击力和武器等级作为保护成员。对于盔甲基类 Armor 也有相同的设置。

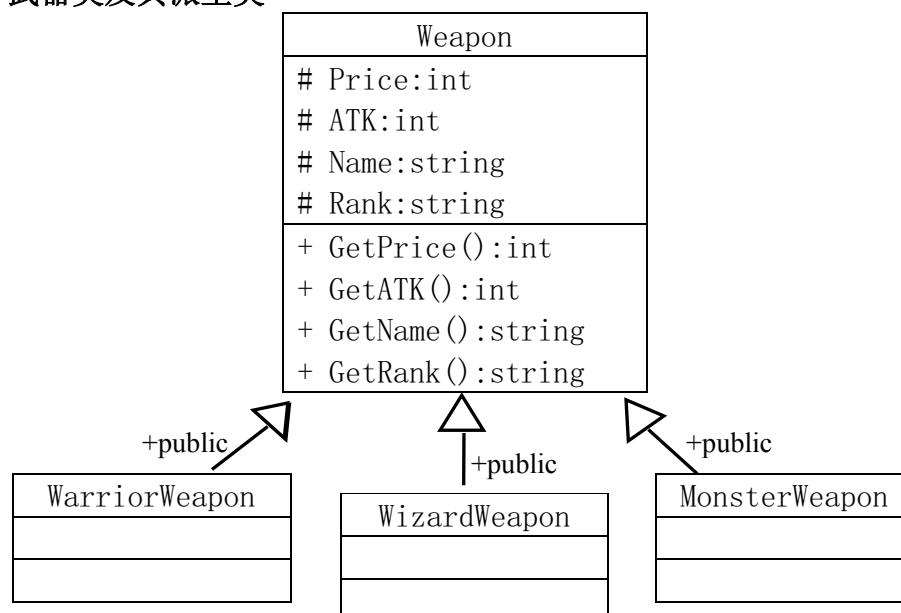
3.2 UML 类图

3.2.1 角色类及其派生类



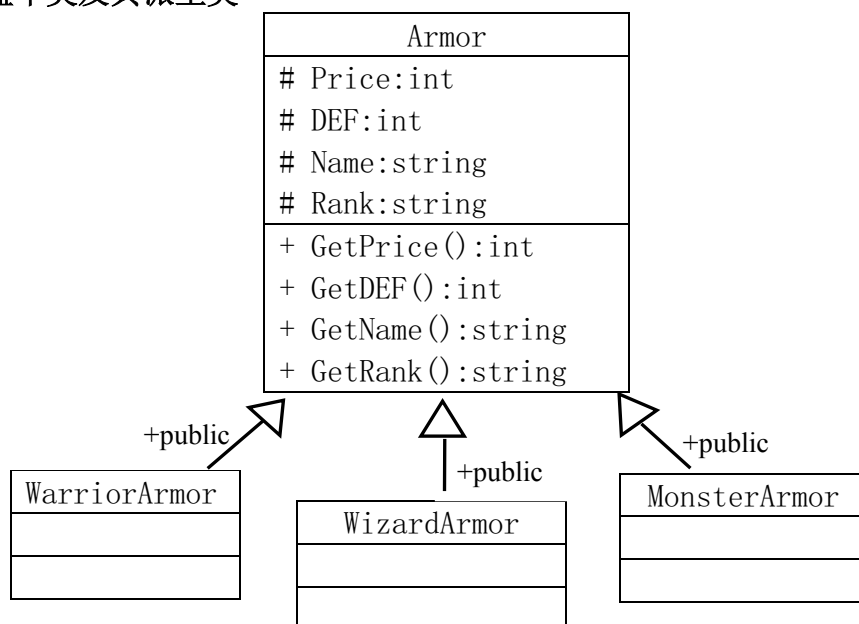
功能：角色基类和三种角色派生类，对不同角色的信息进行存储、管理和显示。

3.2.2 武器类及其派生类



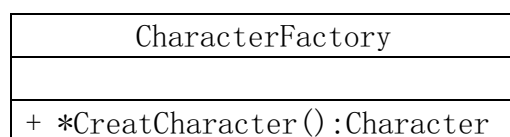
功能：Weapon 作为基类，派生出三种武器类，存储三种不同武器的信息，与三种角色相对应。

