西北大学

面向对象程序设计

上机实验报告

学号：2020115025

班级：计科2班

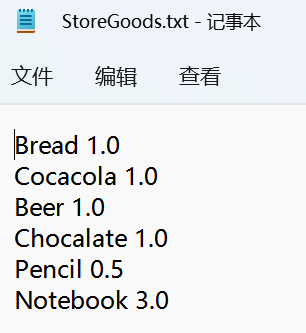
姓名：薄劲阳

日期：2022年5月20日

1. 问题分析

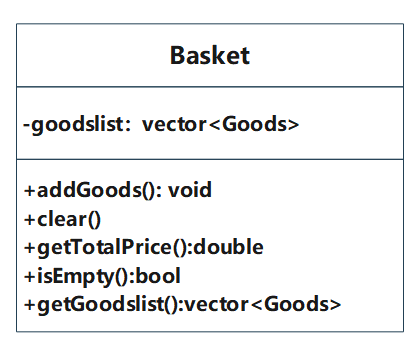
* 购买商品过程中，有三个对象
  + 商店
  + 购物清单
  + 商品

1. 问题建模和流程设计

* 文件格式
  + 
* 类设计
  + 货物类Goods



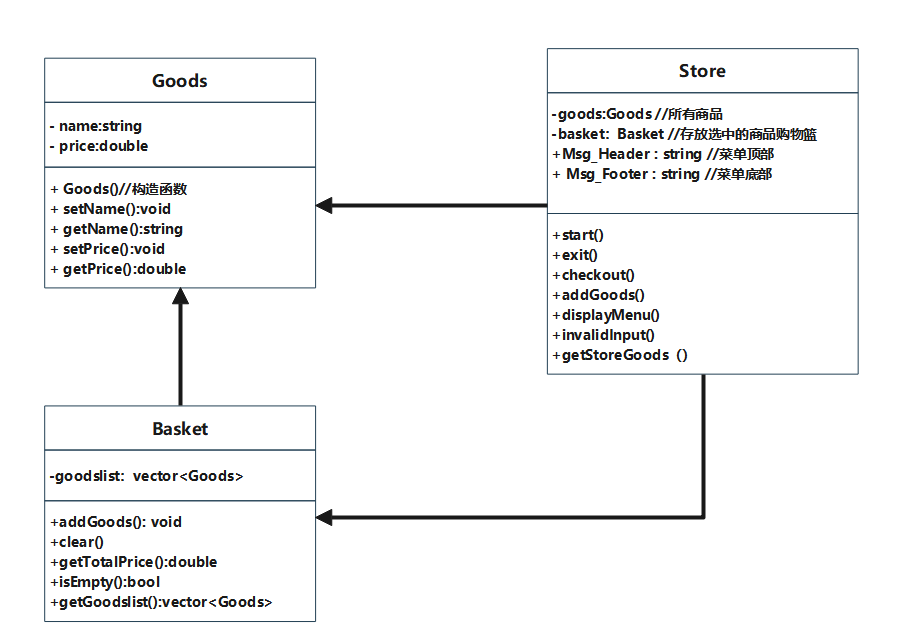
