



Technische
Universität
Braunschweig



Programmieren 1 – Übung #2

Arne Schmidt



Struktogramme



Live Coding

Von Prosa zu Struktogrammen zu Code

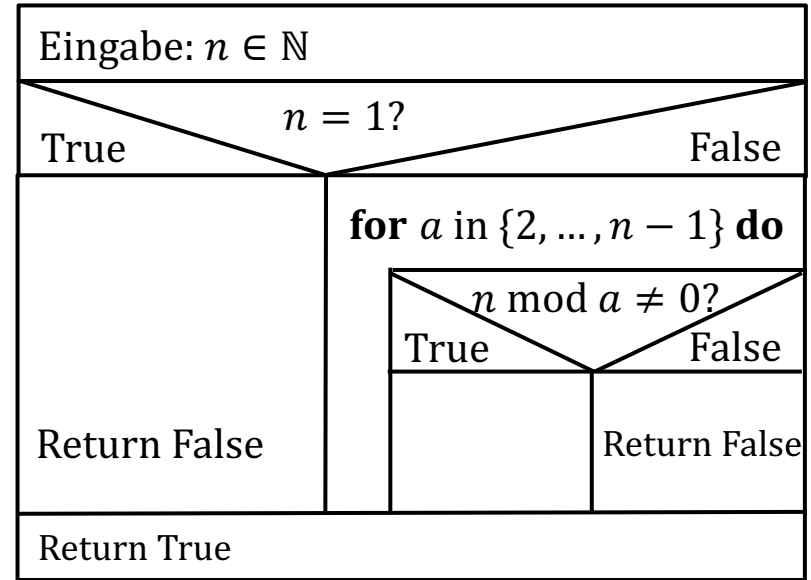
Testen von Primzahlen

Für eine gegebene Zahl n möchten wir testen, ob n eine Primzahl ist. Hat n den Wert 1, können wir *falsch* zurückgeben. Ansonsten prüfen wir jede Zahl zwischen 2 und $n-1$, ob diese n teilen. Falls das für alle Zahlen gilt, gib *true* aus. Ansonsten *false*.

Testen von Primzahlen - Struktogramm

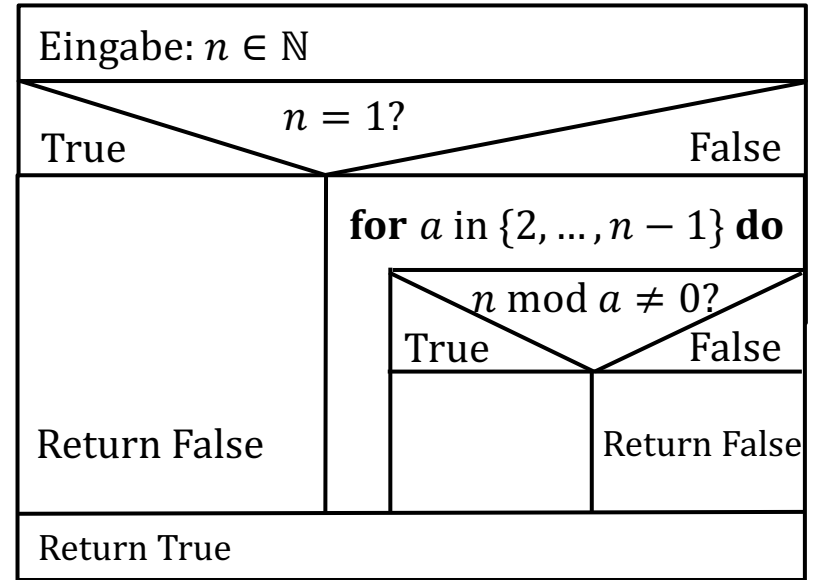
Für eine **gegebene Zahl n** möchten wir testen, **ob n eine Primzahl ist**. **Hat n den Wert 1**, können wir *falsch* zurückgeben. Ansonsten prüfen wir jede Zahl zwischen 2 und $n-1$, **ob diese n teilen**. Falls das für alle Zahlen gilt, gib *true* aus. Ansonsten *false*.

Identifiziere:
Eingabe, **Ausgabe**, **Verzweigungen**,
Schleifen und **Anweisungen**



Testen von Primzahlen - Code

Live Coding!



Sieb des Eratosthenes

Für eine gegebene Zahl n möchten wir alle Primzahlen zwischen 2 und n (inklusive) erhalten. Erstelle ein Array mit den Zahlen 2 bis n . Betrachte jede Zahl x in aufsteigender Reihenfolge. Falls x noch nicht gestrichen, gib x als Primzahl aus und streiche alle seine Vielfachen aus dem Array.

Sieb des Eratosthenes - Struktogramm

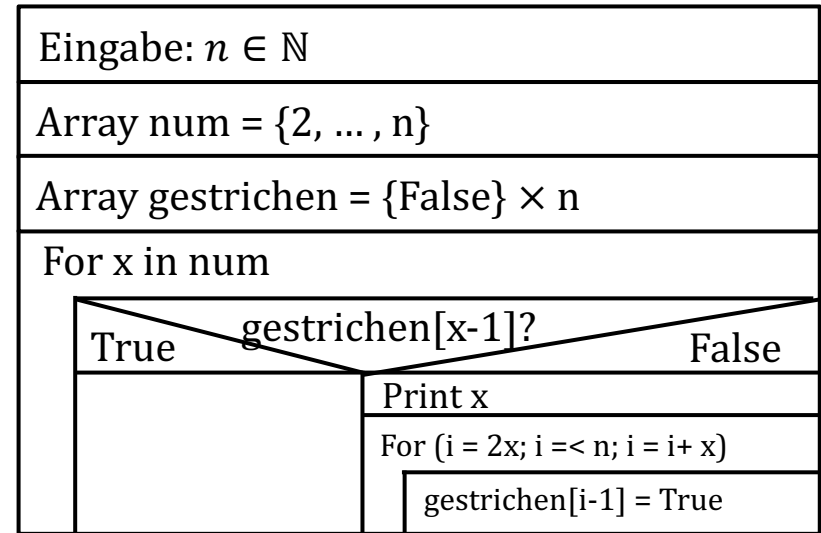
Für eine **gegebene Zahl n** möchten wir **alle Primzahlen zwischen 2 und n** (inklusive) erhalten. **Erstelle ein Array mit den Zahlen 2 bis n** . Betrachte **jede Zahl x in aufsteigender Reihenfolge**. **Falls x noch nicht gestrichen**, gib x als Primzahl aus und **streiche alle seine Vielfachen** aus dem Array.

Überlegungen dazu:

- Kann man Elemente einfach Rausstreichen?
=> zusätzliches "Statusarray"
- Wie iteriert man einfach über Vielfache?

Identifiziere:

Eingabe, **Ausgabe**, **Verzweigungen**, **Schleifen** und **Anweisungen**



Sieb des Eratosthenes - Code

Live Coding!

Identifiziere:

Eingabe, Ausgabe, Verzweigungen,
Schleifen und Anweisungen

