Design For the Tool Rental Application

Overview

- The requirements for this program are spelled out in the functional specification provided. The name of the functional specification is:

```
Technical Assessment (Architecture).pdf
```

- Since this functional specification details the requirements for this application, this document will focus on the code design.
- To keep the design and code as elegant as possible, only Java code is going to be used for this project so that running the program is simpler and does not depend on unnecessary dependencies and libraries. The code should run as-is based on the main function provided. I felt that anything more would introduce unnecessary complexity and burden the user of this code with unnecessary setup steps.

The Computation

- The computation should begin with the day count entered by the user and figure out the number of days the renter is going to be charged for. This is the day count potentially only includes weekends, weekdays, and holidays as configured by the Price information from the functional specification. This can be different for the Ladder, Chainsaw, and Jackhammer.
- The computation should proceed to figure out the total charges prior to the discount being applied.
- The computation should them figure out the discount amount based on the discount percent entered by the user and the total charges.
- The computation should use the discount amount and the total charges to figure out the final change the renter will owe. The total charge is the price of the rental based on the number of chargeable days multiplied by the daily charge, minus the discount amount which is a percentage of the total charge.
- Exact values should be provided in addition to values rounded to the nearest cent.

Design Details

- For code comments that are not at the beginning of the line, tabs should be converted to spaces (so they look nice regardless of the tab stops.)
- For code comments at the beginning of the line, tabs are fine as they will look nice regardless of the tab stops configured by the system.

Tool Class

- The Tool class represents one tool and contains the main tool data as described in the functional specification that includes the:
    - tool code
    - type
    - brand
- Getters and Setters should be provided for these data elements.

- A default constructor should be provided that initializes the instance variables to be null. The values should be set prior to generating the rental agreement or the non-default constructor should be used. This ensures that values will be valid otherwise an exception will be thrown.

Price Class

- The Price class contains more detailed information about the tools that includes the:
    - tool type
    - daily charge
    - weekday charge
    - weekend charge
    - holiday charge
- Note the tool type in the Price class should be associated with the tool code detailed in the tool class. This associates the prices with the correct tool.
- Getters and Setters should be provided for these data elements.
- A default constructor should be provided that initializes the instance variables to be null. The values should be set prior to generating the rental agreement or the non-default constructor should be used. This ensures that values will be valid otherwise an exception will be thrown.

CheckoutInfo Class

- The CheckoutInfo class is the class that stores the tool and price data used by this application.
- Tools and Prices should be stored in an ArrayList.
- Getters should be provided to access the Tools and Prices.

RentalProcessor Class

- This class should be the main place for business logic used by the tool rental application.
- A function, `processRental` should be provided to generate the necessary data used by the application. Please see the section on “The Computation” for more details on how this function should behave.
- This class should also provide a function `displayRentalAgreement` for displaying the values per the specification of the Rental Agreement.
- An instance variable, `processed` should be provided to ensure a rental agreement is only processed once for efficiency and ease-of-use.

ToolType Class

- This class is a simple enum to store the tool types in one place in order to avoid the duplication of data.

ToolRentalUtil Class

- Logic should be provided to determine if a date lies on Independence Day or Labor Day, a weekday, or a weekend.
- Independence Day and Labor Day only are considered holidays by this application. The renter is not charged for holidays or weekends unless the Price information says otherwise.
- A function, `prettyFormat` should be provided to format a string with proper capitalization.
- A function, `getToolTypeDisplay` should be provided to nicely format the Tool Type enum.

ToolRentalMain Class

- This class should contain the entry point for this program as well as functions to interact with the user.
- This class should also contain functions for dealing with the data entered by the user, calculate the checkout date as a date object, the discount percent, and the count of all days in the range specified by the user, which include weekends and holidays, and the chargeable days, which should not include weekends or holidays unless the Price data indicates otherwise.