DP 1: Singleton

Pattern: The pattern used is **Singleton** to avoid accidentally initializing the DateHelper module multiple times. The functionalities stored in DateHelper should only need one instance of the class to exist for the whole program. All information and functionalities in DateHelper should only exist in one instance so that the program can access it anywhere without problems or coupling issues.

Location in Source Tree: src 🡪 DateHelper.py

DP 2: Builder

Pattern: The pattern used is **Builder** for the Report functionality. By using the ReportBuilder and a Report classes, I’m able to build the specifics for a Report() command then simply run the execute() method once the object is built. This also prevents me from having to directly access attributes in the Report() class.

Location in Source Tree: src 🡪 ReportBuilder.py ***and*** src 🡪 Report.py

DP 3: Singleton

Pattern: Similar to DateHelper, I used the **Singleton** design pattern for TimeHelper to avoid having the class initialized more than once anywhere in the program. Any information and functionalities in TimeHelper should only exist in one instance so that it can be used safely throughout the program.

Location in Source Tree: src 🡪 TimeHelper.py