A Calendar-Making Program

Group 7: Aadee Chheda, Devin Riebe, Brayden Wilkins, Bilal Alam

Project Overview

- Calendar that can display different months in the year with user-inputted, specific day-to-day notes and reminders.

- Users have the ability to add and remove notes for a specific day, add multiple notes for a single day, and view multiple months before closing the program.

Requirements and Additions

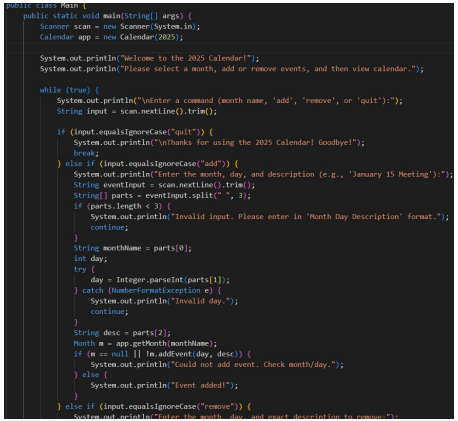
- FR1- Allows the user to enter the month for the program to generate.
- FR2- Displays each month of the calendar separately, with proper alignment for days of the week.
- FR3- Allows the user to add multiple custom events for a specific date.
- FR4- Displays the annotated dates and bulleted events at the bottom of the calendar for the user to view.
- FR5- Allows the user to remove events from a specific date.
- FR6- Allows the user to quit the program.

Design Process

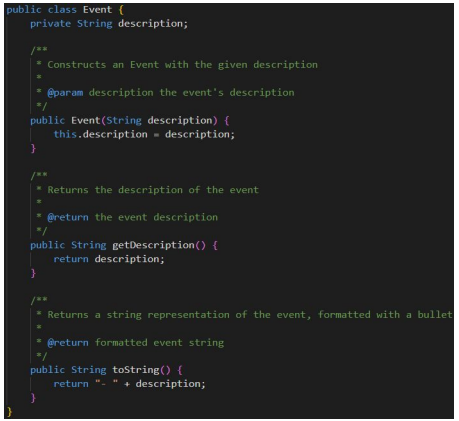
List nouns on what potential classes and fields will be

Create two possible designs with classes and fields and pick the best

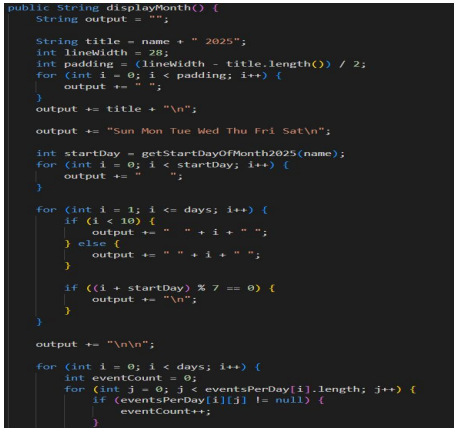
From the design created and approved start the code



```
public class Main {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        Calendar app = new Calendar(2025);

        System.out.println("Welcome to the 2025 Calendar!");
        System.out.println("Please select a month, add or remove events, and then view calendar.");

        while (true) {
            System.out.println("\nEnter a command (month name, 'add', 'remove', or 'quit'):");
            String input = scan.nextLine().trim();

            if (input.equalsIgnoreCase("quit")) {
                System.out.println("\nThanks for using the 2025 Calendar! Goodbye!");
                break;
            } else if (input.equalsIgnoreCase("add")) {
                System.out.println("Enter the month, day, and description (e.g., 'January 15 Meeting'):");
                String eventInput = scan.nextLine().trim();
                String[] parts = eventInput.split(" ", 3);
                if (parts.length < 3) {
                    System.out.println("Invalid input. Please enter in 'Month Day Description' format.");
                    continue;
                }
                String monthName = parts[0];
                int day;
                try {
                    day = Integer.parseInt(parts[1]);
                } catch (NumberFormatException e) {
                    System.out.println("Invalid day.");
                    continue;
                }
                String desc = parts[2];
                Month m = app.getMonth(monthName);
                if (m == null || !m.addEvent(day, desc)) {
                    System.out.println("Could not add event. Check month/day.");
                } else {
                    System.out.println("Event added!");
                }
            } else if (input.equalsIgnoreCase("remove")) {
                System.out.println("Enter the month, day, and exact description to remove:");
```



```
public class Event {
    private String description;

    /**
     * Constructs an Event with the given description
     *
     * @param description the event's description
     */
    public Event(String description) {
        this.description = description;
    }

    /**
     * Returns the description of the event
     *
     * @return the event description
     */
    public String getDescription() {
        return description;
    }

    /**
     * Returns a string representation of the event, formatted with a bullet
     *
     * @return formatted event string
     */
    public String toString() {
        return "- " + description;
    }
}
```



