CST8132 Lab Assignment 04 – To Do List v2

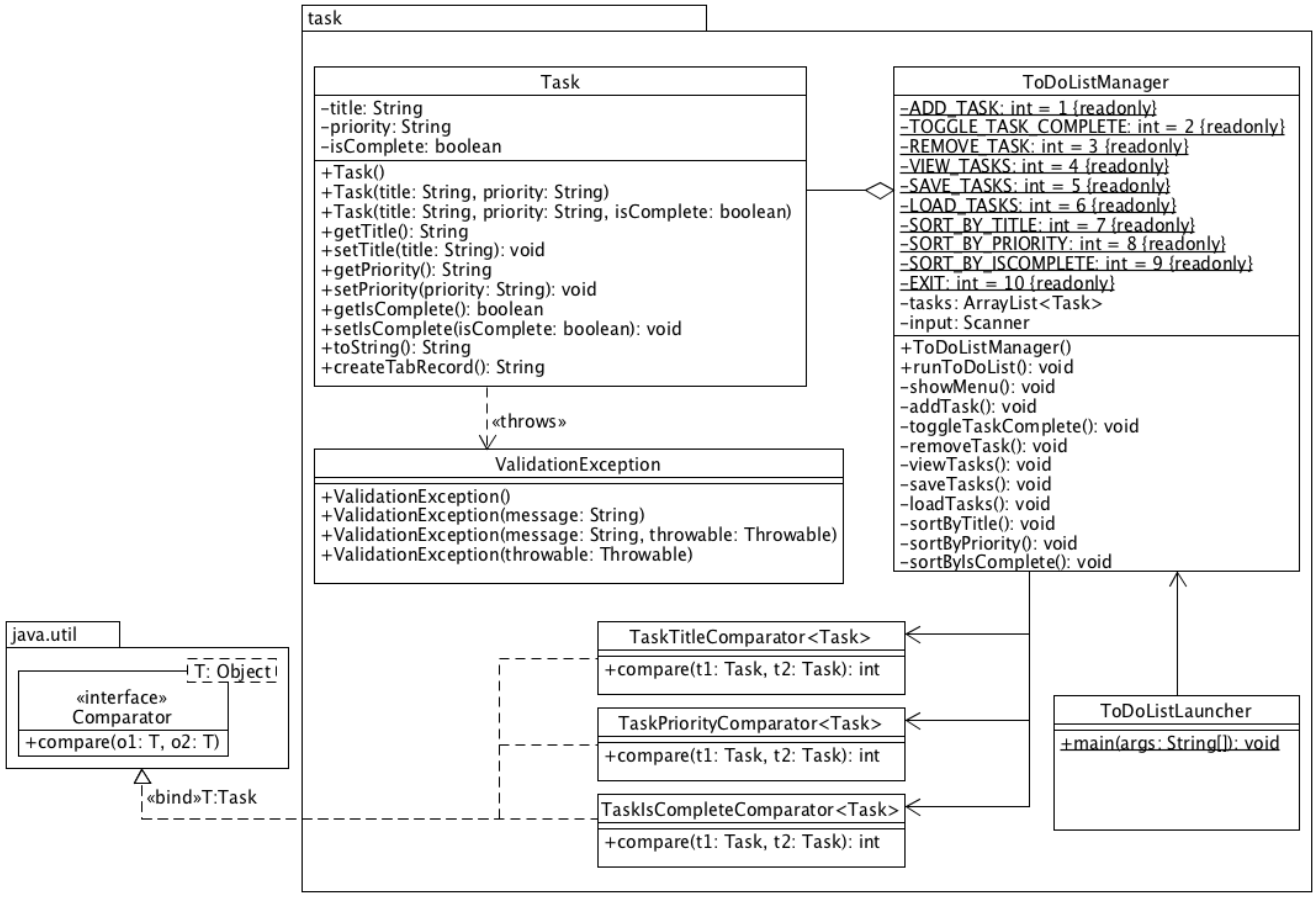
Due: Friday April 22, 2016 by 11:59pm as upload to Blackboard

Problem Description

- In this assignment you will add functionality to the To-Do-List from Assignment 3 so that:
  - The Program will save to, and load records from a text file
  - Comparators can be used to Sort Tasks by title, priority, and isComplete
  - The comparators will be made available to you; you need to use them for sorting.
  - The JUnit test file has been updated with more tests for class Task

