17.04.25 Wkp-HAs OOP Tyra Adeoye

! Ich hatte meinen Laptop zum Bearbeitungszeitp. leider kurzfristig zur Verfügung und musste die Aufgaben diesmal so machen.
→ Ich würde trotzdem um Feedback bitten :)

→ GitHub link für folgende HAs:

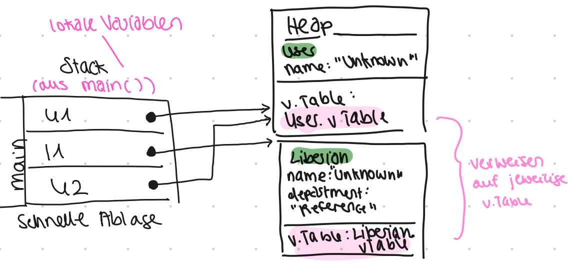
https://github.com/tyraomo/wkp-homeworks.git

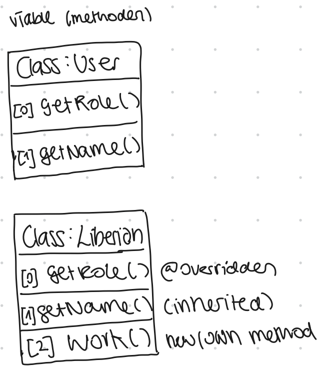
Group & Tutors's name: (according to Ilias)
→ Tuesday 10-12, S91, Adrian Bajraktari

Task 1

```
public class Publication{
     private String title;
     private int year;

  public Publication(String title, int year){
        this.title = title;
        this.year = year;
      }
   public String getInfo(){
        return String.format(" The publication has the title: %, and was published in %.", title, year);
  }
}

class Book extends Publication{
     private String author;

   public Book(String author, String title, int year){
     super(title, year); // ruft Konstruktor von Publication auf
     this.author = author;
   }
  @Override
  public String getInfo(){
       return String.format(" The publication has the title: %, and was published in %. by %.", title, year, author);
 }
}


class Textbook extends Book{
     private String subject;

   public Textbook(String subject, int year, String autor     ){
           super(title, year, author) // Aufruf Konstruktor v. Book
           this.subjekt = subject;
      }
   @Override
    public String getInfo(){
      return String.format(" The publication has the title: %, and was published in %. by %, it's subject is: %.  ", title, year, author, subject);

  }
}
```

Task 2




vTable (methoden)
Class: User
[0] getRole()
[1] getName()
Class: Liberian
[0] getRole() @overridden
[1] getName() (inherited)
[2] work() new/own method

1st print : "General User"

2nd print } points towards the : "Unknown Liberian"
3rd print } same

30-32

u1. getRole()

static indice: User
↳ has getRole() → compiles ✓

l1. getRole()

static indice: Liberian
↳ has getRole() → compiles ✓

u2. getRole()

static indice: User
↳ has getRole() → compiles ✓

34-35

l1. getName()

static indice: Liberian
↳ has getName() → compiles ✓

u1. getName()

static indice: User
↳ has getName() → compiles ✓

37-38

l1. work()

static indice: Liberian
↳ has work() → compiles ✓

u2. work()

static indice: User
↳ doesn't have work() since it's a new method in Liberian → doesn't compile ✗

Task 3

```
public class Mail{
private String sender;
 private String subject;
private String message;
 private String datetime;
 private boolean read; //im Construktor

public Mail(String sender, String subject, Stringe message, String datetime){
  this.sender = sender;
  this.subject = subject;
  this.message = message;
  this.datetime= datetime;
  this.read = false;
 }

 public boolean markAsRead(){
    read = true;
}
```

```
public String getSender(){
     return sender;
}
 public String getSubject(){
     return subject;
 }
public String getDatetime(){
     return datetime;
}
public void print(){
  System.out.println(
"% from % on %. %", subject, sender,
  datetime, message);
```

```
 public String print(){
   return String.format("% from % on % : % ", subject, sender, datetime, message);
  }
}
```

```
public class Inbox {
 private ArrayList<Mail> mails;
                 -> Generics: Liste enhält nur Objekte vom Typ: Mail

     public Inbox(){
      mails = new ArrayList<>();
    }
public void add(Mail mail){
   mails.add(mail);
 }
public void printHeaders(){
  if(read) {
  System.out.Println("read|"subject+"|"+sender+"|"+ datetime);

 }else {
   return "not read!"
  }
}

public void open(int index){
 if (0< index || index >= mail size){ //Gültigkeit prüfen
  System.out.Println("Ungültig");
  return
 }
Mail mail = mails.get(index);
mail.markAsRead();
 mail.print();
}

public int countUnread(){
 int count = 0;
for ( Mail mail : mails){
  if (!mail.markAsRead){
   Count++;
  }
 } return count;
}
```

```
public class MailboxApp{
 public static void main (String[] args){
 Inbox inbox = new Inbox();
 Mail m1= new Mail("Tyra.oa@gmail.com", "Urlaub", "Bin weg!" ,"2025-04-17 10 30");
 Mail m2= new Mail("Sara.gf @gmail.com", "Meeting", "Wann genau?" ,"2025-04-19 14 30");

 inbox.add(m1);
 inbox.add(m2);

  m1.markAsRead(); //eine gelesen markieren

 //ungelesene ausgeben
 System.out.println("Ungelesene Mails:" + inbox.countUnread());

 inbox.open(0); //öffnet 1. Mail -> gelesen

 (inbox.printHeaders();)
 }
}
```