Haley Candia Perez

Apr 21, 2024

CS 210

Project 3 Corner Grocer

This project reflected my implementation of a menu-driven program in C++ that effectively manages grocery item frequencies for Corner Grocer. The way that this code has been designed successfully reflects my ability to utilize good programming practices while also maintaining proper coding structure.

This program allows the company to see how often items are purchased, which facilitates the ability to organize their produce section in the most effective way. When the program received an input file of the items purchased during a specific day, it is able to produce a list of all items along with the number of times that product was purchased. Then, the program produces a histogram that lists all items that were purchased that specific day with a representation of the number of times the item was purchased.

Here, I used C++ to make a program that is maintainable, adaptable, and uses industry standard best practices. I also laid out my code in a manner that would be readable. I used a combination of C++ and Python in this project to take advantage of Python's advanced ability to analyze data, and then uses C++ to output the file back to the user. This creates a program that is most beneficial to the client.

Screenshot from CornerGrocer.cpp of histogram:

```cpp
// Function to display histogram of items
void histogram()
{
    // Calls Python function to read input file and create a new one
    CallProcedure("read_and_create_file");

    // Open file to read
    newFileFS.open("frequency.dat");

    // Check to see if file opened successfully
    if (!newFileFS.is_open())
    {
        cout << "Unable to open file." << endl;
    }

    newFileFS >> item;
    newFileFS >> itemQuantity;
    while (!newFileFS.fail())
    {
        cout << item << " " << nCharString(itemQuantity, '*') << endl;
        newFileFS >> item;
        newFileFS >> itemQuantity;
    }

    // Close the file
    newFileFS.close();

    cout << endl;
}
```

Screenshot from CornerGrocer.cpp of functions to obtain items purchased and number of items:

```cpp
277
278      // Function to get list of items purchased and number purchased for each
279      void itemAndQuantityList()
280      {
281          cout << "*************************************************************************" << endl;
282          cout << "                    Items purchased and quantity of each item" << endl;
283          cout << "*************************************************************************" << endl << endl;
284
285          // Calls Python function to read and display list of items and quantity of each purchased
286          CallProcedure("read_file");
287      }
288
289      // Function to get number of purchases for a certain item obtained from user input
290      void specificItemQuantity()
291      {
292          cin >> userInput;
293          cout << endl;
294
295          // Calls Python function to return number of times specific item was bought
296          quantity = callIntFunc("specific_item_quantity", userInput);
297
298          // If function returns -1, then item was not in list
299          if (quantity == -1)
300          {
301              cout << "Item not found in file or entered incorrectly." << endl;
302          }
303          else
304          {
305              cout << "Total of " << quantity << " " << userInput << " sold today." << endl << endl;
306          }
307      }
308
```

Screenshots of PythonCode.py file:

```python
# Python function to read and display list of items and quantity of each purchased
def read_file():
    # open and read input file, split into words, then get unique products and sort them
    contents = open('CS210_Project_Three_Input_File.txt')
    product_list = contents.read().split()
    products = sorted(set(product_list))

    # Display each product and its quantity
    for word in products:
        print(word + ":", product_list.count(word))
        print()

    # Close input file
    contents.close()
```

```python
# Python function to return number of times a specific item was purchased    ⚠2 ⚠5 ✗1 ⌄
def specific_item_quantity(v):
    # Opens and reads file and sorts items into list
    contents = open('CS210_Project_Three_Input_File.txt', 'r')
    product_list = contents.read().split()
    products = sorted(set(product_list))

    quantity = 0
    num_items_total = 0
    num_items = 0

    # Count total number of items in list
    for word in products:
        num_items_total += 1

    # Obtain number of times item was purchased, if at all
    for word in products:
        temp_word = word
        num_items += 1

        if (temp_word == v.capitalize()):
            quantity = product_list.count(word)
            num_items -= 1
        if (temp_word != v and num_items == num_items_total):
            return -1

    return quantity
```

```python
# Python function to read input file and create a new one
def read_and_create_file():
    # Reads file and sorts items into list
    contents = open('CS210_Project_Three_Input_File.txt', 'r')
    product_list = contents.read().split()
    products = sorted(set(product_list))

    # Open input file
    new_file = open('frequency.dat', 'w')

    # Write data to new file as product and number purchased
    for word in products:
        new_file.write(word)
        new_file.write(" ")
        new_file.write(str(product_list.count(word)))
        new_file.write("\n")

    # Close both input and output files
    contents.close()
    new_file.close()

    # Return output file object
    return new_file
```