```
java
import java.io.*;
import java.util.*;
class TodoItem implements Serializable {
private static final long serialVersionUID = 1L;
int id;
String title;
boolean done;
public TodoItem(int id, String title) {
this.id = id;
this.title = title;
this.done = false;
}
@Override
public String toString() {
return String.format("[%d] %s %s", id, done ? "(done)" : "(todo)", title);
}
}
public class TodoApp {
private static final String DATA_FILE = "todos.ser";
private List<TodoItem> items = new ArrayList<>();
private int nextId = 1;
private Scanner scanner = new Scanner(System.in);
public static void main(String[] args) {
TodoApp app = new TodoApp();
app.load();
app.run();
}
private void run() {
while (true) {
printMenu();
String choice = scanner.nextLine().trim();
switch (choice) {
case "1": list(); break;
case "2": add(); break;
case "3": toggle(); break;
case "4": remove(); break;
case "0": save(); System.out.println("Bye."); return;
default: System.out.println("Unknown menu.");
}
}
}
private void printMenu() {
System.out.println("\n==== TODO List ====");
System.out.println("1. List items");
System.out.println("2. Add item");
System.out.println("3. Toggle done");
System.out.println("4. Remove item");
System.out.println("0. Exit");
System.out.print("Select: ");
}
private void list() {
if (items.isEmpty()) {
System.out.println("No items.");
return;
}
System.out.println("\n[TODO Items]");
for (TodoItem item : items) {
System.out.println(item);
}
}
private void add() {
System.out.print("Title: ");
String title = scanner.nextLine();
if (title.isBlank()) {
System.out.println("Title cannot be empty.");
return;
}
TodoItem item = new TodoItem(nextId++, title);
items.add(item);
System.out.println("Added: " + item);
}
private void toggle() {
System.out.print("Enter ID to toggle: ");
String s = scanner.nextLine();
try {
int id = Integer.parseInt(s);
for (TodoItem item : items) {
if (item.id == id) {
item.done = !item.done;
System.out.println("Updated: " + item);
return;
}
}
System.out.println("Item not found.");
} catch (NumberFormatException e) {
System.out.println("Invalid ID.");
}
}
private void remove() {
System.out.print("Enter ID to remove: ");
String s = scanner.nextLine();
try {
int id = Integer.parseInt(s);
Iterator<TodoItem> it = items.iterator();
while (it.hasNext()) {
TodoItem item = it.next();
if (item.id == id) {
it.remove();
System.out.println("Removed: " + item);
return;
}
}
System.out.println("Item not found.");
} catch (NumberFormatException e) {
System.out.println("Invalid ID.");
}
}
@SuppressWarnings("unchecked")
private void load() {
File f = new File(DATA_FILE);
if (!f.exists()) return;
try (ObjectInputStream ois = new ObjectInputStream(new FileInputStream(f))) {
items = (List<TodoItem>) ois.readObject();
nextId = 1;
for (TodoItem item : items) {
if (item.id >= nextId) nextId = item.id + 1;
}
System.out.println("Loaded " + items.size() + " items.");
} catch (Exception e) {
System.out.println("Failed to load data: " + e.getMessage());
}
}
private void save() {
try (ObjectOutputStream oos = new ObjectOutputStream(new FileOutputStream(DATA_FILE))) {
oos.writeObject(items);
System.out.println("Saved " + items.size() + " items.");
} catch (IOException e) {
System.out.println("Failed to save data: " + e.getMessage());
}
}
}
```