Calculate the difference between two dates

Estimated time needed: 30 minutes

After completing this lab, you will know how to handle addition and subtraction of Dates.

You are currently viewing this lab in a Cloud-based Integrated Development Environment (Cloud IDE). It is a fully-online integrated development environment that is pre-installed with JDK 21, allowing you to code, develop, and learn in one location.

Learning Objectives

After completing this lab, you will be able to:

- Use date manipulation classes to compute the difference in days between these two dates.
- Validate the dates when received as an input from the user.
- Display the difference in dates in days.
- Perform calculations to add days or weeks to a date

Difference between dates

Date is handled with the LocalDate class `java.time` package. In this section of the lab you will write a program that takes the user input for a date and retrieves the difference between the given date and the current date. The time between two dates in years, months and days is handled with a class called `Period` in the `java.time` package.

1. Create a project directory by running the following command.

```
mkdir my_datediff_proj
```

2. Run the following code to create the directory structure.

```
mkdir -p my_datediff_proj/src
mkdir -p my_datediff_proj/classes
mkdir -p my_datediff_proj/test
cd my_datediff_proj
```

3. Now, create a file named `DateDiffCalculator.java` inside the src directory.

```
touch /home/project/my_datediff_proj/src/DateDiffCalculator.java
```

4. Click the following button called **Open DateDiffCalculator.java in IDE** to open the file for editing.

Open **DateDiffCalculator.java** in IDE

5. Read each statement and the accompanying explanations in the following program to understand how to handle input for LocalDate and how to calculate the difference between the current date and the given date. Paste the following content in `DateDiffCalculator.java`.

```
// Import necessary classes for date manipulation, formatting, and user input
import java.time.LocalDate;
import java.time.format.DateTimeFormatter;
import java.util.Scanner;
import java.time.format.DateTimeParseException;
import java.time.Period;
// Define a class named DateDiffCalculator
public class DateDiffCalculator {
    // Method to calculate and return the difference between two dates as a string
    public static String getDiff(LocalDate d1, LocalDate d2) {
        // Calculate the period (difference) between the two dates
        Period period = d1.until(d2);
        // Initialize the result string
        String diffStr = "The difference is ";
        // Add years to the result string if the difference in years is not zero
        if (period.getYears() != 0) {
            diffStr = diffStr + period.getYears() + " years, ";
        }
        // Add months to the result string if the difference in months is not zero
        if (period.getMonths() != 0) {
            diffStr = diffStr + period.getMonths() + " months, ";
        }
        // Add days to the result string if the difference in days is not zero
        if (period.getDays() != 0) {
            diffStr = diffStr + period.getDays() + " day(s)";
        }
        // Return the formatted difference string
        return diffStr;
    }
    // Main method, the entry point of the program
    public static void main(String s[]) {
        // Get the current date
        LocalDate todayDate = LocalDate.now();
        // Define a date formatter for the "dd/MM/yyyy" format
        DateTimeFormatter dateformat = DateTimeFormatter.ofPattern("dd/MM/yyyy");
        // Print the current date in the specified format
        System.out.println("\nThe date is " + todayDate.format(dateformat));
        // Create a Scanner object to read user input
        Scanner scanner = new Scanner(System.in);
        // Prompt the user to input a date in the "dd/MM/yyyy" format
        System.out.println("Input the date in dd/MM/yyyy format");
        String rawDate = scanner.nextLine();
        // Try to parse the user input into a LocalDate object
        try {
            LocalDate givenDate = LocalDate.parse(rawDate, dateformat);
            // Print the given date in the specified format
            System.out.println("\nThe given date is " + givenDate.format(dateformat));
            // Check if the given date is the same as the current date
            if (givenDate.isEqual(todayDate)) {
                System.out.println("Both dates are the same");
            } else {
                // If the given date is before the current date, calculate the difference
                if (givenDate.isBefore(todayDate)) {
                    System.out.println(getDiff(givenDate, todayDate));
                } else {
                    // If the given date is after the current date, calculate the difference
                    System.out.println(getDiff(todayDate, givenDate));
                }
            }
        } catch (DateTimeParseException dte) {
            // Handle invalid date format input
            System.out.println("Invalid input. Please try again!");
        }
    }
}
```

`LocalDate.now()` fetches the current date and time from the system clock.

`getDiff` method calculates the difference between two LocalDate objects (d1 and d2). It uses the Period class to compute the difference in years, months, and days.

The program starts by getting the current date and formatting it as dd/MM/yyyy. It prompts the user to input a date in the same format. The input is parsed into a LocalDate object. If the input is invalid, a DateTimeParseException is caught, and an error message is displayed.
If the input is valid, the program checks if the given date is the same as, before, or after the current date. Depending on the comparison, it calculates and prints the difference between the two dates using the getDiff method.

6. In the IDE, compile the Java program. specifying the destination directory as the `classes` directory that you created. See the following code example.

