Test Plan

Nelson Villatoro

CMSC115

Chapter 3, Project 3

November 4, 2024

**Program Goals & Objectives**

The primary objective of this program is to solicit user input for a month and a year, subsequently displaying the number of days within that month for the specified year. The program accurately incorporates the concept of leap years during its calculations for the number of days in February.

**Program Functional Requirements**

1. The program should prompt the user to enter a month as an integer between 1 and 12.

2. The user should be able to enter the month as a numerical value representing the month (e.g., 1 for January).

3. The program should validate that the entered month is within the valid range (1-12).

4. If the user enters an invalid month, the program should display an appropriate error message and terminate.

5. The program should prompt the user to enter a year as a positive integer.

6. The user should be able to enter the year as a numerical value.

7. The program should calculate whether the year is a leap year.

• A year is a leap year if it is divisible by 4 but not by 100, or it is divisible by 400.

8. The program should determine the number of days in the entered month for the specified year.

9. The program should display the month name, the year, and the number of days in the month.

**Program Pseudocode**

START

Prompt user to "Enter a month in the year (e.g., 1 for Jan): "

Read month

IF month < 1 OR month > 12 THEN

Display "Invalid month"

Terminate program

ENDIF

Prompt user to "Enter a year: "

Read year

Initialize days to 0

Initialize monthName to ""

SWITCH (month)

CASE 1:

monthName = "January"

days = 31

BREAK

CASE 2:

monthName = "February"

IF (year % 4 == 0 AND year % 100 != 0) OR (year % 400 == 0) THEN

days = 29

ELSE

days = 28

ENDIF

BREAK

CASE 3:

monthName = "March"

days = 31

BREAK

CASE 4:

monthName = "April"

days = 30

BREAK

CASE 5:

monthName = "May"

days = 31

BREAK

CASE 6:

monthName = "June"

days = 30

BREAK

CASE 7:

monthName = "July"

days = 31

BREAK

CASE 8:

monthName = "August"

days = 31

BREAK

CASE 9:

monthName = "September"

days = 30

BREAK

CASE 10:

monthName = "October"

days = 31

BREAK

CASE 11:

monthName = "November"

days = 30

BREAK

CASE 12:

monthName = "December"

days = 31

BREAK

ENDSWITCH

Display monthName + " " + year + " has " + days + " days"

END

**Program Flowchart**

**A diagram of a company structure

Description automatically generated**

**Table 1 – Traceability Matrix**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case | Input/Output | Expected Result | Actual Result | Outcome  (Pass/Fail) |
| 1a | Enter a month in the year (e.g., 1 for Jan): 2  Enter a year: 2020 | February 2020 has 29 days | February 2020 has 29 days | Pass |
| 2a | Enter a month in the year (e.g., 1 for Jan): 11  Enter a year: 1989 | November 1989 has 30 days | November 1989 has 30 days | Pass |
| 3a | Enter a month in the year (e.g., 1 for Jan): November | Input validation should handle invalid input | Exception in thread “main” java.util.InputMismatchException | Fail |