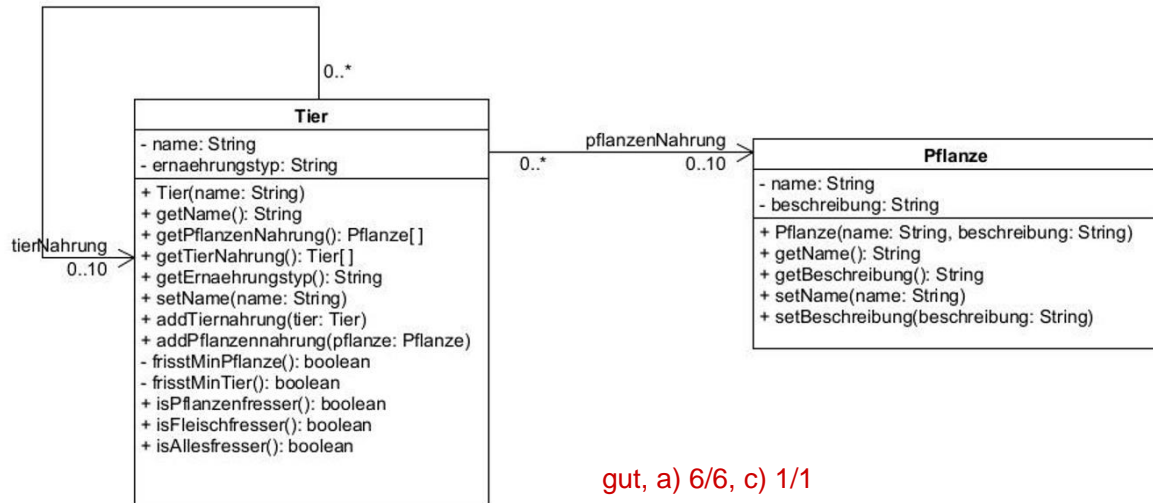


OMP Übungsblatt 01

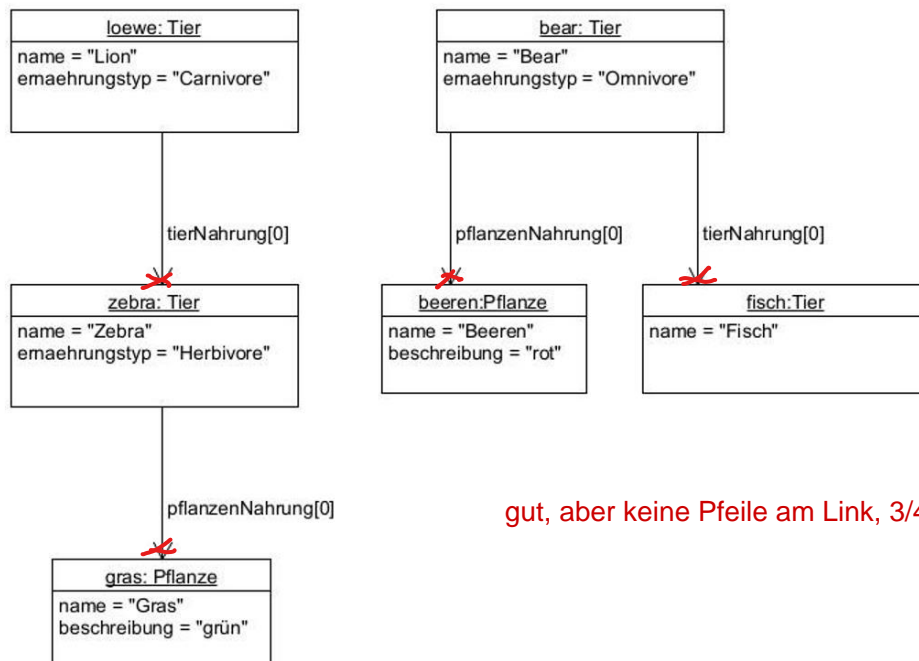
Gruppe: Die Blassnasen
Mail: cora.haase@gmail.com

Abgabe: 20.04.23

Aufgabe 1 a) & c)



