

Task 4 Comprehensive Report

Introduction

In Task 4, we were required to create an organization called "Shoe Store" that implements the provided `LegalEntity` interface in the package `finalexam.task4`. Additionally, we needed to add methods similar to those in Tasks 2 and 3. This task involves creating classes to manage the details of a shoe store, including its address, VAT number, and the list of shoe styles it offers.

Class Descriptions

`LegalEntity` Interface

The `LegalEntity` interface defines the structure that any legal entity must follow. It includes two methods: `getAddress()` and `getVatNumber()`. These methods are expected to return the address and VAT number of the entity, respectively.

```
package finalexam.task4;

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

`ShoeStore` Class

The `ShoeStore` class implements the `LegalEntity` interface. This class represents a shoe store and contains the following functionalities:

- Attributes:**
 - `address`: The address of the shoe store.
 - `vatNumber`: The VAT number of the shoe store.
 - `styles`: A list of shoe styles available in the store.
- Methods:**
 - `getAddress()`: Returns the address of the shoe store.
 - `getVatNumber()`: Returns the VAT number of the shoe store.

- `addStyle(Style style)`: Adds a new shoe style to the store's list.
- `deleteStyle(Style style)`: Removes a shoe style from the store's list.
- `getStyles()`: Returns the list of all shoe styles in the store.

```
package finalexam.task4;

import java.util.ArrayList;
import java.util.List;

public class ShoeStore implements LegalEntity {
    private String address;
    private String vatNumber;
    private List<Style> styles;

    public ShoeStore(String address, String vatNumber) {
        this.address = address;
        this.vatNumber = vatNumber;
        this.styles = new ArrayList<>();
    }
}
```

```
@Override
public String getAddress() {
    return address;
}

@Override
public String getVatNumber() {
    return vatNumber;
}

public void addStyle(Style style) {
    styles.add(style);
}

public boolean deleteStyle(Style style) {
    return styles.remove(style);
}
```

```
    public List<Style> getStyles() {  
        return styles;  
    }  
}
```

Style Class

The `Style` class represents a style of shoe. It contains the following attributes:

- `name`: The name of the shoe style.
- `description`: A brief description of the shoe style.

```
package finalexam.task4;  
  
public class Style {  
    private String name;  
    private String description;  
  
    public Style(String name, String description) {  
        this.name = name;  
        this.description = description;  
    }  
  
    public String getName() {  
        return name;  
    }  
}
```

```
public void setName(String name) {
    this.name = name;
}

public String getDescription() {
    return description;
}

public void setDescription(String description) {
    this.description = description;
}

@Override
public String toString() {
    return "Style{" +
        "name='" + name + '\'' +
        ", description='" + description + '\'' +
        '}';
}
}
```



ShoeStoreTester Class

The `ShoeStoreTester` class is a test class that demonstrates the functionality of the `ShoeStore` class. It includes a `main` method that creates an instance of `ShoeStore`, adds and deletes styles, and prints the list of styles.

```
package finalexam.task4;

public class ShoeStoreTester {
    public static void main(String[] args) {
        ShoeStore store = new ShoeStore("123 Main St", "VAT123456");

        Style style1 = new Style("Sneakers", "Casual sneakers");
        Style style2 = new Style("Boots", "Leather boots");

        store.addStyle(style1);
        store.addStyle(style2);

        System.out.println("Shoe store address: " + store.getAddress());
        System.out.println("Shoe store VAT number: " + store.getVatNumber());
        System.out.println("Styles available: " + store.getStyles());

        store.deleteStyle(style1);
        System.out.println("Styles available after deletion: " + store.getStyles());
    }
}
```