Primzahlsuche: Zahlensieb

Vom griechischen Philosoph und Mathematiker Eratosthenes von Kyrene (3. Jahrhundert v. Chr.) ist ein Verfahren überliefert, Primzahlen bis zu einer beliebigen Grenze schnell zu finden. Das Verfahren ist bekannt als das "Sieb des Erathostenes" oder "Zahlensieb".

Idee

1. Man stellt zunächst eine Liste mit allen Zahlen von 2 bis r gewünschten Obergrenze zusammen.
2. Jetzt streicht man alle Vielfachen von 2, denn das sind ja keine Primzahlen (durch 2 teilbar) und "behält" die 2 als erste Primzahl.
3. Die nächste nicht durchgestrichene Zahl ist die nächste Primzahl - die 3.
4. Jetzt streicht man alle Vielfachen der 3.
5. Jetzt wiederholt man die Schritte ab 3. bis man am Ende des Zahlenbereichs angekommen ist.

Die Zahlen, die dann noch übrig sind, sind die gesuchten Primzahlen.

Modellierung des Problems

Man verwendet ein Array aus booschen-Werten (wahr/falsch). "wahr" soll dabei für Primzahl stehen, "falsch" für "keine Primzahl". Wenn also das Arrayelement `sieb[9]` den Wert `true` hat, würde das bedeuten, dass wir 9 für eine Primzahl halten.

Aufgabe 1

Das Grundgerüst eines BlueJ Projekts kannst du hier herunterladen oder mit dem Befehl `git clone https://gitea.schule.social/QGM-Unterricht/bluej-eratosthenes.git` aus dem Repo klonen.

- Überlege dir zunächst, was die Grenzen für den Index deines Sieb-Arrays sein sollten, damit die den Zahlenbereich von 1 bis zur `grenze` abdecken kannst.
- Vervollständige den Konstruktor: Erzeuge und initialisiere das Sieb-Array.[1)]

Aufgabe 2: Die Gefangenen

Ein König will zu seinem Geburtstag alle 100 Gefangenen, die in Einzelzellen (von 1 bis 100 durchnummeriert) sitzen, freilassen. Dazu schickt er einen Boten, der alle Türen aufschließen soll.

Während der Bote den Befehl ausführt, kommen dem König Bedenken. Er schickt einen zweiten Boten hinterher, der jede zweite Tür (d.h. die Türen 2, 4, 6, usw.) wieder schließt. Bei jedem "Schließvorgang" wird eine Türe entweder aufgeschlossen (wenn zu) oder zugeschlossen (wenn auf).

Danach schickt er einen dritten Boten, der jede dritte Türe (also 3, 6, 9, ...) weiterschließt, usw. bis zum 100. Boten.

Welche Türen sind am Schluss offen? Erstelle ein Programm, das diese Frage beantwortet. Lässt sich das Ergebnis erklären?

```
* Überlege zunächst, wie du die Zellen und ihren Zustand modellieren möchtest.
* Beginne zunächst mit einer Methode, die den ersten Boten repräsentiert, dann einer, die den zweiten Boten repräsentiert.
* Verallgemeinere dann die Methode so, dass sie den i-ten Boten darstellt. Jetzt kannst du das Problem mit einer Schleife lösen.
```

1)

0 und 1 sind keine Primzahlen, die kleinste Primzahl ist 2