import java.time.\*;

import java.util.\*;

public class Calendars {

private static LocalDate *currentDate* = LocalDate.*now*();

//utility class

private Calendars() {

}

//prints the visual calendar

public static void PrintCalendar() {

//reads the Assignments text file before printing, to ensure all values are up to date (accounts for changes made in between prints)

Assignments.*ReadAssignmentsFile*();

//reads the User file before printing, to ensure all values are up to date (accounts for changes made in between prints)

User.*ReadUserFile*();

//2D arrays of the calendar

int[][] calendarDisplayInt = new int[6][7];

String[][] calendarDisplayString = new String[6][7];

//all variables related in the creation of the calendar

int dayOne = 1;

int spacesInFrontOfFirstDay = *currentDate*.withDayOfMonth(1).getDayOfWeek().getValue();

YearMonth yearMonth = YearMonth.*now*();

int numberOfDaysInMonth = yearMonth.lengthOfMonth();

//reads Assignment files, puts values into arrays in Calendar class from the Assignment Class

Assignments.*ReadAssignmentTimesFile*();

ArrayList<String> listOfAssignmentDueDates = Assignments.*GetAssignmentDueDates*();

ArrayList<String> listOfAssignmentSubjects = Assignments.*GetAssignmentSubjects*();

//reads User file, puts values for birthday and user events in Calendar class from User class

User.*ReadUserFile*();

LocalDate birthday = User.*GetBirthday*();

ArrayList<String> listOfYearlyEventNames = User.*GetYearlyEventNames*();

ArrayList<LocalDate> listOfYearlyEventDates = User.*GetYearlyEventDates*();

//prints out the top portion of the calendar

System.*out*.println("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" +

"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

//centers the month and year with formatting functions, prints the top portion of the calendar

String calendarTitle = *currentDate*.getMonth() + " " + *currentDate*.getYear(); //string with month and year

//prints space in front of the month and year, then the space after the month and year

System.*out*.printf("|%" + (((134 - calendarTitle.length()) + 1) / 2 + calendarTitle.length()) + "s", calendarTitle);

//prints the space that comes after the month and year

System.*out*.printf("%" + ((134 - calendarTitle.length()) / 2) + "s", "|\n");

//prints the days of the week

System.*out*.println("————————————————————————————————————————————————————————————————————————————————————————————" +

"——————————————————————————————————————————\n" +

"| Sunday | Monday | Tuesday | Wednesday | Thursday | " +

"Friday | Saturday |\n" +

"———————————————————————————————————————————————————————————————————————————————————————————————————————" +

"———————————————————————————————");

//fills calendarDisplayInt array with values representing the calendar in the proper format for the month

month:

for (int i = 0; i < 6; i++) {

for (int j = 0; j < 7; j++) {

//puts 0 into the calendarDisplayInt array for all the days where they are left as blank in front of the first day

if (--spacesInFrontOfFirstDay > -1) {

continue;

}

//fills the rest of the array with the days, incrementing up to the number of days in the current month

calendarDisplayInt[i][j] = dayOne++;

if (dayOne > numberOfDaysInMonth) {

break month;

}

}

}

//traverses the 2D arrays

for (int i = 0; i < 6; i++) {

for (int j = 0; j < 7; j++) {

//variable to contain the number of assignments on a day, resets with every iteration

int assignmentsPerDay = 0;

//if the value inside the int calendar array is 0, then fill the corresponding value in the String array to be blank

if (calendarDisplayInt[i][j] == 0) {

calendarDisplayString[i][j] = "| ";

//accounts for spacing for one digit days, if statement if the day is less than 10 (all one digit)

} else if (calendarDisplayInt[i][j] < 10) {

//if the current iteration of the day in the int calendar is equal to the current day and month, set

//the corresponding day in the String array to display that day is today

if (calendarDisplayInt[i][j] == *currentDate*.getDayOfMonth() && *currentDate*.getMonth().equals(LocalDate.*now*().getMonth())) {

calendarDisplayString[i][j] = "| " + calendarDisplayInt[i][j] + " ----Today---- ";

} else {

//else make the corresponding value in the String array blank

calendarDisplayString[i][j] = "| " + calendarDisplayInt[i][j] + " ";

}

//if the iteration in the String array is blank

if (calendarDisplayString[i][j].equals("| " + calendarDisplayInt[i][j] + " ")) {

//iterates through the list of assignment due dates arraylist

for (int k = 0; k < listOfAssignmentDueDates.size(); k++) {

//variable that contains the current month in number format, with leading zeroes (in MM)

String currentMonth;

if (*currentDate*.getMonthValue() < 10) { //digit check to have leading zeroes for month format

currentMonth = "0" + *currentDate*.getMonthValue();

