Social Network

In the Algorithms and Data structures project you will implement a **Social Network app** that helps connect people who are friends, allows users to post status updates on their walls, allows companies to post ads, etc.

You will use the data structures you learn about and work within the lectures and exercises throughout the semester. You can use only the code you programmed **yourself**, or the code provided in the context of this project (during the lectures and exercises). You should work on the project throughout the semester, as there is 1 intermediate submission.
You should work on the project alone; this is NOT a team project. You cannot submit code you did not write yourself. All submissions will be checked, and any suspicion of a fraud will be reported to the dean according to the exam regulations.

You cannot use **any** third-party code or Java library (e.g. java.Util) without an explicit and prior confirmation by us.
The project is divided into two parts, and the assignments will be released throughout the semester. There is 1 intermediate deadline, and you need to submit your code into canvas.

Code Guidelines

Please follow Oracle's Java Code Conventions, which can be downloaded here:

http://www.oracle.com/technetwork/java/codeconv---138413.html
http://www.oracle.com/technetwork/java/codeconventions---150003.pdf
Most importantly:

- Class names should be nouns, in mixed case with the first letter of each internal word capitalized. Try to keep your class names simple and descriptive.
  - E.g. *MyClass*
- Methods should be verbs, in mixed case with the first letter lowercase, with the first letter of each internal word capitalized.
  - E.g. *runMyMethod()*
- Variables are in mixed case with a lowercase first letter. Internal words start with capital letters.
  - E.g. *myVariable*
- The names of variables declared class constants should be all uppercase with words separated by underscores ("_").
  - E.g. *MY_CONSTANT*
- Opening curly brackets on the same line, closing curly brackets on a new line.
- Use spacing between statements and operators, not between consecutive brackets.
  - E.g. *if ((a == b) && (c == d)) { …*

Mandatory Classes and methods

Your project should include a class named Main that contains a main method with all your test cases and examples.
All classes should implement the ToString() method.

Part 1: Basic classes

Deadline: 05.12.2025 (via Canvas)

Design and implement classes to represent user profile and status update.

- User profile has username, age of the user and a representation of a wall, i.e. collection of status updates
- A status update contains the content, author, privacy setting, age limit, and timestamp.

The privacy setting is an integer number, where 0 represents public, 1 friends, and 2 private.

Add methods to the user profile class to post a status on the wall and to print the wall.

Besides standard status updates, the network should support advertisements. These advertisement messages are posted by special kind of non-personal profiles related to companies. Add support for distinguishing personal and corporate profiles and support for advertisement messages.

When printing a wall, ads and normal messages are alternated: for every 4 user status updates one ad is inserted. Both status updates and ads are sorted chronologically. The timestamp can be represented with an integer number where a higher number corresponds with a newer message. Please note that messages are not necessarily chronologically posted to the network, i.e. messages can be posted with a timestamp in the past or future.

Each class needs to implement the ToString() method which prints the details of the object in the following format:

- “User Profile: user name, age”
- “Status update: timestamp, author, privacy setting, age limit, content”

Implement the interface iSocialNetwork and provide the implementation for the methods listed in the “Part 1” section, namely:

- `public void createUserProfile(String username, int age);`
- `public void createCorporateProfile(String companyName);`
- `public void printWallOf(String username);`
- `public void postStatus(String username, String status, int privacy, int ageLimit, int timestamp);`
- `public void postAd(String companyName, String ad, int ageLimit, boolean paid, int timestamp);`

Important: Do not change the provided interface. Add the interface file to your project and implement the interface.

Part 2: Connections

Deadline: 09.01.2026 (via Canvas)

So far, there were no connections between users. Implement the following methods of the interface, to connect two users and print list of friends of a given user:

- **void** connect(String username1, String username2);
- **void** printFriendListOf(String username);

Since we can connect friends, add implementation of the following method. It enables users to check the wall of a friend. It should show only status updates with privacy settings public and friends. Do not forget to consider the age limit of status updates as well.

- **void** printWallOfAFriend(String username, String friendUsername);

User profiles can connect also with company profiles, i.e. a person can like/follow a company to see their ads on her/his walls.

- **void** follow(String username, String corporateName);

Complete the described functionality to create a fully operational system. All methods should work as expected (considering privacy and age limits) and work in an efficient way.

You must implement the following new methods of the interface.

- **int** distance(String username1, String username2);
- **void** printPath(String username1, String username2);
- **int** distanceExcludeCorporate(String username1, String username2);
- **void** printPathExcludeCorporate(String username1, String username2);

Important: Do not change the provided interface. Add the interface file to your project and implement the interface.

Interface

```
public interface iSocialNetwork {

    /**************************** PART 1 ****************************/
    /*
     * Create new user profile with given parameters: username, age.
     *
     * @param username
     * @param age
     *
     */
    public void createUserProfile(String username, int age);

    /*
     * Create new corporate profile with given parameters: companyName.
     *
     * @param companyName
     *
     */
    public void createCorporateProfile(String companyName);

    /*
     * Print the wall of the user.
     *
     * @param username
     *
     */
    public void printWallOf(String username);

    /*
     * Create a new status update of a user.
     *
     * @param username of the author of the status update
     * @param status content
     * @param privacy setting of the status
     * @param ageLimit
     * @param timestamp
     *
     */
    public void postStatus(String username, String status, int privacy, int
ageLimit, int timestamp);

    /*
     * Create new ad message.
     *
     * @param companyName of the author of the ad
     * @param ad content
     * @param ageLimit
     * @param paid
     * @param timestamp
     *
     */
    public void postAd(String companyName, String ad, int ageLimit, boolean
paid, int timestamp);
```

```
	/**************************** PART 2 ****************************/
	/*
	 * Connect two user.
	 *
	 * @param username1
	 * @param username2
	 *
	 */
	public void connect(String username1, String username2);

	/*
	 * Print all friends of the user.
	 *
	 * @param username
	 *
	 */
	public void printFriendListOf(String username);

	/*
	 * Print contents of the wall of a friend.
	 * Note: make sure people are connected before printing the wall of the
friend
	 *
	 * @param username
	 * @param friendUsername
	 *
	 */
	public void printWallOfAFriend(String username, String friendUsername);

	/*
	 * Connect user with a company, i.e. user follows this company's content.
	 *
	 * @param username
	 * @param corporateName
	 *
	 */
	public void follow(String username, String corporateName);

	/*
	 * Return distance between two users, i.e. the number of steps required
to go from one user to another.
	 *
	 * @param username1
	 * @param username2
	 *
	 * @return distance
	 *
	 */
	public int distance(String username1, String username2);

	/*
	 * Print path between two users, i.e. usernames of users on the path
between them.
	 *
	 * @param username1
	 * @param username2
	 *
	 */
	public void printPath(String username1, String username2);
	/*
	 * Return distance between two users, i.e. the number of steps required
to go from one user to another,
	 * BUT excluding corporate profiles from the path.
	 *
	 * @param username1
	 * @param username2
	 *
	 * @return distance
	 *
	 */
	public int distanceExcludeCorporate(String username1, String username2);

	/*
	 * Print path between two users, i.e. usernames of users on the path
between them,
	 * BUT excluding corporate profiles from the path.
	 *
	 * @param username1
	 * @param username2
	 *
	 */
	public void printPathExcludeCorporate(String username1, String
username2);
}
```