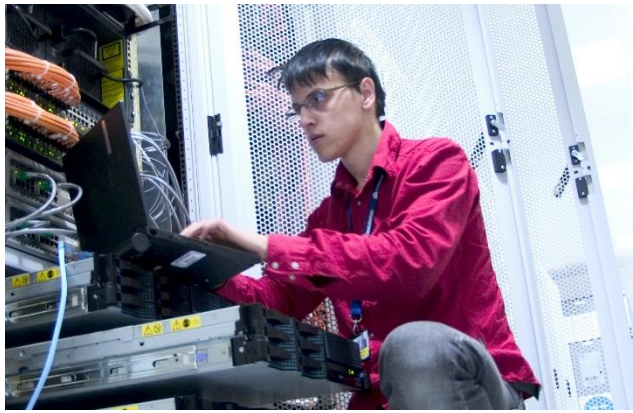
EMIL-VON-BEHRING GYMNASIUM

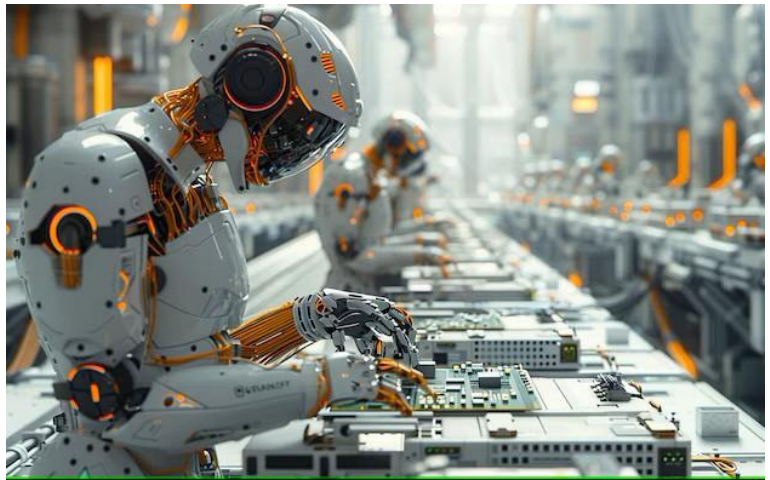
Erzeuger-Verbraucher-Problem



Grafiken © shutterstock.com

Arbeitsaufteilung wie am Fließband

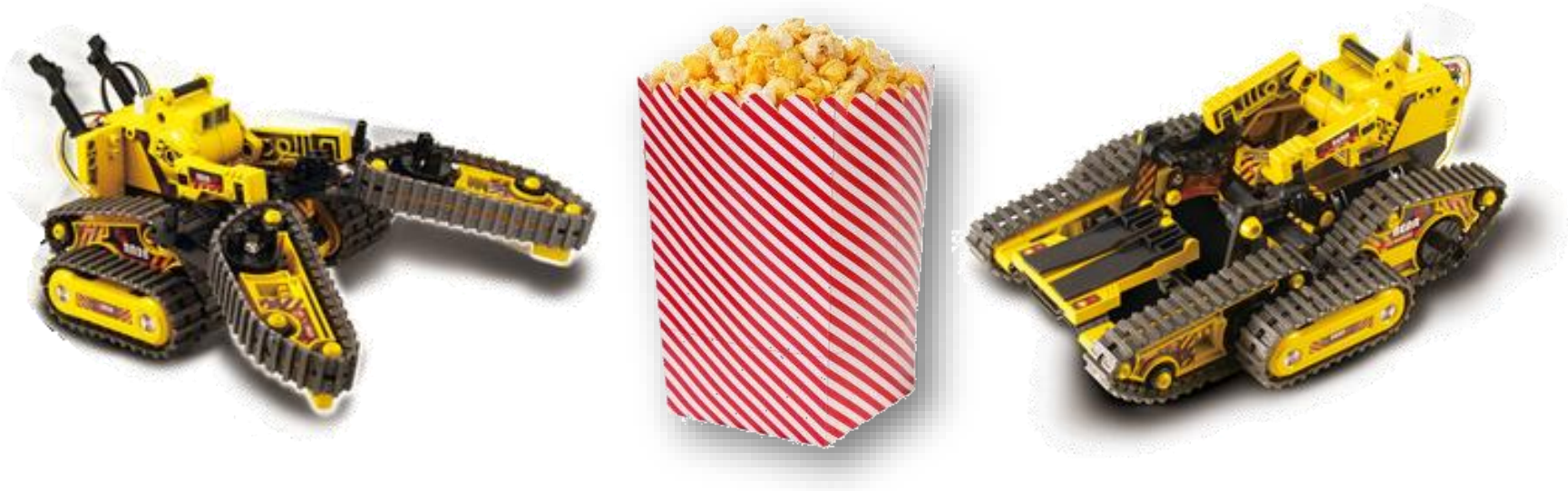
- Bisher haben die vielen Aufgaben auf die Threads verteilt. Man kann aber auch die Threads jeweils eine Teilaufgabe machen lassen
- Thread brauchen daher einen gemeinsamen Puffer/Speicher zum übergeben der Teilergebnisse, damit der nächste Thread weiterarbeiten kann
- Beispiel: Senden/Empfangen von E-Mails
  - Nachrichten werden vom Sender auf Server zwischengespeichert, bis Empfänger sie abholt
- Gekoppelte Prozesse dieser Art bilden ein sog. Erzeuger-Verbraucher-System



