Kevin Jodrie-Black

CS210

Project Three

Grocery List Application Documentation

This program is designed to manage a grocery inventory list. It allows users to:

1. Search for items by name.
2. View the full list of items with their quantities.
3. Display the list as a histogram.
4. Save the list to a file.

The system is divided into three main classes:

1. **GroceryItem**: Represents a single grocery item.
2. **GroceryItemList**: Manages a list of grocery items.
3. **UserMenu**: Provides a user interface for interacting with the grocery list.

### **1. GroceryItem Class**

#### **Purpose**

Represents a single grocery item with a name and quantity.

#### **Attributes**

* m\_itemName: Stores the name of the grocery item.
* m\_itemQuantity: Stores the quantity of the grocery item.

#### **Methods**

* **GroceryItem()**

Default constructor. Initializes an item with no name and a default quantity of 1.

* **GroceryItem(string t\_itemName)**

Parameterized constructor. Initializes an item with the given name and a default quantity of 1.

* **string GetItemName() const**

Returns the name of the grocery item.

* **void IncrementQuantity()**

Increases the quantity of the item by 1.

* **int GetItemQuantity() const**

Returns the quantity of the item.

### **2. GroceryItemList Class**

#### **Purpose**

Manages a list of GroceryItem objects and provides functionality to manipulate and display the list.

#### **Attributes**

* m\_groceryItemArray: A vector that stores all GroceryItem objects.

#### **Methods**

* **void ReceiveItemInput(const string t\_itemName)**

Adds a new item to the list or increments the quantity of an existing item if it already exists.

* **void PrintFullGroceryList()**

Prints the name and quantity of all items in the list.

* **void PrintListHistogram()**

Prints the name of each item followed by a histogram (using \*) representing its quantity.

* **void PrintItemInfo(const GroceryItem& t\_groceryItem)**

Prints the name and quantity of a specific item.

* **void FindItemByName(const string t\_name)**

Searches for an item by name and prints its details if found. If not found, informs the user.

* **void SaveItemListToFile()**

Saves the list of items and their quantities to a file named frequency.txt.

* **void CreateGroceryItem(const string t\_itemName)**

Creates a new GroceryItem and adds it to the list.

### **3. UserMenu Class**

#### **Purpose**

Provides a user interface for interacting with the GroceryItemList.

#### **Methods**

* **void RunUserMenu(GroceryItemList& t\_itemList)**

Starts the user menu system by opening the main menu.

* **string GetStringFromInput(string t\_message)**

Prompts the user for a string input and returns it.

* **int GetIntFromInput(string t\_message, int t\_minRange, int t\_maxRange)**

Prompts the user for an integer input within a specified range. Validates the input and handles errors.

* **void OpenMainMenu(GroceryItemList& t\_itemList)**

Displays the main menu options and handles user choices:

* + Search for an item by name.
  + Print the full grocery list with numeric quantities.
  + Print the grocery list as a histogram.
  + Exit the program.
* **void OpenNameSearchMenu(GroceryItemList& t\_itemList)**

Prompts the user to enter an item name and searches for it in the list.

### **4. Main Program**

#### **Purpose**

The entry point of the program. It initializes the grocery list, reads data from a file, and starts the user menu.

#### **Functions**

* **static void ReadInfoFromFile()**

Reads grocery item data from a file (CS210\_Project\_Three\_Input\_File.txt) and adds it to the GroceryItemList.

* **int main()**

The main function:

* + Calls ReadInfoFromFile() to load data from the input file.
  + Saves the list to an output file using SaveItemListToFile().
  + Starts the user menu system using RunUserMenu().
  + Displays a message when the program exits.