EMPLOYEE MANAGER CLI APPLICATION

package com.INDHIRAcode;

import com.INDHIRAtask.Task;

import java.util.ArrayList;

import java.util.Scanner;

public class Main {

static Scanner scan = new Scanner(System.in);

static ArrayList<Task> tasks = new ArrayList<>();

static int tasksCount = tasks.size();

static void displayTasks(ArrayList<Task> tasks) {

System.out.println("---");

for (int i = 0; i < tasks.size(); i++) {

System.out.println(tasks.get(i));

}

}

static void continueOrExit(String option) {

switch (option) {

case "0":

exit();

break;

default:

showMainMenu();

break;

}

}

static void noTasks() {

System.out.println("You have no tasks yet! " +

"Continue [any key] - Exit [0]");

String option = scan.nextLine();

continueOrExit(option);

}

static void showTasks() {

if(tasks.isEmpty()) {

noTasks();

} else {

System.out.println("Here are your tasks " +

"Continue [any key] - Exit[0]:\n");

displayTasks(tasks);

String option = scan.nextLine();

continueOrExit(option);

}

}

static void addTask() {

displayTasks(tasks);

System.out.println("What task do you want to add? Give a description " +

"(Cancel [0])");

String option = scan.nextLine();

if(option.equals("0")) {

showMainMenu();

} else {

Task task = new Task(++tasksCount, option);

tasks.add(task);

addTask();

}

}

static void deleteTask() {

if(tasks.isEmpty()) {

noTasks();

} else {

displayTasks(tasks);

System.out.println("Which task do you want to delete? " +

"(give the id - Cancel [0])");

int option = scan.nextInt();

if(option == 0) {

showMainMenu();

} else if(Task.isOutOfBounds(tasks, option)) {

deleteTask();

} else {

tasks.removeIf(task -> (task.getId() == option));

deleteTask();

}

}

}

static void markTaskDone() {

if(tasks.isEmpty()) {

noTasks();

} else {

displayTasks(tasks);

System.out.println("Which task do you want to mark as done? " +

"(give the id - Cancel [0])");

int option = scan.nextInt();

if(option == 0) {

showMainMenu();

} else if(Task.isOutOfBounds(tasks, option) || !Task.exists(tasks, option)) {

markTaskDone();

} else if(Task.exists(tasks, option) && Task.getById(tasks, option).isDone()) {

markTaskDone();

} else {

Task.getById(tasks, option).setDone();

markTaskDone();

}

}

}

static void exit() {

System.out.println("Bye Bye ;-)");

System.exit(0);

}

static void updateTask() {

if(tasks.isEmpty()) {

noTasks();

} else {

displayTasks(tasks);

System.out.println("Which task do you want to update? " +

"(give the id - Cancel [0])");

int option = scan.nextInt();

if(option == 0) {

showMainMenu();

} else if(Task.isOutOfBounds(tasks, option) || !Task.exists(tasks, option)) {

updateTask();

} else {

System.out.println("Give a new description for your task " + "(Cancel [0])");

scan.nextLine();

String description = scan.nextLine();

if(description.equals("0")) {

updateTask();

} else {

Task.getById(tasks, option).setDescription(description);

updateTask();

}

}

}

}

static void showMainMenu() {

System.out.println("Welcome to your task manager!\n" +

"1. See all your tasks\n" +

"2. Add a task\n" +

"3. Delete a task\n" +

"4. Mark a task as done\n" +

"5. Update Task\n" +

"6. Exit\n");

String option = scan.nextLine();

switch (option) {

case "1":

showTasks();

break;

case "2":

addTask();

break;

case "3":

deleteTask();

break;

case "4":

markTaskDone();

break;

case "5":

updateTask();

break;

case "6":

exit();

break;

default:

showMainMenu();

}

}

public static void main(String[] args) {

tasks.add(new Task(1, "Go to the gym"));

tasks.add(new Task(2, "Go to the cinema"));

tasks.add(new Task(3, "Get a new monitor"));

tasksCount = 3;

showMainMenu();

}

}