OOAD_project3

Names

Corey Stephens

Daniel Taylor

Language + environment

- Java 8
- Junit 5
- Intellij IDE

Assumptions

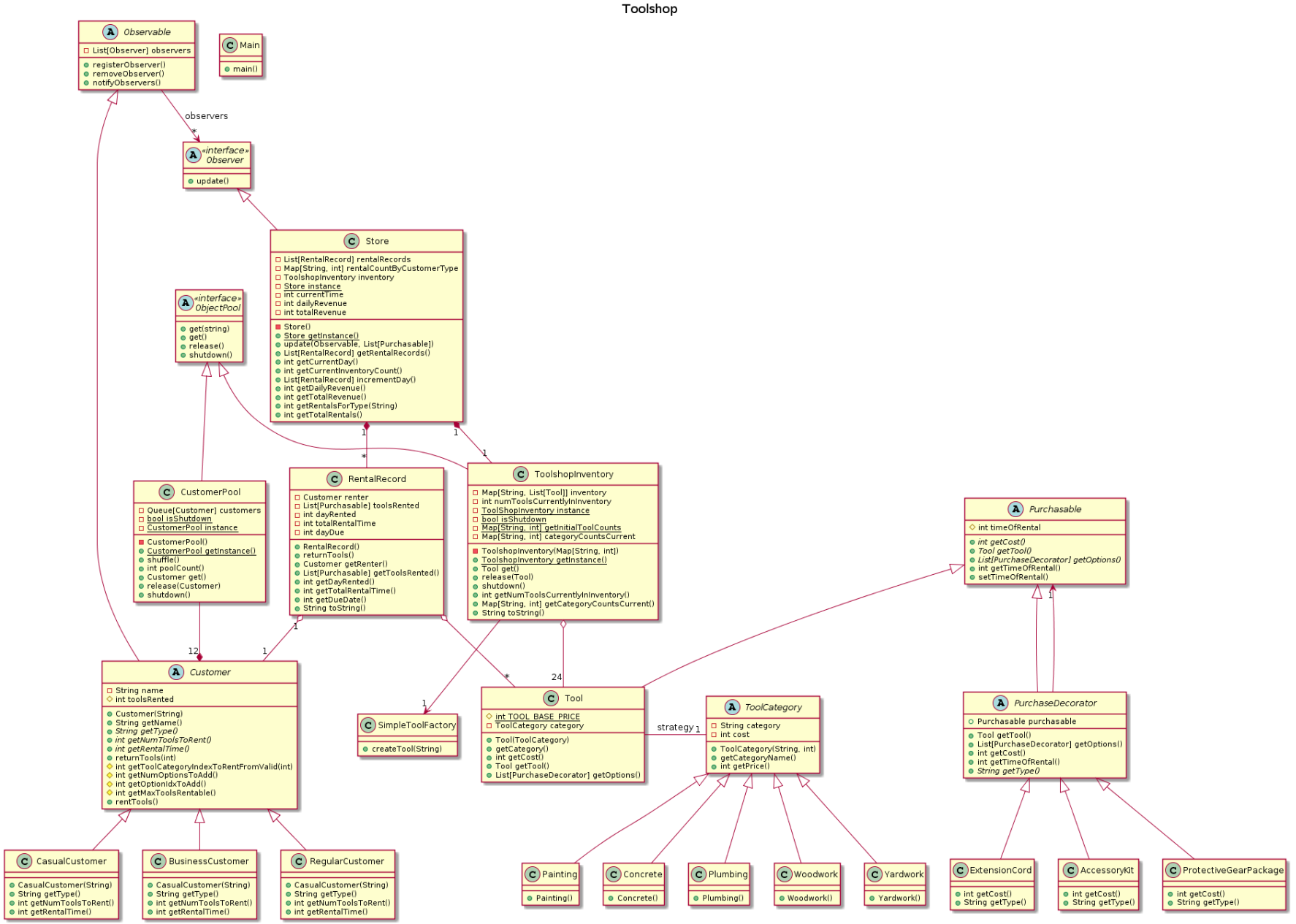
The general design of this program is best understood from the UML diagram (see plantuml for source, class_diagram.svg for a vector image, and class_diagram.png for a png image). As for assumptions, we made as few as seemed reasonable:

- A customer is said to “enter” the store when they begin to add tools to their purchase, not when the method rentTools() is first called. This allows for the program to decide whether a not a customer can rent on a customer-by-customer basis.
- A tool rental has a “base price” of $10 per night rented.
- The quantities and nightly prices for each tool in the inventory (at the beginning) are:
    - 1 concrete tool at $40/night
    - 8 painting tools at $20/night
    - 5 plumbing tools at $30/night
    - 6 woodwork tools at $25/night
    - 4 yardwork tools at $15/night
- The options available for each tool are:
    - Accessory kit for $5 extra total
    - Extension cord for $2 extra total
    - Protective gear package for $1 extra total

Patterns used

- Object pool
  - The tool shop inventory (composed of tools)
  - The customers
- Observer
  - The store acts as an ovserver to each customer so it can create rental records when they make a purchase.
- Decorator
  - Options are added to each tool using decorator.
- Strategy (for deciding which category a tool is in)
  - The tool class delegates its category behavior to the ToolCategory family of classes using strategy.
- Template (for the different types of customers)
  - The abstract Customer class acts as a template for the concrete types of customers, requiring implementations for a method which tells the main renting algorithm how many tools this specific type of customer wants to rent and for how long.
- Singleton
  - The store.
- Simple Factory
  - The SimpleToolFactory allows us to create tools of different categories by simply passing in a string with which category we wish to create.

UML



Class Diagram