

TRABALHO PRÁTICO 4

Gerenciamento de acesso concorrente

- O protótipo de banco de dados possui um simulador de acesso concorrente
- Nesse simulador, o controle de acesso concorrente é gerenciado pela classe **ConcurrencyManager**
- Esta classe não implementa nenhum controle para detecção/prevenção de deadlock

Gerenciamento de acesso concorrente

- Para suportar o tratamento de deadlocks, deve-se estender a classe **ConcurrencyManager**
- Por exemplo, a classe **DeadlockPreventionConcurrencyManager** implementa duas estratégias de prevenção de deadlock
 - Wound-Wait
 - Wait-Die

Objetivo do Trabalho

- O objetivo do trabalho é implementar uma estratégia de prevenção de deadlock baseada em **time out**.
- A classe que implementa o estratégia deve estender **ConcurrentyManager**
- A classe deve se chamar **XXXManager**, onde XXX é o nome do aluno

Objetivo do Trabalho

- Em vez de contar o tempo que uma transação espera
 - a estratégia deve contar a quantidade de vezes (**ticks**) que uma transação precisou aguardar na fila
- Caso a transação tenha esperado um número máximo de vezes (**max_ticks**), ela deve ser abortada
- O número máximo de ticks deve ser fornecido no construtor da classe

Objetivo do Trabalho

- O exemplo a seguir mostra os controles que devem ser implementados e o resultado (schedule) que deve ser gerado
 - Considere **max_ticks = 2**

[illegible]

Transações

T1: read(B); write(A); write(C).	T2: read(A); read(B); read(C).	T3: write(C); read(A); write(B).
-------------------------------------------	-----------------------------------------	-------------------------------------------

Tabela de bloqueio

item	lista
A	
B	
C	

Controle de tempo de espera

transação	tick
T1	
T2	
T3	

[illegible]

Transações

T1: read(B); write(A); write(C).	T2: read(A); read(B); read(C).	T3: write(C); read(A); write(B).
-------------------------------------------	-----------------------------------------	-------------------------------------------

Tabela de bloqueio

item	lista
A	
B	R(1)
C	

Controle de tempo de espera

transação	tick
T1	0
T2	
T3	

T1 conseguiu executar
Seu contador é resetado

[illegible]

Transações

T1: read(B); write(A); write(C).	T2: read(A); read(B); read(C).	T3: write(C); read(A); write(B).
-------------------------------------------	-----------------------------------------	-------------------------------------------

Tabela de bloqueio

item	lista
A	R(2)
B	R(1)
C	

Controle de tempo de espera

transação	tick
T1	0
T2	0
T3	

T2 conseguiu executar
Seu contador é resetado

[illegible]

Transações

T1: read(B); write(A); write(C).	T2: read(A); read(B); read(C).	T3: write(C); read(A); write(B).
-------------------------------------------	-----------------------------------------	-------------------------------------------

Tabela de bloqueio

item	lista
A	R(2)
B	R(1)
C	W(3)

Controle de tempo de espera

transação	tick
T1	0
T2	0
T3	0

T3 conseguiu executar
Seu contador é resetado

[illegible]

Transações

T1: read(B); write(A); write(C).	T2: read(A); read(B); read(C).	T3: write(C); read(A); write(B).
--------------------------------------------------	-----------------------------------------	-------------------------------------------

Tabela de bloqueio

item	lista
A	R(2), W(1)
B	R(1)
C	W(3)

Controle de tempo de espera

transação	tick
T1	1
T2	0
T3	0

T1 não conseguiu executar
Seu contador é incrementado

[illegible]

Transações

T1:	T2:	T3:
read(B);	read(A);	write(C);
write(A);	read(B);	read(A);
write(C).	read(C).	write(B).

Tabela de bloqueio

item	lista
A	R(2), W(1)
B	R(1), R(2)
C	W(3)

Controle de tempo de espera

transação	tick
T1	1
T2	0
T3	0

T2 conseguiu executar
Seu contador é resetado

[illegible]

Transações

T1:	T2:	T3:
read(B);	read(A);	write(C);
write(A);	read(B);	read(A);
write(C).	read(C).	write(B).

