**Step 1: Case Study Scenario**

Create a Class **MenuItem,** which has three instances

1. **name:** name of the item

**2. type:** whether *food* or a *drink*

**3. price:** price of the item

Write a class called **CoffeeShop**, which has three instance variables:

1. **name** : a string (basically, of the shop)

2. **menu** : an list of items (of object type), with each item containing the item (name of the item),

type (whether *food* or a *drink*) and price.

3. **orders** : an empty list of string type.

And a parameterized constructor which takes the name of the CoffeeShop as a parameter.

and eight methods:

1. **addMenuItem:** adds the menu item in the list of menu

2. **addOrder:** adds the name of the item to the end of the orders list if it exists on the menu.

Otherwise, return "This item is currently unavailable!"

3. **fulfillOrder:** if the orders list is not empty, return "The {item} is ready!" and make the list empty.

If the order list is empty, return "All orders have been fulfilled!"

4. **listOrders:** returns the list of orders taken, otherwise null.

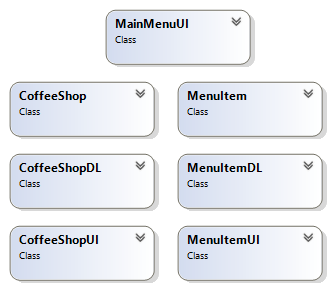
5. **dueAmount:** returns the total amount due for the orders taken.

6. **cheapestItem:** returns the name of the cheapest item on the menu.

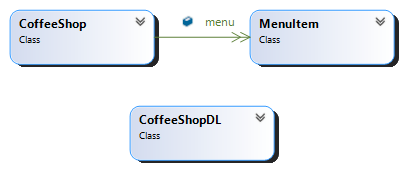
7. **drinksOnly:** returns only the *item* names of *type* drink from the menu.

8. **foodOnly:** returns only the *item* names of *type* food from the menu.

IMPORTANT: Orders are fulfilled in a FIFO (first-in, first-out) order.

**Step 2: Domain Model with Only the Class Names**

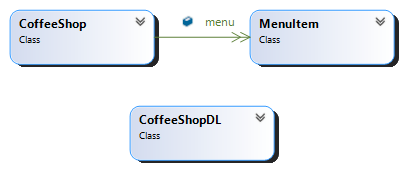
**Step 3: Domain Model with Relations and Constraints**



