Code for LinkedIn

Write object oriented code to implement the design of the LinkedIn problem.

We'll cover the following

- LinkedIn classes
  - Constants
  - Account
  - Person, admin, and user
  - Recommendation, achievement, and analytics
  - Profile, experience, education, and skill
  - Company, job, and group
  - Post, comment, message, and connection invitation
  - Search, catalog, and notification
- Wrapping up

We've gone over the different aspects of LinkedIn and observed the attributes attached to the problem using various UML diagrams. Let's explore the more practical side of things, where we will work on implementing the LinkedIn network using multiple languages. This is usually the last step in an object-oriented design interview process.

We have chosen the following languages to write the skeleton code of the different classes present in LinkedIn:

- Java
- C#
- Python
- C++
- JavaScript

LinkedIn classes

In this section, we will provide the skeleton code of the classes designed in the class diagram lesson.

Note: For simplicity, we are not defining getter and setter functions. The reader can assume that all class attributes are private and accessed through their respective public getter methods and modified only through their public method functions.

Constants

The following code provides the definition of the various enums and custom data types being used in the LinkedIn design:

Note: JavaScript does not support enumerations so we will be using the `Object.freeze()` method as an alternative that freezes an object and prevents further modifications.

```
public class Address {
  private int zipcode;
  private String streetAddress;
  private String city;
  private String state;
  private String country;
}

enum AccountStatus {
  ACTIVE,
  DEACTIVATED,
  BLOCKED,
  DELETED
}

enum ConnectionInvitationStatus {
  PENDING,
  ACCEPTED,
  IGNORED
}

enum JobStatus {
  OPEN,
  ON_HOLD,
  CLOSED
}
```

Constant definitions

Account

The `Account` class refers to an account of a user on LinkedIn. It includes their personal details, such as username, password, etc. It also allows users to reset their existing passwords. The definition of this class is given below:

```
public class Account {
  private String accountId;
  private String password;
  private String username;
  private String email;
  private AccountStatus status;

  public boolean resetPassword();
}
```

The Account class

Person, admin, and user

The `Person` class is an abstract class and contains details like the name, address, phone number, and email. It is derived into the `Admin` and `User` class.

```
// Person will be an abstract class
public abstract class Person {
  private String name;
  private Address address;
  private String email;
  private String phone;
  private Account account;
}

public class Admin extends Person {
  public boolean blockUser(User user);
  public boolean unblockUser(User user);
  public boolean disablePage(CompanyPage page);
  public boolean enablePage(CompanyPage page);
  public boolean deleteGroup(Group group);
}

public class User extends Person {
  private int userId;
  private Date dateOfJoining;
  private Profile profile;
  private List<User> connections;
  private List<User> followedUsers;
  private List<CompanyPage> followCompanies;
  private List<Group> joinedGroups;
  private List<CompanyPage> companyPages;
  private List<Group> createdGroups;

  public boolean sendMessage(Message message);
  public boolean sendInvite(ConnectionInvitation invite);
}
```

The Person, Admin, and User classes

Recommendation, achievement, and analytics

The `Recommendation`, `Achievement`, and `Analytics` classes will provide a user's personal information and make up the `Profile` class. The definition of these classes is given below:

```
public class Recommendation {
  private int userId;
  private Date createdOn;
  private String description;
  private boolean isAccepted;
}

public class Achievement {
  private String title;
  private Date achievedOn;
  private String description;
}

public class Analytics {
  private int searchAppearances;
  private int profileViews;
  private int postImpressions;
  private int totalConnections;
}
```

The Recommendation, Achievement, and Analytics classes

Profile, experience, education, and skill

The `Experience`, `Education`, and `Skill` classes will provide a user's personal information and make up the `Profile` class. The definition of these classes is given below:

```
public class Experience {
  private String title;
  private String company;
  private Address location;
  private Date startDate;
  private Date endDate;
  private String description;
}

public class Education {
  private String school;
  private String degree;
  private Date startDate;
  private Date endDate;
  private String description;
}

public class Skill {
  private string name;
}

public class Profile {
  private String headline;
  private String about;
  private String gender;
  private List<byte> profilePicture;
  private List<User> coverPhoto;

  private List<Experience> experiences;
  private List<Education> educations;
```

The Experience, Education, Skill, and Profile classes

Company, job, and group

LinkedIn users can create groups and company pages. The company page contains information about the company. The company pages will host various job postings. The `Job`, `CompanyPage`, and `Group` classes are shown below:

```
public class Job {
  private int jobId;
  private String jobTitle;
  private Date datePosted;
  private String description;
  private CompanyPage company;
  private String employmentType;
  private Address location;
  private JobStatus status;
}

public class CompanyPage {
  private int pageId;
  private String name;
  private String description;
  private String type;
  private Address address;
  private User createdBy;
  private List<Job> jobs;

  public boolean createJobPosting();
  public boolean deleteJobPosting(Job job);
}

public class Group {
  private int groupId;
  private String name;
  private String description;
  private int totalMembers;
  private List<User> members;
```

The Job, CompanyPage, and Group classes

Post, comment, message, and connection invitation

LinkedIn users can create posts and comments. They can also send messages and connection invitations to other users. The definition of `Post`, `Comment`, `Message`, and `ConnectionInvitation` classes is given below:

```
public class Post {
  private int postId;
  private User postOwner;
  private String text;
  private Date createdOn;
  private int totalShots;
  private int totalShares;
  private List<Comment> comments;

  public boolean updateText();
}

public class Comment {
  private int commentId;
  private User commentOwner;
  private String text;
  private int totalShots;
  private List<Comment> comments;

  public boolean updateText();
}

public class Message {
  private int messageId;
  private String text;
  private List<User> recipients;
  private String text;
  private List<byte> media;

  public boolean addRecipients(List<User> recipients);
```

The Post, Comment, Message, and ConnectionInvitation classes

Search, catalog, and notification

The `SearchCatalog` class contains information on users, company pages, groups, and jobs. It also implements the `Search` interface class to enable the search functionality based on the given criteria (user, company page, group, and job keywords).

The `Notification` class is responsible for sending notifications to users about any new messages, comments, posts, or connection invitations via the built-in notification option.

The definition of these classes is given below:

```
public interface Search {
  // Interface method (does not have a body)
  public List<User> searchUsers(String name);
  public List<CompanyPage> searchCompany(String name);
  public List<Group> searchGroup(String name);
  public List<Job> searchJob(String title);
}

public class SearchCatalog implements Search {
  private HashMap<String, List<User>> users;
  private HashMap<String, List<CompanyPage>> companies;
  private HashMap<String, List<Group>> groups;
  public HashMap<String, List<Job>> jobs;

  public void addNewUser(User user);
  public void addNewCompany(CompanyPage company);
  public void addNewGroup(Group group);
  public void addNewJob(Job job);

  public List<User> searchUsers(String name) {
    // functionality
  }

  public List<CompanyPage> searchCompany(String name) {
    // functionality
  }

  public List<Job> searchJobs(String title) {
    // functionality
  }
}
```

The Search interface and the SearchCatalog and Notification classes

Wrapping up

We've explored the complete design of LinkedIn in this chapter. We've looked at how LinkedIn can be visualized using various UML diagrams and designed using object-oriented principles and design patterns.