3.2.3 盔甲类及其派生类



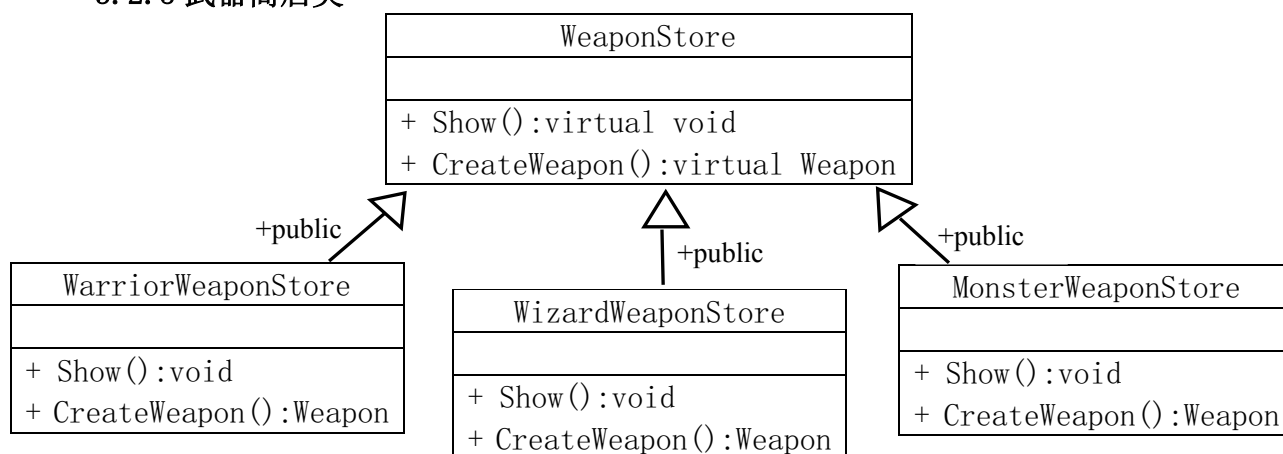
功能：Armor 作为基类，派生出三种盔甲类，存储三种不同盔甲的信息，与三种角色相对应。

3.2.4 角色生产工厂



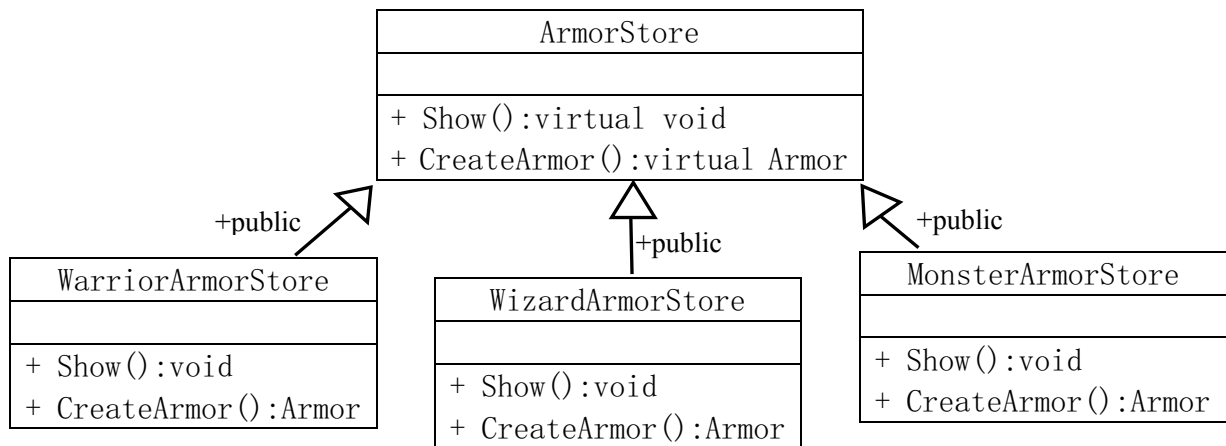
功能：创建玩家想要的角色，在 `CreatCharacter()` 函数中，根据输入的指令可以返回相应的角色类（Warrior, Wizard, Monster 三者之一）。

