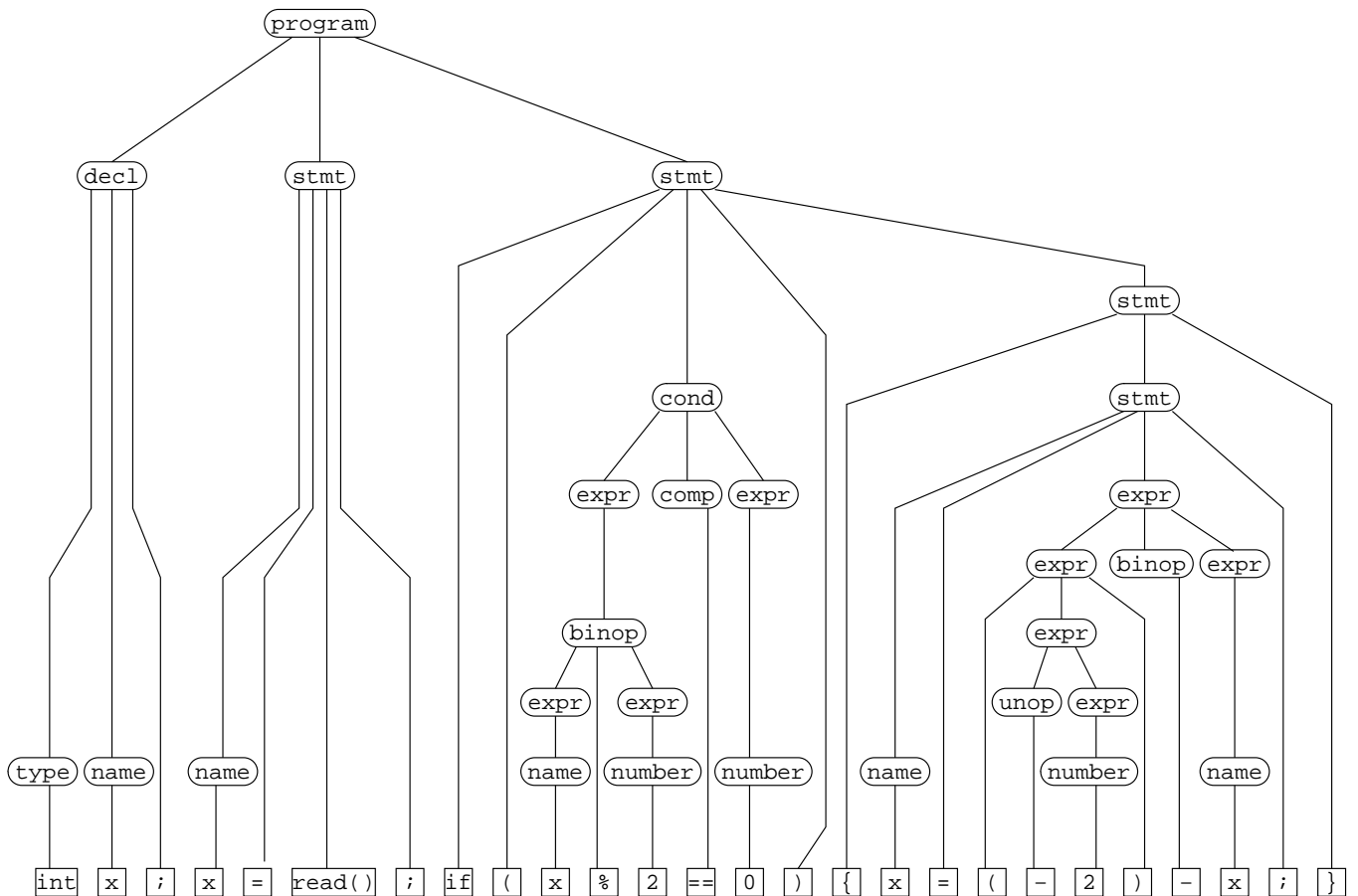
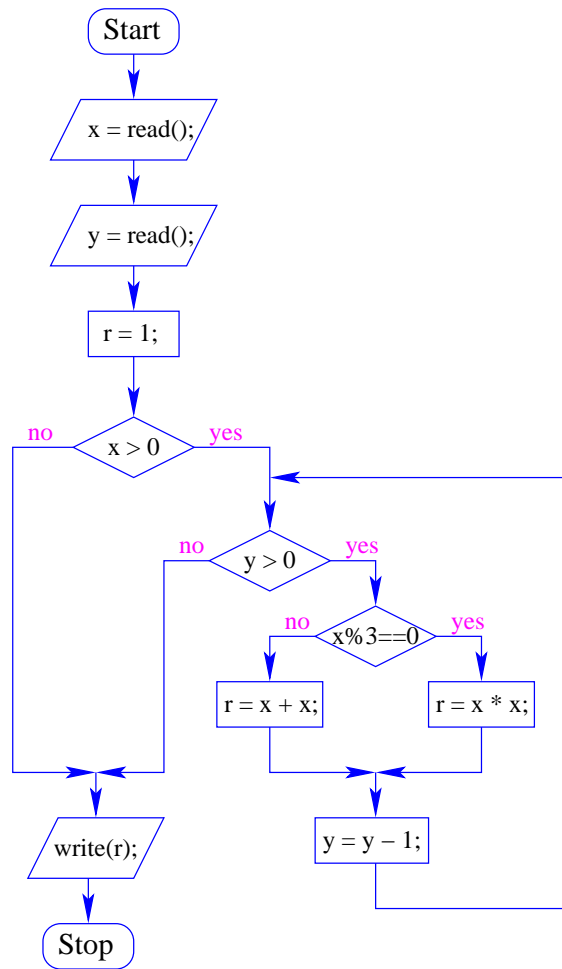