Erzeuger-Verbraucher-System

Beispiel:

- Roboter füllt Kisten und stellt diese auf einem Zwischenspeicherplatz ab
- Weiterer Roboter holt diese Kisten und ordnet sie in ein Regallager ein



Aufgabe: BlueJ-Projekt Erzeuger-Verbraucher-System

Beschreibe den Ablauf und identifiziere die Probleme!

Ablauf/Problem	Beispiel

Aufgabe: BlueJ-Projekt Erzeuger-Verbraucher-System

Beschreibe den Ablauf und identifiziere die Probleme!

Ablauf/Problem	Beispiel
Erzeuger schreibt in Speicher	Roboter1 holt Kiste und stellt sie auf Zwischenspeicher/Palette ab

Aufgabe: BlueJ-Projekt Erzeuger-Verbraucher-System

Beschreibe den Ablauf und identifiziere die Probleme!

Ablauf/Problem	Beispiel
Erzeuger schreibt in Speicher	Roboter1 holt Kiste und stellt sie auf Zwischenspeicher/Palette ab
Verbraucher holt die Daten und verarbeitet diese	Roboter2 holt Kiste von der Palette und ordnet sie in Regallager ein

Aufgabe: BlueJ-Projekt Erzeuger-Verbraucher-System

Beschreibe den Ablauf und identifiziere die Probleme!

Ablauf/Problem	Beispiel
Erzeuger schreibt in Speicher	Roboter1 holt Kiste und stellt sie auf Zwischenspeicher/Palette ab
Verbraucher holt die Daten und verarbeitet diese	Roboter2 holt Kiste von der Palette und ordnet sie in Regallager ein
Verarbeitung dauert unterschiedlich lang	Beide Roboter benötigt mehr/weniger Zeit, je nach Entfernung der Regalplätze

Aufgabe: BlueJ-Projekt Erzeuger-Verbraucher-System

Beschreibe den Ablauf und identifiziere die Probleme!

Ablauf/Problem	Beispiel
Erzeuger schreibt in Speicher	Roboter1 holt Kiste und stellt sie auf Zwischenspeicher/Palette ab
Verbraucher holt die Daten und verarbeitet diese	Roboter2 holt Kiste von der Palette und ordnet sie in Regallager ein
Verarbeitung dauert unterschiedlich lang	Beide Roboter benötigt mehr/weniger Zeit, je nach Entfernung der Regalplätze
Erzeuger will Daten schreiben und überschreibt alte Daten	Roboter1 stellt eine weitere Kiste in die erste Kiste (trotz Begrenzung von 1 Kiste)

Aufgabe: BlueJ-Projekt Erzeuger-Verbraucher-System

Beschreibe den Ablauf und identifiziere die Probleme!

