Jonathan Liao

Dr. Harrison

CS 203

17 Jan 2015

Fortune Teller Project

This basic fortune teller is difficult to get a unique message out at first, since the only questions asked are name, favorite number, birthdays, marriage, and current date. Multiple tries are necessary to figure out the secret fortunes that are found in the combination of answers. At the end of the questionnaire, there is a combination of 9 different fortunes that the computer picks from; an appropriate fortune is chosen based on *if* and *if else* statements that reflect on the answers given during questions. The lucky number is also eccentrically chosen; but influential factors on the number are determined by marriage status and odd or even date.

Because this program is reliant on user input, the line *import java.util.Scanner;* imports essential data to create *Scanner* objects later in the code. This project does not need multiple classes, so only one class called *Fortune\_liao18* has been created that contains the *main method.* Inside the main method, a *Scanner* object called *userInput* is created immediately; after the construction of this object, data variables that will be placeholders for answer values are declared, each named with an appropriate label such as *favNumber* or *todayE.* After the declarations, a *while* loop that is set to *true* (thus making the loop infinite) sets the enclosed code to forever run, so the user can use the program over and over again without exiting the terminal and having to re-invoke the main method. First, directions and the required welcoming message is printed out and following that line, the series of questions are printed out with the computer prompted the user for answers after each one. At the end of the questions, the lucky number is calculated based on answers given and a series of *if* and *else if* statements determine the given fortune. After the fortune has been read, the user may choose to continue playing after given the “Thank you” message by typing EXIT or any random key. To evaluate this program, I have input various combinations of answers to ensure that each fortune (of the 9 customized messages out there) are achievable and that there is no bug that prevents the user from attaining a message that they are supposed to attain. For example, a positive fortune will appear if a user is born in the month of August (8), is married, and has an income of over $100,000 a year; through my tests, I have been able to achieve the intended fortune for these scenarios. By answering that I am born in the month “8”, saying “yes” to my marriage status, and declaring that I make “120000” dollars a year, I get the message saying that my “future is in good hands.”

The most challenging part of this project was learning how to use the .nextLine method correctly. Sometimes, the line *userInput.nextLine()* would be entirely skipped when a question is asked, thus the user never has a chance to answer to input data for an answer. Through research into this method, I have hypothesized it’s because the *Scanner#nextInt* method does not read the *last newline* character of the input, and thus that *newline* is consumed in the next call to *Scanner#nextLine.* Another problem I have encountered in this project is user error and incompatible data types being fed through the *userInput.next()* assignment. This program so far does not contain any *catch* or *throw* statements that would otherwise prevent user error from crashing the program. For example, when a question asks for number of the month a person was born, the code cannot accept any string data such as “October” or “February”. Thus, the program crashes if a *String* data type is forced into an *int* data type where *birthMonth* is stored. To prevent future crashes, a *throwsIOException* statement could be placed in the main method where *catch* statements would tell the program what to do in case of incompatible data types.

Test Case:

User Enters: Output:  
**Lucifer** Because you are the devil, you shall return to your origins  
**666** below the earth one day. You will find much misery in your  **12** quest for suffering and satanic beliefs! **25  
0** Thank you, Lucifer, for using the customized fortune   
**0** teller.To continue and retry, enter any key. If you **Y** would like to quit, enter EXIT in input **N**