Praktische Informatik I | WiSe 2021/22
Tutor:in: Leander Staack | Bearbeiter:in: Julius Walczynki

Übungsblatt 1

Lösungsvorschlag

Aufgabe 1 Primzahlen berechnen

Primzahlen kann man einfach berechnen, indem man eine Folge aller zahlen ab 2 erzeugt,

```
10 static void primes(){
11     IntStream.iterate(2, i -> i+1)
```

nur zahlen behält, die nicht durch kleinere Zahlen teilbar sind

```
12     .filter(i -> IntStream.range(2,i).noneMatch(j -> i % j == 0))
```

und die verbleibenden Zahlen ausgibt

```
14     .forEach(System.out::println);
15 }
```

Test. Nach einem Aufruf von *Primes.primes()* wird jeweils eine Zahl pro Zeile ausgegeben. Die Zahlenfolge 2, 3, 5, 7, 11, 13, 17 usw. sieht korrekt nach Primzahlen aus.

Aufgabe 2 Verbesserungen

Auch wenn das programm aus Aufgabe 1 schnell ist, merkt man, dass es für größere Primzahlen immer langsamer wird. Folgende Verbesserungen sind denkbar:

- Man könnte die 2 separat ausgeben und danach nur noch ungerade Zahlen erzeugen.
- Man muss eigentlich nur testen, ob eine Zahl i durch keine Zahl $j \in \left[2...\lfloor\sqrt{i}\rfloor\right]$ teilbar ist.
- Bei bekannter Obergrenze kann man das Sieb des Eratosthenes [**?**] anwenden (s. Tab. 1).

	2	**3**	4
5	6	**7**	8
9	10	**11**	12
13	14	15	16

Tabelle 1:Sieb des Eratosthenes, Die fett dargestellten Zahlen wurden nicht weggestrichen.

Literatur