Final Project

Option 2:
Community Garden
Program

CIS-18A

Paola Alcala

Goals:

My Community Garden Program displays hours and days of operation as well as activities that take place at the garden. Users are prompted to register with their first, last names, and email. Next the user can choose from either making an appointment or making a donation. The user will only be allowed to make an appointment during the garden's hours of operation. The user will have the option to make either a monetary or other form of donation. After all information is imputed the user will see a message that confirms their general information, appointment date and time, and or donation made. This program was aimed at students, staff, faculty and the surrounding community around MVC.

Below I have provided a program progression:

Version 1 :

I will start with creating a driver program that will output the specific information about The Community Garden. Then I will use the scanner class to get user string input to collect the user's contact information.

Version 2:

In this next version I will be creating an object class to hold the information for the appointment options. Set appointments during operating hours and have a 1 year span.

Version 3:

I will attempt to incorporate the 'JDatePicker: Java Swing Date Picker' feature with a button to save the date picked by user.

Version 4:

I caught 2 potential bugs in my version 3. The probability of the user inputting a date and time on non business hours. Need to include error handling as well as creating a drop down box that provides an array of appropriate times to choose from.

Version 5:

In this version I will be creating a package based structure to add monetary and material donations. I will be creating a class 'Donation' and use inheritance to specify the different types of donations.

Version 6 (Final Version)

I am implementing the final feature that will output the order summary and appointment in a text file. In this version I also went through and made sure that I have met all the rubric requirements. In Donation.java I implemented the use of interfaces and a switch statement.

Strength:

My strength was implementing the 'JDatePicker: Java Swing Date Picker' feature and adding a button.

Weakness:

At the end after either making an appointment or donation the program keeps going and does not prompt users to exit. I ran out of time to debug this issue. To exit the program you will need to input the number 3 and then the program will exit and your information will be saved to the text file.

Pseudo Code:

Summary.java

INTERFACE Summary:

   DEFINE METHOD getSummary() that returns a STRING

END INTERFACE

Donations.java

ABSTRACT CLASS Donations IMPLEMENTS Summary:

   DECLARE protected donorName as STRING

   DECLARE protected date as STRING

   CONSTRUCTOR(name, date):

     SET THIS.donorName to name

     SET THIS.date to date

   END CONSTRUCTOR

   ABSTRACT METHOD displayDonationSummary()

END CLASS

CLASS MonetaryDonation INHERITS FROM Donations:

   DECLARE amount as DOUBLE

   CONSTRUCTOR(name, date, amount):

```
        CALL Parent Constructor(name, date)

        SET THIS.amount to amount

    END CONSTRUCTOR


    OVERRIDE getSummary():

        RETURN "Monetary Donation by " + donorName + ": $" + amount

    END METHOD


    OVERRIDE displayDonationSummary():

        PRINT "--- Monetary Donation Logged ---"

        PRINT result of getSummary()

    END METHOD
END CLASS


CLASS ResourceDonation INHERITS FROM Donations:

    DECLARE materialType as STRING

    DECLARE description as STRING


    CONSTRUCTOR(name, date, type, desc):

        CALL Parent Constructor(name, date)

        SET THIS.materialType to type

        SET THIS.description to desc

    END CONSTRUCTOR


    OVERRIDE getSummary():

        RETURN "Material Donation (" + materialType + ") by " + donorName
```

END METHOD

OVERRIDE displayDonationSummary():

PRINT "--- Material Donation Logged ---"

PRINT result of getSummary()

END METHOD

END CLASS

Garden.java

PROGRAM Garden:

DECLARE global finalReport as STRING (initially empty)

METHOD Main():

DECLARE timeSlots ARRAY containing ["08:00 AM", ..., "08:00 PM"]

DECLARE keepRunning as BOOLEAN = TRUE

PRINT "Welcome to the Community Garden"

INPUT name from user

INPUT email from user

PRINT "Registration Confirmed"

// MAIN LOOP

WHILE keepRunning is TRUE:

PRINT "--- MAIN MENU ---"

PRINT "1. Donation, 2. Appointment, 3. Exit"

INPUT choice from user

```
    // SWITCH STATEMENT (Branching)

    SWITCH choice:

        CASE 1:

            CALL handleDonationMenu(name)

        CASE 2:

            DISPLAY all timeSlots from Array

            CALL launchAppointmentGUI(timeSlots)

        CASE 3:

            CALL saveSummaryToFile(name, email)

            PRINT "Goodbye"

            SET keepRunning = FALSE

        DEFAULT:

            PRINT "Invalid choice"

    END SWITCH

  END WHILE

END METHOD


METHOD handleDonationMenu(name):

  PRINT "Select Type: [1] Money [2] Material"

  INPUT type


  IF type is 1:

    INPUT amt

    IF amt > 0 AND amt <= 50000:

        CREATE new MonetaryDonation object

        CALL displayDonationSummary()
```

APPEND summary to finalReport

END IF

ELSE IF type is 2:

INPUT item and description

CREATE new ResourceDonation object

CALL displayDonationSummary()

APPEND summary to finalReport

END IF

END METHOD

METHOD launchAppointmentGUI(slots):

OPEN window with DatePicker and Time Dropdown

WHEN "Confirm" is clicked:

GET selectedDate and selectedTime

// LOGICAL OPERATORS

IF day is Sunday:

SHOW ERROR "Closed on Sundays"

ELSE IF day is Saturday AND time is after 5PM:

SHOW ERROR "Saturday hours end at 5PM"

ELSE:

CREATE new Appointment object

APPEND summary to finalReport

PRINT "Appointment Saved"

CLOSE window

END IF

END METHOD

        METHOD saveSummaryToFile(user, email):

            TRY:

                CREATE/OPEN file "GardenSummary.txt"

                WRITE user and email info

                IF finalReport is empty:

                    WRITE "No activities recorded"

                ELSE:

                    WRITE finalReport

                END IF

                PRINT "File saved successfully"

            CATCH Errors:

                PRINT "Error saving file"

        END METHOD

END PROGRAM