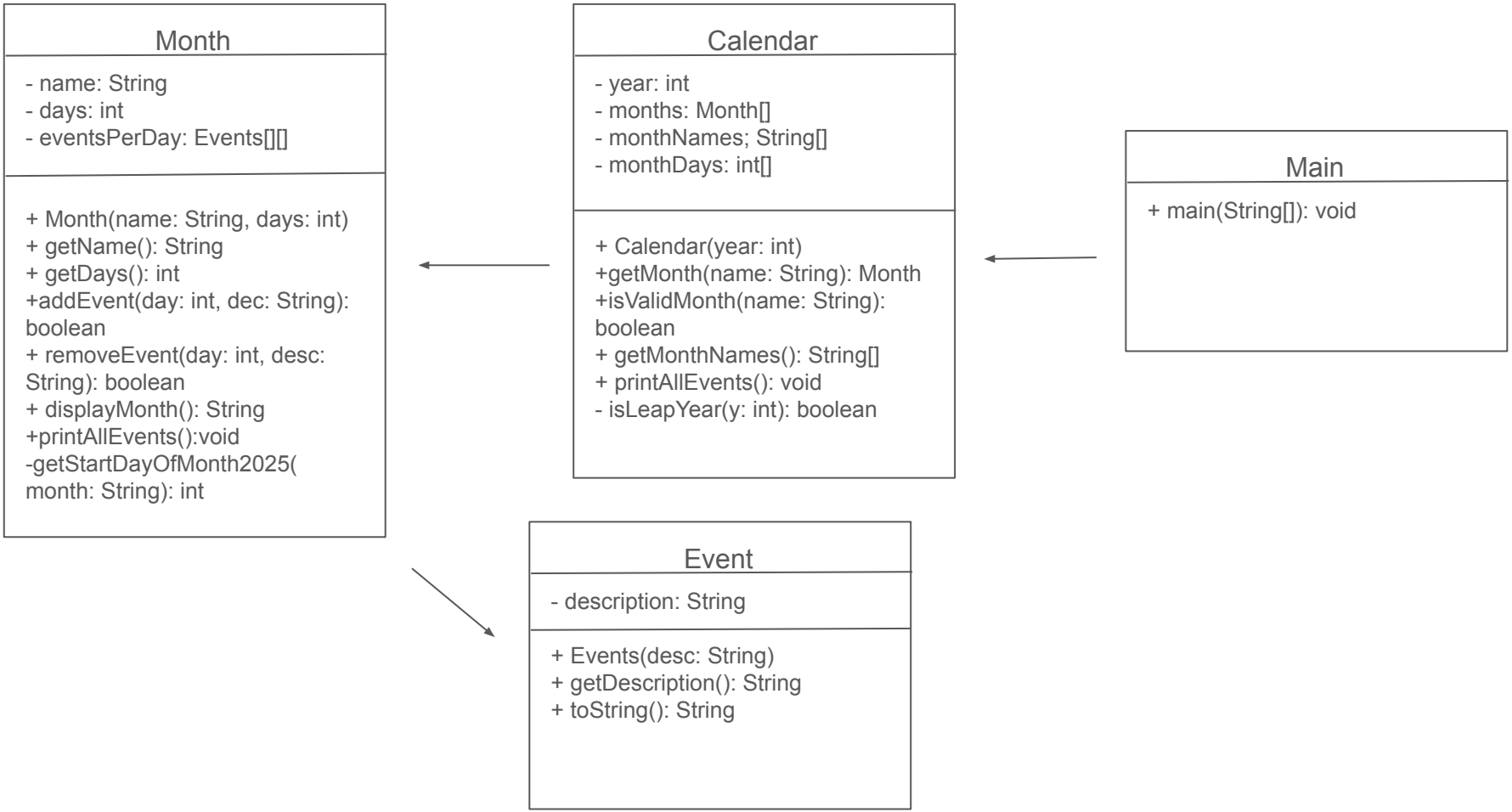
```
public String displayMonth() {
    String output = "";

    String title = name + " 2025";
    int lineWidth = 28;
    int padding = (lineWidth - title.length()) / 2;
    for (int i = 0; i < padding; i++) {
        output += " ";
    }
    output += title + "\n";

    output += "Sun Mon Tue Wed Thu Fri Sat\n";

    int startDay = getStartDayOfMonth2025(name);
    for (int i = 0; i < startDay; i++) {
        output += "    ";
    }

    for (int i = 1; i <= days; i++) {
        if (i < 10) {
            output += "  " + i + " ";
        } else {
            output += " " + i + " ";
        }

        if ((i + startDay) % 7 == 0) {
            output += "\n";
        }
    }

    output += "\n\n";

    for (int i = 0; i < days; i++) {
        int eventCount = 0;
        for (int j = 0; j < eventsPerDay[i].length; j++) {
            if (eventsPerDay[i][j] != null) {
                eventCount++;
            }
```