} else { //else two digit; don't have to worry just set it to equal the month value

currentMonth = String.*valueOf*(*currentDate*.getMonthValue());

}

//format for one digit days

String yearMonthDay = *currentDate*.getYear() + "-" + currentMonth + "-0" + calendarDisplayInt[i][j];

//if one assignment on a day and the due date of the assignment equals the current iteration of the day

if (listOfAssignmentDueDates.get(k).equals(yearMonthDay)) {

//switch case for the subject to print correctly into the calendar, sets String array value to equal the subject name with proper spacing

switch (listOfAssignmentSubjects.get(k)) {

case "Social" -> calendarDisplayString[i][j] = "| " + calendarDisplayInt[i][j] + " Social ";

case "English" -> calendarDisplayString[i][j] = "| " + calendarDisplayInt[i][j] + " English ";

case "Math" -> calendarDisplayString[i][j] = "| " + calendarDisplayInt[i][j] + " Math ";

case "Science" -> calendarDisplayString[i][j] = "| " + calendarDisplayInt[i][j] + " Science ";

case "Other" -> calendarDisplayString[i][j] = "| " + calendarDisplayInt[i][j] + " Other ";

}

//adds 1 to the number of assignments for the day

assignmentsPerDay++;

}

//if there are 2 or more assignments on a day, print the last subject on that day with ellipses on the end (signals multiple)

if (assignmentsPerDay >= 2) {

//switch case for the subject to print correctly into the calendar, sets String array value to equal the subject name with proper spacing

switch (listOfAssignmentSubjects.get(k)) {

case "Social" -> calendarDisplayString[i][j] = "| " + calendarDisplayInt[i][j] + " Social... ";

case "English" -> calendarDisplayString[i][j] = "| " + calendarDisplayInt[i][j] + " English... ";

case "Math" -> calendarDisplayString[i][j] = "| " + calendarDisplayInt[i][j] + " Math... ";

case "Science" -> calendarDisplayString[i][j] = "| " + calendarDisplayInt[i][j] + " Science... ";

case "Other" -> calendarDisplayString[i][j] = "| " + calendarDisplayInt[i][j] + " Other... ";

}

}

}

//prints yearly events if it matches the date on the calendar, overrides display for assignments

//for loop iterates through all the values inside the listOfYearlyEventDates arraylist

for (int l = 0; l < listOfYearlyEventDates.size(); l++) {

if (listOfYearlyEventDates.get(l).getDayOfMonth() == calendarDisplayInt[i][j] && listOfYearlyEventDates.

get(l).getMonth().equals(currentDate.getMonth())) {

String centeredEventName = String.format("%" + (((17 - listOfYearlyEventNames.get(l).length())) /

2 + listOfYearlyEventNames.get(l).length()) + "s", listOfYearlyEventNames.get(l));

centeredEventName = centeredEventName.concat(String.format("%" + ((16 - listOfYearlyEventNames.

get(l).length()) / 2) + "s", ""));

calendarDisplayString[i][j] = "| " + calendarDisplayInt[i][j] + centeredEventName;

}

}

//prints birthday if day matches the current day iterations month and day, overrides displays for assignments and events

if (calendarDisplayInt[i][j] == birthday.getDayOfMonth() && currentDate.getMonth().equals(birthday.getMonth())) {

calendarDisplayString[i][j] = "| " + calendarDisplayInt[i][j] + " BIRTHDAY! ";

}

}

//exact same process as the days below 10, except all the values put into the String array has one space

//subtracted to account for the 1 additional space that 2 digit days take up, compared to 1 digit days

} else if (calendarDisplayInt[i][j] >= 10) {

if (calendarDisplayInt[i][j] == currentDate.getDayOfMonth() && currentDate.getMonth().equals(LocalDate.now().getMonth())) {

calendarDisplayString[i][j] = "| " + calendarDisplayInt[i][j] + " ----Today---- ";

} else {

calendarDisplayString[i][j] = "| " + calendarDisplayInt[i][j] + " ";

}

if (calendarDisplayString[i][j].equals("| " + calendarDisplayInt[i][j] + " ")) {

for (int k = 0; k < listOfAssignmentDueDates.size(); k++) {

String currentMonth;

if (currentDate.getMonthValue() < 10) { //digit check to have leading zeroes for month format

currentMonth = "0" + currentDate.getMonthValue();

} else { //else two digit; don't have to worry

currentMonth = String.valueOf(currentDate.getMonthValue());

}

//format for two digit days

String yearMonthDay = currentDate.getYear() + "-" + currentMonth + "-" + calendarDisplayInt[i][j];