* + 购物篮类Basket



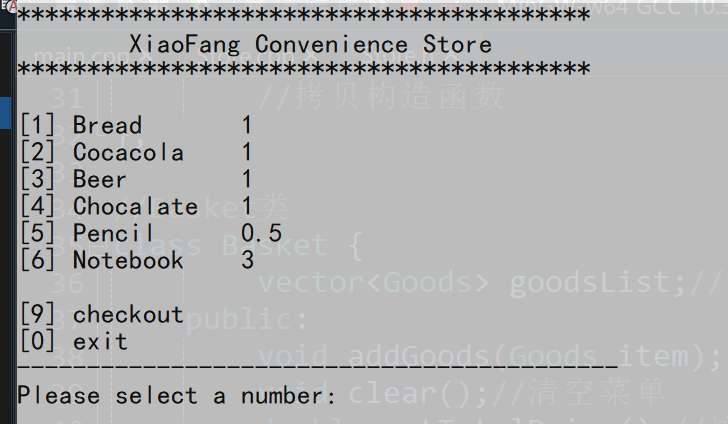
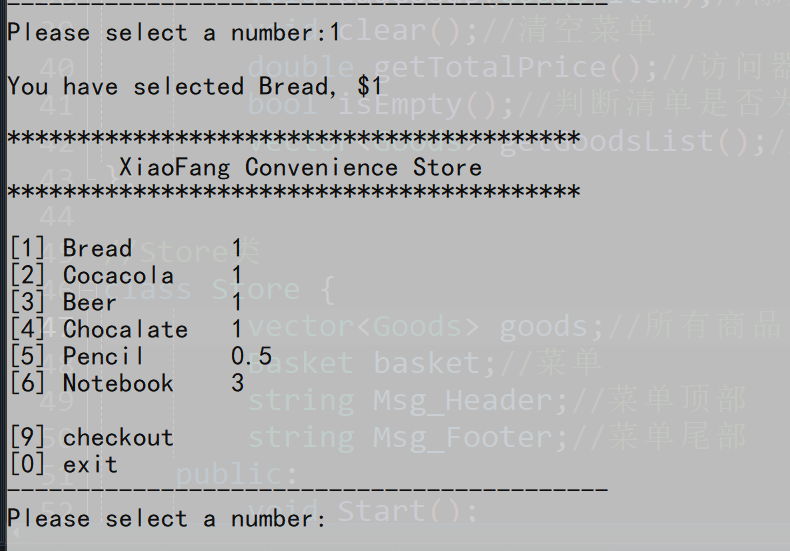
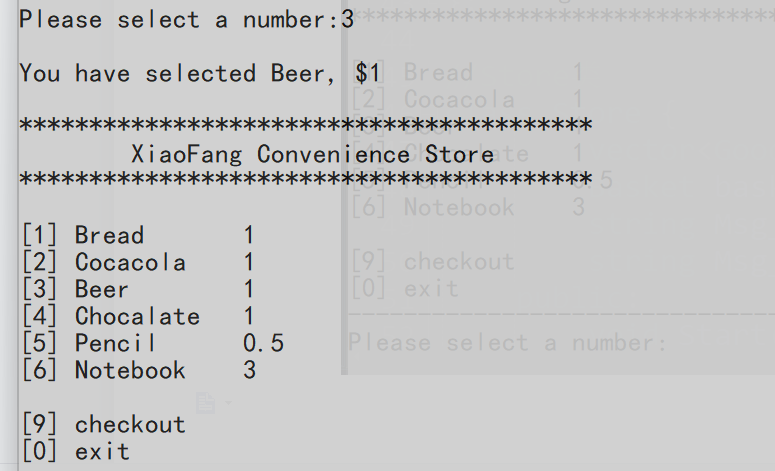
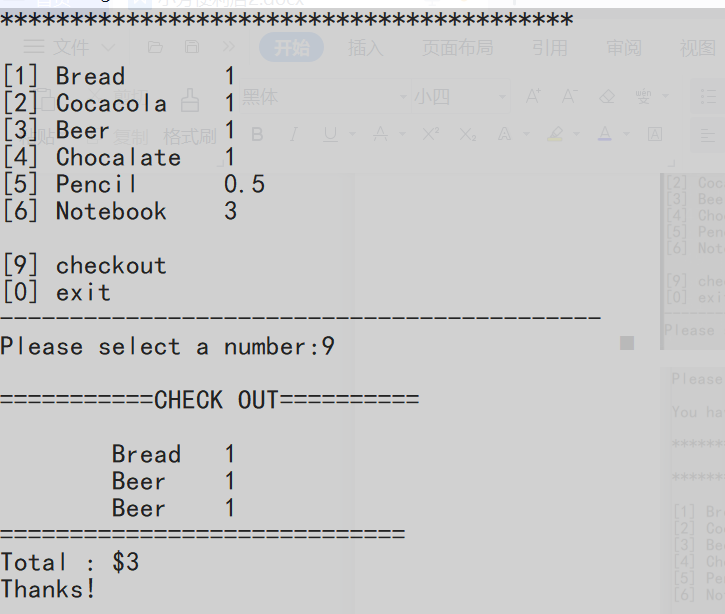
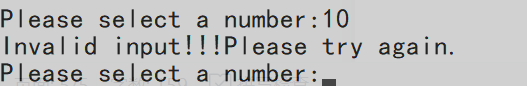
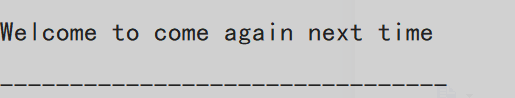
* + 小芳便利店Store