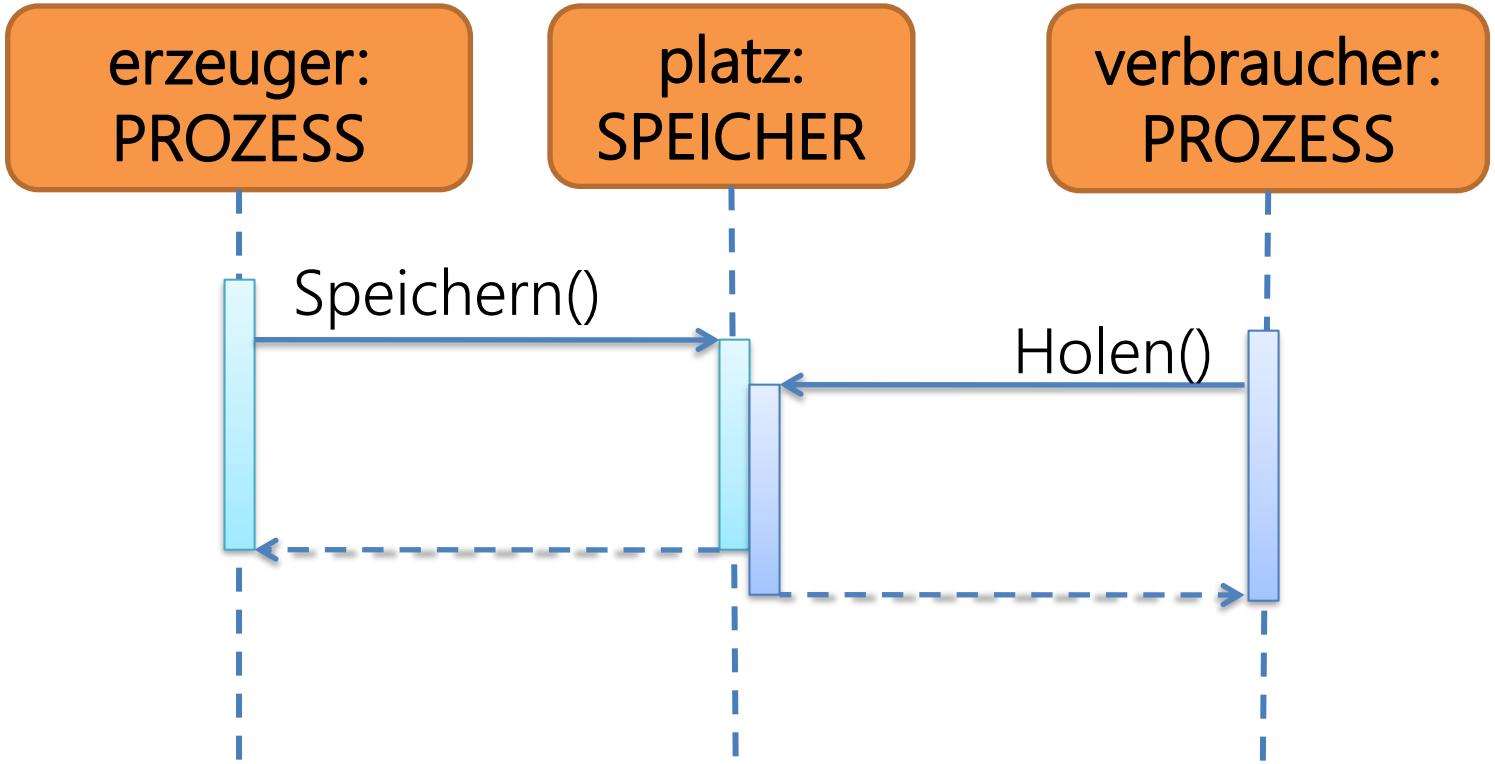
Ablauf/Problem	Beispiel
Erzeuger schreibt in Speicher	Roboter1 holt Kiste und stellt sie auf Zwischenspeicher/Palette ab
Verbraucher holt die Daten und verarbeitet diese	Roboter2 holt Kiste von der Palette und ordnet sie in Regallager ein
Verarbeitung dauert unterschiedlich lang	Beide Roboter benötigt mehr/weniger Zeit, je nach Entfernung der Regalplätze
Erzeuger will Daten schreiben und überschreibt alte Daten	Roboter1 stellt eine weitere Kiste in die erste Kiste (trotz Begrenzung von 1 Kiste)
Verbraucher will Daten lesen, die aber noch nicht vorhanden sind	Roboter2 holt sich eine Kiste, die nicht vorhanden ist → null-Objekt/Kiste

