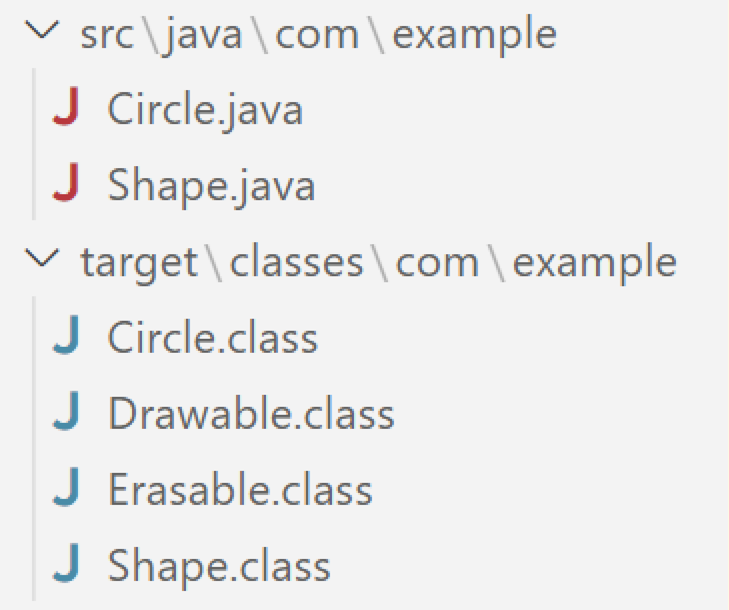
Project Management

Manual Method

Folder Structure

- Terminology:
  - Source folder: Root folder for source code
    - The package names are calculated relatively from this root
  - Destination folder: Root folder for compiled classes
  - Example (see figure):
    - Source folder: `src\java`
    - Destination folder: `target\classes`
    - In `Shape.java`: `package com.example;`
- Compile with source and destination folder information:
  - `javac -s source\folder -d destination\folder source\folder\x\y\File.java`



```
src\java\com\example
    Circle.java
    Shape.java
target\classes\com\example
    Circle.class
    Drawable.class
    Erasable.class
    Shape.class
```

Command Line

- Compile with external libraries:
  - `javac -cp path\to\lib-file.jar;. ClassName.java`
  - `javac -cp "path\to\lib-file.jar;path\to\lib-folder\*;." ClassName.java`
  - Use `:` instead of `;` in Linux
- Run with external libraries:
  - `java -cp path\to\lib-file.jar;. ClassName`
  - `java -cp "path\to\lib-file.jar;path\to\lib-folder\*;." ClassName`
- Create a JAR file:
  - Prepare a subfolder inside the project directory with all necessary resources
  - `jar cf file-name.jar all-input-files`

With VS Code

Project Management

- With help of the **Extension Pack for Java**
- The **Java Projects** view in **Explorer** panel provides some useful functionalities:
  - Add a class: Use + button next to the **Referenced Libraries** tab
  - Create JAR file: Use ↦ button
  - Manage dependencies: Use + button next to the **Referenced Libraries** tab
  - Debugging
- More functions can be found in the **Command Palette**:
  - Press Ctrl+Shift+P then type `>java`

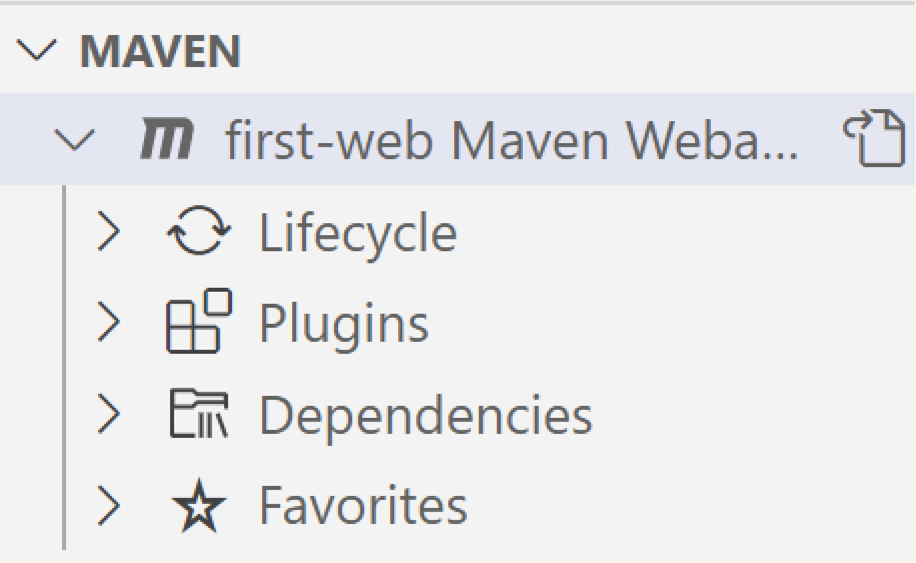
IntelliSense

- For code completion, error checking, documentation as you type
- You can change some settings to help it work correctly: choose source and destination folders, referenced libraries:
  - **Command Palette ➔ Java: Configure Classpath**
  - Or, press ··· button next to **Java Projects** view and choose **Configure Classpath**
  - Or, edit the `.vscode\settings.json` file directly (if you know what you are doing)

With Maven for VS Code

Maven

- Maven is a project manager and build tool, it can work from the command line
  - It uses a `pom.xml` file for project settings
- **Maven for Java** extension is included in the **Extension Pack for Java**, that exposes some functionalities:
  - Create a project: **Command Pallete ➔ Java: Create Java Project…**
  - Add dependencies: **Command Pallete ➔ Maven: Add a dependency**
  - See the Maven tab for more functions: manage dependencies, packaging,…


MAVEN
first-web Maven Weba...
Lifecycle
Plugins
Dependencies
Favorites