Patryk Dziedzic, Aveesh Patel

Lily Chang

CS213 Software Methodology

2 October 2023

Project 1: Event Organizer

|  |  |  |  |
| --- | --- | --- | --- |
| Class Name: Date  Method Signature: public boolean isValid() {} //check if a given date is a valid calendar date | | | |
| Test Case # | Requirement | Test Description and Input Data | Expected Result/Output |
| 1 | The number of days for a date in February (in a non-leap year) is less than or equal to 28. | First checks if the specific day, month, and year are valid in a non-leap year and then takes the date, expected output, and actual output from the test to check if the test passed successfully (TRUE) or not (FALSE)  Input Data: the String “2/29/2023” depicting the Date for the test | FALSE |
| 2 | The number of days for a date in February (in a leap year) is equal to 29 | First checks if the specific day, month, and year are valid in a leap year and then takes the date, expected output, and actual output from the test to determine if the test passed successfully (TRUE) or not (FALSE)  Input Data: the String “2/29/2020” depicting the Date for the test | TRUE |
| 3 | The number corresponding to a month is in the range of 1 to 12 | Checks if the specific month given is within the range of 1-12 by checking if the month (as well as the day and year) are valid and determining if the test passed successfully by taking the date, expected output, and actual output and comparing the expected and actual output to see if they are the same (TRUE) or not (FALSE)  Input Data: the String “13/29/2023” depicting the Date for the test | FALSE |
| 4 | The number corresponding to a specific day is in the range for that given month | First checks if the specific day is valid for the given month and year and then takes the date, expected output, and actual output from the test to determine if the test passed successfully (TRUE) or not (FALSE)  Input Data: the String “4/31/2023” depicting the Date for the test | FALSE |

|  |  |  |  |
| --- | --- | --- | --- |
| Class Name: Event  Method Signature: @Override public boolean equals(Object obj) {} | | | |
| Test Case # | Requirement | Test Description and Input Data | Expected Result/Output |
| 1 | Two given events with the same date, startTime, and location are equal | First constructs two events with the same date, startTime, location, contact, and duration. Then takes the first event, second event, expected output and the actual output of checking if the two events are equal and checks if the expected output is the same as the actual output (TRUE) or different (FALSE)  Input Data: 2 Events constructed with a Date of “01/01/1999”, a Timeslot with a MORNING value, a Location of AB2225, a Contact constructed with the CS Department, and a duration of 80 minutes | TRUE |
| 2 | Two given events are not equal (with the same date and startTime but different locations) | First constructs two events with the same date, startTime, contact, and duration but different Locations. Then takes the first event, second event, expected output and the actual output of checking if the two events are equal and checks if the expected output is the same as the actual output (TRUE) or different (FALSE)  Input Data: 2 Events constructed with a Date of “01/01/1999”, a Timeslot with a MORNING value, differing Locations of AB2225 and TIL232, respectively, a Contact constructed with the CS Department, and a duration of 80 minutes | FALSE |
| 3 | Two events are comparatively equal in terms of Date and TimeSlot (startTime) | First constructs two events with the same date, startTime, location, contact, and duration. Then takes the first event, second event, expected output and the actual output of checking if the two events are comparatively equal and checks if the expected output is the same as the actual output (0) or different (-1 or 1)  Input Data: 2 Events constructed with a Date of “01/01/1999”, a Timeslot with a MORNING value, a Location of AB2225, a Contact constructed with the CS Department, and a duration of 80 minutes | 0 |
| 4 | Two events are comparatively not equal in terms of Date and Timeslot | First constructs two events with different dates, same startTime(s), same contact and duration but different Locations. Then takes the first event, second event, expected output and the actual output of checking if the two events are comparatively equal and checks if the expected output is the same as the actual output (0), if the first event comes before the second event (-1), or if the second event comes before the first event (1)  Input Data: Events constructed with the dates “01/01/1999” and “05/27/2000” respectively, a Timeslot with a MORNING value, differing Locations of AB2225 and TIL232 respectively, a Contact constructed with the CS Department, and a duration of 80 minutes | -1 |