Task 4 Report

Nikoloz Beridze - nikoloz.beridze.3@iliauni.edu.ge

General Introduction

This project involves the development of a system to manage and track charitable organizations, their legal entities, and beneficiaries. Each class in the system is designed to represent a specific entity with distinct attributes and methods to facilitate the management of data and operations within the system. The following sections provide a detailed description of the functionality and goals of each class.

Class Descriptions

1. CharityOrganization

Purpose: The `CharityOrganization` class represents a charitable organization within the system. It manages the organization's details and operations related to its charitable activities.

Attributes:

- `name`: A string representing the name of the charity organization.
- `registrationNumber`: A string representing the unique registration number of the charity organization.
- `address`: A string representing the address of the charity organization.
- `legalEntities`: A list of `LegalEntity` objects associated with the charity organization.

Methods:

- `CharityOrganization(String name, String registrationNumber, String address)`: Constructor to initialize the charity organization with its name, registration number, and address.
- `addLegalEntity(LegalEntity entity)`: Adds a legal entity to the charity organization.
- `getLegalEntities()`: Returns the list of legal entities associated with the charity organization.
- `toString()`: Returns a string representation of the charity organization.

Goal: The primary goal of the `CharityOrganization` class is to encapsulate all relevant information and operations related to a charitable organization, enabling efficient management and retrieval of data.

```java
package finalexam.task4;

import java.util.ArrayList;
import java.util.List;
public class CharityOrganization implements LegalEntity {
   private String name;
   private String address;
   private String vatNumber;
   private List<Beneficiary> beneficiaries;

   public CharityOrganization(String name, String address, String vatNumber) {
       this.name = name;
       this.address = address;
       this.vatNumber = vatNumber;
       this.beneficiaries = new ArrayList<>();
   }

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

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

   public String getName() {
       return name;
   }

   public void addBeneficiary(Beneficiary beneficiary) {
       beneficiaries.add(beneficiary);
   }

   public void removeBeneficiary(Beneficiary beneficiary) {
       beneficiaries.remove(beneficiary);
   }

   public List<Beneficiary> getBeneficiaries() {
       return new ArrayList<>(beneficiaries);
   }

   @Override
   public String toString() {
       StringBuilder sb = new StringBuilder();
       sb.append("CharityOrganization {\n");
```

```
        sb.append("    Name          : ").append(name).append(",\n");
        sb.append("    Address       : ").append(address).append(",\n");
        sb.append("    VAT Number    : ").append(vatNumber).append(",\n");
        sb.append("    Beneficiaries : \n");

        for (Beneficiary beneficiary : beneficiaries) {
            sb.append("        ").append(beneficiary).append("\n");
        }

        sb.append("}");
        return sb.toString();
    }

}
```

2. LegalEntity (Interface)

Purpose: The `LegalEntity` interface defines the methods that any legal entity associated with a charity organization must implement. This abstraction allows different types of legal entities to be managed in a consistent manner.

Methods:

- `String getEntityName()`: Returns the name of the legal entity.
- `String getEntityType()`: Returns the type of the legal entity.
- `String getRegistrationDate()`: Returns the registration date of the legal entity.
- `List<Beneficiary> getBeneficiaries()`: Returns the list of beneficiaries associated with the legal entity.
- `void addBeneficiary(Beneficiary beneficiary)`: Adds a beneficiary to the legal entity.

Goal: The goal of the `LegalEntity` interface is to provide a common structure for all legal entities, ensuring that they implement the necessary methods to interact with the rest of the system.

3. Beneficiary

Purpose: The `Beneficiary` class represents an individual or group that benefits from the charity's activities. It holds information about the beneficiaries and their relationship with the legal entities.

Attributes:

- `name`: A string representing the name of the beneficiary.
- `beneficiaryType`: A string representing the type of beneficiary (e.g., individual, community).
- `contactInfo`: A string representing the contact information of the beneficiary.
- `supportAmount`: A double representing the amount of support provided to the beneficiary.

Methods:

- `Beneficiary(String name, String beneficiaryType, String contactInfo, double supportAmount)`: Constructor to initialize the beneficiary with their name, type, contact information, and support amount.
- `toString()`: Returns a string representation of the beneficiary.

Goal: The goal of the `Beneficiary` class is to store and manage details about the individuals or groups receiving support from the charity organization, ensuring transparency and accountability in the distribution of aid.

```
package finalexam.task4;

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

class Beneficiary {
   private String name;
   private String description;

   public Beneficiary(String name, String description) {
       this.name = name;
       this.description = description;
   }

   public String getName() {
       return name;
   }

   public String getDescription() {
       return description;
   }

   @Override
   public String toString() {
       return "Beneficiary {\n" +
```

```
                "    Name        : " + name + ",\n" +
                "    Description : " + description + "\n" +
                "}";
    }

}
```

Conclusion

This report provides a detailed overview of the components involved in managing a charity organization system. The `CharityOrganization` class manages the overall organization, the `LegalEntity` interface defines the necessary structure for legal entities associated with the charity, and the `Beneficiary` class represents the recipients of the charity's support. This structure ensures a consistent and efficient way to manage data and operations related to charitable activities.