PROO/SEW 3 - Assignment: Christmas Observer

Objective

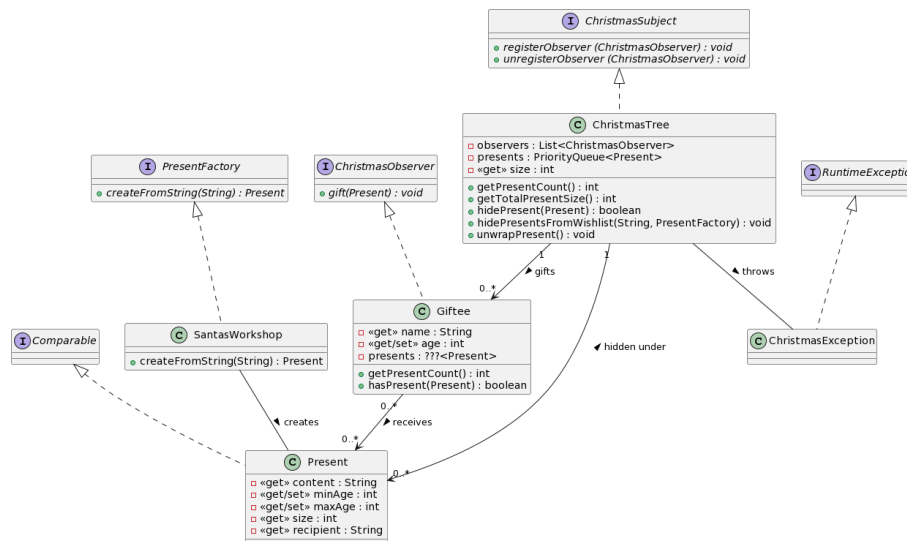Implement the Christmas gift-giving using the *Observer pattern*.

Things To Learn

- Creating a *Maven* project.
- Implementing the *Observer* and *Factory* patterns.
- Importing, configuring and logging with *Log4j 2*.
- Parsing poorly structured data.

Submission Guidelines

- Your implemented solution as **zipped** *IntelliJ*-project.

Task

Implement the traditional ceremony on Christmas eve consisting of *presents* beneath a *tree*, that is *observed* by the *giftees*.

The program consists of the following parts, listed in the suggested order of implementation:

1. `Present`
   - Apart from the `content` and an *optional* label with the name of a `recipient`, each *present* has a *positive* `size` attribute.

- Additionally, each present specifies a minimum and maximum age of the recipient between 1 and 99. Of course, the maximum age can't be below the minimum age.
    - If an illegal age is passed to the according *setters*, the original age remains unchanged!
- Presents can be compared according to their size, with smaller presents coming first.

2. `SantasWorkshop`
- *Santa's workshop* is a concrete implementation of the `PresentFactory` *interface* (allowing easy extensibility). This legendary *factory* allows creating presents based on their `String` representation from a wishlist:
    - The size is graphically represented as a *one-dimensional gift-wrap* with a ribbon on top - e.g. `[===]*` for a *present* with size 3 or `[=====]*` for size 5.
    - It is followed by the *content* and the *age* range in parentheses: If the *max age* is 99 it's just the *min age* followed by a `+`, else both ages are given, seperated by a `-`.
    - Optionally, the present's *recipient* is specified after the word `for`.
- If something *unexpected* happens when producing the present, no *exception* should be thrown - this is where *logging* will come into play.

3. `Giftee`
- A *giftee* has a name, an age *capped* between 1 and 99, and a collection of unique, received gifts.
- Each *giftee* implements the `ChristmasObserver` *interface*, and thus has to implement the `gift`-method, that is called when presents are unwrapped.
- A *giftee* knows that a present is for them, if they fit into the age range of the gift and the label either says their name or is *blank*.

4. `ChristmasTree`
- A *christmas tree* is defined by its `size`, that limits the number of presents that can be hidden underneath (in a `PriorityQueue`).
- Additionally it implements the `ChristmasSubject` *interface* for registering and unregistering giftees (i.e. `ChristmasObserver`s), that can be stored in a simple *linked list*.
- Presents can be hidden under the tree using `hidePresent`, if there's still enough space and *the present has not been added yet*.
    - It's a disaster if somebody else already hid the same present - triggering an *unchecked* `ChristmasException`.
    - Use a `boolean` to indicate whether adding the present was successful.
- Multiple presents from a *wishlist* - a `.txt`-file is provided for this assignment - can be added at once using `hidePresentsFromWishlist`. To allow extensibility the `PresentFactory` to be used for parsing is also passed as parameter.
    - While faulty lines should be ignored (see above), a possibly occuring `IOException` should be wrapped in a `ChristmasException`

and rethrown with an appropriate exception message.
- The actual *gift-giving* is done by calling `unwrapPresent`. The method removes the smallest present from the pile (if there are still presents left) and notifies all the potential giftees.
5. *Log4j 2*
   - Implement *logging* for the methods `hidePresent`, `hidePresentsFromWishlist`, `createFromString` and the *setters* inside `Present` using *Log4j 2*.
   - Play around with different *logging levels* to enable the distinction between the importance (or *severity*) of various events - from `FATAL` for the most critical errors up to `TRACE` for e.g. *parsing* details.
      - An *exemplary* log is provided as inspiration!

Merry Christmas!