

Übung 2

Aufgabe 1

c)

Ich habe das Programm nach den Angaben der Aufgabestellung erstellt, allerdings kam es bei mir auch mit notify() nicht zu einem Deadlock.

Wäre das Programm allerdings in ein Deadlock geraten, läge das an der Coffman Kondition:

1. Wechselseitiger Ausschluss (Mutual Exclusion):

Der Zugriff auf den Stack wird durch den synchronized-Block geschützt.

Dadurch kann immer nur ein Thread gleichzeitig auf den Stack zugreifen.

2. Halte-und-Warte-Bedingung (Hold and Wait):

Ein Thread, der eine Ressource hält (den Monitor des Stack-Objekts), wartet darauf, dass eine Bedingung erfüllt wird, ohne den Monitor freizugeben.

3. Keine Präemptierung (No Preemption):

Ein Thread gibt den Monitor nicht freiwillig frei, sondern nur, wenn er entweder:

Die wait()-Methode aufruft (in diesem Fall bleibt der Thread blockiert), oder

Seine Arbeit vollständig erledigt hat.

4. Zirkuläre Warteschlange (Circular Wait):

Es entsteht eine zyklische Abhängigkeit zwischen den Threads:

Der Producer wartet darauf, dass der Consumer den Stack leert, bevor er fortfahren kann.

Gleichzeitig wartet der Consumer darauf, dass der Producer ein Element hinzufügt, bevor er fortfahren kann.

Da die Threads sich gegenseitig blockieren und notify() nur einen zufälligen Thread weckt, kann es passieren, dass immer wieder ein unpassender Thread geweckt wird (z. B. ein weiterer Producer, wenn der Stack voll ist). Dadurch bleibt die Blockade bestehen.

Aufgabe 2

V1: Version wo pro Thread 1 Kolonne berechnet wird.

V2: Version wo pro Thread 50 Kolonnen berechnet werden.

V3: Version wo pro Thread 100 Kolonnen berechnet werden.

Generations	Threads	Time (ms)	ms/gen	$T_{Avg}(X_{10,1,Ver})/T_{Avg}(X_{gen,Threads,Ver})$	Survivors	Version
10	1	50105	5010,5	5010,5	20006327	original
10	1	49345	4934,5	1,0	20026380	
100	4	107562	1075,6	4,6	9455262	V1
100	6	73543	735,4	6,7	9484405	V1
1000	6	720699	720,7	6,8	4355415	V1
10	1	49562	4956,2	1,0	20011557	V2
100	4	106902	1069,0	4,6	9451271	V2
100	6	71469	714,7	6,9	9452072	V2
1000	6	708335	708,3	7,0	4333704	V2
10	1	48605	4860,5	1,0	20022316	V3
100	4	104697	1047,0	4,6	9446630	V3
100	6	73920	739,2	6,6	9463949	V3
1000	6	676814	676,8	7,2	4336874	V3