* + UML设计图
    - Store类依赖Goods类、Basket类
    - Basket类依赖Goods类



1. 程序实现和测试

* 界面展示
  + 
* 购物
  + 
  + 
* 结账
  + 
* 错误输入
  + 
* 退出
  + 

1. 实验源码

* Stores.h文件

1. #ifndef STORE
2. #define STORE
4. #include<string>
5. #include<vector>
6. **using** **namespace** std;
8. //类的声明
9. //Goods类
10. **class** Goods {
11. string name;
12. **double** price;
13. **public**:
14. //成员函数在.h文件中只声明
16. //构造函数——初始化类
17. Goods() {
18. name = "NULL";    //默认构造函数
19. price = 0;
20. }
21. Goods(string \_name, **double** \_price); //构造函数
22. Goods(**const** Goods & item);//拷贝构造函数
23. //修改器——姓名
24. **void** setName(string \_name);
25. //访问器——姓名
26. string getName();
27. //修改器——价钱
28. **void** setPrice(**double** \_price);
29. //访问器——价钱
30. **double** getPrice();
31. //拷贝构造函数
32. };
34. //Basket类
35. **class** Basket {
36. vector<Goods> goodsList;//清单表
37. **public**:
38. **void** addGoods(Goods item);//添加商品
39. **void** clear();//清空菜单
40. **double** getTotalPrice();//访问器
41. **bool** isEmpty();//判断清单是否为空
42. vector<Goods> getGoodsList();//访问器
43. };
45. //Store类
46. **class** Store {
47. vector<Goods> goods;//所有商品
48. Basket basket;//菜单
49. string Msg\_Header;//菜单顶部
50. string Msg\_Footer;//菜单尾部
51. **public**:
52. **void** Start();
53. **void** exit();
54. **void** checkout(Basket List);//确认购物结束
55. **void** addGoods(Goods item);//向商品菜单中添加商品
56. **void** displayMenu();//打印商品菜单
57. **void** invalidInput();//错误输入
58. vector<Goods> getStoreGoods();//获取商品清单
59. };
61. #endif

* Stores.cpp文件

1. #include<iostream>
2. #include<windows.h>
3. #include<string>
4. #include<vector>
5. #include"Store.h"
7. **using** **namespace** std;
9. //Goods类成员函数定义
10. //构造函数——类内部成员函数可以直接访问类内部私有成员
11. Goods::Goods(string \_name, **double** \_price) {
12. name = \_name;
13. price = \_price;
14. }
15. //拷贝构造函数——浅拷贝(不考虑指针引用啥的)
16. Goods::Goods(**const** Goods & item) {
17. name = item.name;
18. price = item.price;
19. }
20. //访问器和修改器
21. //姓名
22. **void** Goods::setName(string \_name) {
23. name = \_name;
24. }
26. string Goods::getName() {
27. **return** name;
28. }
29. //价格
30. **void** Goods::setPrice(**double** \_price) {
31. //输入检查——价格必须不小于零
32. **if** (\_price >= 0) {
33. price = \_price;
34. } **else** {
35. cout << "输入不合法" << endl;
36. }
37. }
39. **double** Goods::getPrice() {
40. **return** price;
41. }

44. //Basket类成员函数定义
45. //添加商品
46. **void** Basket::addGoods(Goods item) {
47. goodsList.push\_back(item);
48. }
50. //判断菜单是否为空
51. **bool** Basket::isEmpty() {
52. **if** (goodsList.empty()) {
53. **return** **true**;
54. } **else** {
55. **return** **false**;
56. }
57. }
58. //清空菜单
59. **void** Basket::clear() {
60. //不为空进行清空
61. **if** (!isEmpty()) {
62. goodsList.clear();
63. }
64. }
66. //获得购物清单总价
67. **double** Basket::getTotalPrice() {
68. **double** Total = 0;
70. **for** (**int** i = 0; i < goodsList.size(); i++) {
71. Total += goodsList[i].getPrice();
72. }
74. **return** Total;
75. }
77. //查看清单
78. vector<Goods> Basket::getGoodsList() {
79. **if** (!isEmpty()) {
80. **return** goodsList;
81. }
82. }
84. //Store类成员函数定义
85. //退出程序
86. **void** Store::exit() {
87. cout << endl << "Welcome to come again next time" << endl;
88. }
89. //确认购物结束
90. **void** Store::checkout(Basket List) {
91. cout << endl << "===========CHECK OUT==========" << endl << endl;
92. vector<Goods> check = List.getGoodsList();
93. //输出购物清单
94. **for** (**int** i = 0; i < check.size(); i++) {
95. cout << "\t" << check[i].getName() << "\t" << check[i].getPrice() << endl;
96. }
97. cout << "=============================" << endl;
98. //输出总价
99. cout << "Total : $" << List.getTotalPrice() << endl << "Thanks!" << endl << endl;
100. }
101. //向商品表中添加商品
102. **void** Store::addGoods(Goods item) {
103. goods.push\_back(item);
104. }
105. //菜单输出
106. **void** Store::displayMenu() {
107. //界面打印
108. SetConsoleTitle("XiaoFang Convenience Store");
109. system("color 70");
110. cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;
111. cout << "\tXiaoFang Convenience Store" << endl;
112. cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl << endl;
113. //遍历动态数组并进行输出
114. **for** (**int** i = 0; i < goods.size(); i++) {
115. cout << "[" << i + 1 << "] " << goods[i].getName() << "\t" << goods[i].getPrice() << endl;
116. }
117. cout << endl << "[9] checkout" << endl;
118. cout << "[0] exit" << endl;
119. cout << "-------------------------------------------" << endl;
120. }
121. //获取商品清单
122. vector<Goods> Store::getStoreGoods() {
123. **return** goods;//返回商品单
124. }
125. //错误输入
126. **void** Store::invalidInput() {
127. cout << "Invalid input!!!Please try again." << endl; //错误输入提示
128. }