e)



```

1 package ue01;
2
3 public class Tier {
4     private String name;
5     private Pflanze[] pflanzenNahrung;
6     private Tier[] tierNahrung;
7     private String ernaehrungstyp;
8
9     // Konstruktor
10    public Tier(String name) {
11        this.name = name;
12        this.pflanzenNahrung = new Pflanze[10];
13        this.tierNahrung = new Tier[10];
14    }
15
16    // Getter
17    public String getName() {
18        return name;
19    }
20
21    public Pflanze[] getPflanzenNahrung() {
22        return pflanzenNahrung;
23    }
24
25    public Tier[] getTierNahrung() {
26        return tierNahrung;
27    }
28
29    public String getErnaehrungstyp() {
30        if(this.isFleischfresser()) {
31            this.ernaehrungstyp = "Carnivore";
32        } else if(this.isPflanzenfresser()) {
33            this.ernaehrungstyp = "Herbivore";
34        } else if(this.isAllesfresser()){
35            this.ernaehrungstyp = "Omivore";
36        }
37        return ernaehrungstyp;
38    }
39
40    // Setter
41    public void setName(String name) {
42        this.name = name;
43    }
44
45    // Methode zum Hinzufügen von Pflanzennahrung
46    public void addPflanzennahrung(Pflanze plant) {
47        for (int i = 0; i < pflanzenNahrung.length; i++) {
48            if (pflanzenNahrung[i] == null) {
49                pflanzenNahrung[i] = plant;
50                return;
51            }
52        }
53    }
54
55    // Methode zum Hinzufügen von Tiernahrung
56    public void addTiernahrung(Tier tier) {
57        for (int i = 0; i < tierNahrung.length; i++) {
58            if (tierNahrung[i] == null) {
59                tierNahrung[i] = tier;
60                return;
61            }
62        }
63    }
64
65    // Methode zum Prüfen, ob Tier Pflanzen-, Fleisch- oder Allesfresser ist.
66    public boolean isPflanzenfresser() {
67        return (!frisstMinTier()) && (frisstMinPflanze());

```

Attribut ernaehrungstyp nicht notwendig, wird sowieso bei jedem Zugriff neu berechnet

```
68     }
69
70     public boolean isFleischfresser() {
71         return (!frisstMinPflanze()) && (frisstMinTier());
72     }
73
74     public boolean isAllesfresser() {
75         return (frisstMinPflanze()) && (frisstMinTier());
76     }
77
78     // Hilfsmethode um zu prüfen, ob das Tier mindestens eine Pflanze frisst
79     private boolean frisstMinPflanze() {
80         for (int i = 0; i < getPflanzenNahrung().length; i++) {
81             if (getPflanzenNahrung()[i] != null) {
82                 return true;
83             }
84         }
85         return false;
86     }
87
88     // Hilfsmethode um zu prüfen, ob das Tier mindestens ein Tier frisst
89     private boolean frisstMinTier() {
90         for (int i = 0; i < getTierNahrung().length; i++) {
91             if (getTierNahrung()[i] != null) {
92                 return true;
93             }
94         }
95         return false;
96     }
97
98 }
99
100
101
```

b) 6/6

```

1 package ue01;
2
3 public class BioTest {
4
5     public static void main(String[] args) {
6         // Erstelle Pflanzen
7         Pflanze gras = new Pflanze("Gras", "grün");
8         Pflanze beeren = new Pflanze("Beeren", "rot");
9
10        // Erstelle Tiere und füge Nahrung hinzu
11        Tier zebra = new Tier("Zebra");
12        zebra.addPflanzennahrung(gras);
13
14        Tier loewe = new Tier("Lion");
15        loewe.addTiernahrung(zebra);
16
17        Tier fisch = new Tier("Fisch");
18
19        Tier baer = new Tier("Bear");
20        baer.addPflanzennahrung(beeren);
21        baer.addTiernahrung(fisch);
22
23
24        //Vorgegebene Ausgabe:
25        System.out.println(gras.getName() + " ist " + gras.getBeschreibung() + ".");
26        System.out.println(beeren.getName() + " sind " + beeren.getBeschreibung() + ".");
27
28        Pflanze[] zebraFood = zebra.getPflanzenNahrung();
29        System.out.println(zebra.getName() + "s fressen " + zebraFood[0].getName() + ".");
30    };
31
32    Tier[] loewenFood = loewe.getTierNahrung();
33    System.out.println(loewe.getName() + "s fressen " + loewenFood[0].getName() + "s.");
34    };
35
36    //Tier und Pflanzenarrays in neuer Variable zwischenspeichern, um die Namen der
37    // Objekte an einer
38    // gewünschten Indexstelle auszugeben.
39    Tier[] baerenNahrungTier = baer.getTierNahrung();
40    Pflanze[] baerenNahrungPflanze = baer.getPflanzenNahrung();
41
42    //Weiter mit aus Aufgabe vorgegebener Ausgabe:
43    System.out.println(baer.getName() + "en fressen " + baerenNahrungTier[0].getName() + "e.");
44    System.out.println(baer.getName() + "en fressen " + baerenNahrungPflanze[0].getName() + ".");
45
46    System.out.println(zebra.getName() + " is an " + zebra.getErnaehrungstyp() + ".");
47    };
48    System.out.println(loewe.getName() + " is a " + loewe.getErnaehrungstyp() + ".");
49    System.out.println(baer.getName() + " is an " + baer.getErnaehrungstyp() + ".");
50
51 }
52 }

```

gut, 3/3

```
1 package ue01;
2
3 public class Pflanze {
4     private String name;
5     private String beschreibung;
6
7     //Konstruktor
8     public Pflanze(String name, String beschreibung) {
9         this.name = name;
10        this.beschreibung = beschreibung;
11    }
12    // Getter
13    public String getName() {
14        return name;
15    }
16
17    public String getBeschreibung() {
18        return beschreibung;
19    }
20    // Setter
21    public void setName(String name) {
22        this.name = name;
23    }
24
25    public void setBeschreibung(String beschreibung) {
26        this.beschreibung = beschreibung;
27    }
28 }
29
```