Program Design (Updated)

(Diagram created using UMLet)



Implementation Notes

Class Task

- Add an overloaded constructor that has 3 parameters, one per field
- Update the validation for Title to throw ValidationException if the user enters a tab character as part of a Title, e.g. “There are tabs here”
  (Priorities must be “high”, “medium” or “low”, this already prevents tab characters)
- Add a method named createTabRecord() that returns a String with the Title, Priority, and IsComplete separated by Tab Characters. Tip: Don’t forget about \t
  (We will be saving Tasks as text-based records, using tabs to separate fields)
  Tip: The **String** class has a **method** named **contains()**, that you can **pass "\t" into** to check for tabs.

Class ValidationException / Class ToDoListLauncher

- No changes needed to these files from Assignment 3

Class ToDoListManager

- Updates Needed:
  - Update the class constants adding the new ones as per the UML above
  - Update the runToDoList and showMenu methods to align with the updated constants
  - Add methods as seen in the UML for the new program features: saveTasks(), loadTasks(), sortByTitle, sortByPriority(), sortByIsComplete()
- When saving
  - If there are no tasks in the ArrayList tell the user there is nothing to save
  - If there are tasks, save them into a file named "Tasks.txt" replacing the older file if it exists Loop over the ArrayList and write one line per Task into the file
  - Tip: write lines to the file, each one with the values for title, priority, isComplete as Strings separated by tabs (but no spaces between values) See method createTabRecord above
- When loading
  - Clear the ArrayList of any Tasks
  - Read the records in from the text file (tip: loop)
  - Split each line into a String array, splitting on the tab character
  - Create a new Task using the data spit from the line
  - Add the new Task to the ArrayList
- When sorting
  - If there are no tasks tell the user there is nothing to sort
  - Use Collections.sort on the ArrayList with the appropriate Comparator in each method

Tip:

- This program **still** uses one Scanner(System.in) with a mix of calls to nextInt() and nextLine(), make sure you call nextLine() after every nextInt() to remove leftover line-terminator characters from the input Stream (System.in).
- Also, if an InputMismatchException is thrown and caught make sure your nextLine() on the Scanner to clean out bad data.

Assignment Tasks

- An Updated Unit test file is provided, it tests class Task for:
  validation for no Tab characters, the createTabRecordMethod(), the new constructor
- Use the UML class diagram, comparator code, and implementation notes to update the program.
- Find one web tutorial on file-IO in Java and cite with APA style.
- Find one web tutorial on interfaces and interface based polymorphism in Java and cite with APA style.
- Comment all parts of the program, old code, new code, with Javadoc comments.
  - **NOTE: You are not required to write comments for the Unit Tests, or the Comparator classes**
- Generate Javadoc docs for the project, with private members documented also
- Write one-page essay, using MS Word, that addresses the discussion questions (below) as a guide to discuss what was accomplished in the assignment, cite and reference any sources used online to help you answer questions. (You may use more than one page, but professors will stop reading after 3 pages).

Discussion Questions (Cite and reference any sources you use)

- What happens if you do not write source code to close an open file in a computer program?
    - What is a resource leak?
    - Why are resource leaks bad?
- Did you use try-catch-finally with file.close() or try-with-resources?
    - What are some advantages and drawbacks of either approach?
- Is Collections.sort(List<T>, Comparator<T>) an example of Inheritance based polymorphism or interface based polymorphism?
    - How do you know?

Submission Requirements

- Place source code files, essay document, and generated Javadoc docs folder into a zip archive and upload it to Blackboard by the due date.
- The essay should be an electronic document, Microsoft Word format, include your name inside the document.
- Your lab professor will indicate any additional submission requirements to you in the lab

Notes on citations and references

- Please do not cite or reference other students, ask for the original sources from them and cite and reference those instead
- You will not get credit for an assignment or project if large portions are copied directly from other sources, even if you cite and reference the source(s) correctly. Assignments are to be your own original work; other works can be used for help in solving problems or as small pieces of your larger program. Determinations on this are up to the discretion of the professor, if in doubt check with your professor.
- Note: One exception to this is in the case of lecture handouts, and code samples posted to Blackboard. You are free to use these as a starting point just cite + reference them as a personal communication from your professor e.g. Stanley Pieda (2015) personal communication.