* main.cpp文件

1. #include <iostream>
2. #include<fstream>
3. #include<sstream>
4. #include<vector>
5. #include<string>
6. #include"Store.h"
8. **using** **namespace** std;

11. //商店商品信息读入——文件读入
12. **void** StoreGoodsInput(Store &oneStore) {
13. //从文件中读取数据并存入Store对象中的goods
14. ifstream fin("StoreGoods.txt");
16. //未到达文件最后一行 一直读取
17. **while** (!fin.eof()) {
18. string buffer;
19. getline(fin, buffer); //读取一个商品的信息
20. **if** (buffer.empty()) { //如果读进来的为空行 则略过
21. **continue**;
22. }
23. istringstream is(buffer);//利用字符串流逐项提取学生的数据——利用空格进行划分
24. string Gname;
25. **double** Gprice;
26. is >> Gname >> Gprice;
27. Goods one(Gname, Gprice);
28. oneStore.addGoods(one);//向商店的商品清单中添加商品
29. }
31. fin.close();
32. }
34. //判断输入的号码为哪个商品
35. Goods SearchBuyGood(Store & oneStore, **int** no) {
36. vector<Goods> StoreGoods = oneStore.getStoreGoods();
38. **for** (**int** i = 0; i < StoreGoods.size(); i++) {
39. **if** (i + 1 == no) {
40. **return** StoreGoods[i];
41. }
42. }
43. }
45. **int** main(**int** argc, **char**\*\* argv) {
47. Store XiaoFang;//定义一个小芳便利店
48. Basket List\_XiaoFang;//定义小芳便利店的清单
49. vector<Goods> GoodsSheet;
51. //读取商店菜单
52. StoreGoodsInput(XiaoFang);
53. //购物
54. Goods buy;
55. **while** (1) {
56. XiaoFang.displayMenu();
57. loop:
58. cout << "Please select a number:";
59. **int** n;
60. cin >> n;
61. **switch** (n) {
62. **case** 0:
63. XiaoFang.exit();
64. exit(0);
65. **case** 1:
66. buy = SearchBuyGood(XiaoFang, 1);
67. cout << endl << "You have selected " << buy.getName() << ", $" << buy.getPrice() << endl << endl;
68. List\_XiaoFang.addGoods(buy);
69. **break**;
70. **case** 2:
71. buy = SearchBuyGood(XiaoFang, 2);
72. cout << endl << "You have selected " << buy.getName() << ", $" << buy.getPrice() << endl << endl;
73. List\_XiaoFang.addGoods(buy);
74. **break**;
75. **case** 3:
76. buy = SearchBuyGood(XiaoFang, 3);
77. cout << endl << "You have selected " << buy.getName() << ", $" << buy.getPrice() << endl << endl;
78. List\_XiaoFang.addGoods(buy);
79. **break**;
80. **case** 4:
81. buy = SearchBuyGood(XiaoFang, 4);
82. cout << endl << "You have selected " << buy.getName() << ", $" << buy.getPrice() << endl << endl;
83. List\_XiaoFang.addGoods(buy);
84. **break**;
85. **case** 5:
86. buy = SearchBuyGood(XiaoFang, 5);
87. cout << endl << "You have selected " << buy.getName() << ", $" << buy.getPrice() << endl;
88. List\_XiaoFang.addGoods(buy);
89. **break**;
90. **case** 6:
91. buy = SearchBuyGood(XiaoFang, 6);
92. cout << endl << "You have selected " << buy.getName() << ", $" << buy.getPrice() << endl;
93. List\_XiaoFang.addGoods(buy);
94. **break**;
95. **case** 9:
96. XiaoFang.checkout(List\_XiaoFang);
97. **break**;
98. **default**:
99. XiaoFang.invalidInput();
100. **goto** loop;//输入错误 跳转到loop处 重新输入
101. }
102. }



107. **return** 0;
108. }