Aufgabe: BlueJ-Projekt Erzeuger-Verbraucher-System

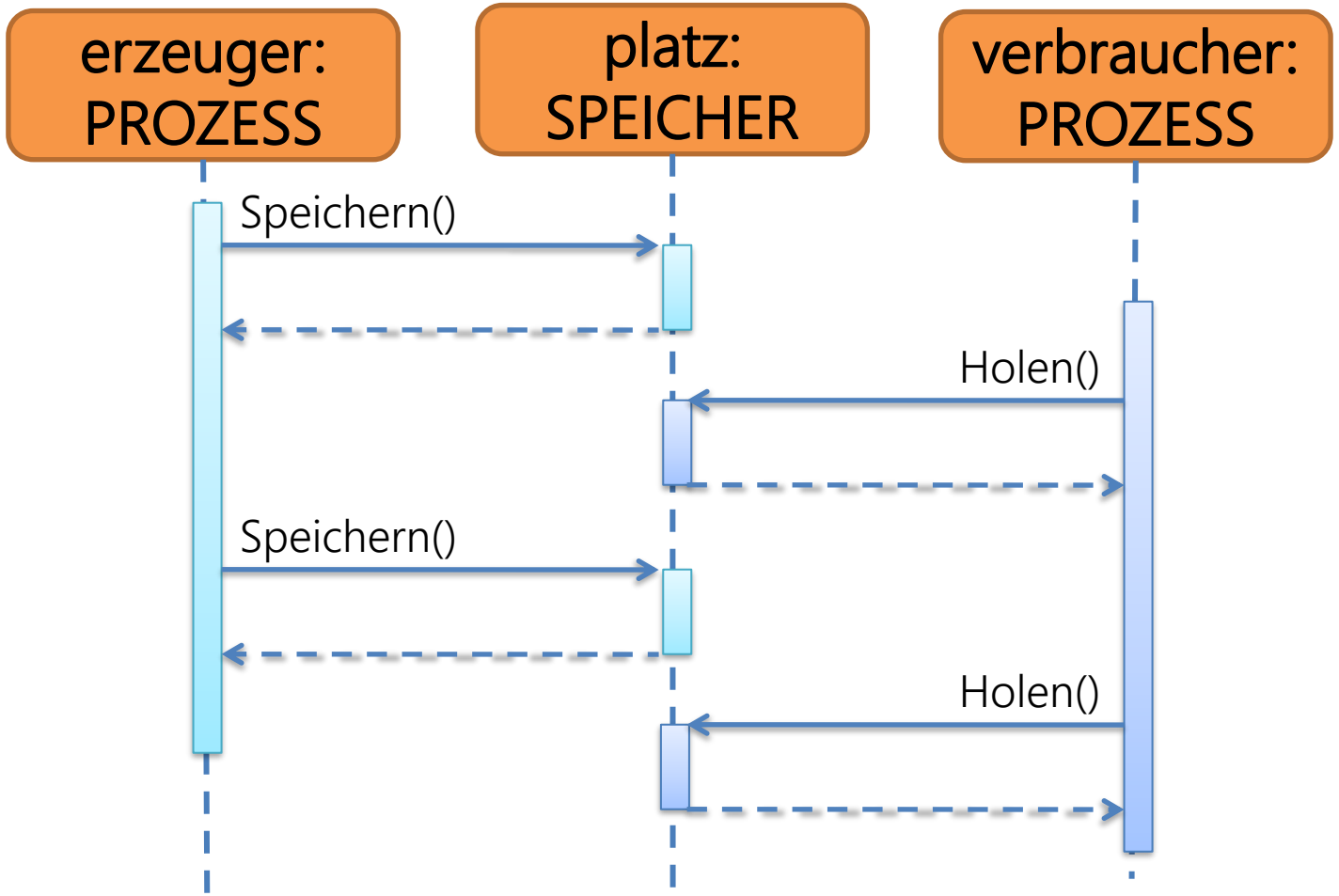
Beschreibe den Ablauf und identifiziere die Probleme!