Lösungsvorschläge der Zwischenklausur zu Einführung in die Informatik I

Aufgabe 1 Syntax-Baum (Lösungsvorschlag)



Aufgabe 2 Kontrollfluss-Diagramm (Lösungsvorschlag)



Aufgabe 3 Übersetzung von MiniJava nach MiniJVM (Lösungsvorschlag)

```
a)          ALLOC 3

            READ      // n=read();
            STORE 0

            CONST 2   // i = 2;
            STORE 1

            CONST 1   // d = 1;
            STORE 2

_while:     LOAD 1    // (i<=n)?
            LOAD 0
            LEQ

            FJUMP _end_while

            LOAD 0    // (n%i == 0)?
            LOAD 1
            MOD
            CONST 0
            EQ

            FJUMP _end_if

            LOAD 2    // d = d+1;
            CONST 1
            ADD
            STORE 2

_end_if:    LOAD 1    // i = i+1;
            CONST 1
            ADD
            STORE 1

            JUMP _while

_end_while: LOAD 2    // write(d);
            WRITE
```

b) Das Programm berechnet die Anzahl der Teiler einer positiven natürlichen Zahl.

Aufgabe 4 Felder in Java (Lösungsvorschlag)

a) Bestimmung von Minimum und Maximum

```
public class MinMax{
    public static void printMaxAndMin(int [] array){
        int max = array[0];
        int min = array[0];
        for (int index=1;index<array.length;index++) {
            if (array[index]>max) max = array[index];
            if (array[index]<min) min = array[index];
        }
        System.out.println(max);
        System.out.println(min);
    }
    public static void main(String [] args){
        int [] test = { 3, 5, 6, 2, 7, 12, 400, -100 };
        printMaxAndMin(test);
    }
}
```

b) Umkehrung von Feldern

```
public class Umkehrung{
    public static void reverse(String [] array){
        for (int index=0;index< array.length/2;index++){
            String tmp = array[index];
            array[index]=array[array.length-1-index];
            array[array.length-1-index]=tmp;
        }
    }
    public static void main(String [] args){
        String[] test =
            { "Hamster", "Hund", "Katze", "Goldfisch", "Maus" };
        reverse(test);
        for (int index=0;index< test.length;index++)
            System.out.println(test[index]);
    }
}
```

Aufgabe 5 Mensa-Theke (Lösungsvorschlag)

```
a) public class Person {
    private boolean mitarbeiter;
    private String vorname;
    private String nachname;

    public Person(String vorn, String nachn, boolean istMitarbeiter) {
        mitarbeiter = istMitarbeiter;
        vorname = vorn;
        nachname = nachn;
    }

    public boolean istMitarbeiter() {
        return mitarbeiter;
    }
}
```

```
b) public class MensaTheke {
    private Element wartende;

    public MensaTheke() {
    }

    public void anstellen(Person p) {
        // Studenten am Anfang einhängen
        // Mitarbeiter bei leerer Liste ebenfalls
        if (wartende==null || !p.istMitarbeiter() || wartende.person.istMitarbeiter()) {
            // am Ende der List einhängen
            wartende = new Element(p, wartende);
            return;
        }
        // Mitarbeiter
        // suche wartenden Mitarbeiter oder Listenende
        Element elem = wartende;
        while (elem.next != null && !elem.next.person.istMitarbeiter())
            elem = elem.next;
        elem.next = new Element(p, elem.next);
    }

    public Person bedienen() {
        Person p = null;

        // leere Liste = niemand zu bedienen
        if (wartende == null)
            return null;

        // nur eine wartende Person
        if (wartende.next == null) {
            p = wartende.person;
            wartende = null;
            return p;
        }

        // Ende der List suchen
        Element elem = wartende;
        while (elem.next.next != null)
            elem = elem.next;
        // person entnehmen
        p = elem.next.person;
        // letzten Listeneintrag löschen
        elem.next = null;
        return p;
    }
}
```