Name of the Course : Complete Java SE8 Developer Bootcamp

Level : Medium

Tool Stack: Java 8

Problem Statement : A Employer registration application collects employer birth date and need to validate the same . The business rules are the

1. the age should be greater than 18 , if less than 18 print “Too young to apply”. Else print candidate accepted
2. the date format can be one of the three following formats month[2 digit]/date[2 digit]/year [4 digit]
3. the date separators can be dot . ,slash/ and hypen -

Description : Create class DateValidator with following methods .

1.method static boolean validateParseDate(String dateString)

return true if the date is valid .

2.method static int calculateAge(String dateString)

return age .

3.Create a MainClass to have a main method and get the date string as user input and call the validateparserDate(String detestr) and calculateAge(String datestr) .

Test Data1

Sample input:

11.04.1978

sample output:

Valid Date

Test Data2

sample input :

01.01.2020

sample output :

Invalid Date

Code:

**import** java.time.LocalDate;

**import** java.time.Month;

**import** java.time.Period;

**import** java.time.format.DateTimeFormatter;

**import** java.time.format.DateTimeParseException;

**import** java.time.format.ResolverStyle;

**import** java.util.Scanner;

**public** **class** DateValidator {

**public** **static** **void** main(String[] args) {

Scanner sc = **new** Scanner(System.***in***);

System.***out***.println("Enter date in month date year format");

String dateString = sc.nextLine();

**if**( *validateParseDate*(dateString)&& *calculateAge*(dateString) >=18 )

System.***out***.println("Valid Date");

**else**

System.***out***.println("Invalid Date");

}

//Java 8 - Use DateTimeFormatter (thread-safe)

**public** **static** **boolean** validateParseDate(String dateStr )

{

LocalDate date = **null**;

String dateFormat;

**if** (dateStr.contains("-"))

dateFormat = "MM-dd-uuuu";

**else** **if** (dateStr.contains("."))

dateFormat = "MM.dd.uuuu";

**else** dateFormat = "MM/dd/uuuu";

DateTimeFormatter dateFormatter = DateTimeFormatter.*ofPattern*(dateFormat)

.withResolverStyle(ResolverStyle.***STRICT***);

**try** {

date = LocalDate.*parse*(dateStr, dateFormatter);

} **catch** (DateTimeParseException e) {

// e.printStackTrace();

**return** **false**;

}

**return** **true**;

}

**public** **static** **int** calculateAge(String dateStr) {

LocalDate currentDate = LocalDate.*now*();

//Today's date

LocalDate date = **null**;

String dateFormat;

**if** (dateStr.contains("-"))

dateFormat = "MM-dd-uuuu";

**else** **if** (dateStr.contains("."))

dateFormat = "MM.dd.uuuu";

**else** dateFormat = "MM/dd/uuuu";

DateTimeFormatter dateFormatter = DateTimeFormatter.*ofPattern*(dateFormat)

.withResolverStyle(ResolverStyle.***STRICT***);

// DateTimeFormatter formatter = DateTimeFormatter.ofPattern("M/d/uuuu");

LocalDate birthDate;

//convert String to LocalDate

**try** {

birthDate = LocalDate.*parse*(dateStr, dateFormatter); //Birth date

} **catch** (Exception e) {

System.***out***.println(e.getLocalizedMessage());

**return** 0;

}

**if** ((birthDate != **null**) && (currentDate != **null**)) {

**return** Period.*between*(birthDate, currentDate).getYears();

} **else** {

**return** 0;

}

}

}

Junit Testing

**import** **static** org.junit.Assert.*assertEquals*;

**import** org.junit.Test;

**import** handson.DateValidator;

**public** **class** TestDateValidator {

@Test

**public** **void** testValidateDates() {

*assertEquals*(**true** ,DateValidator.*validateParseDate*("11.04.1978"));

*assertEquals*(**false**,DateValidator.*validateParseDate*("22.4.1978"));

*assertEquals*(**true** ,DateValidator.*validateParseDate*("11-04-1978"));

*assertEquals*(**false**,DateValidator.*validateParseDate*("22/4/1978"));

*assertEquals*(**false** ,DateValidator.*validateParseDate*("15.04.1978"));

*assertEquals*(**true**,DateValidator.*validateParseDate*("12.24.2020"));

*assertEquals*(41 ,DateValidator.*calculateAge*("11.04.1978"));

*assertEquals*(41,DateValidator.*calculateAge*("12.04.1978"));

*assertEquals*(41 ,DateValidator.*calculateAge*("11-04-1978"));

*assertEquals*(41,DateValidator.*calculateAge*("12/04/1978"));

*assertEquals*(42 ,DateValidator.*calculateAge*("07.04.1978"));

*assertEquals*(0,DateValidator.*calculateAge*("01.01.2020"));

}

}

Test Data1

Sample input:

11.04.1978

sample output:

Valid Date

Test Data2

sample input :

01.01.2020

sample output :

Invalid Date

Learning outcome: Participant could able to learn how to use LocalDate , DateTimeFormatter classes.