

CompSci 251

Assignment 2

Due Sep. 24th, 2018

Program Specification

This assignment will focus on strengthening your skills with manipulating Strings. The String class provides many useful methods, in which this assignment will give you further practice with. You will read in dates in a variety of different formats, parse the dates, and then print them out in a converted format. The dates will be entered in one line by the user, and they will be separated by the word “and”.

Date formats

Your program will parse a line of dates written in three styles: day first, month first, and year first.

Day first: The day of the month is first, followed by the month, followed by the year. The month string must be at least 3 letters long, and can be a mix of upper and lowercase letters. Spaces should separate the three parts. Spaces are not allowed between numbers or month names.

ex: 8 Aug 2015

24 February 1988

Month first: The month is first, followed by the day, a comma, then the year. The month may be fully spelled out or an abbreviation at least 3 letters long and be upper or lower case. A space must separate the month and day, but there may be multiple spaces between the two.

ex: March 15, 1967

Jan 17, 1990

All numbers: Month, day, and year are entered as numbers with dashes between them. There may be many spaces around the dashes or none. Spaces are not allowed between the numbers.

ex: 12-4-2008

1-8-2003

Requirements

- Your program must read a single line that may contain many dates in many formats separated by the word “and”. It will parse out the dates, and print them all in the correct format. The correct format is day first, a space, followed by the full name of the month with the first letter capitalized, a space, and then the year.
- Your program must give an error message if a date is invalid. It should give a specific error message for the following cases:
 - A month name or abbreviation is misspelled or too short (less than three characters)

- A day or month number is incorrect. Months must be between 1 and 12. Days must be valid for the month (ignore leap years)
 - A year number is too low or high (1900 to 2018).
 - There are too many dashes in All Numbers format.
- It is ok if your program crashes when calling Integer.parseInt and the argument is not valid.

Implementation

There are some specific requirements for how you write the program.

- Use the below build-in methods
 - String class: substring, trim, split, toLowerCase, indexOf, or lastIndexOf
 - Integer class: parseInt
- In the main() method, you must prompt the user to enter the dates, read the line of dates, break the line into separate Strings based on the delimiter ("and"), and for each date print "Date <number>: <correct format>". You must also call the appropriate parser here.
- One method for parsing each kind of date. These methods either output an error message or the corresponding standard date string.
 - public static void parseDayFirst(String dateStr)
 - public static void parseMonthFirst(String dateStr)
 - public static void parseAllNumbers(String dateStr)
- public static boolean isValidMonthDay(int day, int month)
 - returns true if the month and day numbers form a valid day of the year (leap years excluded)
- public static int monthToNumber(String monthStr)
 - Takes a string that should be the name or abbreviation of the month name and returns the number of the month. It returns zero if the name doesn't match any month or is too short. This method is provided for you.
- public static boolean isValidMonthAbbr(String month)
 - Takes a month as a String and returns boolean value. True means the passed in month string represents a valid month string (the length of the trimmed month string is greater than or equals 3, and the month string abbreviation is correctly spelled!!!).
 - Hint: indexOf along with the provided static String array fullNameMonths declared at the top of the class may be useful here for.

- `public static boolean isValidYear(int year)`
 - returns true if the year is valid (1900 to 2018)
- `public static String standardDateString(int day, int month, int year)`
 - takes a day, month, and year as numbers and returns a String containing the date in the correct format, Day Month Year. (ex. 13 March 2006)

Grad Students

You will be responsible for adding 2 or 3 more Junit test cases to each `Testing.java` method listed below. Pay attention to the test cases provided. What are they testing? What are they not testing?

- `test isValidMonthDay1()`
- `test isValidYear1()`
- `test monthToNum()`

Sample Output

Welcome to the CS251 Date Converter!

Enter line of dates: 8 Aug 2015 and March 15, 1967 and 12-4-2008

Date 1: 8 August 2015

Date 2: 15 March 1967

Date 3: 4 December 2008

Goodbye!

Welcome to the CS251 Date Converter!

Enter line of dates: 15 xyz 2000 and 2-2-22222-and J 10, 2004 and Feb 45 , 2012

Date 1: ERROR: Invalid month string

Date 2: ERROR: Too many dashes

Date 3: ERROR: Invalid month string

Date 4: ERROR: Invalid month or day number

Goodbye!

Welcome to the CS251 Date Converter!

Enter line of dates:

ERROR: Empty input line

Goodbye!

Welcome to the CS251 Date Converter!

Enter line of dates:and 2-0-2222 and Jan 13, 1190 and March 12,2222 and 9 - 9-2018

Date 1: ERROR: No date entered

Date 2: ERROR: Invalid month or day number

Date 3: ERROR: Invalid Year-too low or hi

Date 4: ERROR: Invalid Year-too low or hi

Date 5: 9 September 2018

Goodbye!