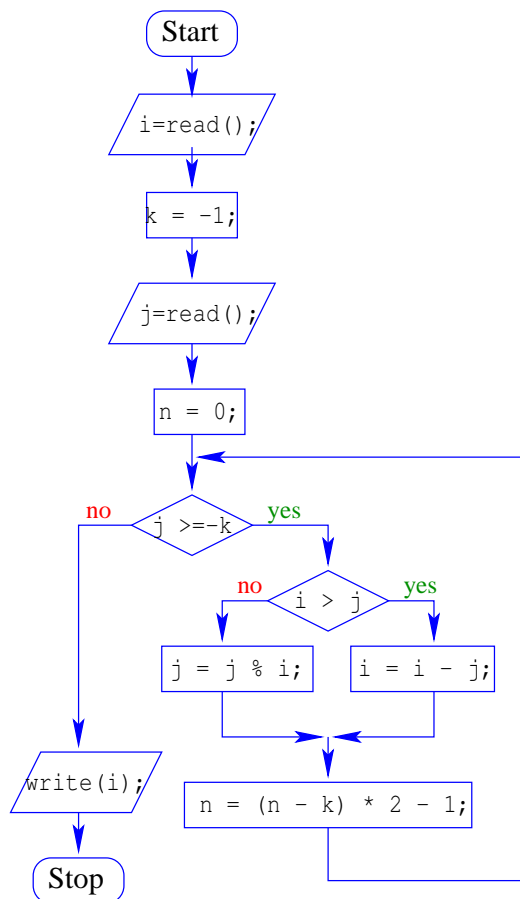


Übungen zu Praktikum Grundlagen der Programmierung

Aufgabe 14 Kontrollfluss-Graph (Lösungsvorschlag)



Aufgabe 15 Minimum und Maximum (Lösungsvorschlag)

```
public class MinMax extends MiniJava {
    public static void main(String[] args) {

        // Anzahl der einzugebenden Zahlen abfragen
        int eingaben = readInt("Wie_viele_Zahlen_möchten_Sie_eingeben?");
        if (eingaben < 1)
            return;

        // Array anlegen
        int[] zahlen = new int[eingaben];

        // Array mit Zahlen füllen
        for (int i = 0; i < eingaben; i++) {
            int zahl = readInt("Bitte_geben_Sie_die_" + (i+1) + "._Zahl_ein.");
            zahlen[i] = zahl;
        }

        // min und max berechnen
        int min = zahlen[0];
        int max = zahlen[0];
        for (int i = 1; i < eingaben; i++) {
            if (zahlen[i] < min)
                min = zahlen[i];
            if (zahlen[i] > max)
                max = zahlen[i];
        }

        // Ausgeben
        write ("Die_kleinste_Zahl_war_" + min);
        write ("Die_größte_Zahl_war_" + max);
    }
}
```

Aufgabe 16 (Ü) Tanzpaarbildung

```

public class Tanzpaare extends MiniJava{

    public static void main(String[] args){
    int c = 10;
    String[] men = new String[c];
    String[] women = new String[c];
    String[][] pair = new String[c][2];

    for(int i = 0; i<women.length; i++){
        women[i] = readString();
    }
    for(int i = 0; i<men.length; i++){
        men[i] = readString();
    }

    for(int j=0;j<women.length;j++){

        int bigman =0;
        int bigwoman =0;
        for(int i= 1; i<men.length; i++){
            if(men[bigman].length() <men[i].length()) bigman = i;
        }
        for(int i= 1; i<women.length; i++){
            if(women[bigwoman].length()<women[i].length()) bigwoman = i;
        }
        pair[j][0]=women[bigwoman];
        pair[j][1]=men[bigman];
        women[bigwoman] = "";
        men[bigman]="";
    }
    write("es_ergibt_sich_folgende_Tanzpaarbildung:");
    for(int i= 0; i<men.length; i++){
        write("frau:_"+pair[i][0]);
        write("mann:_"+pair[i][1]);
    }
    }
}

```

Aufgabe 17 Sieb des Eratosthenes (Lösungsvorschlag)

```

public class Eratosthenes extends MiniJava {

    public static void main(String[] args) {
        // Eingabe
        int n = read();

        if (n < 0) {
            // fehlerhafte Eingabe
            write("Nur_positive_Eingaben_erlaubt.");
        } else {
            // korrekte Eingabe; Berechnung kann erfolgen

            // Sieb anlegen
            boolean prime[] = new boolean[n + 1];
            for (int i = 2; i < prime.length; i++)
                prime[i] = true;

            // Primzahlen berechnen
            for (int i = 2; i < prime.length; i++) {
                if (prime[i]) {
                    // Primzahl ausgeben
                    System.out.println(i);
                    // alle Vielfachen streichen
                    for (int j = i * i; j < prime.length; j = j + i) {
                        prime[j] = false;
                    }
                }
            }
        }
    }
}

```

Mögliche Optimierungen:

- Nur ungerade Zahlen berücksichtigen
- Berechnung bis zur Wurzel der oberen Grenze ist ausreichend