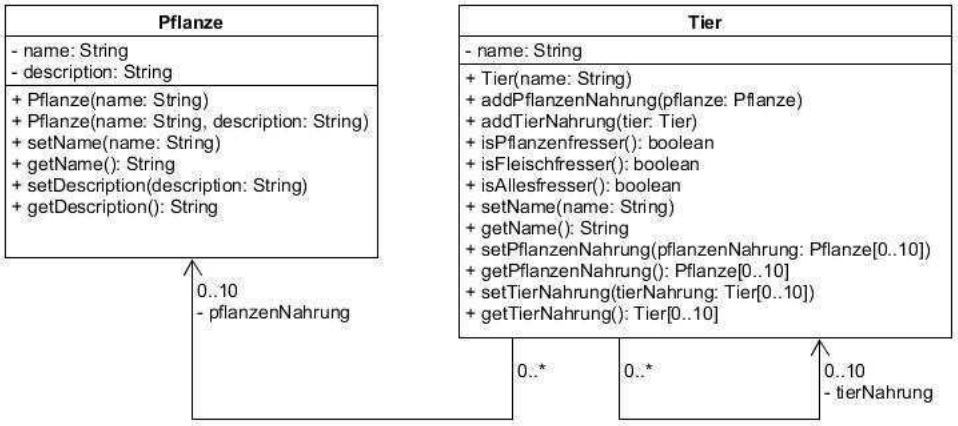
Abgabe OMP Blatt 01

Weerts, Steffen, steffen.weerts@uni-oldenburg.de. Gruppenname: 17

Aufgabe 1

(a) UML-Klassendiagramm:

6/6



(b) Java-Implementierung der Klassen Pflanze und Tier:
6/6 (Der Quellcode befindet sich auch als .java Datei im Anhang)

```java
class Pflanze {
    // Attribute
    private String name;
    private String description;

    // Methoden
    public Pflanze(String name) {
        this.name = name;
    }

    public Pflanze(String name, String description) {
        this.name = name;
        this.description = description;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getName() {
        return name;
    }

    public void setDescription(String description) { this.description = description; }

    public String getDescription() {
        return description;
    }
}
```

```java
class Tier {
    // Attribute
    private String name;
    private Pflanze[] pflanzenNahrung = new Pflanze[10];
    private Tier[] tierNahrung = new Tier[10];

    public Tier(String name) {
        this.name = name;
    }

    public void addPflanzenNahrung(Pflanze pflanze) {
        for (int i = 0; i < 10; i++) {
            if(this.pflanzenNahrung[i] == null) {
                this.pflanzenNahrung[i] = pflanze;
                break;
            }
        }
    }

    public void addTierNahrung(Tier tier) {
        for (int i = 0; i < 10; i++) {
            if(this.tierNahrung[i] == null) {
                this.tierNahrung[i] = tier;
                break;
            }
        }
    }

    public boolean isPflanzenfresser() {
        return isstPflanze() && !isstTier();
    }

    public boolean isFleischfresser() {
        return !isstPflanze() && isstTier();
    }

    public boolean isAllesfresser() {
        return isstPflanze() && isstTier();
    }

    // Hilfsmethoden
    private boolean isstPflanze() {
     1) if(pflanzenNahrung[0] != null) {
            return true;
        } else {
```

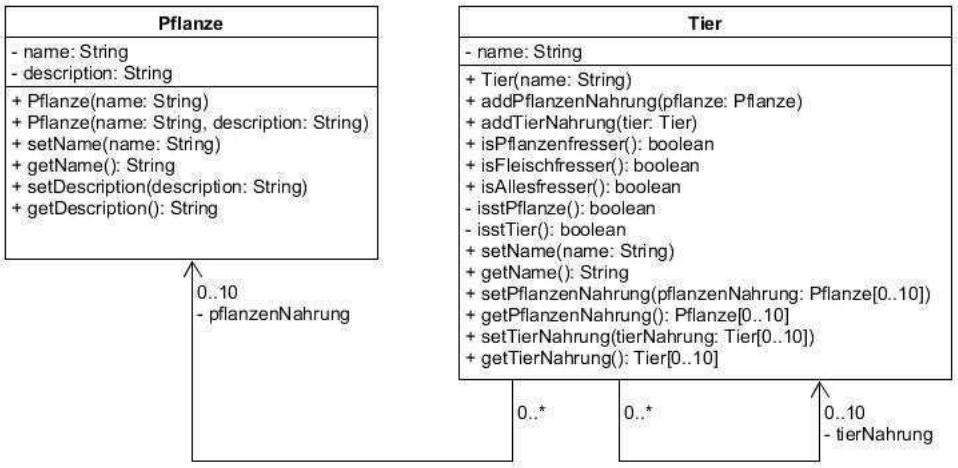
(Fortsetzung auf nächster Seite)