Containmss

**Step 4: Domain Model with Multiplicity**

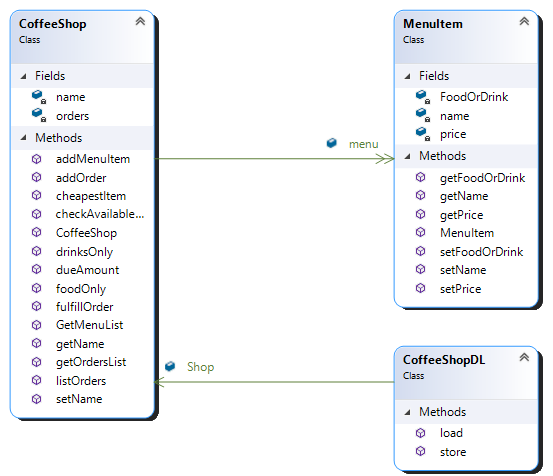
Containmss



1

∞

**Step 5: Class Diagram Attributes and Functions**



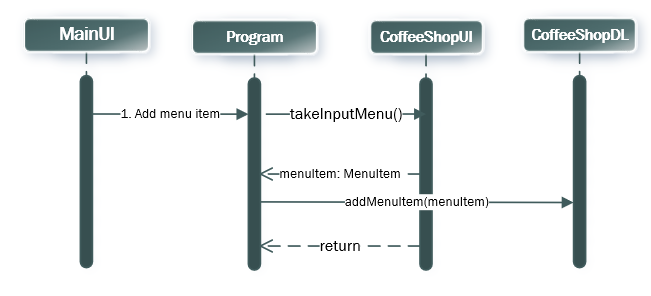
Containmss

∞

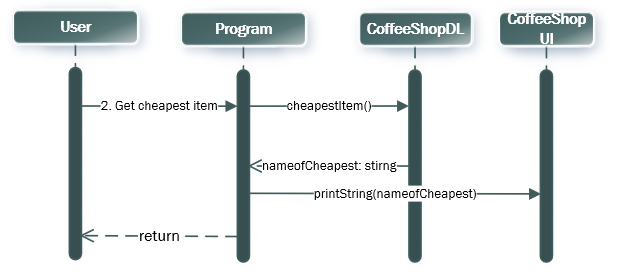
1

**Step 6: Sequence Diagram**

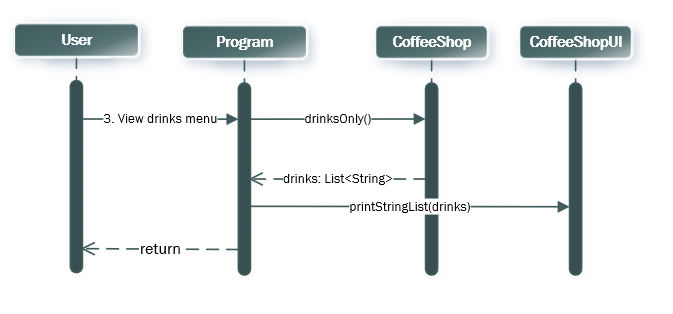
**Option 1: Add New Item:**

****

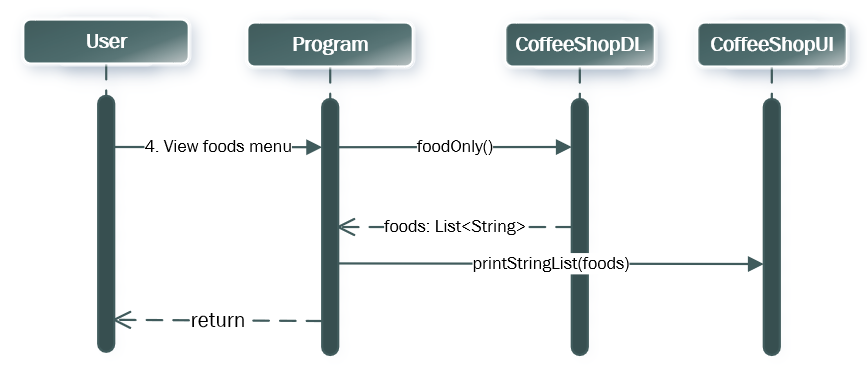
**Option 2: View the Cheapest item in the menu:**

****

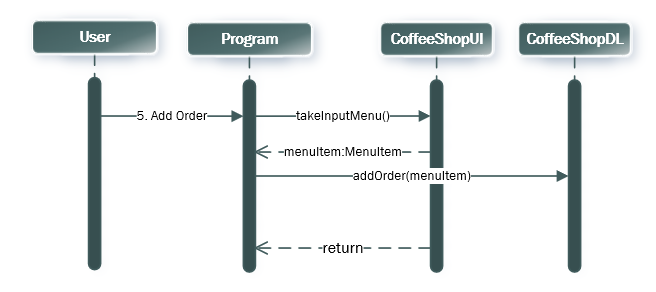
**Option 3: View the Drink’s Menu**:

****

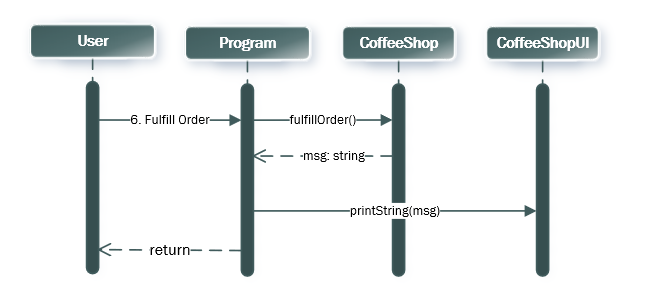
**Option 4: View the Food’s Menu:**

****

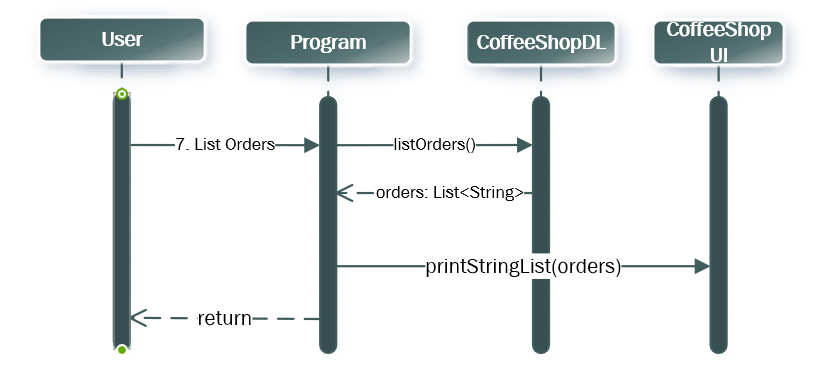
**Option 5: Add Order:**

****

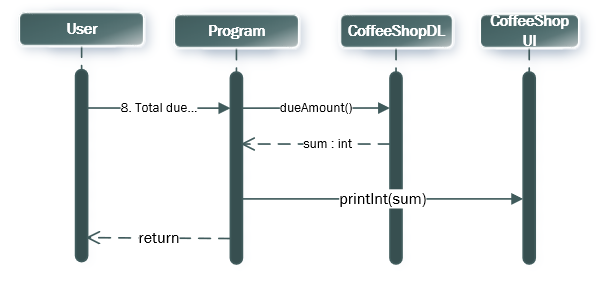
**Option 6: FulFill Order:**

****

**Option 7: View the Orders List:**

****

**Option 8: Total Payable Amount:**

****

**BL Code: CoffeeShop.cs**

class CoffeeShop

{

private string name;

public List<MenuItem> menu = new List<MenuItem>();

private List<string> orders = new List<string>();

public List<MenuItem> GetMenuList()

{

return menu;

}

public List<string> getOrdersList()

{

return orders;

}

public string getName()

{

return name;

}

public void setName(string name)

{

this.name = name;

}

public CoffeeShop(string name)

{

this.name = name;

}

public void addMenuItem(MenuItem item)

{

menu.Add(item);

}

public bool addOrder(string order)

{

bool check = checkAvailableOnMenu(order);

if (check == true)

{

orders.Add(order);

return true;

}

return false;

}

public bool checkAvailableOnMenu(string order)

{

foreach(MenuItem item in menu)

{

if (order == item.getName()) {

return true;

}

}

return false;

}

public string fulfillOrder()

{

if (orders.Count != 0)

{

string s = $"The {orders[0]} is Ready";

orders.Remove(orders[0]);

return s;

}

else {

return "All orders have been fulfilled";

}

}

public List<string> listOrders() {

if(orders.Count > 0)

{

return orders;

}

return null;

}

public int dueAmount()

{

int sum = 0;

foreach (MenuItem item in menu)

{

sum = sum + item.getPrice();

}

return sum;

}

public string cheapestItem()

{

int cheapest = menu[0].getPrice();

string nameOfCheapest = menu[0].getName();

for(int i = 0; i < menu.Count; i++)

{

if(menu[i].getPrice() < cheapest)

{

cheapest = menu[i].getPrice();

nameOfCheapest = menu[i].getName();

}

}

return nameOfCheapest;

}

public List<string> drinksOnly()

{

List<string> drinks = new List<string>();

foreach (MenuItem item in menu)

{

if(item.getFoodOrDrink() == "drink")

{

drinks.Add(item.getName());

}

}

return drinks;

}

public List<string> foodOnly()

{

List<string> food = new List<string>();

foreach (MenuItem item in menu)

{

if (item.getFoodOrDrink() == "food")

{

food.Add(item.getName());

}

}

return food;

}

}

**BL Code: ship.cs**

**DL Code: shipDL.cs**

**UI Code: shipUI.cs**

**BL Code: MenuItem.cs**

class MenuItem

{

private string name;

private string FoodOrDrink;

private int price;

public MenuItem(string name,string FoodorDrink,int price)

{

this.name = name;

this.FoodOrDrink = FoodorDrink;

this.price = price;

}

public string getName() {

return name;

}

public string getFoodOrDrink()

{

return FoodOrDrink;

}

public int getPrice()

{

return price;

}

public void setName(string name)

{

this.name = name;

}

public void setFoodOrDrink(string FoodOrDrink)

{

this.FoodOrDrink = FoodOrDrink;

}

public void setPrice(int price)

{

this.price = price;

