

Objektorientierte Programmierung in Java

Vorlesung 10 - Parallel Computing

Emily Lucia Antosch

HAW Hamburg

07.11.2024

Inhaltsverzeichnis

1. Einleitung	2
2. Parallelverarbeitung	6
3. Klassenbasierte Threads	14
4. License Notice	24

1. Einleitung

- In der letzten Vorlesung haben wir uns mit der Ausgabe und Eingabe beschäftigt
- Sie können nun
 - Ausgabe auf der Konsole in den richtigen Kanal senden und formatieren,
 - Eingabe vom User einfordern
 - und Dateien in Java einlesen.
- Heute geht es weiter mit der **Parallelverarbeitung**.

1.1 Wo sind wir gerade?

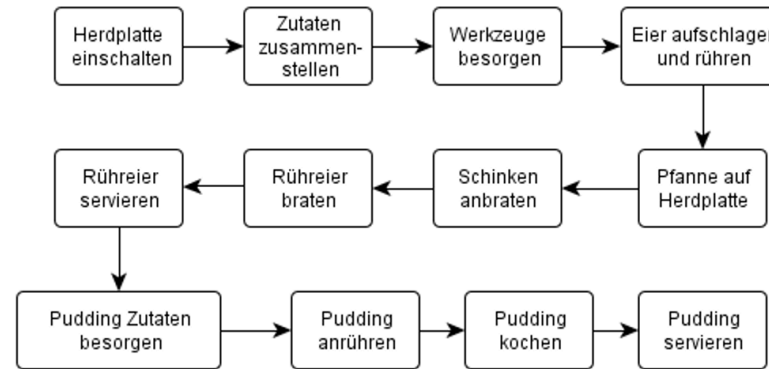
1. Imperative Konzepte
2. Klassen und Objekte
3. Klassenbibliothek
4. Vererbung
5. Schnittstellen
6. Graphische Oberflächen
7. Ausnahmebehandlung
8. Eingaben und Ausgaben
9. **Multithreading (Parallel Computing)**

- Sie führen Programmcode zeitgleich in nebenläufigen Ausführungssträngen (Threads) aus.
- Sie verändern die Zustände aktiver Threads zur Erzeugung der geforderten Funktionalität.
- Sie synchronisieren Threads und Objekte, um fehlerhafte Datenzustände durch nicht korrekte Ausführungsreihenfolgen zu verhindern.

2. Parallelverarbeitung

2.1 Beispiel: Rührei und Pudding

- Sie machen Rührei und Pudding.
- Möglicher Ablauf:



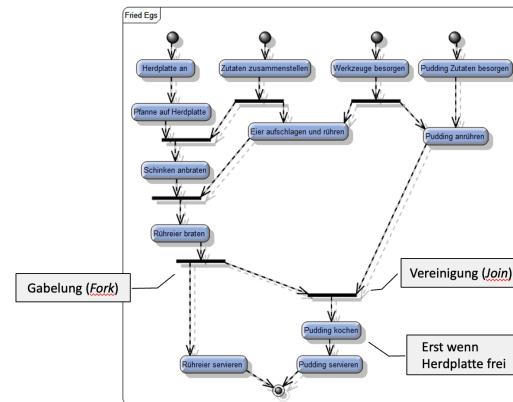
? Frage

- Wie könnte der Ablauf aussehen, wenn Sie zu viert kochen?
- Einschränkung: Es gibt nur eine Herdplatte

2.1 Beispiel: Rührei und Pudding

2. Parallelverarbeitung

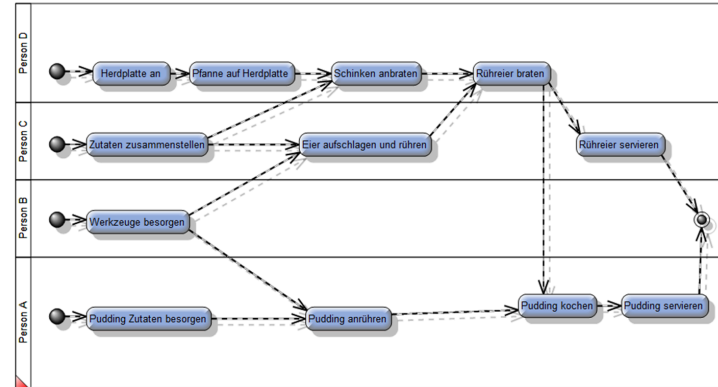
- Mögliche Reihenfolge
- Ressourcenkonflikt: Herdplatte



2.1 Beispiel: Rührei und Pudding

2. Parallelverarbeitung

- Mögliche Aufteilung auf vier Personen



2.1 Beispiel: Rührei und Pudding

2. Parallelverarbeitung

- Aufgabe wird in Teilaufgaben zerlegt, die parallel ausgeführt werden können
- Ergebnisse der Teilaufgaben müssen ausgetauscht werden
- Probleme:
 - Abhängigkeiten: Teilaufgaben benötigen Ergebnisse anderer Teilaufgaben
 - Ressourcenkonflikt: Teilaufgaben benötigen dieselbe Ressource
 - Kommunikations-Overhead: Austausch von Ergebnissen benötigt Ressourcen und Zeit
- Aufgaben können nicht beliebig oder automatisch parallelisiert werden.

