Notes & Password Manager

Java Implementation Overview

July 05, 2025

Overview

- Two modules: Notes & Password storage
- Simple command-line interface (Scanner)
- Data persisted in local text files
- Reverse-string encryption for passwords
- Object-oriented, easily extensible

Class: Note

- Fields: title, content
- Methods:
  - toString()
  - toFileFormat()
  - fromFileFormat()

Class: PasswordEntry

- Fields: site, username, encryptedPassword
- Constructor encrypts password
- Methods:
  - encrypt() & decrypt()
  - toString()
  - toFileFormat() & fromFileFormat()

Encryption Logic

- Very simple: reverse the password string
- Example: "secret" ⇒ "terces"
- ✔ Easy to implement
- ✘ NOT secure for production use
- Replace with AES or bcrypt in real apps

File Persistence

- notes.txt & passwords.txt created in working dir
- BufferedWriter/Reader used for I/O
- Append mode preserves existing records
- Format: field1::field2(::field3)
- try-with-resources recommended for safety

CLI Main Loop

- Infinite while-loop displays menu
- switch-case handles chosen action
- Scanner for user input
- IOException caught for file errors
- Option 5 cleanly exits program

Sample Run

===== Notes and Password Manager =====

1. Add Note

2. View Notes

3. Add Password

4. View Passwords

5. Exit

Choose: 1

Enter note title: Grocery

Enter content: Buy milk and bread

Note saved.

Future Improvements

- Strong encryption (AES, PBKDF2)
- Update / delete entries
- Mask password input (Console)
- GUI (Swing / JavaFX)
- Use JSON or database for storage

Source Code (snippet)

```
import java.io.*;

import java.util.*;


public class NotesPasswordManager {

   static class Note {

      String title;

      String content;


      Note(String title, String content) {

         this.title = title;

         this.content = content;

      }


      public String toString() {

         return "Title: " + title + "\nContent: " + content;
```