}

}

**DL Code: CoffeeShopDL.cs**

class CoffeeShopDL

{

public static CoffeeShop Shop = new CoffeeShop("Tesha's");

public static void store()

{

StreamWriter file = new StreamWriter("Coffee.txt");

file.WriteLine(Shop.getName());

foreach(var v in Shop.GetMenuList())

file.WriteLine(v.getName() + ','+v.getPrice() + ',' + v.getFoodOrDrink());

file.Flush();

file.Close();

}

public static void load()

{

StreamReader file = new StreamReader("Coffee.txt");

Shop.setName(file.ReadLine());

string r;

while((r = file.ReadLine()) != null)

{

string[] line = r.Split(',');

Shop.menu.Add(new MenuItem(line[0], line[2], int.Parse(line[1])));

}

file.Close();

}

}

**U**

**UI Code: CoffeeShopUI.cs**

class CoffeeShopUI

    {

        public static CoffeeShop myCoffeeShop = new CoffeeShop("Alpha");

        public static void loadItemsFromFile ()

        {

            MenuItemDL.loadFromFile("MenuItem.txt" , myCoffeeShop);

        }

        public static void loadShopFromFile ()

        {

            CoffeeShopDL.loadFromFile("CoffeeShop.txt" , myCoffeeShop);

        }

        public static void storeShopIntoFile ()

        {

**UI Code: CoffeeShopUI.cs**

class CoffeeShopUI

{

public static MenuItem takeInputMenu()

{

Console.WriteLine("Enter name of Item:");

string name = Console.ReadLine();

Console.WriteLine("Enter Type:");

string type = Console.ReadLine();

while(type != "food" && type != "drink")

{

Console.WriteLine("Enter Type:");

type = Console.ReadLine();

}

Console.WriteLine("Enter Price");

int price = int.Parse(Console.ReadLine());

MenuItem menuItem = new MenuItem(name,type,price);

return menuItem;

}

public static string TakeOrder()

{

Console.WriteLine("Enter item you want to order:");

string order = Console.ReadLine();

return order;

}

public static void printStringList(List<string> ListToPrint)

{

foreach(string v in ListToPrint)

{

Console.WriteLine(v);

}

}

public static void printIntValue(int intValue)

{

Console.WriteLine(intValue);

}

public static void printStringValue(string stringValue)

{

Console.WriteLine(stringValue);

}

}

**UI Code: MainUI.cs**

class MainUI

{

public static int menu()

{

int option=0;

Console.WriteLine("Welcome to the Tesha’s Coffee Shop\n\n");

Console.WriteLine("1. Add a Menu item");

Console.WriteLine("2. View the Cheapest Item in the menu");

Console.WriteLine("3. View the Drink’s Menu");

Console.WriteLine("4. View the Food’s Menu");

Console.WriteLine("5. Add Order");

Console.WriteLine("6. Fulfill the Order");

Console.WriteLine("7. View the Orders’s List");

Console.WriteLine("8. Total Payable Amount");

Console.WriteLine("9. Exit");

option = int.Parse(Console.ReadLine());

return option;

}

**Driver Program: Program.cs**

class Program

{

static void Main(string[] args)

{

int option=0;

do

{

option = MainUI.menu();

if (option == 1)

{

CoffeeShopDL.Shop.addMenuItem(CoffeeShopUI.takeInputMenu());

}

else if (option == 2)

{

CoffeeShopUI.printStringValue(CoffeeShopDL.Shop.cheapestItem());

}

else if (option == 3)

{

CoffeeShopUI.printStringList(CoffeeShopDL.Shop.drinksOnly());

}

else if (option == 4)

{

CoffeeShopUI.printStringList(CoffeeShopDL.Shop.foodOnly());

}

else if (option == 5)

{

CoffeeShopDL.Shop.addOrder(CoffeeShopUI.TakeOrder());

}

else if (option == 6)

{

CoffeeShopUI.printStringValue(CoffeeShopDL.Shop.fulfillOrder());

}

else if (option == 7)

{

CoffeeShopUI.printStringList(CoffeeShopDL.Shop.listOrders());

}

else if (option == 8)

{

CoffeeShopUI.printIntValue(CoffeeShopDL.Shop.dueAmount());

}

}

while (option != 9);

}

}