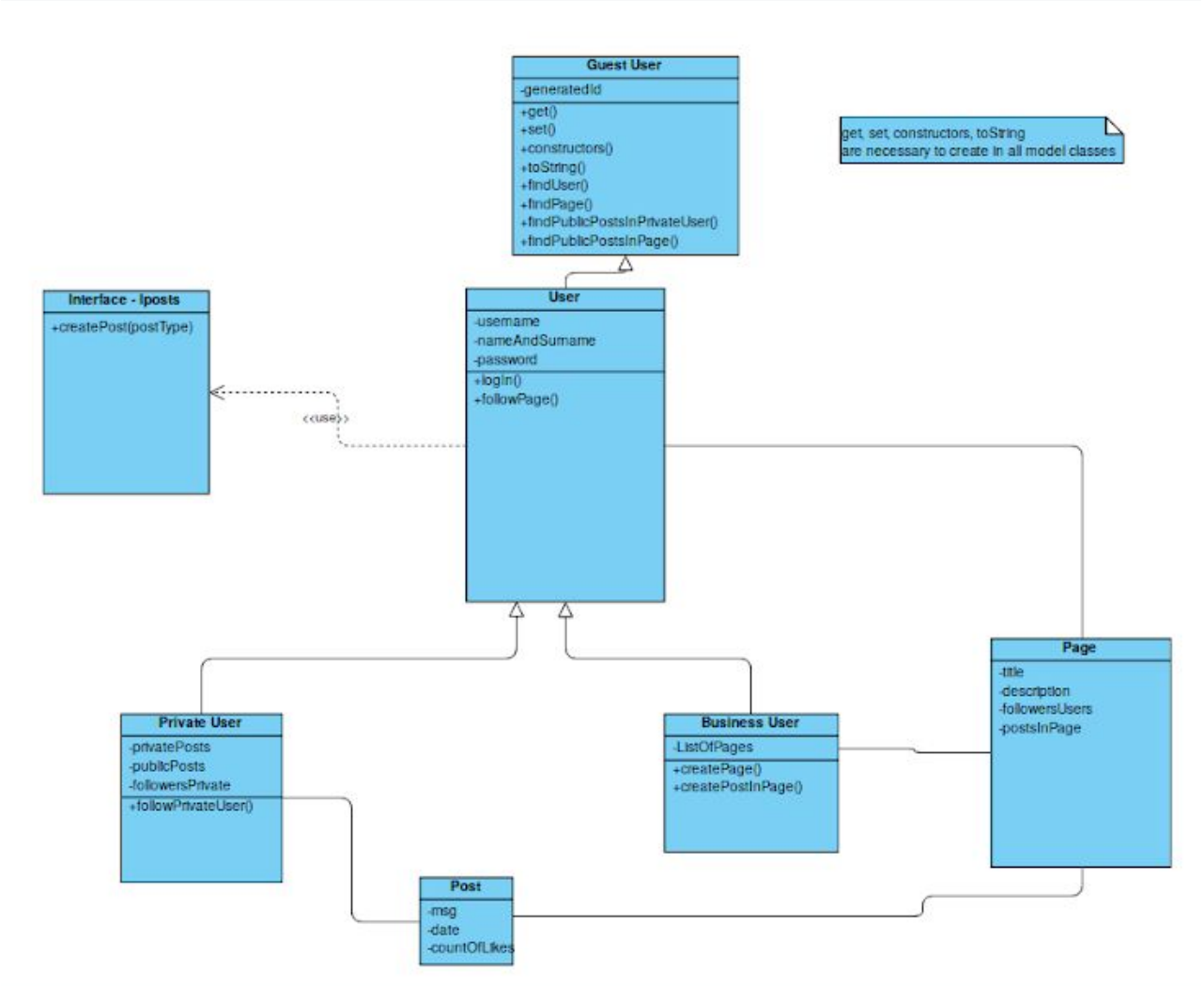
VENTSPILS AUGSTSKOLA in collaboration with accenture

Seminar 3

Classes, Objects, Inheritance, Interfaces

- **Open Eclipse(or NetBeans, or ...), create new empty Java Application Project "Seminar_3" and create a miniFaceBook information system (see an example below - the changes of UML diagram is possible)**