   ```
   javac -d classes src/DateDiffCalculator.java
   ```

7. Set the `CLASSPATH` variable.

   ```
   export CLASSPATH=$CLASSPATH:/home/project/my_datediff_proj/classes
   ```

8. Now, when you run the Java program, the Java program will run seamlessly as expected.

```
java DateDiffCalculator
```

Here is a sample output:

```
The date is 13/02/2025
Input the date in dd/MM/yyyy format
25/10/2025
The given date is 25/10/2025
The difference is 8 months,12 day(s)
```

Calculate difference between two given dates

Now you will do a similar exercise but this time, take any two dates in a desired format from the user and calculate the difference between then in years, months and days.

1. Click the following button called **Open DateDiffCalculator.java in IDE** to open the file for editing if the IDE is not already open.

Open **DateDiffCalculator.java** in IDE

2. Read each statement in the following program to understand how to calculate the difference between two dates in any format. Paste the following content in `DateDiffCalculator.java`.

```
// Import necessary classes for date manipulation, formatting, and user input
import java.time.LocalDate;
import java.time.format.DateTimeFormatter;
import java.util.Scanner;
import java.time.format.DateTimeParseException;
import java.time.Period;
```

```
// Define a class named DateDiffCalculator
public class DateDiffCalculator {
    // Method to calculate and return the difference between two dates as a string
    public static String getDiff(LocalDate d1, LocalDate d2) {
        // Calculate the period (difference) between the two dates
        Period period = d1.until(d2);
        // Initialize the result string
        String diffStr = "The difference is ";
        // Add years to the result string if the difference in years is not zero
        if (period.getYears() != 0) {
            diffStr = diffStr + period.getYears() + " years, ";
        }
        // Add months to the result string if the difference in months is not zero
        if (period.getMonths() != 0) {
            diffStr = diffStr + period.getMonths() + " months, ";
        }
        // Add days to the result string if the difference in days is not zero
        if (period.getDays() != 0) {
            diffStr = diffStr + period.getDays() + " day(s)";
        }
        // Return the formatted difference string
        return diffStr;
    }
    // Main method, the entry point of the program
    public static void main(String s[]) {
        // Create a Scanner object to read user input
        Scanner scanner = new Scanner(System.in);
        // Prompt the user to enter the date format they want to use (e.g., dd/MM/yyyy)
        System.out.println("Enter the format in which you want to feed the dates in:");
        String dateFormatStr = scanner.nextLine();
        // Create a DateTimeFormatter object using the user-provided format
        DateTimeFormatter dateformat = DateTimeFormatter.ofPattern(dateFormatStr);
        // Try to parse the user input into LocalDate objects
        try {
            // Prompt the user to input the first date in the specified format
            System.out.println("Input the date in " + dateFormatStr + " format:");
            String rawDate1 = scanner.nextLine();
            // Parse the input into a LocalDate object
            LocalDate givenDate1 = LocalDate.parse(rawDate1, dateformat);
            // Prompt the user to input the second date in the specified format
            System.out.println("Input another date in " + dateFormatStr + " format:");
            String rawDate2 = scanner.nextLine();
            // Parse the input into a LocalDate object
            LocalDate givenDate2 = LocalDate.parse(rawDate2, dateformat);
            // Print the first date in the specified format
            System.out.println("\nThe given date is " + givenDate1.format(dateformat));
            // Print the second date in the specified format
            System.out.println("\nThe other given date is " + givenDate2.format(dateformat));
            // Check if the two dates are the same
            if (givenDate1.isEqual(givenDate2)) {
                System.out.println("Both dates are the same");
            } else {
                // If the first date is before the second date, calculate the difference
                if (givenDate1.isBefore(givenDate2)) {
                    System.out.println(getDiff(givenDate1, givenDate2));
                } else {
                    // If the first date is after the second date, calculate the difference
                    System.out.println(getDiff(givenDate2, givenDate1));
                }
            }
        } catch (DateTimeParseException dte) {
            // Handle invalid date format input
            System.out.println("Invalid input. Please try again!");
        } catch (Exception e) {
            //Handle any other exception such as unknown formats
            System.out.println("Invalid input. Please try again!");
        }
    }
}
```

Here's an explanation of the code:

The program starts by prompting the user to enter a date format. If the format input is invalid, the exception is caught, and an error message is displayed. Else, it creates a DateTimeFormatter object using the user-provided format. The user is prompted to input two dates in the chosen format. The program parses the input into LocalDate objects. If the input is invalid, a DateTimeParseException is caught, and an error message is displayed.

If the input is valid, the program checks if the two dates are the same or if one is before the other.

3. Compile the Java program, specifying the destination directory as the `classes` directory that you created.

   ```
   javac -d classes src/DateDiffCalculator.java
   ```

4. Now, run the Java program.

   ```
   java DateDiffCalculator
   ```

Please see the following sample output.

```
Enter the format in which you want to feed the dates in
dd MMM yy
Input the date in dd MMM yy format
02 Feb 25
Input the another date in dd MMM yy format
25 Dec 25
The given date is 02 Feb 25
The other given date is 25 Dec 25
The difference is 10 months,23 day(s)
```

You can try running with various permuations and combinations.

