

## VL01, Aufgabe 1 (Übung)

Das „Sieb des Eratosthenes“ ist ein von dem griechischen Philosophen Eratosthenes (276-195 v. Chr.) entwickelter Algorithmus zur Berechnung aller Primzahlen bis zu einer vorgegebenen natürlichen Zahl  $n$ . Der Algorithmus in Umgangssprache (angelehnt an den „Duden der Informatik“):

1. Man schreibe alle Zahlen von 1 bis  $n$  hin und streiche die Zahl 1 durch.
2. Sei  $i$  die kleinste noch nicht durchgestrichene und nicht eingerahmte Zahl. Solange  $i$  existiert und  $i^2 \leq n$  ist, rahme man  $i$  ein und streiche alle Vielfachen von  $i$  durch (die Vielfachen von  $i$  werden „ausgesiebt“).
3. Die eingerahmten und die nicht durchgestrichenen Zahlen sind die Primzahlen von 1 bis  $n$ .

a) Führen Sie den Algorithmus zunächst auf einem Blatt Papier für die Zahlen von 1 bis 25 durch:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
✗	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Im Pseudocode kann der Algorithmus wie folgt formuliert werden:

```
1. einlesen n           // Obergrenze n einlesen
2. for a ← 1 to n do    // Schreibe Zahlen von 1 bis n auf
    hinschreiben(a)
  end for
3. durchstreichen(1)    // Streiche die Zahl 1 durch
4. a ← 2
5. while a*a ≤ n do
    if not istDurchgestrichen(a)
    then                // Streiche alle Vielfachen von a durch
      durchstreichenVielfache(a, n)
    end if
    a ← a+1
  end while
6. for a ← 1 to n do    // Ausgabe der Primzahlen
    if not istDurchgestrichen(a)
    then
      ausgeben a
    end if
  end for
```

- b) Notieren Sie die Methode `durchstreichenVielfache(a, n)` als Pseudocode.
- c) Die abstrakte Datenstruktur „Sieb des Eratosthenes“ besitzt also die Methoden `hinschreiben`, `durchstreichen` und `istDurchgestrichen`. Wie könnten diese Methoden durch eine (aus *Einführung in die Programmierung* bekannte) konkrete Datenstruktur in Java realisiert werden?

## VL01, Aufgabe 2 (Praktikum)

Schreiben Sie die Java-Klasse `SiebDesEratosthenes`, die den obigen Algorithmus realisiert, und alle Primzahlen von 1 bis zu einer vom Benutzer eingegebenen Obergrenze berechnet und ausgibt. Programmieren Sie die Klasse möglichst kompakt und effizient!