For: Dominic “Seth” Jones-Jackson

Assignment: Exercise 13.4 Display Calendars

GitHub URL: https://github.com/sethveeper/Spring21\_ProjectStarbucks/tree/master/Calendar\_Assignment

Student: Please answer the questions, then use the Insert, Screenshot option in Word to snip an appropriate sample of your executing program’s output.

Copy the code from your .java file(s) into the code section below. Your code should match the code submitted in GitHub.

Be sure to review your graded assignment for instructor comments!

|  |
| --- |
| **Analysis** |
| The assignment calls for us to take a month and year as input, generate a “calendar block”, and return the number of days in that month.  A potential problem is input validation – suppose someone inputs a month or year that isn’t valid, or tries to input the month’s name rather than a number? |
|  |

|  |
| --- |
| **Design** |
| Getting the calendar block and the number of days seems straightforward enough – the Calendar class is explicitly designed with this in mind.  Input is allowed from either the command line to run the app, or from within the app itself. A strong argument could be made for implementing both.  Validating the input as numbers is also reasonably straightforward – for months, we only need to check that the input month is between 1 and 12. The year is slightly stickier – if a user only puts two digits in, is it an abbreviation, or do they actually mean that literal year? The latter is probably a simpler and more reliable interpretation.  Validating the month is slightly convoluted. I propose the use of a String array of each month’s name. If the input month isn’t a number, it can check each array index to see if the input is contained in that String – this way, we can catch “Feb” as well as “February”, giving the user a little more margin. If the input is found in the array, the matching index is returned to provide the appropriate number input. |
|  |

|  |
| --- |
| **Testing** |
| 1. Accept and validate inputs. - Can successfully determine if command line inputs were given. (Update: This feature became deprecated when I arbitrarily decided not to bother with it.) - Can successfully establish a “String” input as opposed to an “int” input. - Can successfully determine if the “String” input is an actual month. - Can successfully determine if the “int” input is an actual month. 2. Create a Calendar. - Can successfully instantiate a Calendar object. - Can successfully set the Calendar to the input values. 3. Generate output. - Can successfully get the number of days in a provided month. - Can successfully create a “block”. - Can correctly show the month and year at the head of the block. (Although it’s slightly clunky…) |
|  |

|  |
| --- |
| **Screenshot(s)** |
|  |
|  |

|  |
| --- |
| **Code** |
| import java.util.Scanner;  import java.util.Calendar;  public class Program {  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Author: Dominic "Seth" Jones-Jackson (She/They)  Initialized: January 25, 2021    Abstract: The assignment calls for us to take a month and year as input,  generate a “calendar block”, and return the number of days in that month.    \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  public static Scanner *scan* = new Scanner(System.***in***);  public static void main(String[] args) {  // **TODO** Auto-generated method stub  int month = 0;  int year = 2021;  System.***out***.println("Hello, fox. Let's play with calendars.");    month = *InputMonth*();  year = *InputYear*();    Calendar cal = Calendar.*getInstance*();  cal.set(year, month, 1);  // Create a Calendar object, and set it to our inputs.    *PrintOut*(cal);  *scan*.close();  }  // End of main method    public static void PrintOut(Calendar input)  {  String[] aryMonth = {  "JAN",  "FEB",  "MAR",  "APR",  "MAY",  "JUN",  "JUL",  "AUG",  "SEP",  "OCT",  "NOV",  "DEC"  };  String output = "\n" + aryMonth[input.get(Calendar.***MONTH***)] + " " + input.get(Calendar.***YEAR***) + "\n";  output += "-----------------------------\n";  output += " Sun Mon Tue Wed Thu Fri Sat\n";  // Setting up a header    int month = input.get(Calendar.***MONTH***);  int count = 0;  // Declare an exit condition and an iteration counter    while(month == input.get(Calendar.***MONTH***))  {  for(int i = 1; i < input.get(Calendar.***DAY\_OF\_WEEK***); i++)  output += "|---";  // End of For loop (Fill in blank spaces if needed)    for(int i = input.get(Calendar.***DAY\_OF\_WEEK***); i <= 7; i++)  {  // Fill in the rest of the row  String date = String.*valueOf*(input.get(Calendar.***DAY\_OF\_MONTH***));  // Making the current day a little easier to handle    if (month != input.get(Calendar.***MONTH***))  break; // Stop if we've gone too far  else if (date.length() == 1)  output += "| " + date; // Add extra space for a single digit  else if (date.length() == 2)  output += "| " + date; // Add less space for two digits    count++;  input.add(Calendar.***DATE***, 1);  // Increase the total and roll the calendar up a day  }  // End of For loop (Fill in the row)    output += "\n";  // Add a new row  }  // End of While loop    System.***out***.println(output);  System.***out***.println("Total days in this month: " + count);  // Print the results!  }  // End of Get Block method    public static int InputMonth()  {  int output = -1;  // Declared a scanner and an output.    String[] month = {  "january",  "february",  "march",  "april",  "may",  "june",  "july",  "august",  "september",  "october",  "november",  "december"  };  // Declared an array of months.    System.***out***.println("Please enter the month:");  String input = *scan*.nextLine();  // Input received.    try  {  output = Integer.*parseInt*(input);  // Attempt to cast    if(output < 1)  output = 0;  else if (output > 11)  output = 11;  else  output--;  // End of If/ElseIf block (Check for appropriate month range)  }  // End of Try (Cast String to int)  catch(Exception e)  {  input = input.toLowerCase();  for(int i = 0; i < month.length; i++)  {  // Check each month against the input...  if (month[i].contains(input))  {  // If it's the appropriate month, make the output equal that month.  output = i;  break;  }  // End of If statement (Is this the right month?)  }  // End of For loop    if (output == -1)  {  System.***out***.print("That's not an actual month! Try again.");  output = *InputMonth*();  }  // End of If statement (Was the month found?)  }  // End of Catch (String can't be cast as int)    return output;  // All done!  }  // End of Input method    public static int InputYear()  {  int output = 0;    System.***out***.println("Please enter the year:");  String input = *scan*.nextLine();  // Input received.    try  {  output = Integer.*parseInt*(input);  // Attempt to cast  }  // End of Try (Cast String to int)  catch(Exception e)  {  System.***out***.println("That's not a valid input! Please enter an integer.");  output = *InputYear*();  }  // End of Catch (String can't be cast as int)    return output;  }  // End of Input method  }  // End of Program class |
|  |