Association Management Software

A project by **Gianlorenzo Lucioni**

Introduction

My name is Gianlorenzo Lucioni, and I am an undergraduate student in Computer Science and Automation Engineering. I developed this software out of a personal need: to provide the small association I am a part of with a simple, zero-cost tool to manage the organizational activities of a patron saint festival. This project has become a valuable learning experience, blending academic theory with practical problem-solving.

Project Abstract

"Association Management Software" is a cross-platform desktop application developed in pure Java. It operates via a command-line interface (CLI) to ensure maximum lightness and compatibility. The software addresses the issue of disorganization in the management of small associations by offering a centralized solution to track member records and, most importantly, accounting. It automates the recording of income and expenses and generates, on demand, a detailed financial report in CSV format, ready for analysis and for delivery to an accountant, thereby simplifying management and reducing costs.

Key Features

- **Member Roster Management:** Allows for adding and viewing association members, with permanent data saving.
- **Simple and Persistent Accounting:** Records every financial transaction. Data is saved locally, ensuring the complete history is always available at every startup.
- **Analysis by Category:** Each transaction can be categorized (e.g., "Beverage Sales," "Material Costs"), enabling a detailed analysis of cash flows and profits for each individual activity.
- **Professional Report Generation:** Exports a complete and formatted balance sheet in CSV format, which can be opened with any spreadsheet software (Excel, Google Sheets). The report includes a category summary and an automatic calculation of the operating surplus.
- **Cross-Platform:** Being based on Java, the executable .jar file runs on Windows, macOS, and Linux without modification.

Technology Stack

- **Language:** Java (JDK 11 or higher)
- **Interface:** Command-Line Interface (CLI)
- **Data Format:** Text (.txt for members) and CSV (.csv for accounting)
- **IDE:** Developed with Visual Studio Code

Getting Started

Prerequisites

Ensure you have **Java** installed on your computer (version 11 or later). You can check this by opening a terminal and typing:

```
java -version
```

Running the Pre-compiled Version (.jar)

This is the recommended method for end-users.

1. Download the latest Gestionale.jar file from the **"Releases"** section of this repository.
2. Create a folder for the program (e.g., FestivalManager) and move the Gestionale.jar file into it.
3. Open a terminal in that same folder.
4. Run the program with the following command:

   ```
   java -jar Gestionale.jar
   ```
5. On the first save, the program will automatically create a dati/ subfolder to store the data.

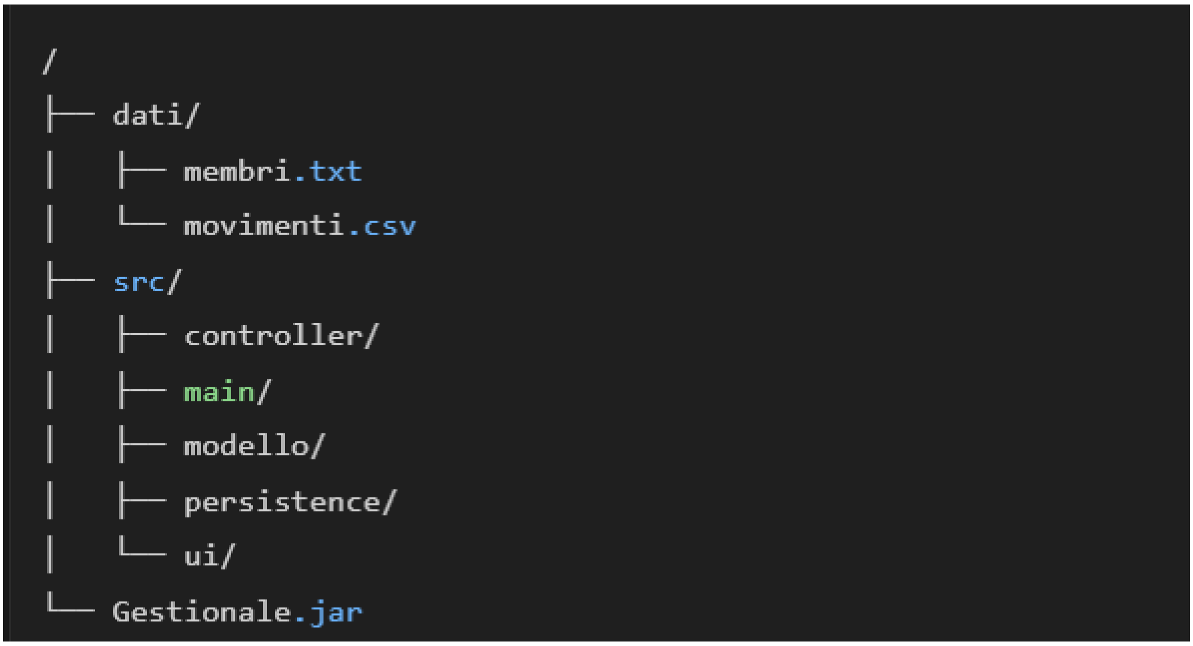
User Guide

Once started, the program will display an interactive menu:

- **Option 1 & 2:** Manage the association's member roster.
- **Option 3 & 4:** Register new income or expenses. It is crucial to use consistent categories for effective analysis.
- **Option 5:** Generate the Rendiconto_Festa.csv report. This file will be created in the same folder as the .jar file.
- **Option 0: (IMPORTANT)** Always use this option to exit. It allows the software to correctly save all data entered before closing.

Project Structure (For Developers)

The software architecture follows the **Separation of Concerns** principle, organizing the code into specific packages for maintainability and clarity.



```
/
├── dati/
│   ├── membri.txt
│   └── movimenti.csv
├── src/
│   ├── controller/
│   ├── main/
│   ├── modello/
│   ├── persistence/
│   └── ui/
└── Gestionale.jar
```

- modello: Contains the POJO classes that represent the data (Membro, MovimentoContabile).
- persistence: Manages reading and writing data to files.
- controller: Contains the business logic that orchestrates the application.
- ui: Manages the command-line interface and user input.
- main: Contains the application's entry point.

Contributing

Contributions are welcome! If you wish to improve the project, you are invited to:

1. Fork the repository.
2. Create a new branch for your feature (git checkout -b feature/AmazingFeature).
3. Commit your changes (git commit -m 'Add some AmazingFeature').
4. Push to the branch (git push origin feature/AmazingFeature).
5. Open a Pull Request.

License and Intellectual Property

This means you are free to use, copy, modify, merge, publish, distribute, and/or sell copies of the software, provided that you include the original license and this copyright notice in all copies or substantial portions of the software.

It is a very permissive license that encourages collaboration and knowledge sharing while protecting the recognition of the original author.

Contacts

Gianlorenzo Lucioni [email: lucioni2002@gmail.com]

LinkedIn

https://www.linkedin.com/in/gianlorenzo-lucioni-714a42359?utm_source=share&utm_campaign=share_via&utm_content=profile&utm_medium=android_app