Assignment 2: Decomposing a Programming Problem

Hide Assignment Information

Instructions

Decomposing a Programming Problem

In this assignment, you will gain experience planning the structure of your program solution given a real-world problem statement and application. For th part of this assignment, you will be planning, and then describing the structure of a program that solves the problem outlined below. At this stage, you w writing the code - just planning. In the second part of the assignment, you will write the code for your program based on the plan that you developed in t part.

Problem Statement: In this assignment, your task is to build a console version of a municipal call center records system. Each day, the City of Springfield hundreds of calls from customers - some are compliments and some are complaints. In the past, they've used paper to keep track of this information but skyrocketing prices of paper (not to mention the environmental costs!), the municipality has hired you to build them a computer system that will keep tra calls and retrieve information from the database.

Modular Design: In this assignment, you will be designing the structure of and writing your program by using modules to divide up the functionality of th program into separate units. A program that is not divided into modules will receive minimal marks.

The Data: You are provided with a data file which provides a log of some of the correspondence received by the City from the public. Each row of the dat correspondence artifact (i.e. email, phone call, social media message). The dataset is not an exhaustive list of all of the calls that they have received, but r a generous sampling of a variety of their correspondence logs. The columns (or features) of the dataset are as follows:

- identification number (5-digit integer) - the call's unique identifier.
- department (string) - the department responsible for following up on the call.
- priority (int) - an integer in the range of 1 to 5 that denotes the priority (urgency) of the call. 1 is least priority and 5 is highest priority.
- call type (string) - a complaint or positive feedback.
- resolution (boolean) - True if the issue has been resolved, False otherwise.
- communication feed (string) - whether the correspondence was received by phone call, email, social media.

Functionality of the Software: Your teammates consulted the City of Springfield to perform stakeholder analysis to determine what features and functio each stakeholder would like to see included in the final product. Here are the notes of what they discovered:

1. **Input of Data:** Your program will initially read the data from the comma-separated (CSV) file into an appropriate data structure (maybe a list of lists of dictionaries? The choice is up to you as the programmer). The filename will always be the same. This means that the file name for the CSV file c hard-coded into your program. All updates are made to the data structure - not to the CSV file. Check out this resource for tips on reading a CSV f Python program.
2. **Create New Record:** Your program should allow the user to add a new call to the database. When a new call is added to the database, remember t adding it to the data structure (i.e. the list) that you chose to use to organize the information. The user should be prompted to input all required information: identification number, department, priority level, call type, resolution status and communication feed.

3. **Retrieve and Display Records:** The user should be able to search for records by the call identification number, department, priority level, call type (complaint or compliment), or resolution status. Upon finding one or more matching records, all information for those record(s) should be printed tc console. Here's an overview of the required retrieval and display operations:
    - There should be an option to display all of the information for all calls in the system - no filtering based on the attributes.
    - There should be an option to list all calls by a given identification number, department, priority level, call type, resolution status, or commun method..
        - For example, if the user elects to search by "department" and enters "Roads", **all** of the information (identification number, departmer priority, call type) for each of the calls associated with the roads department would be printed to the console.
4. **Update Records:** Your program should allow the user to make the following changes to the vehicle information. First, the user should be prompted unique identification number for the call that they would like to update. Then, the user is prompted to indicate which field (or attribute) they woul update by entering the new information.
    - update the department name associated with the call
    - update the priority level.
    - update the call type.
    - update the resolution type.
    - update the communication method.

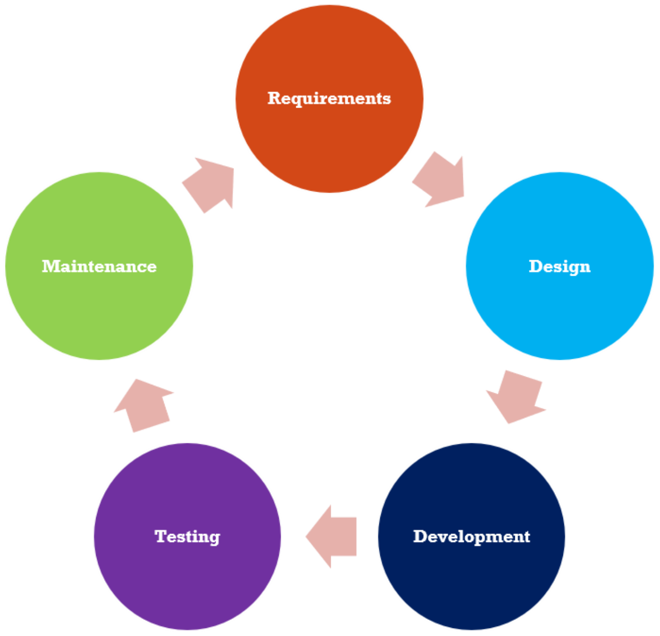
    Note: Your program should reflect any changes related to the call information directly within the data structure that you cho: use. Your program is not required to re-write to the CSV after every update or addition to the data structure. You are welcon add this feature as an additional (and worthwhile!) practice exercise.
5. **Remove Records:** Given a unique call identification number, the program should remove the entire record from the data structure for the call assoc the identification number. The call would remain in the CSV file.
6. **Output:** The user should have a choice to write the call listings to a text file. If the user chooses to have the information written to a text file, your should prompt the user for the name of the file to which they wish to "write" the information. If the user requests to write the call listings to a tex updates (edits, additions, or removals) that were made to the data structure since the program started and the data was initially read into the data will now be reflected in the text file.

