Report: Analysis of task 4

Introduction

This report provides an analysis of three Java classes: `PaperFactory`, `Person`, and `FactoryTester`. These classes are designed to simulate a factory management system and provide testing capabilities for its functionality.

1. PaperFactory Class

Functionality:

The `PaperFactory` class manages a list of `Person` objects representing the staff. It provides methods to add (`addPerson`), delete (`deletePerson`), list all staff, save the staff list to a file, and load the staff list from a file. The class emphasizes data management and persistence through file I/O operations.

Goal:

The goal of the `PaperFactory` class is to simulate a factory's human resources management system, enabling CRUD (Create, Read, Update, Delete) operations on staff records and ensuring data persistence across sessions.

2. Person Class

Functionality:

The `Person` class represents an individual with attributes such as name, surname, and personal number. It provides getters and setters for these attributes and overrides the `toString` method to provide a string representation of a `Person` object.

Goal:

The goal of the `Person` class is to encapsulate the attributes and behaviors of an individual staff member within the `PaperFactory`. It serves as a basic data structure to store and manipulate staff information.

3. FactoryTester Class

Functionality:

The `FactoryTester` class provides a testing environment for the `PaperFactory` class. It demonstrates the functionality of adding persons to the factory, saving and loading staff lists from a file, and ensuring data integrity through CRUD operations.

Goal:

The goal of the `FactoryTester` class is to validate the functionality and correctness of the `PaperFactory` class. It serves as a tool to verify that the implemented methods within `PaperFactory` work as expected, including file handling operations.

Conclusion

In conclusion, the `PaperFactory`, `Person`, and `FactoryTester` classes collectively simulate a factory management system in Java. They demonstrate principles of object-oriented programming, encapsulation, and file I/O operations for data persistence. These classes provide a foundational structure for managing staff records within a simulated factory environment.