- Begriffe:
 - Thread (engl. für „Faden“): Ausführungsstrang innerhalb eines Programmes
 - Multithreading: Mehrere (parallele) Ausführungsstränge innerhalb eines Programmes
- Speicher:
 - Threads teilen sich Speicherbereich des Programmes:
 - Teilen sich daher Variablen und Objekte
 - Können effizient (aber unsicher!) über Variablen und Objekte kommunizieren
- Aber: Jeder Thread hat eigenen Aufruf-Stack der aufgerufenen Methoden

? Frage

- Kleines Rätsel zwischendurch:
- Zumindest einen parallelen Thread haben wir bereits kennengelernt. Welchen?

-

-

? Frage

- Kleines Rätsel zwischendurch:
 - Zumindest einen parallelen Thread haben wir bereits kennengelernt. Welchen?
-
- Antwort:
 - Garbage Collector (Speicher nicht referenzierter Objekte freigeben)
 - Beachte:
 - Java-Programme erzeugen beim Start einen main-Thread
 - Setzen hierbei main() als unterste Methode auf den Aufruf-Stack
 - Bei Bedarf zusätzlich ein Thread für den Garbage Collector gestartet
 - Programm beendet, sobald der letzte zugehörige Thread beendet wurde

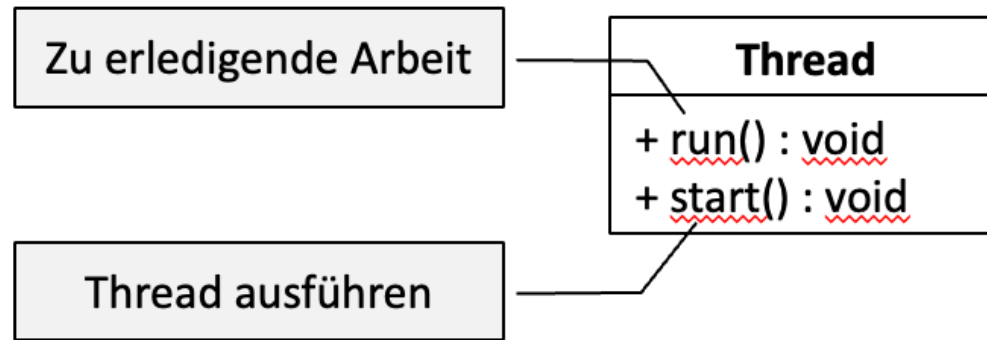
2.1 Beispiel: Rührei und Pudding

2. Parallelverarbeitung

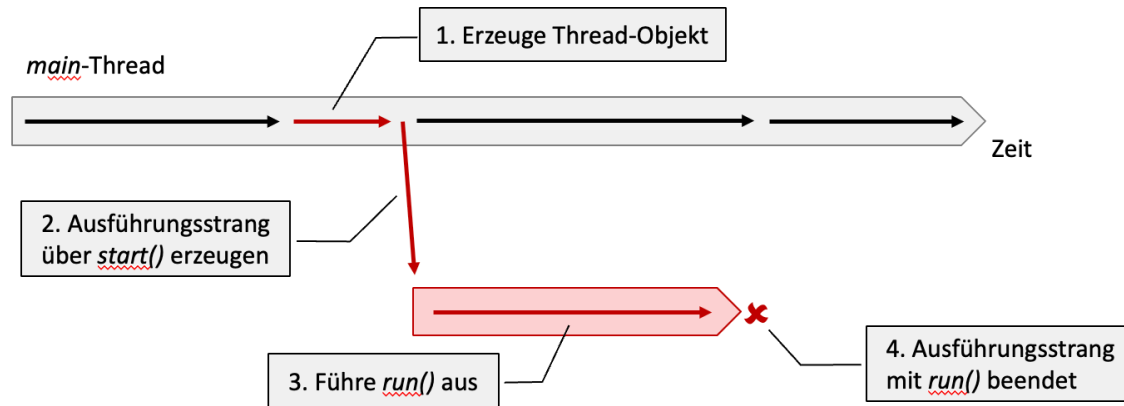
- Teilt Programmen und Threads Rechenzeit (d.h. Prozessoren bzw. Prozessorkerne) zu
- Wartezeiten anderer Threads oder Programme genutzt
- Pseudo-Parallelität:
 - Falls mehr parallele Ausführungsstränge als Prozessoren bzw. Prozessorkerne
 - Scheduler verteilt Rechenzeit scheibchenweise:
 - Ausführung im zeitlichen Wechsel
 - Eindruck, dass Dinge parallel prozessiert werden