Extension and Additions

- The project's extension challenge was allowing the user to add multiple events to a single date.

- Our group added our own challenge, additionally implementing the users ability to remove events.

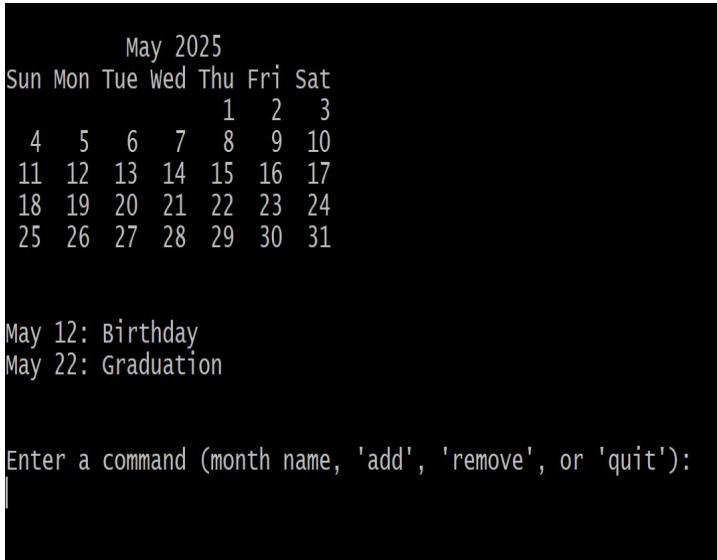
```
bilal@LAPTOP-190E17MB MINGW64 ~/csc116/csc116-003-CE-07/Ce (main)
$ java -cp bin Main
Welcome to the 2025 Calendar!
Please select a month, add or remove events, and then view calendar.

Enter a command (month name, 'add', 'remove', or 'quit'):
add
Enter the month, day, and description (e.g., 'January 15 Meeting'):
May 12 Birthday
Event added!
```


Month
- name: String
- days: int
- eventsPerDay: Events[][]
+ Month(name: String, days: int)
+ getName(): String
+ getDays(): int
+addEvent(day: int, dec: String): boolean
+ removeEvent(day: int, desc: String): boolean
+ displayMonth(): String
+printAllEvents():void
-getStartDayOfMonth2025(month: String): int
Calendar
- year: int
- months: Month[]
- monthNames; String[]
- monthDays: int[]
+ Calendar(year: int)
+getMonth(name: String): Month
+isValidMonth(name: String): boolean
+ getMonthNames(): String[]
+ printAllEvents(): void
- isLeapYear(y: int): boolean
Main
+ main(String[]): void
Event
- description: String
+ Events(desc: String)
+ getDescription(): String
+ toString(): String

Live Demo

Testing

- Invalid Add Inputs
  - Handles non-integer inputs for the day
  - Handles inputs that are not month names for the month.
  - Handles inputs of null for the month and day.
- Invalid Remove Inputs
  - Handles non-integer inputs for the day.
  - Handles inputs that are not month names for the month.
  - Handles inputs of null for the month and day.
  - Handles inputs of non-exact descriptions.
  - Handles inputs of null descriptions.
- Quit
  - Handles input for quitting the program.



Lessons Learned

- If there is a schedule provided to what to get done each day that is the bare minimum of what the group should accomplish
- In person group time is the most productive time to get work done
- Take breaks while working on code

A Calendar-Making Program

Group 7: Aadee Chheda, Devin Riebe, Brayden Wilkins, Bilal Alam