2024

Homework Assignment

NEXTDATE-TEST CASE DESIGN

ASHWIN PREM

SOFE3980U

The NextDate functions has three inputs: Day, Month, Year

Characteristics:

- Day
- Month
- Year

Domain:

- Day (1-31)
- Month (1-12)
- Year (1812 – 2212)

Blocks:

- Normal value – Entering a date will output the next day
- Max value – Entering a date that is the last value of the domain would output a date that resets to the minimum value of the domain
- Invalid value – Entering a date outside of the domain would yield an error
- Leap Year value – Takes into consideration leap years, such that February's domain is increased by one day.
- Leap century value – years are that divisible by 100 but not by 400 are not leap years

Table 1: Testing critical dates that would validate the normal functionality of the method/function

Test Case	**Day**	**Month**	**Year**	**Expected value (DD/MM/YYYY)**	**Notes**
1	1	1	1912	02/01/1912	Results next day
2	31	1	1912	01/02/1912	Day loops back to 1. Month increments.
3	28	2	1913	01/03/1913	Non-leap-year. 28 is the last day of February
4	31	12	1912	01/01/1913	Day and Month loops back to 1. Year increments

Table 2: Tests most out of bounds inputs for all domains.

Test Case	Day	Month	Year	Expected value (DD/MM/YYYY)	Notes
5	1	1	1711	(--/--/----)	Invalid, out of year’s domain
6	1	1	2213	(--/--/----)	Invalid, out of year’s domain
7	0	1	1912	(--/--/----)	Invalid, out of day’s domain
8	32	1	1912	(--/--/----)	Invalid, out of day’s domain
9	1	0	1912	(--/--/----)	Invalid, out of month’s domain
10	1	13	1912	(--/--/----)	Invalid, out of month’s domain

Table 3: Testing if day 31 constitutes as invalid entry for months that don't contain 31 days.

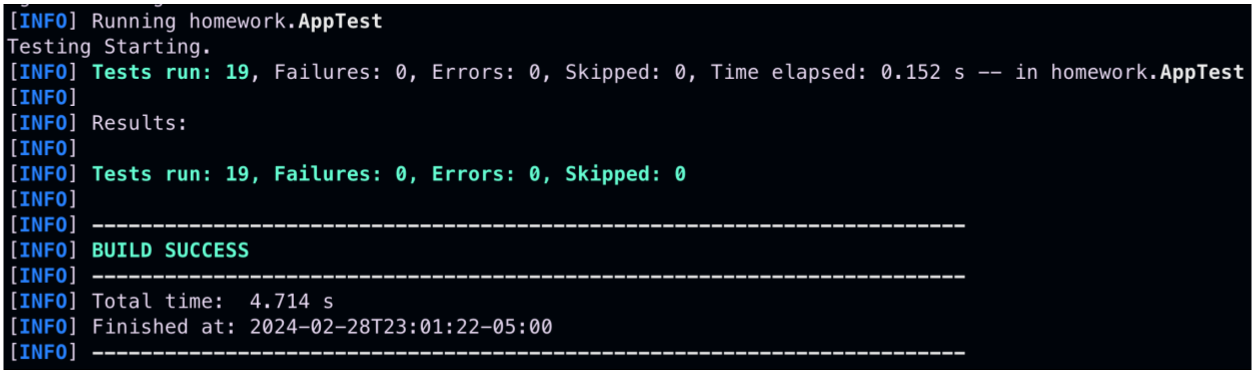
Test Case	Day	Month	Year	Expected value (DD/MM/YYYY)	Notes
11	30	2	1912	(--/--/----)	Invalid, day out of bound for month of feb
12	31	4	1912	(--/--/----)	Invalid, day out of bound for month of april
13	31	6	1912	(--/--/----)	Invalid, day out of bound for month of June
14	31	8	1912	(--/--/----)	Invalid, day out of bound

					for month of August
15	31	9	1912	(--/--/----)	Invalid, day out of bound for month of September
16	31	11	1912	(--/--/----)	Invalid, day out of bound for month of November

Table 4: Tests to validate the correct functionality of leap years and leap centuries