1) Das funktioniert so nicht mehr, sobald das Element auf Position 0 wieder entfernt/auf null gesetzt wird.
Die Methode gibt dann false zurück, theoretisch könnten auf Position 1-9 aber noch Pflanzen gespeichert sein.

```
            return false;
        }
    }

    private boolean isstTier() {
        if(tierNahrung[0] != null) {
            return true;
        } else {
            return false;
        }
    }

    // get und set Methoden
    public void setName(String name) {
        this.name = name;
    }

    public String getName() {
        return name;
    }

    public void setPflanzenNahrung(Pflanze[] pflanzenNahrung) {
        this.pflanzenNahrung = pflanzenNahrung;
    }

    public Pflanze[] getPflanzenNahrung() {
        return pflanzenNahrung;
    }

    public void setTierNahrung(Tier[] tierNahrung) {
        this.tierNahrung = tierNahrung;
    }

    public Tier[] getTierNahrung() {
        return tierNahrung;
    }
}
```

(c) UML-Klassendiagramm

1/1



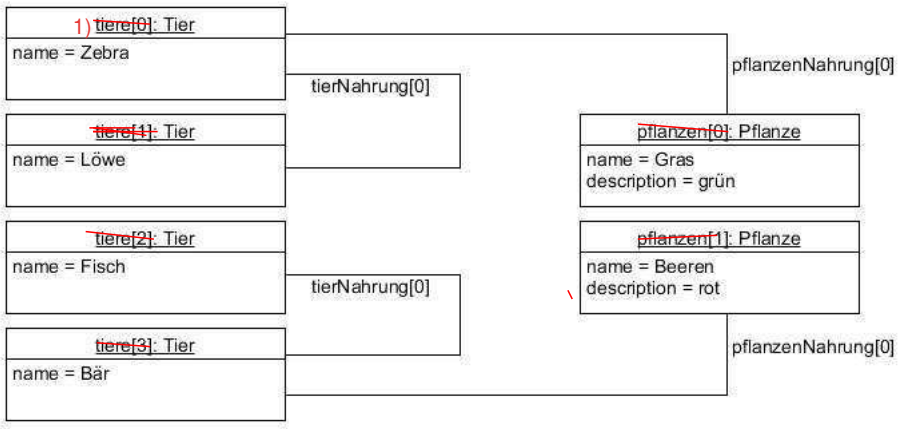
(d) Java-Implementierung des Programms BioTest, welches die obigen Klassen verwendet:
(Der Quellcode befindet sich auch als .java Datei im Anhang)

3/3

```java
public class BioTest {

    public static void main(String[] args) {

        Pflanze[] pflanzen = new Pflanze[2];
        pflanzen[0] = new Pflanze("Gras", "grün");
        pflanzen[1] = new Pflanze("Beeren", "rot");

        Tier[] tiere = new Tier[4];
        tiere[0] = new Tier("Zebra");
        tiere[0].addPflanzenNahrung(pflanzen[0]);
        tiere[1] = new Tier("Löwe");
        tiere[1].addTierNahrung(tiere[0]);
        tiere[2] = new Tier("Fisch");
        tiere[3] = new Tier("Bär");
        tiere[3].addTierNahrung(tiere[2]);
        tiere[3].addPflanzenNahrung(pflanzen[1]);

        for (Tier tier : tiere) {
            if (tier.isPflanzenfresser()) {
                System.out.println(tier.getName() + " ist Pflanzenfresser.");
            } else if (tier.isFleischfresser()) {
                System.out.println(tier.getName() + " ist Fleischfresser.");
            } else if (tier.isAllesfresser()) {
                System.out.println(tier.getName() + " ist Allesfresser.");
            }
        }
    }
}
```

(e) UML-Objektdiagramm am Ende der Ausführung des Programms:

3,5/4



Strings müssen in Anführungszeichen gesetzt werden

1) Vor dem Doppelpunkt steht (falls vorhanden) der Name des Objekts.
(bspw. Pflanze gras = new Pflanze("Gras", "grün"); -> im Objektdiagramm: gras: Pflanze. -0,5

-> hier also jeweils nur :Tier / :Pflanze

19,5/20 gut! :)