



# CalendarHelper

**Java Demo Project – Date Conversion Utility**

Presented by [Spyridon Mavrommatis] • 2025



# Project Goals

- **Convert dates to day-of-year** (e.g., 30, 6, 2045 → 181st of the year)
- **Convert day-of-year back to full date** (e.g., 69, 2000 → 2/9/2000)



# Example 1 Output

```
"C:\Program Files\Java\jdk-17\bin\java.exe" "-javaagent:C:\Users\Spiri\AppData\Local\JetBrains\IntelliJ I
Choose an option:
  Convert a date to day of year
  Convert a day of year to date

Choose by number: 1 or 2

Your choice: 1
Enter number of day: 30
Enter number of month: 6
Enter year: 2045
30 June 2045 is the 181st day of the year.

Process finished with exit code 0
```

*Input*

*Outcome*



# Example 2 Output

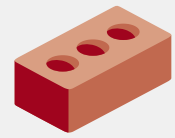
```
"C:\Program Files\Java\jdk-17\bin\java.exe" "-javaagent:C:\Users\Spiri\AppData\Local\JetBrain
Choose an option:
  Convert a date to day of year
  Convert a day of year to date

Choose by number: 1 or 2

Your choice: 2
Enter day of year: 69
Enter year: 2000
The 69th day of 2000 is 3/9/2000.

Process finished with exit code 0
```

*3 inputs*  
*outcome*



# Project Structure

The code is organized in a **modular way** using multiple Java classes:

- `Main.java` → *The Orchestrator*
- `DisplayHelper.java` → *The Director*
- `DateConverter.java` → *The Parser*
- `MathUtils.java` → *The Worker*
- `DateConstants.java` → *The Reference Library*

See: `calendar_modules_architecture.png`



# Technologies Used

- Input handling using Scanner
- Input validation using try / catch
- Control flow with switch-case and while loops
- Use of static utility methods
- Method overloading and decomposition
- Leap year logic using conditional structures
- Use of arrays for month and day data
- Encapsulation of constants in a dedicated class



# Learning Outcomes

- Practiced **modular design** in Java
- Gained hands-on experience with **date manipulation**
- Strengthened understanding of **input validation**
- Applied **control flow, loops, and methods**
- Learned to structure a utility tool from scratch