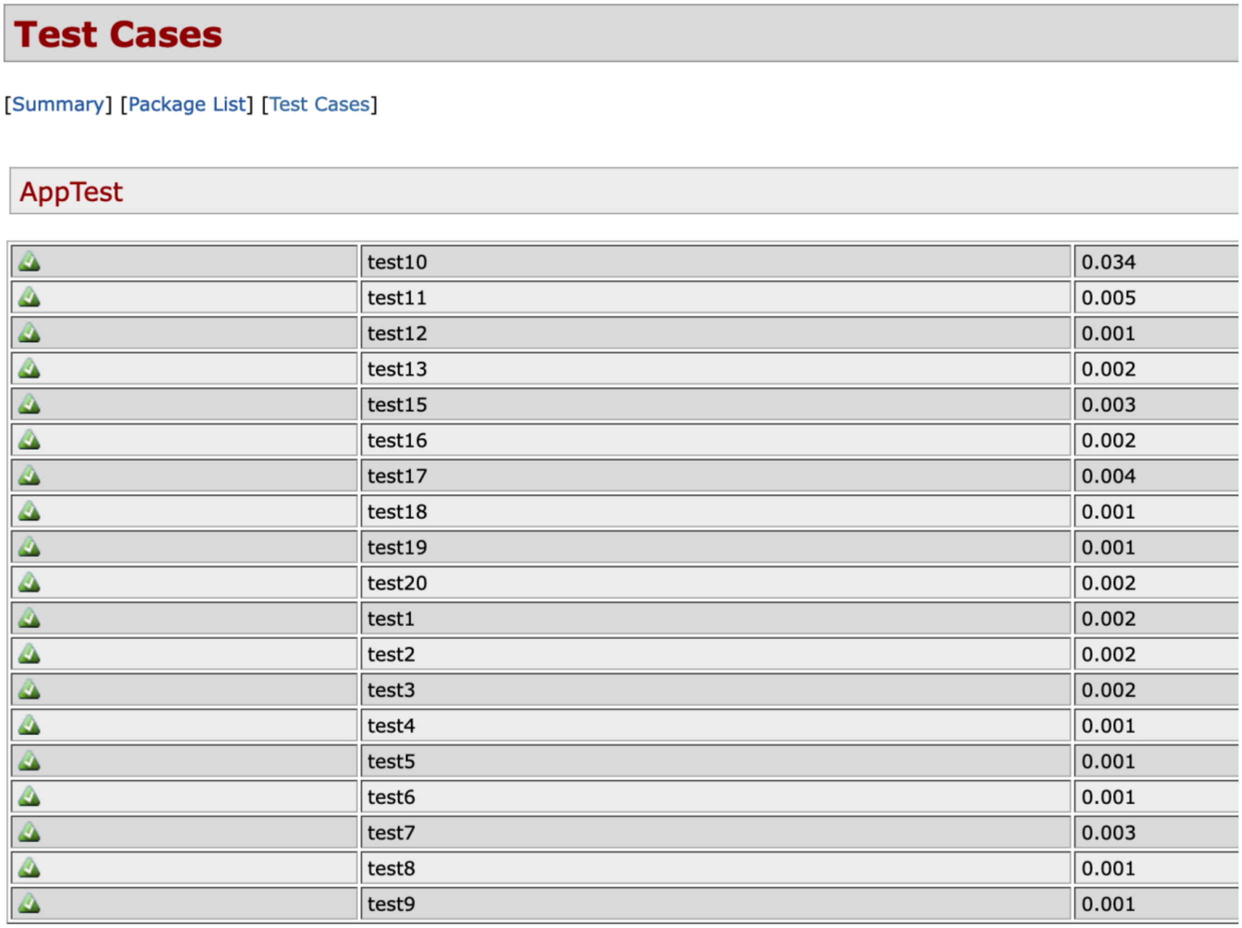
Test Case	**Day**	**Month**	**Year**	**Expected value (DD/MM/YYYY)**	**Notes**
17	28	2	2024	(29/02/2024)	Checking if years divisible by 4 are correctly leap years.
18	29	2	1912	(01/03/1912)	Checking if Day 29 for February is accepted as valid for leap years, and successfully increments
19	28	2	1900	(01/03/1900)	Checking if centuries divisible by 100 but not 400 correctly aren’t leap years.
20	28	2	2000	(29/02/2000)	Checking if centuries divisible by both 100 and 400

					correctly are leap years.



```
[INFO] Running homework.AppTest
Testing Starting.
[INFO] Tests run: 19, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.152 s -- in homework.AppTest
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 19, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  4.714 s
[INFO] Finished at: 2024-02-28T23:01:22-05:00
[INFO] ------------------------------------------------------------------------
```

Figure 1: Successfully built and ran the tests.

