SocialMedia 2\src\socialmedia\SocialMedia.java

```
package socialmedia;

import java.io.IOException;
import java.io.File;
import java.io.FileWriter;
import java.util.ArrayList;

import java.util.*;

import java.io.BufferedWriter;
import java.io.BufferedReader;
import java.io.FileReader;


public class SocialMedia implements SocialMediaPlatform {

    ArrayList<Account> accounts = new ArrayList<>();
    ArrayList<FeedDenizen> orphanage = new ArrayList<>(); // comments wherer the post has
been deleted


    public Account findAccountFromID(int id) throws AccountIDNotRecognisedException{
        for (Account i : accounts){
            if(i.getID() == id){
                return i;
            }
        }
        throw new AccountIDNotRecognisedException();
    }

    public Account findAccountFromHandle(String handle) throws
HandleNotRecognisedException{
        for (Account i : accounts){
            if(handle.equals(i.getHandle())){
                return i;
            }
        }
        throw new HandleNotRecognisedException();
    }


    public Account findAccountFromKnownID(int id){
        for (Account i : accounts){
            if(i.getID() == id){
                return i;
            }
        }
        return new Account(0, null, null);
    }


    public int[] unpackID(int id){
        String strID = String.valueOf(id);
        int postID = Integer.parseInt(strID.substring(strID.length() -4, strID.length()));
        int accountID = Integer.parseInt(strID.substring(0, strID.length()-4));
```

```
        int[] result = {accountID, postID};
        return result;
    }


    public FeedDenizen findPostFromID(int id) throws PostIDNotRecognisedException{
        String strID = String.valueOf(id);
        int postID = Integer.parseInt(strID.substring(strID.length() -4, strID.length()));
        int accountID = Integer.parseInt(strID.substring(0, strID.length()-4));
        Account account = findAccountFromKnownID(accountID);
        FeedDenizen post = account.getPost(postID);
        return post;
    }


    public FeedDenizen findPostFromKnownID(int id){
        String strID = String.valueOf(id);
        int postID = Integer.parseInt(strID.substring(strID.length() -4, strID.length()));
        int accountID = Integer.parseInt(strID.substring(0, strID.length()-4));
        Account account = findAccountFromKnownID(accountID);
        FeedDenizen post = account.getKnownPost(postID);
        return post;
    }


    @Override
    public int createAccount(String handle) throws IllegalHandleException,
InvalidHandleException {
        return createAccount(handle, "");
    }

    @Override
    public int createAccount(String handle, String description) throws
IllegalHandleException, InvalidHandleException {
        if (!(0 < handle.length() && handle.length() < 31)){
            throw new InvalidHandleException();
        }
        char[] array = handle.toCharArray();
        for (char i : array ){
            int ascii = i;
            if (!(64 < ascii && ascii < 123)){
                throw new InvalidHandleException();
            }
        }
        for (Account i : accounts){
            if (handle.equals(i.getHandle())) {
            throw new IllegalHandleException();
                }
        }
        int next_index = accounts.size();
        accounts.add(new Account(next_index, handle, description));
        return next_index;
    }

    @Override
    public void removeAccount(int id) throws AccountIDNotRecognisedException {
        Account i = findAccountFromID(id);
        List<Integer> orphans = i.delete();
        for (Integer childID : orphans){
```

```
            orphanage.add(findPostFromKnownID(childID));
        }
        accounts.remove(i);
    }

    @Override
    public void removeAccount(String handle) throws HandleNotRecognisedException {
        Account i = findAccountFromHandle(handle);
        List<Integer> orphans = i.delete();
        for (Integer childID : orphans){
            orphanage.add(findPostFromKnownID(childID));
        }
        accounts.remove(i);
    }

    @Override
    public void changeAccountHandle(String oldHandle, String newHandle)
            throws HandleNotRecognisedException, IllegalHandleException,
InvalidHandleException {
        if (!(0 < newHandle.length() && newHandle.length() < 31)){
            throw new InvalidHandleException();
        }
        int acc = -1;
        for (int i = 0; i > accounts.size(); i++){
            if (newHandle.equals(accounts.get(i).getHandle())) {
                throw new IllegalHandleException();
            }
            if (oldHandle.equals(accounts.get(i).getHandle())) {
                acc = i;
            }
            if ((i == accounts.size() - 1) && (acc > -1)) {
                accounts.get(acc).setHandle(newHandle);
            }
        }
        throw new HandleNotRecognisedException();
    }

    @Override
    public void updateAccountDescription(String handle, String description) throws
HandleNotRecognisedException {
        findAccountFromHandle(handle).setDescription(description);
    }

    @Override
    public String showAccount(String handle) throws HandleNotRecognisedException {
        Account i = findAccountFromHandle(handle);
