Elene Skhulukhia 01.07.2024
Eleneskhulukhia.1@iliauni.edu.ge

Comprehensive Report for Task 4

This report provides an overview of the classes implemented for Task 4, focusing on managing an `AccountingFirm` and handling invoices. The report discusses the `AccountingFirm` class, its integration with the `LegalEntity` interface, and usage demonstrated in the `FirmTester` class.

Classes

`AccountingFirm` Class

- **Purpose**:

  Manages information related to an accounting firm, including its address, VAT number, and a list of invoice numbers.

- **Functionality**:

  Implements the `LegalEntity` interface, providing methods to retrieve the firm's address and VAT number.

  Allows adding (`addInvoice`) and deleting (`deleteInvoice`) invoice numbers from the list.

  Supports serialization (`saveInvoicesToFile`, `loadInvoicesFromFile`) to save and load the list of invoices from a file.

- **Key Methods**:

  `addInvoice(String invoiceNumber)`: Adds an invoice number to the list.

  `deleteInvoice(String invoiceNumber)`: Deletes an invoice number from the list.

  `saveInvoicesToFile(String filename)`: Saves the list of invoices to a specified file using serialization.

  `loadInvoicesFromFile(String filename)`: Loads the list of invoices from a specified file using deserialization.

Java implementation:

java Copy code

```java
// AccountingFirm.java
package finalexam.task4;

import java.io.*;
import java.util.ArrayList;
import java.util.List;

public class AccountingFirm implements LegalEntity {
    private String address;
    private String vatNumber;
    private List<String> invoices = new ArrayList<>();

    public AccountingFirm(String address, String vatNumber) {
        this.address = address;
        this.vatNumber = vatNumber;
    }

    public String getAddress() {
        return address;
    }

    public void setAddress(String address) {
        this.address = address;
    }

    public String getVatNumber() {
        return vatNumber;
    }

    public void setVatNumber(String vatNumber) {
        this.vatNumber = vatNumber;
    }

    public void addInvoice(String invoiceNumber) {
        invoices.add(invoiceNumber);
    }

    public boolean deleteInvoice(String invoiceNumber) {
        return invoices.remove(invoiceNumber);
    }
```

```
    public List<String> getInvoices() {
        return new ArrayList<>(invoices);
    }

    public void saveInvoicesToFile(String filename) throws IOException {
        try (ObjectOutputStream oos = new ObjectOutputStream(new FileOutputStream(
            oos.writeObject(invoices);
        }
    }

    public void loadInvoicesFromFile(String filename) throws IOException, ClassNotF
        try (ObjectInputStream ois = new ObjectInputStream(new FileInputStream(file
            invoices = (List<String>) ois.readObject();
        }
    }

    @Override
    public String toString() {
        return "AccountingFirm{" +
                "address='" + address + '\'' +
                ", vatNumber='" + vatNumber + '\'' +
                ", invoices=" + invoices +
                '}';
    }
}
```

`LegalEntity` Interface

- **Purpose**:

  Defines methods to retrieve the address and VAT number of a legal entity.

- **Functionality**:

  Contains two methods: `getAddress()` and `getVatNumber()`.

  Implemented by classes like `AccountingFirm` to provide specific details for a legal entity.

  Java implementation:

```
// LegalEntity.java
package finalexam.task4;

public interface LegalEntity {
    String getAddress();
    String getVatNumber();
}
```

`FirmTester` Class

- **Purpose**:

  Provides a demonstration of how to use the `AccountingFirm` class functionalities through basic operations such as adding, deleting, saving, and loading invoices.

- **Usage**:

  Instantiates an `AccountingFirm` object with sample address and VAT number.

  Adds invoices (`INV001`, `INV002`), demonstrates saving and loading invoices to/from file (`invoices.dat`), and verifies operations through console output.

3. Implementation Details

- **Serialization**:

  Uses `ObjectOutputStream` and `ObjectInputStream` for serialization and deserialization of the list of invoices.

  Proper exception handling (`IOException`, `ClassNotFoundException`) ensures robust file operations.

4. Relationships and Dependencies

- **Dependencies**:

  `AccountingFirm` class depends on the `LegalEntity` interface for implementing methods related to legal entity information (`getAddress`, `getVatNumber`).

  Utilizes Java standard library classes for file handling (`FileInputStream`, `FileOutputStream`) and serialization (`ObjectOutputStream`, `ObjectInputStream`).

Java implementation:

```java
// FirmTester.java
package finalexam.task4;

import java.io.IOException;

public class FirmTester {

    public static void main(String[] args) {
        AccountingFirm firm = new AccountingFirm("123 Main St", "123456789");

        firm.addInvoice("INV001");
        firm.addInvoice("INV002");

        System.out.println("Firm invoices after adding: " + firm.getInvoices());

        try {
            firm.saveInvoicesToFile("invoices.dat");
            firm.deleteInvoice("INV001");
            System.out.println("Firm invoices after deleting: " + firm.getInvoices(

            firm.loadInvoicesFromFile("invoices.dat");
            System.out.println("Firm invoices after loading from file: " + firm.get
        } catch (IOException | ClassNotFoundException e) {
            e.printStackTrace();
        }
    }
}
```

Conclusion

This report outlines the design and functionality of the `AccountingFirm` class, demonstrating its integration with the `LegalEntity` interface and practical usage in the `FirmTester` class. The implementation showcases basic operations for managing invoices and maintaining legal entity information, adhering to standard Java practices for file handling and serialization.