3. Klassenbasierte Threads

- Threads werden durch Objekte der Klasse Thread erzeugt:
- Methode `start()` erzeugt und startet parallelen Ausführungsstrang
- Methode `run()` enthält Code, der in Ausführungsstrang ausgeführt werden soll
- Ausführungsstrang wird beendet, sobald `run()` beendet wird



- Veranschaulichung



☰ Aufgabe 1

- Lassen Sie uns dies implementieren:
- Schreiben Sie ein Programm, das einen zusätzlichen Thread erzeugt.

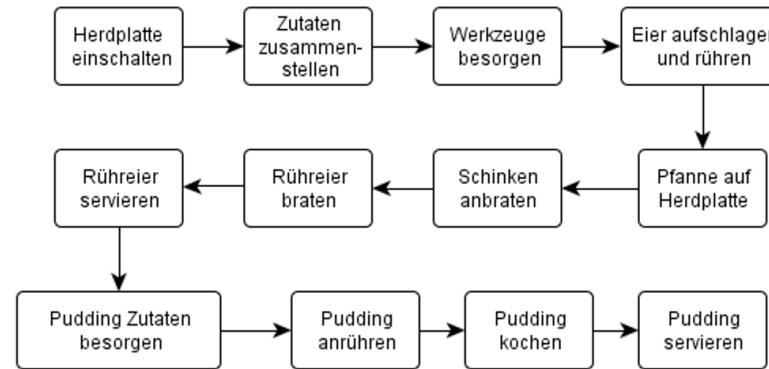
```
1 public class RunThread1 {  
2     public static void main(String[] args) {  
3         Thread thread = new Thread();  
4         System.out.println("Objekt erzeugt");  
5         thread.start();  
6         System.out.println("Thread gestartet");  
7     }  
8 }
```

 Java

? Frage

- Aber man sieht ja gar nichts vom Thread!
 - Die run()-Methode der Klasse Thread ist „leer“.
 - Wie können wir den Thread einen Text auf Konsole ausgeben lassen?

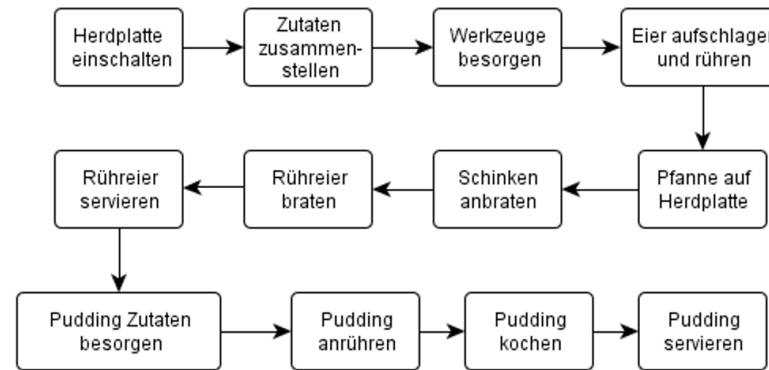
- Sie machen Rührei und Pudding.
- Möglicher Ablauf:



? Frage

- Wie könnte der Ablauf aussehen, wenn Sie zu viert kochen?
- Einschränkung: Es gibt nur eine Herdplatte

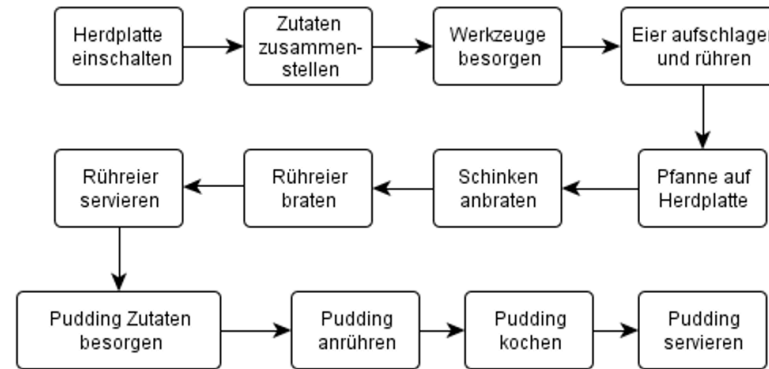
- Sie machen Rührei und Pudding.
- Möglicher Ablauf:



? Frage

- Wie könnte der Ablauf aussehen, wenn Sie zu viert kochen?
- Einschränkung: Es gibt nur eine Herdplatte

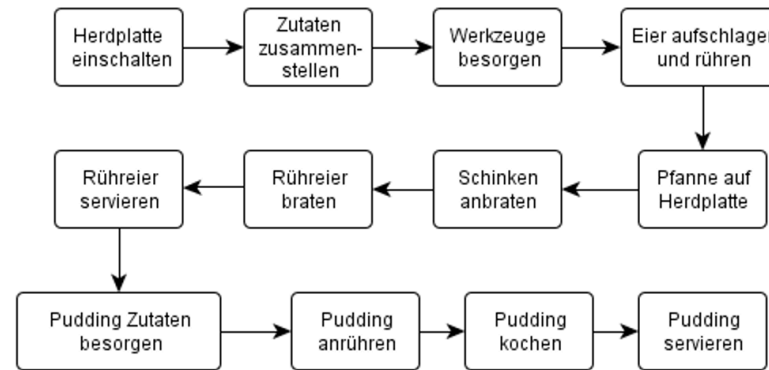
- Sie machen Rührei und Pudding.
- Möglicher Ablauf:



? Frage

- Wie könnte der Ablauf aussehen, wenn Sie zu viert kochen?
- Einschränkung: Es gibt nur eine Herdplatte

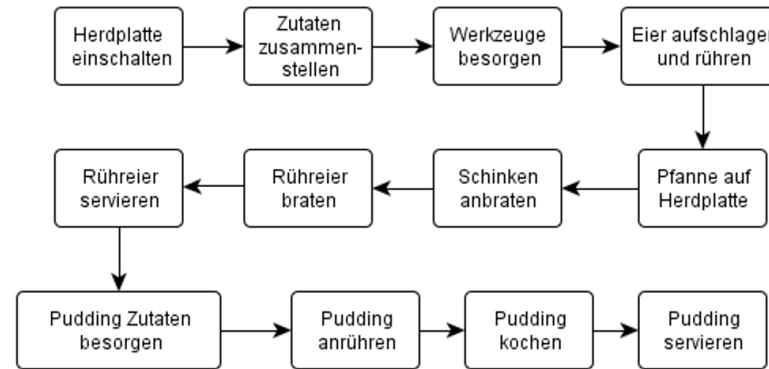
- Sie machen Rührei und Pudding.
- Möglicher Ablauf:



? Frage

- Wie könnte der Ablauf aussehen, wenn Sie zu viert kochen?
- Einschränkung: Es gibt nur eine Herdplatte

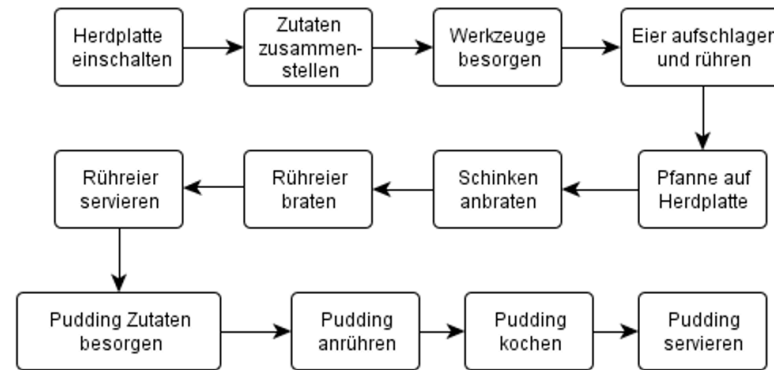
- Sie machen Rührei und Pudding.
- Möglicher Ablauf:



? Frage

- Wie könnte der Ablauf aussehen, wenn Sie zu viert kochen?
- Einschränkung: Es gibt nur eine Herdplatte

- Sie machen Rührei und Pudding.
- Möglicher Ablauf:



? Frage

- Wie könnte der Ablauf aussehen, wenn Sie zu viert kochen?
- Einschränkung: Es gibt nur eine Herdplatte

4. License Notice

4.1 Attribution

- This work is shared under the CC BY-NC-SA 4.0 License and the respective Public License
- link(„<https://creativecommons.org/licenses/by-nc-sa/4.0/>“)
- This work is based off of the work Prof. Dr. Marc Hensel.
- Some of the images and texts, as well as the layout were changed.
- The base material was supplied in private, therefore the link to the source cannot be shared with the audience.