        return  "ID: " + i.getID() + System.lineSeparator() +
                "Handle: " + handle + System.lineSeparator() +
                "Description: " + i.getDescription() + System.lineSeparator() +
                "Post count: " + i.getNumberOf("Post") + System.lineSeparator() +
                "Endorce count: " + i.getNumberOf("Endorcement");
    }

    @Override
    public int createPost(String handle, String message) throws
HandleNotRecognisedException, InvalidPostException {
        if (!(0 < message.length() && message.length() < 100)){
            throw new InvalidPostException();
        }
```

```
        Post post = findAccountFromHandle(handle).post(message);
        return post.getID();
    }

    @Override
    public int endorsePost(String handle, int id) throws HandleNotRecognisedException,
PostIDNotRecognisedException, NotActionablePostException {
        FeedDenizen post = findPostFromID(id);
        if (post.getClass().getSimpleName().equals("Post")){
            Endorsement end = findAccountFromHandle(handle).endorse(id);
            post.endorse();
            return end.getID();
        }
        throw new NotActionablePostException();
    }

    @Override
    public int commentPost(String handle, int id, String message) throws
HandleNotRecognisedException, PostIDNotRecognisedException, NotActionablePostException,
InvalidPostException {
        if (!(0 < message.length() && message.length() < 100)){
            throw new InvalidPostException();
        }
        FeedDenizen post = findPostFromID(id);
        if (!post.getClass().getSimpleName().equals("Endorsement")){
            Comment comment = findAccountFromHandle(handle).comment(message, id);
            return comment.getID();
        }
        throw new NotActionablePostException();
    }

    @Override
    public void deletePost(int id) throws PostIDNotRecognisedException {
        int[] ids = unpackID(id);
        Account account = findAccountFromKnownID(ids[0]);
        List<Integer> orphans = account.deletePost(ids[1]);
        for (Integer childID : orphans){
            orphanage.add(findPostFromKnownID(childID));
        }
    }

    @Override
    public String showIndividualPost(int id) throws PostIDNotRecognisedException {
        int[] ids = unpackID(id);
        Account account = findAccountFromKnownID(ids[0]);
        FeedDenizen post = findPostFromID(id);
        return  "ID: " + post.getID() + System.lineSeparator() +
                "Account: " + account.getHandle() + System.lineSeparator() +
                "No. endorsements: " + post.getTotalEndorsments()  + "| No. comments: " +
post.getTotalComments() + System.lineSeparator() +
                post.getText();

    }

    @Override
    public StringBuilder showPostChildrenDetails(int id)throws
PostIDNotRecognisedException, NotActionablePostException {
        List<Integer> posts = new ArrayList<>();
        posts.add(id);
```

```
        StringBuilder str = new StringBuilder();
        String indent = "    ";

        int indentLevel = 0;
        int index = 0;
        while (index < posts.size()){
            if (posts.get(index) != 0){
                String[] comment =
showIndividualPost(posts.get(index)).split(System.lineSeparator());
                for (String line : comment){
                    if ((indentLevel > 0) && (line.substring(0,3).equals("ID:"))){
                        str.append(indent.repeat(indentLevel -1));
                        str.append("| > ");
                        str.append(line);
                        str.append(System.lineSeparator());
                    } else {
                        str.append(indent.repeat(indentLevel));
                        str.append(line);
                        str.append(System.lineSeparator());
                    }
                }


                FeedDenizen post = findPostFromID(posts.get(index));
                List<Integer> children = post.getChildren();
                if (children.size() > 0){
                    str.append(indent.repeat(indentLevel));
                    str.append("|");
                    str.append(System.lineSeparator());

                    children.add(0);
                    posts.addAll(index+1, children);
                    indentLevel += 1;
                }

            } else {
                indentLevel -=1;
            }
            index += 1;
        }


        return str;
    }

    @Override
    public int getNumberOfAccounts() {
        return accounts.size();
    }

    @Override
    public int getTotalOriginalPosts() {
        int count= 0;
        for (Account i : accounts){
            count += i.getNumberOf("Post");
        }
        return count;
    }
```

```java
@Override
public int getTotalEndorsmentPosts() {
    int count= 0;
    for (Account i : accounts){
        count += i.getNumberOf("Endorsement");
    }
    return count;
}

@Override
public int getTotalCommentPosts() {
    int count= 0;
    for (Account i : accounts){
        count += i.getNumberOf("Comment");
    }
    return count;
}

@Override
public int getMostEndorsedPost() {
    int previous = 0;
    int previousID = 0;
    for (Account account : accounts){
