



Aufgabe 1

20 Punkte

Primzahlen können wie folgt berechnet werden:

- In einer Folge ganzer Zahlen >1 wird jede einzelne darauf untersucht, ob sie durch (mindestens) eine Zahl, die kleiner ist als sie selbst, ganzzahlig teilbar ist
- Existiert keine solche Zahl, dann handelt es sich um eine Primzahl, sonst nicht

Überlegen Sie sich eine Implementierung *ohne Arrays* für diesen Algorithmus:

- Lesen Sie eine ganze Zahl von der Console ein. Geben Sie dazu eine Eingabeaufforderung aus und lesen Sie eine Zahl mit der Methode `readInt()` ein; Beispiel:

```
int n = readInt();
```

Nach dieser Zuweisung enthält die Variable "n" die eingelesene ganze Zahl.

- Geben Sie alle Primzahlen aus, die kleiner oder gleich dieser eingelesenen Zahl sind.

Hinweis:

- Die Primzahlen sollen nicht gespeichert, sondern nur ausgegeben werden.
- Es spielt keine Rolle, ob die Primzahlen aufsteigend oder absteigend ausgegeben werden.

Aufgabe 2

25 Punkte

Implementieren Sie die als "Sieb des Eratosthenes" bekannte Primzahlberechnung, wie sie normalerweise realisiert wird: mit Hilfe eines Arrays. Konzept des Algorithmus:

Ein Array repräsentiert alle ganzen Zahlen bis zu einer maximalen Zahl n (die wie in Aufgabe 1 von der Console gelesen wird). Nun werden nacheinander alle Vielfachen aller Zahlen, die kleiner als $n/2$ sind, im Array markiert. Diejenigen Zahlen im Array, die danach noch nicht markiert sind, sind Primzahlen.

Orientieren Sie sich mit Ihrer Lösung nicht an Aufgabe 1! Entwickeln Sie stattdessen das Programm neu, indem Sie nach Bedarf diejenigen Sprachkonstrukte nutzen, die Sie bis jetzt kennengelernt haben; überlegen Sie sich eine jeweils geeignete Größe für das benötigte Array.

Hinweis

Sie können den Algorithmus "Sieb des Eratosthenes" auch an anderer Stelle nachlesen.

Hinweis zum gesamten Übungsblatt

Legen Sie ein neues Verzeichnis `uebung04` an.

Legen Sie in diesem Verzeichnis für jede Aufgabe eine eigene Klasse an.