

Finance Management System

Coding Task 1

Object Oriented Programming

Create the model/entity classes corresponding to the schema within package entity with variables declared private, constructors(default and parametrized) and getters, setters)

This task involves designing and implementing the core model or entity classes for a **Finance Management System**. Each entity class represents a specific component of the system such as users, expenses, categories, suggestions, and administrators. The models are defined within the entity package, and each class contains private fields, a default constructor, a parameterized constructor, getter and setter methods for encapsulation, and a meaningful toString() method for easy data display.

Task 1.1 User Class

Task 1.2 Suggestion Class

Task 1.3 ExpenseCategory Class

Task 1.4 Expense Class

Task 1.5 Admin Class

These entity classes form the foundation for the application's data layer, enabling interaction with databases through well-structured object-oriented representations.

Task 1.1 User Class

The User class represents the primary users of the finance management system. It holds essential information such as userId, username, email, and password. This class provides the foundational structure for user authentication and identification across the platform. With proper encapsulation and methods, it ensures secure and consistent handling of user data.

```
package entity;

public class User {
    private int userId;
    private String username;
    private String email;
    private String password;

    public User() {
    }
}
```

```

    public User(int userId, String username, String email, String password) {
        this.userId = userId;
        this.username = username;
        this.email = email;
        this.password = password;
    }

    public int getUserId() { return userId; }

    public void setUserId(int userId) { this.userId = userId; }

    public String getUsername() { return username; }

    public void setUsername(String username) { this.username = username; }

    public String getEmail() { return email; }

    public void setEmail(String email) { this.email = email; }

    public String getPassword() { return password; }

    public void setPassword(String password) { this.password = password; }

    @Override
    public String toString() {
        return "UserId : " + userId +
            ", Username : '" + username + '\'' +
            ", Email : " + email + '\'';
    }
}

```

Task 1.2 Suggestion Class

The Suggestion class is used to store feedback or improvement ideas submitted by users. It includes the suggestionId, the associated userId, and the suggestionText. This model supports a feedback loop within the application, promoting user engagement and system enhancement by allowing users to share their thoughts or recommendations.

```

package entity;

public class Suggestion {
    private int suggestionId;
    private int userId;
    private String suggestionText;

    public Suggestion() {
    }
}

```

```

public Suggestion(int suggestionId, int userId, String suggestionText) {
    this.suggestionId = suggestionId;
    this.userId = userId;
    this.suggestionText = suggestionText;
}

public int getSuggestionId() { return suggestionId; }
public void setSuggestionId(int suggestionId) { this.suggestionId = suggestionId; }
public int getUserId() { return userId; }
public void setUserId(int userId) { this.userId = userId; }
public String getSuggestionText() { return suggestionText; }
public void setSuggestionText(String suggestionText) { this.suggestionText = suggestionText; }

@Override
public String toString() {
    return "SuggestionId : " + suggestionId +
        ", UserId : " + userId +
        ", Suggestion : '" + suggestionText + '\'';
}
}

```

Task 1.3 ExpenseCategory Class

The ExpenseCategory class defines various categories under which users can classify their expenses, such as food, travel, or utilities. It contains the categoryId and categoryName. This class is crucial for organizing user expenses in a meaningful way and enabling analytical features like category-based expense tracking.

```

package entity;

public class ExpenseCategory {
    private int categoryId;
    private String categoryName;

    public ExpenseCategory() {
    }

    public ExpenseCategory(int categoryId, String categoryName) {
        this.categoryId = categoryId;
        this.categoryName = categoryName;
    }

    public int getCategoryId() { return categoryId; }

    public void setCategoryId(int categoryId) { this.categoryId = categoryId; }

    public String getCategoryName() { return categoryName; }

    public void setCategoryName(String categoryName) { this.categoryName = categoryName; }

    @Override
    public String toString() {
        return "CategoryId : " + categoryId +
            ", CategoryName : '" + categoryName + '\'';
    }
}

```

Task 1.4 Expense Class

The Expense class captures detailed expense records of users, including expenseId, userId, categoryId, amount, date, and a description. It optionally includes the categoryName for display purposes. This class acts as the core of the application, enabling users to log, view, and manage their financial transactions. It supports both detailed and summary reporting features.

```
package entity;

import java.sql.Date;

public class Expense {
    private int expenseId;
    private int userId;
    private int categoryId;
    private double amount;
    private Date date;
    private String description;

    public Expense() {
    }

    public Expense(int userId, int categoryId, double amount, Date date, String note) {
        this.userId = userId;
        this.categoryId = categoryId;
        this.amount = amount;
        this.date = date;
        this.description = note;
    }

    public Expense(int expenseId, int userId, int categoryId, double amount, Date date, String description) {
        this.expenseId = expenseId;
        this.userId = userId;
        this.categoryId = categoryId;
        this.amount = amount;
        this.date = date;
        this.description = description;
    }

    public int getExpenseId() { return expenseId; }

    public void setExpenseId(int expenseId) { this.expenseId = expenseId; }

    public int getUserId() { return userId; }

    public void setUserId(int userId) { this.userId = userId; }

    public int getCategoryId() { return categoryId; }

    public void setCategoryId(int categoryId) { this.categoryId = categoryId; }

    public double getAmount() { return amount; }

    public void setAmount(double amount) { this.amount = amount; }

    public Date getDate() { return date; }

    private String categoryName;

    public String getCategoryName() { return categoryName; }

    public void setCategoryName(String categoryName) { this.categoryName = categoryName; }
```

```

    public void setDate(Date date) { this.date = date; }

    public String getDescription() { return description; }

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

    @Override
    public String toString() {
        return "ExpenseId : " + expenseId +
            ", UserId : " + userId +
            ", CategoryId : " + categoryId +
            ", Amount : " + amount +
            ", Date : " + date +
            ", Description : '" + description + '\'';
    }
}

```

Task 1.5 Admin Class

The Admin class represents system administrators with fields like adminId, adminName, and password. This model is used to authenticate and authorize administrative users who can manage categories, view reports, and maintain system-level settings. It ensures proper segregation of roles between regular users and administrators.

```

package entity;

public class Admin {
    private int adminId;
    private String adminName;
    private String password;

    public Admin() {
    }

    public Admin(int adminId, String adminName, String password) {
        this.adminId = adminId;
        this.adminName = adminName;
        this.password = password;
    }

    public int getAdminId() { return adminId; }

    public void setAdminId(int adminId) { this.adminId = adminId; }

    public String getAdminName() { return adminName; }

    public void setAdminName(String adminName) { this.adminName = adminName; }

    public String getPassword() { return password; }

    public void setPassword(String password) { this.password = password; }
}

```

```
@Override
public String toString() {
    return "AdminId : " + adminId +
        ", AdminName : '" + adminName + '\'' +
        ", Password : '" + password + '\'';
}
```

Conclusion :

The entity classes designed in this task form the backbone of the Finance Management System, providing a clear and organized structure to represent real-world financial components such as users, expenses, categories, suggestions, and administrative roles. By applying object-oriented principles like encapsulation and abstraction, these classes ensure modularity, reusability, and maintainability in the application. They lay the groundwork for further development such as database integration, business logic implementation, and user interface interaction, ultimately enabling a reliable and scalable financial management solution.