UML Class Diagram refresher / review:

Donald Bell. (2004). UML basics: The class diagram. Retrieved from http://www.ibm.com/developerworks/rational/library/content/RationalEdge/sep04/bell/#N100A5

JUnit Testing – Learning Resources (See Blackboard for larger list)

Lars Vogel. (2015). Unit Testing with JUnit – Tutorial. Retrieved from http://www.vogella.com/tutorials/JUnit/article.html

Java Coding Conventions (Reference only)

Oracle. (April 20, 1999). Code Conventions for the Java Programming Language. [webpage] Retrieved from http://www.oracle.com/technetwork/java/index-135089.html
You will need to follow links and use the table of contents.
(PDF version available here: http://www.oracle.com/technetwork/java/codeconventions-150003.pdf)

Grading Guide

Criteria* (Equal Weight)	Needs Work (0)	Poor (1)	Intermediate (2)	Excellent (3)	Value Scored
Code Conventions	Java coding conventions are not followed	Java coding conventions are not well followed.	Java coding conventions are followed with several inconsistencies	Java coding conventions are closely followed.	
Comments **(Javadoc required Assign 4.) (Comments for unit test class and Comparators not required.)**	Javadoc comments missing or incorrect.	Many classes and / or class members missing meaningful Javadoc comments	Very few classes and / or members missing Javadoc comments, Javadoc comments are meaningful. @param, and @return used where expected	Nearly everything is Javadoc commented, Javadoc comments are meaningful, brief, well written. @param, @return, @author, @version, @since all used where expected	
Generated Javadoc files	Missing	Only one or two classes had Javadocs generated	Javadocs generated for all classes but private members not output into the documentation	Javadocs generated for all classes, private members included within Javadocs	
Discussion	Missing or only repeats the questions	Provides brief answers to questions with no explanations or examples	Questions are answered, explained, and examples are used from the code to illustrate	Questions are answered and explained with examples from code, student discusses how concepts in this program will be beneficial in future programing.	
Citations Minimum 2 references (File-IO, Interface Polymorphism)	No referenced work cited when it needed to be. See Algonquin College Policy AA20 on Plagiarism.	References and citations present but not APA style, e.g. only the URL provided.	References and citations loosely follow APA style	References and citations closely follow APA style	
Compiles	Program does not compile, too many syntax mistakes for the professor to track or debug without major re-write.	Program does not compile, has several syntax mistakes	Program does not compile, has a few small syntax mistakes	Program compiles	
Programming Concepts (File-IO, Interface Polymorphism via API-based sorting)	Concepts are not demonstrated, program does not follow UML design.	One or two concepts demonstrated, program loosely follows UML design	Program demonstrates most of the concepts, closely follows UML design	Program demonstrates understanding and application of concepts notably File-IO, Interface Polymorphism via API-based sorting and follows UML design exactly.	
Execution	Program does not run, or crashes on startup	Program runs but crashes on unexpected input	Program runs and does not crash, outputs are a close match to handout example	Program runs, does not crash, outputs are an exact match to handout example.	
Unit Tests	Unit tests omitted from student submission (or modified to pass tests that would normally fail)	Provided unit test has many failed tests	Provided unit test has few failed tests	Provided unit test has no failed tests.	
				Total:	

Max Points: 27

Appendix – Additional Notes

Program Comments Note:

• At the top of each source code file include the following comment header, adding the needed information:	`/* File Name:`
	` * Course Name:`
	` * Lab Section:`
	` * Student Name:`
	` * Date:`
	` * Author: Stanley Pieda (2015) Personal Communication`
	` * Updated By: (Student's name)`
	` */`

• Classes and class members (class level fields, constructors, methods) should have a brief description as a comment immediately above in the code listing. **Javadoc Example, /** */**	`/**`
	` * Provides interaction with the user in the`
	` * form of a console menu system. Options`
	` * include Practice test, Print Report,`
	` * Exit Program and more.`
	` */`
	`public void runMenu(){`

Javadoc comments are required for Assignment 4

Appendix: Sample run of program for new features

Note:

- user inputs were made black font, bolded, and highlighted in yellow
- System.err.println() outputs were made black font, underlined, and bolded

```
1 to add a task
2 to toggle a task's is completed
3 to remove a task
4 to view all tasks
5 to save tasks to file
6 to load tasks from file
7 to sort tasks by title
8 to sort tasks by priority
9 to sort tasks by is-complete
10 to exit program
5
No tasks to save
1 to add a task
2 to toggle a task's is completed
3 to remove a task
4 to view all tasks
5 to save tasks to file
6 to load tasks from file
7 to sort tasks by title
8 to sort tasks by priority
9 to sort tasks by is-complete
10 to exit program
6
1 to add a task
2 to toggle a task's is completed
3 to remove a task
4 to view all tasks
5 to save tasks to file
6 to load tasks from file
7 to sort tasks by title
8 to sort tasks by priority
9 to sort tasks by is-complete
10 to exit program
```

File not foundtasks.txt (The system cannot find the file specified)
7
Nothing to sort
1 to add a task
2 to toggle a task's is completed
3 to remove a task
4 to view all tasks
5 to save tasks to file
6 to load tasks from file
7 to sort tasks by title
8 to sort tasks by priority
9 to sort tasks by is-complete
10 to exit program
8
Nothing to sort
1 to add a task
2 to toggle a task's is completed
3 to remove a task
4 to view all tasks
5 to save tasks to file
6 to load tasks from file
7 to sort tasks by title
8 to sort tasks by priority
9 to sort tasks by is-complete
10 to exit program
9
Nothing to sort
1 to add a task
2 to toggle a task's is completed
3 to remove a task
4 to view all tasks
5 to save tasks to file
6 to load tasks from file
7 to sort tasks by title
8 to sort tasks by priority
9 to sort tasks by is-complete
10 to exit program
1
Please enter task title
(Title cannot be empty, max 25 characters)
**I	like	tabs**
Please enter task priority (high, medium, low)
low
1 to add a task
2 to toggle a task's is completed
3 to remove a task
4 to view all tasks
5 to save tasks to file
6 to load tasks from file
7 to sort tasks by title
8 to sort tasks by priority
9 to sort tasks by is-complete
10 to exit program
There was a problem adding a task:
Title cannot contain tab characters
Please try again.
1 to add a task
2 to toggle a task's is completed
3 to remove a task
4 to view all tasks
5 to save tasks to file
6 to load tasks from file
7 to sort tasks by title

```
8 to sort tasks by priority
9 to sort tasks by is-complete
10 to exit program
1
Please enter task title
(Title cannot be empty, max 25 characters)
aaa
Please enter task priority (high, medium, low)
low
1 to add a task
2 to toggle a task's is completed
3 to remove a task
4 to view all tasks
5 to save tasks to file
6 to load tasks from file
7 to sort tasks by title
8 to sort tasks by priority
9 to sort tasks by is-complete
10 to exit program
1
Please enter task title
(Title cannot be empty, max 25 characters)
bbb
Please enter task priority (high, medium, low)
medium
1 to add a task
2 to toggle a task's is completed
3 to remove a task
4 to view all tasks
5 to save tasks to file
6 to load tasks from file
7 to sort tasks by title
8 to sort tasks by priority
9 to sort tasks by is-complete
10 to exit program
1
Please enter task title
(Title cannot be empty, max 25 characters)
ccc
Please enter task priority (high, medium, low)
high
1 to add a task
2 to toggle a task's is completed
3 to remove a task
4 to view all tasks
5 to save tasks to file
6 to load tasks from file
7 to sort tasks by title
8 to sort tasks by priority
9 to sort tasks by is-complete
10 to exit program
4
index: 0 aaa (low) is not complete
index: 1 bbb (medium) is not complete
index: 2 ccc (high) is not complete
1 to add a task
2 to toggle a task's is completed
3 to remove a task
4 to view all tasks
5 to save tasks to file
6 to load tasks from file
7 to sort tasks by title
8 to sort tasks by priority
```

```
9 to sort tasks by is-complete
10 to exit program
2
Please enter index number of task to toggle complete / incomplete
0
1 to add a task
2 to toggle a task's is completed
3 to remove a task
4 to view all tasks
5 to save tasks to file
6 to load tasks from file
7 to sort tasks by title
8 to sort tasks by priority
9 to sort tasks by is-complete
10 to exit program
4
index: 0 aaa (low) is complete