//if one assignment on a day

if (listOfAssignmentDueDates.get(k).equals(yearMonthDay)) {

switch (listOfAssignmentSubjects.get(k)) {

case "Social" -> calendarDisplayString[i][j] = "| " + calendarDisplayInt[i][j] + " Social ";

case "English" -> calendarDisplayString[i][j] = "| " + calendarDisplayInt[i][j] + " English ";

case "Math" -> calendarDisplayString[i][j] = "| " + calendarDisplayInt[i][j] + " Math ";

case "Science" -> calendarDisplayString[i][j] = "| " + calendarDisplayInt[i][j] + " Science ";

case "Other" -> calendarDisplayString[i][j] = "| " + calendarDisplayInt[i][j] + " Other ";

}

assignmentsPerDay++;

}

//if multiple assignments in a day, print with ellipses on end

if (assignmentsPerDay >= 2) {

switch (listOfAssignmentSubjects.get(k)) {

case "Social" -> calendarDisplayString[i][j] = "| " + calendarDisplayInt[i][j] + " Social... ";

case "English" -> calendarDisplayString[i][j] = "| " + calendarDisplayInt[i][j] + " English... ";

case "Math" -> calendarDisplayString[i][j] = "| " + calendarDisplayInt[i][j] + " Math... ";

case "Science" -> calendarDisplayString[i][j] = "| " + calendarDisplayInt[i][j] + " Science... ";

case "Other" -> calendarDisplayString[i][j] = "| " + calendarDisplayInt[i][j] + " Other... ";

}

}

}

//prints yearly events if it matches the date on the calendar, overrides display for assignments

for (int l = 0; l < listOfYearlyEventDates.size(); l++) {

if (listOfYearlyEventDates.get(l).getDayOfMonth() == calendarDisplayInt[i][j] && listOfYearlyEventDates.get(l)

.getMonth().equals(currentDate.getMonth())) {

String centeredEventName = String.format("%" + (((15 - listOfYearlyEventNames.get(l).length())) / 2

+ listOfYearlyEventNames.get(l).length()) + "s", listOfYearlyEventNames.get(l));

centeredEventName = centeredEventName.concat(String.format("%" + ((16 - listOfYearlyEventNames.get(l)

.length()) / 2) + "s", ""));

calendarDisplayString[i][j] = "| " + calendarDisplayInt[i][j] + centeredEventName;

}

}

//prints birthday if day matches date on calendar, overrides other displays for assignments and events

if (calendarDisplayInt[i][j] == birthday.getDayOfMonth() && currentDate.getMonth().equals(birthday.getMonth())) {

calendarDisplayString[i][j] = "| " + calendarDisplayInt[i][j] + " BIRTHDAY! ";

}

}

}

//puts the line at the end of the week, indents to next line, and prints a line to separate the weeks

System.out.print(calendarDisplayString[i][j]);

if (j == 6) {

System.out.print("|\n" +

"----------------------------------------------------------------------------------" +

"----------------------------------------------------\n");

}

}

}

}

//returns current date in Day Year DD, YYYY with first letter of month and day capitalized

public static String GetFormattedCurrentDate() {

String dayOfTheWeek = currentDate.getDayOfWeek().toString().toLowerCase();

String formattedDayOfTheWeek = dayOfTheWeek.substring(0, 1).toUpperCase() + dayOfTheWeek.substring(1);

String month = currentDate.getMonth().toString().toLowerCase();

String formattedMonth = month.substring(0, 1).toUpperCase() + month.substring(1);

return (formattedDayOfTheWeek + " " + formattedMonth + " " + currentDate.getDayOfMonth() + ", " + currentDate.getYear());

}

public static void DisplayExactTime() {

System.*out*.println("It is currently " + LocalDateTime.*now*());

System.*out*.println("Please ensure this time is correct for program to function properly.");

}

//changes display into the previous month

public static void PreviousMonth() {

*currentDate* = *currentDate*.minusMonths(1); //subtracts one month to the currentDate variable

String month = *currentDate*.getMonth().toString().toLowerCase();

//makes first letter of month capitalized with the rest lower case

String formattedMonth = month.substring(0, 1).toUpperCase() + month.substring(1);

System.*out*.println("Here is the calendar for the month of " + formattedMonth + ".");

*PrintCalendar*(); //prints the calendar

}

//changes display into the next month

public static void NextMonth() {

*currentDate* = *currentDate*.plusMonths(1); //adds one month to the currentDate variable

String month = *currentDate*.getMonth().toString().toLowerCase();

//makes first letter of month capitalized with the rest lower case

String formattedMonth = month.substring(0, 1).toUpperCase() + month.substring(1);

System.*out*.println("Here is the calendar for the month of " + formattedMonth + ".");

*PrintCalendar*(); //prints the calendar

}

//returns the current date variable

public static LocalDate GetCurrentDate() { return *currentDate*; }

}