

Programming Assignment 2

Due at your recitation session on September 8-12, 2014

Reading

- Chapter 14 in the textbook
- Guidelines for the use of strings, floats, doubles, and BigIntegers (covered in class)
- Section 19.1 in the textbook (excluding “Forming Boolean Expression Positively”, “Guidelines for Comparison to 0”, “In C++, consider ...”)
- Section 15.1 (“Plain if-then Statements” only)
- Sections 17.1, 19.4
- Section 17.3 (“Rewrite with a status variable” only)

Programming

You are in charge of developing the data structures for the new Linked With social network.

The class `User` represents a user of the social network. For the time being, the `User` consists only of an identifying number, but the project will add more data as the social network evolves. The class `User` implements the following methods:

- `User()` creates a new invalid user. It must throw no exception.
- `boolean setID(String id)` sets the user identifier and marks the user as valid. If the identifier had already been set (that is, if the user was already valid), it does nothing and returns false, otherwise it returns true. If the identifier is null, throw a `NullPointerException`.
- `String getID()` returns the identifier corresponding to the current user, or null if the user is still invalid.
- `boolean isValid()` returns whether this user is valid.
- `String toString()` returns a human readable code for this user. If the user is invalid, the method should return the String “Invalid User: Uninitialized ID”.

Two users can be connected with a `Link`. A `Link` is between two users and is never really removed. Instead, the links are labelled with a series of `Dates` that represent the times when the connection was established and removed. For example:

Dates	Meaning	Status (Date)
1/1/2014	Link established on January 1	Active (1/1/2014)
1/1/2014, 2/1/2014	Link established on January 1, torn down on February 1	Inactive (2/1/2014)
1/1/2014, 2/1/2014, 3/1/2014	Link established on January 1, torn down on February 1, re-established on March 1	Active (3/1/2014)
1/1/2014, 2/1/2014, 3/1/2014, 3/1/2014	Link established on January 1, torn down on February 1, re-established on March 1, and torn down again on the same day	Inactive (3/1/2014)

The `Link` class implements the following methods:

- `Link()` creates an invalid link. It throws no exception.
- `boolean setUsers(Set<User> users)` set this `Link` to connect the two users and marks this `Link` as valid. If this link had already been associated with two users (that is, if the link was already valid) or if the `users` argument is anything but two distinct users, it does nothing and returns false, otherwise it returns true.
- `boolean isValid()` returns whether the current link is valid.
- `Set<User> getUsers()` returns the two connected users. If the link is invalid, throw a `UninitializedObjectException` (see below).
- `boolean establish(Date date)` establish the link at the given date. If the link is invalid, throw a `UninitializedObjectException` (see below). If the link was already active or if the given date precedes the last date on record, make no change to the link and return false. In case of success, it returns true.
- `boolean tearDown(Date date)` tear down the link at the given date. If the link is invalid, throw a `UninitializedObjectException` (see below). If the link was already inactive or if the given date precedes the last date on record, make no change to the link and return false. In case of success, it returns true.
- `boolean isActive(Date date)` returns whether this link is active at the given date. If there are multiple events on the same date, the last event

on that date will determine whether the link is active or not. If the link is invalid, throw a `UninitializedObjectException` (see below).

- `Date firstEvent()` returns the date of the first event recorded in this `Link`. It returns null if no event has been established yet, and it throws a `UninitializedObjectException` if the link is invalid.
- `Date nextEvent(Date date)` returns the date of the next event after the date given as argument. It returns null if no such event exists, and it throws a `UninitializedObjectException` if the link is invalid.
- `String toString()` returns a human readable code for this `Link`. If the `Link` is invalid, the method should return the String “Invalid Link: Uninitialized IDs”.

These methods throw a `NullPointerException` if any argument is null.

The `UninitializedObjectException` class is a subclass of `Exception` with the four standard constructors:

- `UninitializedObjectException()`,
- `UninitializedObjectException(String message)`,
- `UninitializedObjectException(String message, Throwable cause)`, and
- `UninitializedObjectException(Throwable cause)`.

The class `SocialNetwork` contains the collection of all users in the social network and their connections. It implements the following methods:

- `SocialNetwork()` creates a social network with no users. It must throw no exception.
- `boolean addUser(User user)` adds the given user to the user set. It returns true if the collection changed because of the call and false if the `SocialNetwork` already contains a user with the same identifier.
- `boolean isMember(String id)` returns true if the `SocialNetwork` contains a user with the given identifier, false otherwise.
- `User getUser(String id)` returns the user with the given identifier, or null if no such user is present in the social network.
- `boolean establishLink(Set<String> ids, Date date)` establish a link between the two users with the given identifiers at the given date. In case of error (if `ids` is anything but two distinct users that are already present in this `SocialNetwork`, if the given date precedes the last date on record) or if the link was already active, make no change to this `SocialNetwork` and return false. In case of success, it returns true.
- `boolean tearDownLink(Set<String> String ids, Date date)` tear down the link between the two users with the given identifiers at the given date. In case of error (if `ids` is anything but two distinct users that are already present in this `SocialNetwork`, if the given date precedes the last

date on record) or if the link was already inactive, make no change to this `SocialNetwork` and return false. In case of success, it returns true.

- `boolean isActive(Set<String> ids, Date date)` returns whether the link between the users with the given identifiers exists and is active at the given date.

These methods throw a `NullPointerException` if any argument is null. Links are bidirectional: a link from `id1` to `id2` is to be regarded as the same as a link from `id2` to `id1`.

Additionally, these classes may contain as many auxiliary private methods as you see fit, and additional helper classes may be defined.

Your classes must be documented in JavaDoc. Test cases must be created in JUnit to test all its public methods.

Discussion Guidelines

The class discussion will focus on:

- High-level code organization
- The design and documentation of the implementation
- Straight-line code, conditional code

On-Line Submission

Bring a copy to recitation to display on a projector. Additionally, submit an electronic copy of your program to blackboard. Turn in with your code:

- A README file explaining how to compile and run the code.
- A Make or Ant build file, with at least two targets:
 - Compile the code without running it (for example, `javac`, `ruby -c`, etc.)
 - Run the test cases
- A comment at the top of the file containing your name, email address, a one-sentence description of the file, and if necessary a longer comment describing the design of the file.

The code should be handed in a zip, tar.bz2, or tar.gz archive. Archives in 7z cannot be accepted.