1. Aufgabe:

(8 Punkte)

Folgende Klassen sind gegeben:

```
abstract class EitherOr {
    public void call(EitherOr that) {
    if (this != that) {
        this.meth();
    }
        that.meth();
    }
    public abstract void meth();
}

class Either extends EitherOr {
    public void meth() {
        System.out.println("one");
    }
}

class Or extends EitherOr {
    public void meth() {
        System.out.println("two");
    }
}

public class Test {
    public static void main(String[] args) {
        Either x = new Either();
        x.call(new Or());
        x.call(x);
    }
}
```

Schreiben Sie in das Textfeld den Output, den die Ausführung des Programms durch `java Test` erzeugt!

Folgende Programmausschnitte geben Beziehungen zwischen Typen (dargestellt als Klassen und Interfaces) vor. Gehen Sie davon aus, dass die Semantik der Objekte dieser Typen den umgangssprachlichen Bedeutungen der Typnamen entspricht. Wählen Sie "ja" wenn der jeweilige Programmausschnitt das Ersetzbarkeitsprinzip erfüllt und "nein" wenn er es nicht erfüllt.

`interface Bildungseinrichtung { ... }`
`interface Forschungseinrichtung { ... }`
`interface Universitaet extends Bildungseinrichtung,`
` Forschungseinrichtung { ... }`
	○

`interface Haus { ... }`
`class Tuer implements Haus { ... }`
`class Fenster implements Haus { ... }`
	○

`interface Tante { ... }` `class Nichte implements Tante { ... }`	○

`interface Maikaefer { ... }`
`interface Kaefer extends Maikaefer { ... }`
`class Marienkaefer implements Kaefer { ... }`
	○

2. Aufgabe:

(9 Punkte)

Folgende Klasse ist gegeben:

```
public class Cycle {

    private int elem;
    private Cycle next;

    public Cycle(int value, Cycle cycle) {
        elem = value;
        if (cycle == null) {
            next = this;
        } else {
            next = cycle.next;
            cycle.next = this;
        }
    }

    public int prodOdd() {
        // Gibt das Produkt aller ungeraden
        // Zahlen in der Datenstruktur zurück.

        // Diese Methode müssen Sie selbst implementieren.
    }

    public String toString() {
        String res = "" + elem;
        Cycle cycle = next;
        while (cycle != this) {
            res += "," + cycle.elem;
            cycle = cycle.next;
        }
        return "[" + res + "]";
    }

    public static void main(String[] args) {
        Cycle cycle = new Cycle(1, null);
        for (int i = 2; i < 5; i++) {
            cycle = new Cycle(i, cycle);
            System.out.println(cycle);
        }
    }
}
```

Schreiben Sie in das Textfeld die noch fehlende Implementierung der Methode `prodOdd`. Sie soll das Produkt aller ungeraden Zahlen in der Datenstruktur zurückgeben.

Schreiben Sie in das Textfeld den Output, den die Ausführung des Programms durch `java Cycle` erzeugt (drei Zeilen).

3. Aufgabe:

(2 Punkte)

Erklären Sie den Begriff *Data-Hiding* und beschreiben Sie, wie man in Java Data-Hiding erreicht.

4. Aufgabe:

(2 Punkte)

Beschreiben Sie Unterschiede zwischen *lokalen Variablen* und *Objektvariablen* hinsichtlich *Sichtbarkeitsbereich* und *Initialisierung*.

5. Aufgabe:

(2 Punkte)

Die objektorientierte Programmierung eignet sich vor allem für *umfangreiche und langlebige Programme*. Begründen Sie, warum das so ist.