        if (account.mostEndorsments()[0] - previous > 0){
            previous = account.mostEndorsments()[0];
            previousID = account.mostEndorsments()[1];
        }
    }
    return previousID;
}

@Override
public int getMostEndorsedAccount() {
    int previous = 0;
    int previousID = 0;
    for (Account account : accounts){
        if (account.mostEndorsments()[0] - previous > 0){
            previous = account.mostEndorsments()[0];
            previousID = account.getID();
        }
    }
    return previousID;
}


@Override
public void erasePlatform() {
    accounts.removeAll(accounts);
    orphanage.removeAll(orphanage);

}

@Override
public void savePlatform(String filename) throws IOException {
    try{
        File f = new File(filename);
        if (f.exists()){
            f.delete();
        }
```

```
            f.createNewFile();
            FileWriter fWriter = new FileWriter(filename);
            BufferedWriter bWriter = new BufferedWriter(fWriter);
            for (Account account : accounts){
                StringBuffer line = new StringBuffer();
                String[] chars =  {String.valueOf(account.getID()), account.getHandle(),
account.getDescription()};
                for (String i : chars){
                    line.append(i + ",");
                }
                line.append("(");
                for (FeedDenizen post : account.getTimeline()){
                    char type = post.getClass().getSimpleName().charAt(0);
                    line.append(type + "," + post.getID() + "," + post.getText() + "," +
post.getTotalComments() + "," + post.getTotalEndorsments() + "," + post.getParent() + ",");

                    line.append("(");
                    for (Integer child : post.getChildren()){
                        line.append(child + "|");
                    }
                    line.append(")/");
                }
                line.append(")");
                bWriter.write(line.toString());
                bWriter.flush();
            }
            bWriter.close();
        } catch (IOException e){
            throw new IOException();
        }


    }

    @Override
    public void loadPlatform(String filename) throws IOException, ClassNotFoundException {
        BufferedReader reader;
        String line = null;
        try{
            reader = new BufferedReader(new FileReader(filename));

            while ((line = reader.readLine()) != null) {
                String[] accDesc = line.split(",", 4);
                String strPosts = accDesc[3].substring(1, accDesc[3].length()-1);
                String[] posts = strPosts.split("/");
                int id = Integer.parseInt(accDesc[0]);
                Account account = new Account(id, accDesc[1], accDesc[2]);
                for (String p : posts){
                    String[] splitPost = p.split(",");
                    if (splitPost[0].equals("P")){
                        Post newpost = account.post(splitPost[2]);
                        int endorcements = Integer.parseInt(splitPost[4]);
                        newpost.endorcements = endorcements;
                        int noComments = Integer.parseInt(splitPost[3]);
                        if (noComments > 0){
                            String strComments = splitPost[6].substring(1,
splitPost[6].length() - 1);
                            String[] comments = strComments.split("|");
                            List<Integer> children = new ArrayList<>();
```

```
                            for (String i : comments){
                                int y = Integer.parseInt(i);
                                children.add(y);
                            }
                            newpost.children = children;
                        }
                    }
                    if (splitPost[0].equals("C")){
                        int parent = Integer.parseInt(splitPost[5]);
                        Comment newpost = account.comment(splitPost[2], parent);
                        int endorcements = Integer.parseInt(splitPost[4]);
                        newpost.endorcements = endorcements;
                        int noComments = Integer.parseInt(splitPost[3]);
                        if (noComments > 0){
                            String strComments = splitPost[6].substring(1,
splitPost[6].length() - 1);
                            String[] comments = strComments.split("|");
                            List<Integer> children = new ArrayList<>();
                            for (String i : comments){
                                int y = Integer.parseInt(i);
                                children.add(y);
                            }
                            newpost.children = children;
                        }
                    }
                    if (splitPost[0].equals("E")){
                        int parent = Integer.parseInt(splitPost[5]);
                        try {
                            account.endorse(parent);
                        } catch (Exception e) {
                        }

                    }
                }
            }

            reader.close();

        } catch (IOException e){
            throw new IOException();
        }


    }

}
```