3.2.5 武器商店类



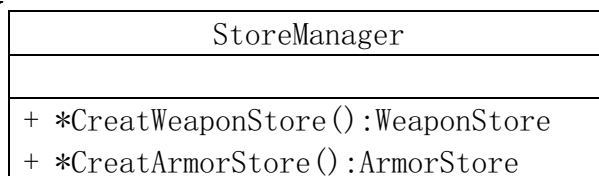
功能：WeaponStore 作为虚基类派生出三种武器商店类，对应三种角色。Show() 函数用来显示武器信息，CreateWeapon() 函数用来返回不同等级的武器类。

3.2.6 盔甲商店类



功能：ArmorStore 作为虚基类派生出三种盔甲商店类，对应三种角色。Show() 函数用来显示盔甲信息，CreateArmor() 函数用来返回不同等级的盔甲类。

3.2.7 管理商店类

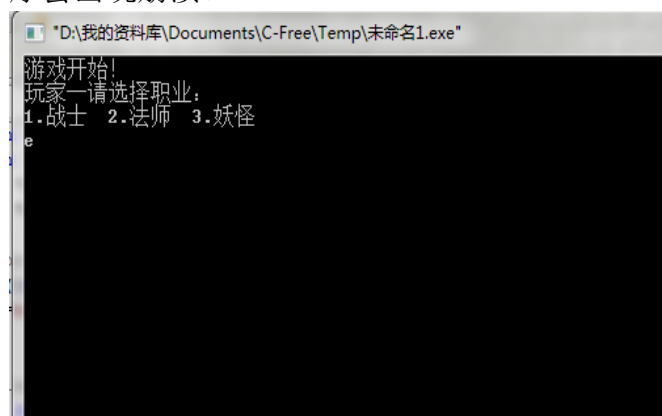


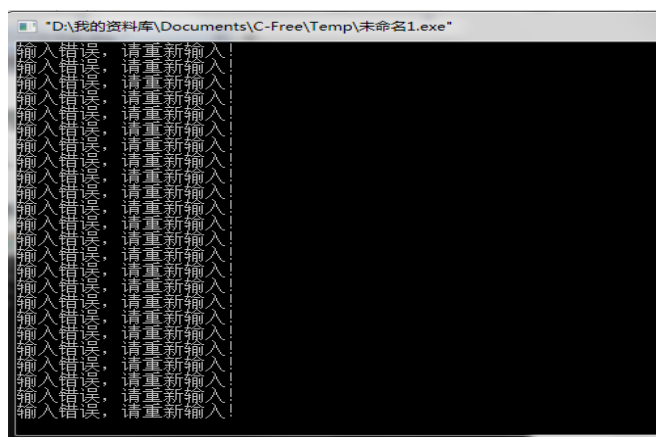
功能：StoreManager 类用来管理 WeaponStore 类和 ArmorStore 类。CreatWeaponStore() 函数根据角色职业名称返回相应的武器商店类，CreatArmorStore() 函数根据角色职业名称返回相应的盔甲商店类。