Ablauf/Problem	Beispiel
Erzeuger schreibt in Speicher	Roboter1 holt Kiste und stellt sie auf Zwischenspeicher/Palette ab
Verbraucher holt die Daten und verarbeitet diese	Roboter2 holt Kiste von der Palette und ordnet sie in Regallager ein
Verarbeitung dauert unterschiedlich lang	Beide Roboter benötigt mehr/weniger Zeit, je nach Entfernung der Regalplätze
Erzeuger will Daten schreiben und überschreibt alte Daten	Roboter1 stellt eine weitere Kiste in die erste Kiste (trotz Begrenzung von 1 Kiste)
Verbraucher will Daten lesen, die aber noch nicht vorhanden sind	Roboter2 holt sich eine Kiste, die nicht vorhanden ist → null-Objekt/Kiste
Wettlaufsituation an der Palette	Beide greifen gleichzeitig auf die Palette zu → Unfall an der Gabel

Problemfall beim E-V-S



Synchronisierung der Prozesse beim E-V-S



- Aufgabe 1/3: Nutze ein geeigneten synchronized-Block, um die Wettlaufsituation zu beheben! → kein Unfall mehr!

Erzeuger-Verbraucher-Problem

Dennoch Konflikte möglich:

- Begrenzter Zwischenspeicher und unterschiedliche Arbeitsgeschwindigkeit von E und V
- **Schneller Erzeuger**
  - Überlaufsituation im Speicher
  - Erzeuger muss Arbeit unterbrechen
- **Schneller Verbraucher**
  - Speicher ständig leer
  - Verbraucher in Wartephase

- Programmiertechnische Umsetzung?

Aktives Warten

Aufgabe 2/3: → palette.istKisteAufPalette()

- Der Erzeuger darf nur synchronisiert abladen(), falls noch keine Kiste auf der Palette ist.
- Der Verbrauchter darf nur synchronisiert aufladen(), falls eine Kiste auf der Palette ist.

Aufgabe 3/3 Aktives Warten: → kiste oder this.kiste

- Solange der Erzeuger noch seine Kiste hat, soll er immer wieder versuchen sie synchronisiert abzuladen.
- Solange der Verbraucher noch keine Kiste hat, soll er immer versuchen eine Kiste synchronisiert aufzuladen.

Probleme beim aktiven Warten?

- Sehr rechenintensiv, weil die Threads immer wieder (vergeblich) versuchen ihre Kiste abzuladen bzw. aufzuladen.
- → Wir brauchen eine geschicktere Lösung!
- Synchronisierung mit wait() and notify() in Java
  - wait() wird auf einem synchronisierten Objekt aufgerufen
  - und versetzt den aufrufenden Thread in den Wartezustand (er gibt die Marke/Token des Objekts frei).
  - Der Thread bleibt blockiert, bis ein anderer Thread notify() oder notifyAll() auf demselben Objekt aufruft, was den ersten Thread wieder „aufweckt".
  - Beide Methoden wait() und notify() müssen innerhalb eines synchronisierten Blocks aufgerufen werden, da der Thread die Marke/Token des Objekts besitzen muss.

Passives Warten

Aufgabe 3/3 Passives Warten: → palette.wait() und palette.notify()

- Falls schon eine Kiste auf der Palette ist, soll der Erzeuger warten. Falls er nicht warten muss bzw. informiert wurde soll er seine Kiste abladen und wiederrum den Verbraucher über die Palette informieren.
- Falls keine Kiste auf der Palette ist, soll der Verbraucher warten. Falls er nicht warten muss bzw. informiert wurde soll er die Kiste aufladen und wiederrum den Verbraucher über die Palette informieren.

Einschub: paralleler Sichtbarkeitsfehler

- Es gibt 3 parallele Fehler: Wettlaufsituation, Verklemmung und Sichtbarkeitsfehler.
  - Der Sichtbarkeitsfehler ist nicht Teil des Schulstoffs!
- Sieh dir das BlueJ-Projekt Erzeuger-Verbraucher-System Sichtbarkeitsfehler an!
  - Was beobachtest du?
- Erklärung: Ein Sichtbarkeitsfehler tritt auf, wenn Änderungen einer gemeinsam genutzten Variable durch einen Thread für andere Threads nicht sofort sichtbar sind. Dies geschieht, weil Threads ihre eigenen Cache-Kopien von Variablen haben, anstatt direkt auf den Arbeitsspeicher zuzugreifen.
- Lösung: mit Schlüsselwort volatile oder ein synchronized-Block
  - Variablenänderungen werden dabei immer in Arbeitsspeicher geschrieben und von den anderen Threads auch dort gelesen!