index: 1 bbb (medium) is not complete
index: 2 ccc (high) is not complete
1 to add a task
2 to toggle a task's is completed
3 to remove a task
4 to view all tasks
5 to save tasks to file
6 to load tasks from file
7 to sort tasks by title
8 to sort tasks by priority
9 to sort tasks by is-complete
10 to exit program
9
Sorting by is-complete completed
1 to add a task
2 to toggle a task's is completed
3 to remove a task
4 to view all tasks
5 to save tasks to file
6 to load tasks from file
7 to sort tasks by title
8 to sort tasks by priority
9 to sort tasks by is-complete
10 to exit program
4
index: 0 bbb (medium) is not complete
index: 1 ccc (high) is not complete
index: 2 aaa (low) is complete
1 to add a task
2 to toggle a task's is completed
3 to remove a task
4 to view all tasks
5 to save tasks to file
6 to load tasks from file
7 to sort tasks by title
8 to sort tasks by priority
9 to sort tasks by is-complete
10 to exit program
8
Sorting by priority completed
1 to add a task
2 to toggle a task's is completed
3 to remove a task
4 to view all tasks
5 to save tasks to file
6 to load tasks from file
```

```
7 to sort tasks by title
8 to sort tasks by priority
9 to sort tasks by is-complete
10 to exit program
4
index: 0 ccc (high) is not complete
index: 1 bbb (medium) is not complete
index: 2 aaa (low) is complete
1 to add a task
2 to toggle a task's is completed
3 to remove a task
4 to view all tasks
5 to save tasks to file
6 to load tasks from file
7 to sort tasks by title
8 to sort tasks by priority
9 to sort tasks by is-complete
10 to exit program
7
Sorting by title completed
1 to add a task
2 to toggle a task's is completed
3 to remove a task
4 to view all tasks
5 to save tasks to file
6 to load tasks from file
7 to sort tasks by title
8 to sort tasks by priority
9 to sort tasks by is-complete
10 to exit program
4
index: 0 aaa (low) is complete
index: 1 bbb (medium) is not complete
index: 2 ccc (high) is not complete
1 to add a task
2 to toggle a task's is completed
3 to remove a task
4 to view all tasks
5 to save tasks to file
6 to load tasks from file
7 to sort tasks by title
8 to sort tasks by priority
9 to sort tasks by is-complete
10 to exit program
5
Saved 3 records to file tasks.txt
1 to add a task
2 to toggle a task's is completed
3 to remove a task
4 to view all tasks
5 to save tasks to file
6 to load tasks from file
7 to sort tasks by title
8 to sort tasks by priority
9 to sort tasks by is-complete
10 to exit program
3
Please enter index number of task to remove
0
1 to add a task
2 to toggle a task's is completed
3 to remove a task
4 to view all tasks
```

```
5 to save tasks to file
6 to load tasks from file
7 to sort tasks by title
8 to sort tasks by priority
9 to sort tasks by is-complete
10 to exit program
3
Please enter index number of task to remove
0
1 to add a task
2 to toggle a task's is completed
3 to remove a task
4 to view all tasks
5 to save tasks to file
6 to load tasks from file
7 to sort tasks by title
8 to sort tasks by priority
9 to sort tasks by is-complete
10 to exit program
3
Please enter index number of task to remove
0
1 to add a task
2 to toggle a task's is completed
3 to remove a task
4 to view all tasks
5 to save tasks to file
6 to load tasks from file
7 to sort tasks by title
8 to sort tasks by priority
9 to sort tasks by is-complete
10 to exit program
4
Nothing to show, no tasks
1 to add a task
2 to toggle a task's is completed
3 to remove a task
4 to view all tasks
5 to save tasks to file
6 to load tasks from file
7 to sort tasks by title
8 to sort tasks by priority
9 to sort tasks by is-complete
10 to exit program
6
Loaded 3 records from file tasks.txt
1 to add a task
2 to toggle a task's is completed
3 to remove a task
4 to view all tasks
5 to save tasks to file
6 to load tasks from file
7 to sort tasks by title
8 to sort tasks by priority
9 to sort tasks by is-complete
10 to exit program
4
index: 0 aaa (low) is complete
index: 1 bbb (medium) is not complete
index: 2 ccc (high) is not complete
1 to add a task
2 to toggle a task's is completed
3 to remove a task
```

```
4 to view all tasks
5 to save tasks to file
6 to load tasks from file
7 to sort tasks by title
8 to sort tasks by priority
9 to sort tasks by is-complete
10 to exit program
10
program will exit
```