Neil Elkadi | 01/26/2025 | Software Development II

Code

App.java

```
/*
 * This source file was generated by the Gradle 'init' task
 */
package org.example;

import java.util.Scanner;

public class App {
  private static TodoList todoList;
  public static Scanner scanner;

  public static void main(String[] args) {
    todoList = new TodoList();
    scanner = new Scanner(System.in);

    menu();
  }

  public static void menu() {
    printMenu();

    char choice;

    try {
      choice = scanner.nextLine().toLowerCase().charAt(0);
    } catch (Exception e) {
      System.out.println("Bad input!\n");
      menu();
      return;
    }

    switch (choice) {
      case 'a':
        todoList.addTodo(new TodoItem());
        System.out.println();
        break;
      case 'd':
        todoList.deleteTodo();
        System.out.println();
        break;
      case 'p':
        System.out.println(todoList.toString());
        break;
      case 'q':
```

```
          System.out.println("Goodbye!");
          return;
        default:
          System.out.println("Invalid!\n");
          break;
      }

      menu();
    }

    public static void printMenu() {
      System.out.println("[A]dd to do item to list");
      System.out.println("[D]elete item from todo list");
      System.out.println("[P]rint to do items");
      System.out.println("[Q]uit");
      System.out.print("Preform an action: ");
    }
  }
```

TodoItem.java

```
package org.example;

import java.util.Date;

import org.example.enums.Priority;

public class TodoItem {
  String todo;
  long dueDate;
  Priority priority;

  public TodoItem(String todo) {
    setTodo(todo);
    setDueDate(new Date().getTime());
    setPriority(Priority.LOW);
  }

  public TodoItem(String todo, long date, Priority priority) {
    setTodo(todo);
    setDueDate(date);
    setPriority(priority);
  }

  public TodoItem() {
    setTodo();
    setDueDate();
    setPriority();
  }

  private void setPriority() {
```

```
    boolean invalid = true;

    do {
      try {
        invalid = false;
        System.out.print("Please set the priority ([L]ow, [M]edium,
[H]igh): ");
        char priority = App.scanner.nextLine().toLowerCase().charAt(0);

        switch (priority) {
          case 'l':
            this.priority = Priority.LOW;
            break;
          case 'm':
            this.priority = Priority.MEDIUM;
            break;
          case 'h':
            this.priority = Priority.HIGH;
            break;
          default:
            invalid = true;
            break;
        }
      } catch (Exception e) {
        System.out.println("Invalid. Try again.");
        invalid = true;
      }
    } while (invalid);
  }

  private void setDueDate() {
    boolean invalid = true;

    do {
      try {
        invalid = false;

        System.out.print("How many days from now will this be due?: ");
        int days = App.scanner.nextInt();
        App.scanner.nextLine();

        long date = new Date().getTime() + (days * 24 * 60 * 60 * 1000);
        setDueDate(date);
      } catch (Exception e) {
        System.out.println("Invalid. Try again.");
        invalid = true;
      }
    } while (invalid);
  }

  private void setTodo() {
    System.out.println("What is your todo item?");
    String todo = App.scanner.nextLine();
```

```
    setTodo(todo);
  }

  public String getTodo() {
    return todo;
  }

  public void setTodo(String todo) {
    this.todo = todo;
  }

  public long getDueDate() {
    return dueDate;
  }

  public void setDueDate(long dueDate) {
    this.dueDate = dueDate;
  }

  public Priority getPriority() {
    return priority;
  }

  public void setPriority(Priority priority) {
    this.priority = priority;
  }
}
```

TodoList.java

```
package org.example;

import java.util.ArrayList;
import java.util.Date;

/**
 * TodoList
 */
public class TodoList {
  private ArrayList<TodoItem> todoItems;

  public TodoList() {
    this.todoItems = new ArrayList<TodoItem>();
  }

  public void addTodo(TodoItem item) {
    this.todoItems.add(item);
  }

  public boolean deleteTodo(int id) {
    try {
```

```
      this.todoItems.remove(id);
      return true;
    } catch (Exception e) {
      return false;
    }
  }

  @Override
  public String toString() {
    String result = "";
    int index = 0;

    long now = new Date().getTime();

    for (TodoItem item : todoItems) {
      result += item.todo + "\n";
      result += "ID: " + index++ + "\n";
      result += "Due in days from now: " + (int)
Math.ceil((item.getDueDate() - now) / (double) 86400000) + "\n";
      result += "Priority: " + item.getPriority() + "\n\n";
    }

    return result;
  }

  public void deleteTodo() {

    try {
      System.out.print("Enter the id of the item you would like to delete:
");
      int id = App.scanner.nextInt();
      App.scanner.nextLine();

      if (!deleteTodo(id)) {
        System.out.println("Invalid!");
        return;
      }

      System.out.println("Item deleted.");
    } catch (Exception e) {
      System.out.println("Invalid!");
    }

  }
}
```

Priority.java

```
package org.example.enums;

public enum Priority {
```

```
    LOW, MEDIUM, HIGH
}
```

Short paragraph

There wasn't a lot I can say about this project. I must have done this exact sort of work at least 30 times over during my AS and this was no different. Just tried to work through it as quickly as possible using what I've done before. A couple of slightly odd things were: one, I choose to use one scanner object and reference it globally due to copy-paste issues otherwise. Two, I used a "days from now" date rather than a month day year date for convenience. And three, I somewhat wanted to separate all user input collection into its own class, but by the time the idea had come I'd already had sunk costs; may attempt in a future assignment if the opportunity is presented.