CSE 2221 - Project 4

Important note: The next project will build directly on your solution to this project. It is essential that you get a decent and working solution for this project before you start working on the next one. Please do not delay work on this project because a failure on this one is likely to result in a failure on the next one as well.

Task

Gain familiarity of the **XMLTree** class, **while** loops, static methods, and RSS technology by parsing an RSS feed URL into HTML

Original Project Instructions

Project 4 Instructions from CSE2221 Project Site

Program Requirements

- Ask the user for a URL of an RSS 2.0 feed
- Ask the user for the name of an output file, it is the users responsibility to include the ".html" in the output file name
- Reads the RSS 2.0 feed from the URL into an **XMLTree** object
- Parses through the created **XMLTree** object to generate a neatly formatted HTML output
- HTML output requirements:
  - The `<channel>`'s `<title>` as the HTML page title (or "Empty Title" if the `<title>` tag has no children)
  - In the page, a heading tag of the `<channel>`'s `<title>` that is a link to the `<channel>`'s `<link>` tag (or "Empty Title" that links to the `<channel>`'s `<link>` tag if the `<title>` tag has no children)
  - A paragraph tag of the `<channel>`'s `<description>` (or "No Description" if the `<description>` tag has no children)
  - A table with each row being one `<item>` tag and the following columns:
    - Publication date, if present, or "No Date Available"
    - Source, if present, which should be a link to the source's url, or "No Source Available"
    - Title, if present and not empty (i.e. has a text child), which should be a link to the `<item>`'s `<link>`
    - If the title is not present or is empty, use the description, which should still be a link to the `<item>`'s `<link>`
    - If neither the title or description are present and not empty, use "No Title Available" and still link to the `<item>`'s `<link>`
- An example of the HTML output is given here. Remember the "View Page Source" hint from the first project!

Example RSS Feed

```
<rss version="2.0">
  <channel>
    <title>Title goes here</title>
    <link>Link goes here</link>
    <description>Description goes here</description>
    <item>
      <title>Optional title goes here</title>
      <description>Optional description goes here</description>
      <link>Optional link goes here</link>
      <pubDate>Optional publication date goes here</pubDate>
```

```
      <source url="the source URL">Optional source goes here</source>
      ...
    </item>
    ...
  </channel>
</rss>
```

Properties of an RSS 2.0 feed

- Children of `<channel>` tag can occur in any order (i.e. child(0) could be a `<title>`, it could be a `<link>`, or it could be a `<description>`, etc.)
- Children of `<item>` tag can occur in any order (i.e. child(0) could be a `<pubDate>`, it could be a `<source>`, or it could be a `<title>`, etc.)
- Other children may exist besides the ones given above, however we will not do anything with them
- The following children of a `<channel>` tag must exist, therefore you may assume they are present:
  - `<title>`
  - `<link>`
  - `<description>`
- The following children of a `<channel>` tag may be blank (i.e. they may not have a text child):
  - `<title>`
  - `<description>`
- All children of a `<item>` tag are optional and don't have to exist, therefore you cannot assume they exist and must check for their presence first before accessing
- Either a `<title>` tag or a `<description>` tag must exist as a child of an `<item>` tag
- Even if a `<title>` tag or a `<description>` tag exists, they may be blank (i.e. have no text child)
- If a `<source>` tag appears as a child of an `<item>` tag, it must have a `url` attribute

Things to Keep in Mind

If your program successfully creates a **XMLTree** object from the user-inputted URL, all you know is that the input was a valid XML document. You must still check the following:

- The root label of the **XMLTree** is a `<rss>` tag
- The `<rss>` tag, if found, has a `version` attribute and the `version` attribute has a value of "2.0"

Once the above conditions have been checked, you can assume the input is a valid RSS 2.0 feed and has the structure shown in the example. In other words, the `<rss>` tag has a `<channel>` child and the `<channel>` tag has `<title>` , `<link>` , and `<description>` children, you can assume these children exist

Do not make any other assumptions. In particular, make sure to check for the presence of children of `<title>` and `<description>` tags before attempting to access these children

If a `<link>` or `<pubDate>` tag exists, the children of these tags must exist and you can assume they exist

Check slide 9 of the RSS slides for a snapshot of all these requirements

Steps

1. Copy and paste *ProjectTemplate* to create a new project folder for this project
2. Name the project *RSSReader*
3. Open the *src* folder, then open *(default package)*
4. Pick any of the four files to keep, delete the other three
5. Rename the kept file to *RSSReader.java*
6. Open *RSSReader.java*
7. Remove all code from *RSSReader.java*
8. Go to this page, copy all the code there, and paste into *RSSReader.java*
9. Update the JavaDoc comments above the class declaration (i.e. author name, leave the program description)
10. Fill in the methods in the program skeleton to satisfy all program requirements given in the Program Requirements section
11. Create a zip file of your *RSSReader* project
12. Rename the zip file (not your project folder) using the naming scheme "FirstName_LastName_DotNumber_ProjectNumber.zip", for example mine would be "Logan_Frank_580_4.zip"
13. Submit to Carmen