四、测试报告

4.1 第一次测试

在程序基本设计好的时候，我们进行了第一次调试。发现当输入错误时，程序会出现崩溃。





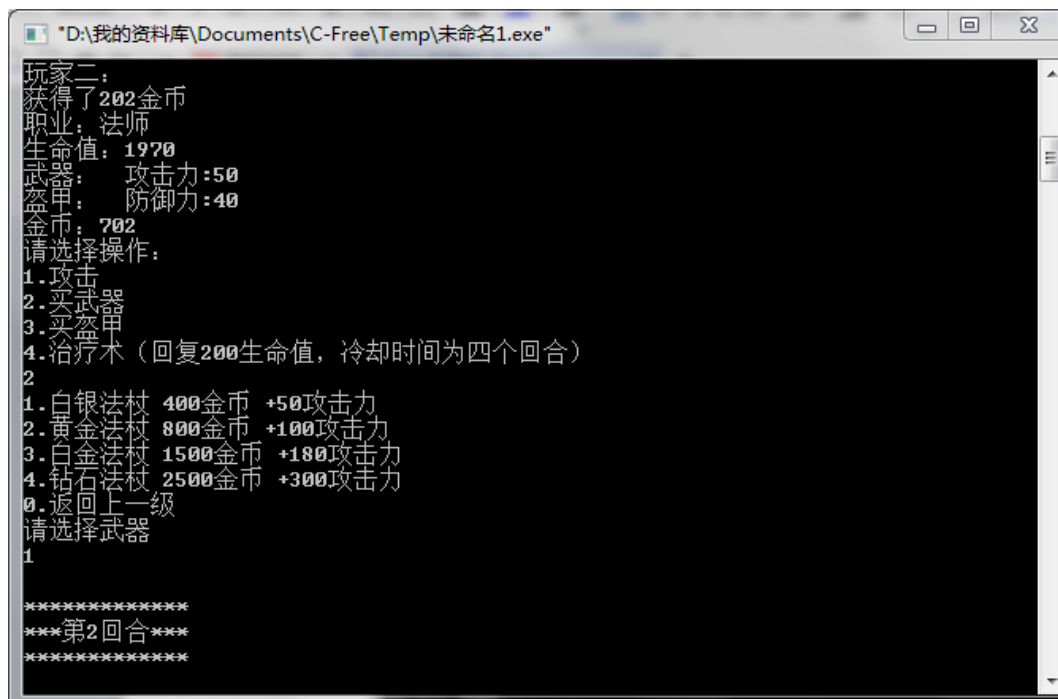
修改方案：增加了以下代码：

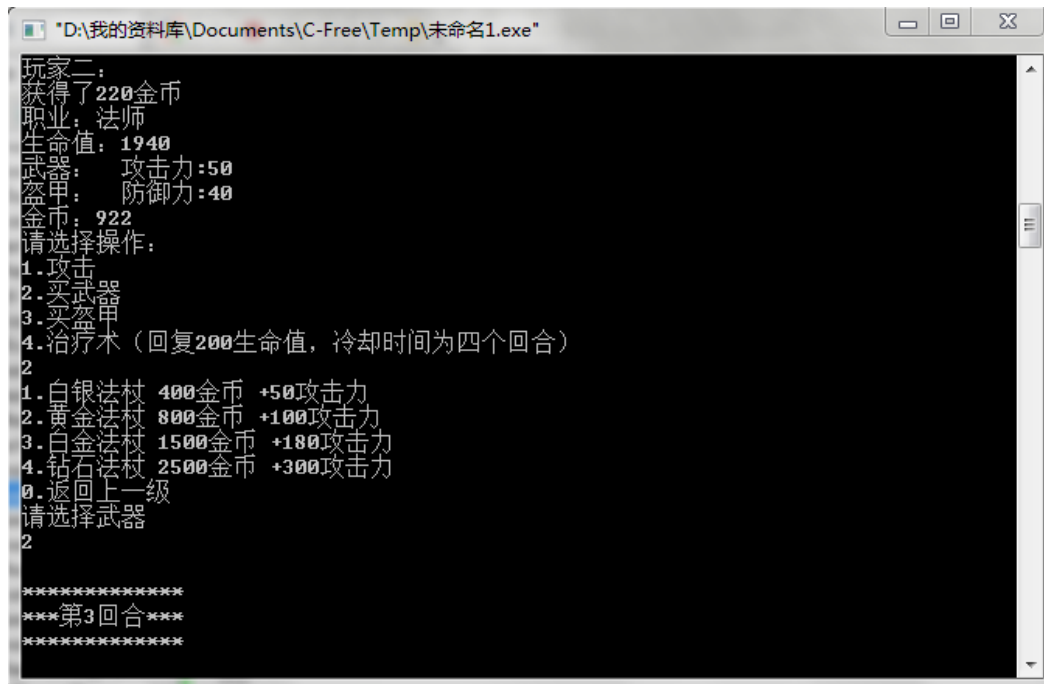
```
if(!cin)           //输入类型错误
{
    cin.clear();    //防止输入字母后程序崩溃，将之前的输入错误标识改为0，
                    //即输入无错误
    cin.sync();     //清除之前错误的数据流，以便重新输入
    cout<<"输入错误请重新输入！\n"<<endl;
    continue;
}
```

并且把角色类中的 void Show() 定义在基类中，而不是以虚函数在派生类中重新定义，简化了代码。

4.2 第二次测试

修改了第一次测试中发现的问题，随即进行了第二次调试程序，发现了一个重大问题，即玩家二可以执行购买武器的操作，但是没有成功购买武器、攻击力没有增加、金币也没有减少。





检查程序后发现存在一个 bug，修改后解决了玩家二不能购买武器的问题。

4.3 第三次测试

把前两次测试出现的问题进行修改后，该程序已大致确定下来了。对程序进行最后的调试，已无错误出现。