Using ChronoUnit (Optional)

`java.time.temporal.ChronoUnit` is an enum in Java. `Enum` is a special type of class that represents a fixed set of constants. `ChronoUnit` is an `enum` that represents different units of time. It is used to measure the amount of time between two temporal objects (objects that represent various aspects of time), such as LocalDate, LocalTime, or ZonedDateTime.

Here are the different units of time represented by ChronoUnit:

1. NANOS: Represents a nanosecond (one billionth of a second).
2. MICROS: Represents a microsecond (one millionth of a second).
3. MILLIS: Represents a millisecond (one thousandth of a second).
4. SECONDS: Represents a second.
5. MINUTES: Represents a minute.
6. HOURS: Represents an hour.
7. HALF_DAYS: Represents half a day (12 hours).
8. DAYS: Represents a day.
9. WEEKS: Represents a week.
10. MONTHS: Represents a month.
11. YEARS: Represents a year.
12. DECADES: Represents a decade (10 years).
13. CENTURIES: Represents a century (100 years).
14. MILLENNIA: Represents a millennium (1000 years).
15. ERAS: Represents an era (for example, BCE/CE).

You can use ChronoUnit to perform various temporal calculations, such as:

- Calculating the difference between two dates or times.
- Adding or subtracting a specified amount of time from a date or time.
- Determining whether a date or time is before or after another.

1. Click the following button called **Open DateDiffCalculator.java in IDE** to open the file for editing if the IDE is not already open.

Open **DateDiffCalculator.java** in IDE

2. Read each statement in the following program to understand how to calculate the difference between two dates in any format. Paste the following content in `DateDiffCalculator.java`.

```
// Import necessary classes
import java.time.ZonedDateTime;
import java.time.format.DateTimeFormatter;
import java.util.Scanner;
import java.time.format.DateTimeParseException;
import java.time.Period;
import java.time.temporal.ChronoUnit;
public class DateDiffCalculator {
    public static void main(String s[]) {
        // Create a scanner for user input
        Scanner scanner = new Scanner(System.in);
        // Prompt the user for the date format
        System.out.println("Enter the format in which you want to feed the dates in");
        String dateFormatStr = scanner.nextLine();
        // Create a DateTimeFormatter with the specified format
        DateTimeFormatter dateformat = DateTimeFormatter.ofPattern(dateFormatStr);
        try {
            // Prompt the user for the first date
            System.out.println("Input the date in " + dateFormatStr + " format");
            String rawDate1 = scanner.nextLine();
            // Parse the first date
            ZonedDateTime givenDate1 = ZonedDateTime.parse(rawDate1, dateformat);
            // Prompt the user for the second date
            System.out.println("Input the another date in " + dateFormatStr + " format");
            String rawDate2 = scanner.nextLine();
            // Parse the second date
            ZonedDateTime givenDate2 = ZonedDateTime.parse(rawDate2, dateformat);
            // Print the parsed dates
            System.out.println("\nThe given date is " + givenDate1.format(dateformat));
            System.out.println("\nThe other given date is " + givenDate2.format(dateformat));
            // Check if the dates are equal
            if (givenDate1.isEqual(givenDate2)) {
                System.out.println("Both dates are the same");
            } else {
                System.out.println("The diff is " +
                    ChronoUnit.DAYS.between(givenDate1, givenDate2)+" day(s)");
            }
        } catch (DateTimeParseException dte) {
            // Handle invalid date input
            System.out.println("Invalid input. Please try again!");
        }
    }
}
```

`ChronoUnit.DAYS.between(givenDate1, givenDate2)` - Calculates the difference between the two given dates in `days`.

3. Compile the Java program, specifying the destination directory as the `classes` directory that you created.

   ```
   javac -d classes src/DateDiffCalculator.java
   ```

4. Now, run the Java program.

   ```
   java DateDiffCalculator
   ```

Please see the following sample output.

```
Enter the format in which you want to feed the dates in
dd MMM yy HH:mm:ss Z
Input the date in dd MMM yy HH:mm:ss Z format
15 Feb 22 14:30:00 +0930
Input the another date in dd MMM yy HH:mm:ss Z format
15 Feb 22 14:30:00 -1730
The given date is 15 Feb 22 14:30:00 +0930
The other given date is 15 Feb 22 14:30:00 -1730
The diff is 1 day(s)
```

You can try running with various permuations and combinations.

Practice Exercise

Exercise 1

1. Create `ZonalDateTimeOperation` application with same code as used in the lab for `DateDiffCalculator.java`. Add code in it to run infintely, as long as the user wants to.
2. Add code to take date input for different time zone.
3. Add/subtract days, weeks, months or years to the date.
4. Calculate the difference between the current date and newly calculated date.

▶ Click here for solution

Conclusion

In this lab, you learned:

- To compute the difference in days/months/year between two dates.
- To validate the dates when received as an input from the user.
- Perform calculations to add days or weeks to a date

Author(s)

Lavanya