Exercise 1- Post class

- **Create a new package named model**
- **Create a new class Post (no main() method in it) containing:**
  - **variables:**
    - **String msg;**
    - **Date date;**
    - **int countOfLikes;**
  - **no-argument and argument constructors;**
  - **methods:**
    - **set and get;**
    - **toString()**

Exercise 2- Page class

- **Create a new class Page (in package model) containing:**
  - **variables:**
    - **String title;**
    - **String description;**
    - **ArrayList<User> followers; //see User class description below**
    - **ArrayList<Post> postsInPage;**
  - **no-argument and argument constructors:**
  - **methods:**
    - **set and get;**
    - **toString().**

Exercise 3- Create classes related to User.

There are three types of User:

1. Default User, which can see all public Posts and Pages and find the User and Pages; Info: Page is similar as Groups;
2. Private User, which can register, login, create posts (private or public), follow a page and follow a private user;
3. Business User, which can register, login, follow a page, create a page and create a public post in the page.

Exercise 3.1 - create a package model.users

- **Create a new package named model.users**

Exercise 3.2 - create GuestUser class

- **Create a new class GuestUser containing:**
  - **variables:**
    - **int generatedId;**
  - **no-argument constructor;**
  - **methods (will be added extra methods later):**
    - **set and get;**
    - **toString()**

Exercise 3.3 - create User class

- **Create a new abstract class User (derived from GuestUser class) containing:**
  - **variables:**
    - **String nameAndSurname**
    - **String username**
    - **String password (can be created using MD5 Hashing cryptographic method)**
  - **no-argument and argument constructors;**
  - **methods:**
    - **set and get;**
    - **toString();**
    - **login();**
    - **followPage().**

Exercise 3.4 - create PrivateUser class

- **Create a new class PrivateUser (derived from User class) containing:**
  - **variables:**
    - **ArrayList<Post> privatePosts;**
    - **ArrayList<Post> publicPosts;// all posts can be stored in one ArrayList if PostType enum is used (see at Exercise 4.2)**
    - **ArrayList<User> followers;**
  - **no-argument and argument constructors;**
  - **methods:**
    - **set and get;**

- toString();
- followPrivateUser().

Exercise 3.5 - create BusinessUser class

- Create a new class BusinessUser (derived from User class) containing:
  - variables:
    - ArrayList<Pages> listOfPages;
  - no-argument and argument constructors;
  - methods:
    - set and get;
    - toString();
    - createPage().

Exercise 4 - interface IPost creation and usage

Exercise 4.1 - create package

- Create new package named ifaces

Exercise 4.2 - create interface

- Create a new interface IPost (can be titled as IPostCreation) containing:
- method:
  - createPost();

It is possible to create an enum PostType where two constants are stored - *private* and *public* and to pass PostType object as an input parameter in createPost() method.

Exercise 4.3 - interface implementation

- **Add interface IPost implementation in class User. Create the functionality part of createPost() method in PrivateUser and BusinessUser classes;**
- **Create createPostInPage() method in BusinessUser class and call IPost method createPost() in mentioned method.**

Exercise 5 - additional

- **Create new methods in class GuestUser:**
  - **findUser()**
  - **findPage()**
  - **findPublicPostsInPrivateUser()**
  - **findPublicPostInPage()**

Exercise 6 - IS usage

- **Add method, where multiple objects will be created and stored in Arrays or ArrayLists:**
  - **Create at least 2 objects of GuestUsers;**
  - **Create at least 2 objects of PrivateUser;**
  - **Create at least 2 objects of BusinessUser;**
  - **Create at least 2 objects of Post for each PrivateUser (public and private);**
  - **Create at least 1 object of Page for each BusinessUser and add at least 2 posts in each Page.**
- **Add method, where *following* functionality for PrivateUser - followPrivateUser() and Page - followPage() will be demonstrated;**
- **Add method, where findUser(), findPage(), findPublicPostsInPrivateUser(), findPublicPostInPage() usage will be demonstrated;**
- **Add method, where register() and login() usage will be demonstrated;**
- **Add method, where all posts (private and public) of specific PrivateUser will be shown;**
- ***Add method, where all public posts will be sorted by countOfLikes;**
- ***Add method, where all posts will be sorted by date.**