Input Checking: When the user enters the information for a new record, your program should verify that the values are legal. If any user input is invalid, p print a meaningful error message and re-prompt the user for a valid entry.

- The identification number is numeric (no other characters) and consists of only 5 digits.
- The department type is one of: "ROADS", "BYLAW", "WASTE", "UTILITIES", "PARKING", "RECREATION", "FIRE".
- The priority level should be an integer between 1 and 5 (inclusive).
- The call type is one of the following values: "COMPLAINT" or "COMPLIMENT".
- The communication channel should be one of the following values: "PHONE", "EMAIL", "SOCIALMEDIA"
- The resolution status is one of the following values: "RESOLVED" or "UNRESOLVED".
    - *Note that the information for department type, call type, resolution status or communication feed should be case inse the user should be able to enter "ROADS" or "roads" or "Roads".*
- The identification number for the calls should be unique. When the user attempts to ad a new call to the database, your program should ensure th identification number that they entered does not already exist in the database. If it does, print a meaningful message to the user identifying the iss prompt the user for a new identification number.

The Requirements Process

In the real-world, software engineers are developing products for "stakeholders". For example, if you are developing software for a bank, the managers of would be your "stakeholders". It is often the case that the problem statement given to you (the developer) by the stakeholders will have some ambiguous The Software Design process is circular. You begin by collecting **requirements** (specification) information from the stakeholders. The specification that th stakeholder would provide you tells you what the software should be able to do. You would collect this information through meeting with the stakeholde as by reading through specifications documents provided by the stakeholders. Once you have a clear picture of what the software should do, you move i **design phase**. As you begin designing the software, you may need to seek clarification from the stakeholders regarding the software's requirements (i.e. h software should be used or what functions it should perform). This is why the Software Design Process is circular. Even after the program is developed a the requirements stage is revisited in order to make sure that the final product meets the requirements of the stakeholder team. Your focus for the first p assignment is on the **requirements and design stages.**



For this assignment, the professor will be your stakeholder/customer. I have opened up a discussion forum called "Stakeholder Questions". In this forum, ask me questions to clarify any of the requirements for the software.

Deliverables:

Task # 1 - Presentation: You are to create a presentation to explain the design/plan for this program. You should show the modules that you would create each module a name) and list the functions (or "tasks") that would belong in each module. For each function, you will include the name of the function, pr brief comment for the function describing its purpose, show the parameters that the function will take and indicate what the function returns. There sho NO code in the presentation. You are to present only your design/plan.

There are many ways that you can present this information - you may choose to record a narrated PowerPoint, create an audio recording or create a non diagram that accurately depicts your program's design. However you choose to present your information, you must submit an explanation of how you we translating the problem statement into a design (or plan) of the program, as well as a document (or a PowerPoint) similar to the word cloud step-wise refi document.

Here is an example of what we are looking for (modeling an incomplete calculator application): Calculator Example

---

Task # 2 - Code: You are to write the code for the program that you planned out in the first part of the assignment.

Your program must be written in a modular format. You may, for example, choose to have 4 modules; one for your user interface (this one will only really your main() method), one for file handling (such as reading the data from the text file and writing new information to the text file), one for the commands updating call information and printing the information to the console) and a module for the database functionality (such as retrieving information for a sp call).

You will notice that there are no names specified for the functions that you write. The TAs will run your program from the console when they are marki work -- be sure that the user interface is user-friendly and intuitive for them to understand how to use your program.

What Do I Hand In? Please ensure that you submit two compressed folders (i.e. a "zip" folder) - each of the compressed folders should contain the files a with each part of the assignment (plan and code implementation):

- Part 1: Submit the "plan" for your solution.
- Part 2: Submit the ".py" files for the modules that you created.

Click here to access the "Stakeholder Questions" forum.

Due Date

Jul 26, 2021 11:30 PM

Attachments

CallLogs.csv (3.87 KB)

Download All Files

Hide Rubrics

Rubric Name: Decomposing a Programming Problem - Assignment 2

Oops! We're having trouble connecting you. You might want to refresh the page, or try again later. **refresh the page**

Submit Assignment

Files to submit *

(0) file(s) to submit

After uploading, you must click Submit to complete the submission.

Add a File Record Audio Record Video

Comments

Paragraph Font Famil Font Size

Submit Cancel