Sarah walker

Homework Survey Program

5/5/15

## **Functional Overview**

My project will be a Java version of the homework survey. It will run as a command line program, but hopefully next year I will be able to expand on it and make it into a web app or another usable program. The intended audience of this program is Lakeside students. I, along with many other students, felt that the homework survey we took in March did not reflect well the actual homework trends as that was a light week, and the questions to try to factor that in (“Is this more or less than your typical amount of homework?”) were very subjective. This will hopefully give a more concrete picture of a Lakeside student’s homework.

## **Design Overview**

Describe your design - What will the program run like? What are the expected inputs? How will the user interact with the program? Do you have any sketches of what you think this will look like?

The program will collect a month’s worth of data for the student. The student will initially give how many classes they are taking so I know how much memory to allocate for the data. The student will then each night input the amount of hours they spent doing homework for each class. The student will then have the option to generate up to three graphs: one of the hours they spent that night, one of the current week, and one of that month up until that point. In order to make sure the student inputs the correct data, the program will double check with the user that the number of hours they just put in is the correct value. At the end of the month, the user will have the option to print out all of their data for the month in a nicely formatted table.

## **Design Details**

2 Files: HomeworkSurveyClient and Day Class

Public Class HomeworkSurveyClient

{

Public static Day[] thisMonth;

//String array with the names of the classes

//main method

//call other methods

Public static void setup

Ask user for how many classes they are taking

Use AskUserInt method (from previous programs)

Make sure that the user gives an int and that it is between 1 and 8

Instantiate the string array to be of that length

For that number of classes

Ask the user what the name of the class is and store it in the String array

Ask the user what month it is

Instantiate thisMonth with the number of days that this month has

Public static void addTodaysHours

Ask the user for the day of the month

Make sure it is within the size of the month

For the number of classes the student is taking

Use askUserInt method to ask the student for the number of hours they spent doing

homework for that class

make sure that the user gives an int between 0 and 12 hours

add that number to that class for this day

Public static void showGraphs

Ask the user if they want to see the graph for today

Use AskUserString method (from previous programs)

If yes, call show graph in the day class

Ask the user if they want to see the graph for the month

If yes, call show graph method in the month class

If it is the end of the month, call showData

Public static void showData

Use printf to print a 2D table with the data from each day (the day will

show the total number of homework hours from that day)

}

Public class Day

{

Encapsulated field int array hours

Encapsulated field int total

//constructor

//take one parameter of the number of classes the student is taking

public Day

initialize hours to be an array with the length of the number of classes

initialize total to be 0

public void addTime //takes parameter which is the class # and one that is the number of hours

for that class number, add the number of hours to the int array

add the number of hours to total

public void getHours

return hours

public void getTotal

return total

}

## **Testing**

I will test my client program to make sure it is robust by…

1. Making sure that the student is taking at least one class and not more than 8, if not, the user will be reprompted
2. Make sure the user enters an actual month, and if not, the user will be reprompted
3. Make sure if the user wants to see a graph, it generates (it will generate for any input (case insensitive) that starts with y) and if not, then nothing is generated
4. Make sure that if it is the end of the month, all of the month’s data is printed out, but not if it is not the end of the month
5. Check that the user is not able to enter homework hours less than 0 or more than 12
6. For all things that take an int, make sure that if anything else is inputed that the user is reprompted
7. I will do a mini-test program to make sure my Day object works before using it for the big program
8. Check that the graph’s data matches the inputted data

## **Grading Rubric**

1. Does the program match the specification? (14 points)
   1. If part of the design was not able to be achieved, was there a good reason for it and did you find a creative solution to work around it?
2. Is the code well documented and readable (6 points)
3. Does the code match all test cases/is it robust? (6 points)
4. Is there functional decomposition? (5 points)
5. Do the graphs display as expected? (4 points)
6. Was there a good class hierarchy used (3 points)?
7. Is the code efficient (2 points)?

## **Proposed Implementation Schedule**

5/9-5/10: download/learn jFreeChart (see how compatible it will be with my Day objects)

5/11-5/15: implement client and Day objects

5/18-5/22: implement graphs

5/23-5/26: finalize code/make any last minute edits

## **Potential Showstoppers**

* JFreeChart might not be compatible with my Day objects
  + If so, I will not use Day objects but instead use a 2D array or other data structure that is compatible with the charts

## **Open Questions**

* How do you use JFreeChart? Will it be compatible with my Day objects?

## **Resources**

I plan to use jFreeChart to help me generate my graphs.