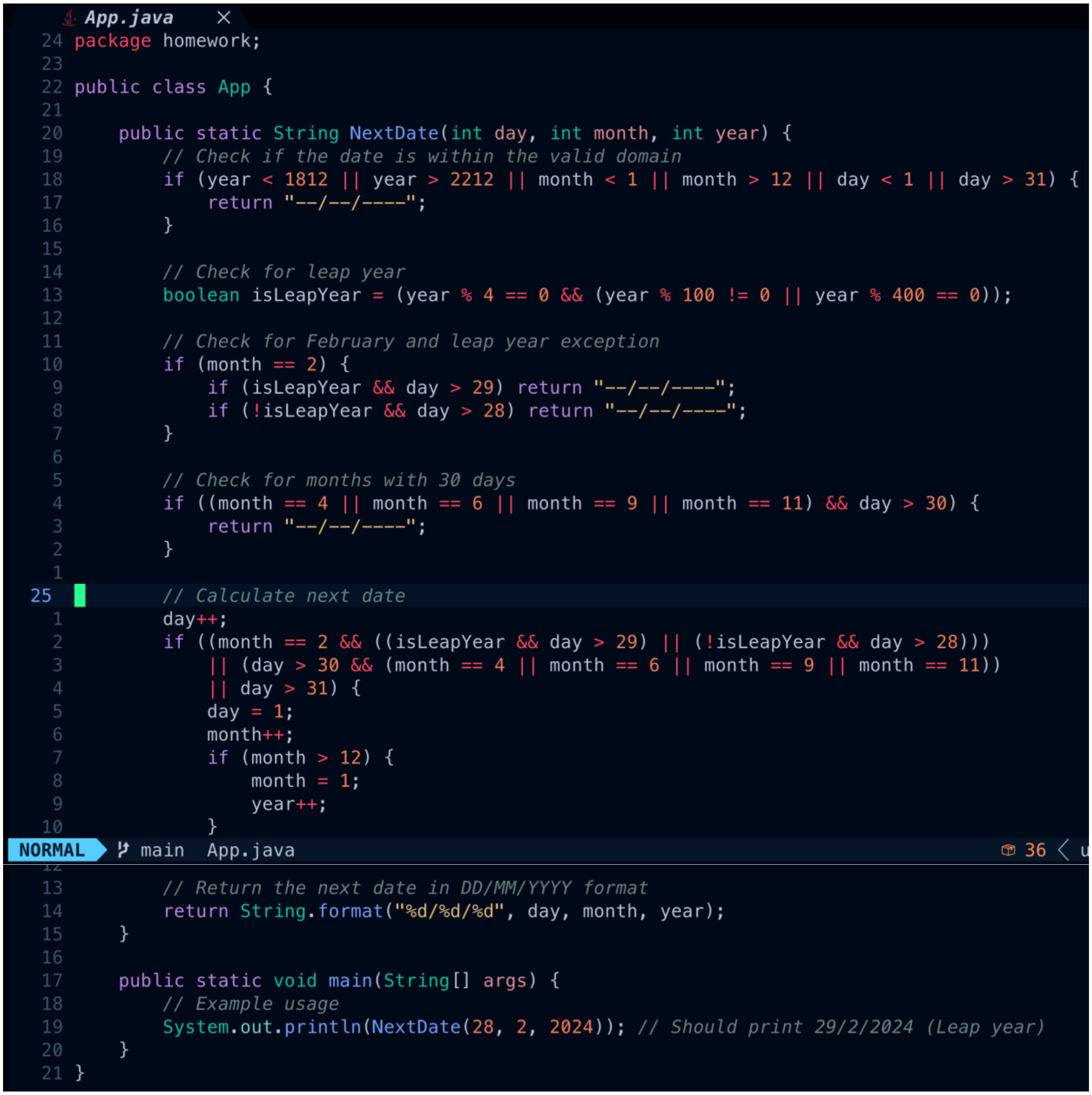


Test Cases

[Summary] [Package List] [Test Cases]

AppTest

✓	test10	0.034
✓	test11	0.005
✓	test12	0.001
✓	test13	0.002
✓	test15	0.003
✓	test16	0.002
✓	test17	0.004
✓	test18	0.001
✓	test19	0.001
✓	test20	0.002
✓	test1	0.002
✓	test2	0.002
✓	test3	0.002
✓	test4	0.001
✓	test5	0.001
✓	test6	0.001
✓	test7	0.003
✓	test8	0.001
✓	test9	0.001



```java
package homework;

public class App {

    public static String NextDate(int day, int month, int year) {
        // Check if the date is within the valid domain
        if (year < 1812 || year > 2212 || month < 1 || month > 12 || day < 1 || day > 31) {
            return "--/--/----";
        }

        // Check for leap year
        boolean isLeapYear = (year % 4 == 0 && (year % 100 != 0 || year % 400 == 0));

        // Check for February and leap year exception
        if (month == 2) {
            if (isLeapYear && day > 29) return "--/--/----";
            if (!isLeapYear && day > 28) return "--/--/----";
        }

        // Check for months with 30 days
        if ((month == 4 || month == 6 || month == 9 || month == 11) && day > 30) {
            return "--/--/----";
        }

        // Calculate next date
        day++;
        if ((month == 2 && ((isLeapYear && day > 29) || (!isLeapYear && day > 28)))
            || (day > 30 && (month == 4 || month == 6 || month == 9 || month == 11))
            || day > 31) {
            day = 1;
            month++;
            if (month > 12) {
                month = 1;
                year++;
            }
```

```java
        // Return the next date in DD/MM/YYYY format
        return String.format("%d/%d/%d", day, month, year);
    }

    public static void main(String[] args) {
        // Example usage
        System.out.println(NextDate(28, 2, 2024)); // Should print 29/2/2024 (Leap year)
    }
}
```