

Übungen zum Propädeutikum Programmierung in der Bioinformatik

Blatt 7

Termin: Dienstag, 3. Dezember 2019

Übung 1 *Vererbung*

Implementiere eine Klasse **Person**:

1. mit den `private` Attributen `name` und `age`, die über Getter und Setter zugänglich sein sollen.
2. mit einem geeigneten Konstruktor, der die Attribute befüllt.
3. Überschreibe die `String toString()` Methode, so dass die Informationen der Person ausgegeben werden.
4. Überschreibe die `boolean equals(Object obj)` Methode, welche überprüfen soll ob das Objekt im Parameter Teil der Klasse Person ist und ob 2 Personen anhand von Name und Alter identisch sind.

Implementiere eine Klasse **Student**, die von **Person** erbt:

1. mit dem zusätzlichen Attribut `String matrikelnummer` samt Getter und Setter
2. einem geeigneten Konstruktor (erst `super` Konstruktor aufrufen dann Matrikelnummer setzen)
3. Überschreibe hier auch die `boolean equals(Object obj)` Methode, welche erst `Equals`-Methode ihrer Superklasse ausführt und nachträglich noch überprüft ob `obj` ein Student ist und ob die Matrikelnummern gleich sind.

Implementiere eine Klasse **Professor** die genau wie **Student** von **Person** erbt:

1. Statt einer Matrikelnummer hat ein Professor eine `ArrayList` von `Strings`, in dem die Namen der aktuell von ihm gehaltenen Veranstaltungen enthalten sein sollen.
2. Überschreibe die `String toString()` Methode, sodass sie nicht nur die **Person**-Informationen liefert, sondern auch die Namen der gehaltenen Veranstaltungen.

Teste deine Klassen mit Hilfe einer Klasse namens **Runner**, welche eine `main`-Methode enthält. Lege 3 Professoren des Lehrstuhls samt ihrer Veranstaltungen (siehe Instituts-Website, ein oder zwei reichen), dich selbst, drei deiner Kommilitonen und drei deiner Lieblings-Prominenten an. Benutze `Collections` und `Schleifen`, um alle 100 paarweisen `equals`-Vergleiche dieser 10 Personen durchzuführen. Für jeden Vergleich sollen beide Personen und das Vergleichsergebnis ausgegeben werden. (Tipp: `ArrayList<Person>` mit allen 10 Personen und 2 Schleifen darüberlegen)

Lösung 1

Person-Klasse: (ohne Getter und Setter)

```
1 public class Person {
2     private int age;
3     private String name;
4
5     public Person(int age, String name) {
6         this.age = age;
7         this.name = name;
8     }
9
10    @Override
11    public String toString() {
12        return String.format(name + " age: " + age + " Years old");
13    }
14
15    @Override
16    public boolean equals(Object obj) {
17        if(obj instanceof Person){
18            if(((Person) obj).age == this.age && ((Person) obj).name.equals(this.name)) return true;
19            else return false;
20        }else return false;
21    }
22 }
```

Student-Klasse: (ohne Getter und Setter)

```
1 public class Student extends Person {
2     String matrikelnummer;
3     public Student(int age, String name, String matrikelnummer) {
4         super(age,name);
5         this.matrikelnummer = matrikelnummer;
6     }
7
8     @Override
9     public boolean equals(Object obj) {
10        if(super.equals(obj) && obj instanceof Student){
11            if(((Student) obj).matrikelnummer.equals(this.matrikelnummer)) return true;
12            else return false;
13        }else return false;
14    }
15 }
```

Professor-Klasse: (ohne Getter und Setter)

```
1 public class Professor extends Person {
2     ArrayList<String> veranstaltungen;
3     public Professor(int age, String name){
4         super(age ,name);
5     }
6
7     @Override
8     public String toString() {
9         String output = super.toString() + " hält folgende Veranstaltungen: ";
10        for(int i=0; i<veranstaltungen.size(); i++){
11            output += veranstaltungen.get(i) + " ";
12        }
13        return output;
14    }
15 }
```

Runner-Klasse:

```
1 public class Runner {
2     public static void main(String[] args) {
3         ArrayList<Person> persons = new ArrayList<>();
4         Professor baumbach = new Professor(45, "Baumbach");
5         persons.add(baumbach);
6         Student chris = new Student(23, "Christian Hoffmann", "123");
7         Person mario = new Person(40, "Mario");
8         // 7 andere Personen und immer alle zur ArrayList adden
9
10        for(int i=0; i<persons.size(); i++){
11            for(int j=0; j<persons.size(); j++){
12                System.out.println(persons.get(i) + " vs " + persons.get(j) + " : "
13                    + persons.get(i).equals(persons.get(j)));
14            }
15        }
16    }
17 }
```

Übung 2 Vererbungen bei Sequenzen

Überlege dir selbst eine **sinnvolle** Vererbungsstruktur für verschiedene biologische Sequenzen. Du kannst dir Inspiration aus den letzten Übungsblättern holen (z.B. DNA, RNA, Protein), bzw diese Klassen auf eine Vererbungsstruktur ummodellieren. Wenn du deinen Code korrigiert haben willst schicke uns die Javafiles per Mail!