Tabela de bloqueio

item	lista
A	R(2), W(1), R(3)
B	R(1), R(2)
C	W(3)

Controle de tempo de espera

transação	tick
T1	1
T2	0
T3	1

T3 não conseguiu executar
Seu contador é incrementado

[illegible]

Transações

T1:	T2:	T3:
read(B); write(A); write(C).	read(A); read(B); read(C).	write(C); read(A); write(B).

Tabela de bloqueio

item	lista
A	R(2), W(1), R(3)
B	R(1), R(2)
C	W(3)

Controle de tempo de espera

transação	tick
T1	2
T2	0
T3	1

T1 não conseguiu executar
Seu contador é incrementado

[illegible]

Transações

T1: read(B); write(A); write(C).	T2: read(A); read(B); read(C).	T3: write(C); read(A); write(B).
--------------------------------------------------	-----------------------------------------	-------------------------------------------

Tabela de bloqueio

item	lista
A	R(2), R(3)
B	R(2)
C	W(3)

Controle de tempo de espera

transação	tick
T1	0
T2	0
T3	1

O contador de T1 atingiu o seu limite.
T1 aborta e o contador é resetado

[illegible]

Transações

T1:	T2:	T3:
read(B);	read(A);	write(C);
write(A);	read(B);	read(A);
write(C).	read(C).	write(B).

Tabela de bloqueio

item	lista
A	R(2), R(3)
B	R(2)
C	W(3), R(2)

Controle de tempo de espera

transação	tick
T1	0
T2	1
T3	1

T2 não conseguiu executar
Seu contador é incrementado

[illegible]

Transações

T1:	T2:	T3:
read(B);	read(A);	write(C);
write(A);	read(B);	read(A);
write(C).	read(C).	write(B).

Tabela de bloqueio

item	lista
A	R(2), R(3)
B	R(2)
C	W(3), R(2)

Controle de tempo de espera

transação	tick
T1	0
T2	1
T3	0

T3 conseguiu executar
Seu contador é resetado

[illegible]

Transações

T1: read(B); write(A); write(C).	T2: read(A); read(B); read(C).	T3: write(C); read(A); write(B).
-------------------------------------------	-----------------------------------------	-------------------------------------------

Tabela de bloqueio

item	lista
A	R(2), R(3)
B	R(2), R(1)
C	W(3), R(2)

Controle de tempo de espera

transação	tick
T1	0
T2	1
T3	0

T1 conseguiu executar
Seu contador é resetado

[illegible]

Transações

T1:	T2:	T3:
read(B);	read(A);	write(C);
write(A);	read(B);	read(A);
write(C).	read(C).	write(B).

Tabela de bloqueio

item	lista
A	R(2), R(3)
B	R(2), R(1)
C	W(3), R(2)

Controle de tempo de espera

transação	tick
T1	0
T2	2
T3	0

T2 não conseguiu executar
Seu contador é incrementado

[illegible]

Transações

T1: read(B); write(A); write(C).	T2: read(A); read(B); read(C).	T3: write(C); read(A); write(B).
-------------------------------------------	------------------------------------------------	-------------------------------------------

Tabela de bloqueio

item	lista
A	R(3)
B	R(1)
C	W(3)

Controle de tempo de espera

transação	tick
T1	0
T2	0
T3	0

O contador de T2 atingiu o seu limite.
T2 aborta e o contador é resetado

[illegible]

Transações

T1:	T2:	T3:
read(B);	read(A);	write(C);
write(A);	read(B);	read(A);
write(C).	read(C).	write(B).

Tabela de bloqueio

item	lista
A	R(3)
B	R(1), W(3)
C	W(3)

Controle de tempo de espera

transação	tick
T1	0
T2	0
T3	1

T3 não conseguiu executar
Seu contador é incrementado

[illegible]

Transações

T1: read(B); write(A); write(C).	T2: read(A); read(B); read(C).	T3: write(C); read(A); write(B).
--------------------------------------------------	-----------------------------------------	-------------------------------------------

Tabela de bloqueio

item	lista
A	R(3), W(1)
B	R(1), W(3)
C	W(3)

Controle de tempo de espera

transação	tick
T1	1
T2	0
T3	1

T1 não conseguiu executar
Seu contador é incrementado

[illegible]

Transações

T1: read(B); write(A); write(C).	T2: read(A); read(B); read(C).	T3: write(C); read(A); write(B).
-------------------------------------------	-----------------------------------------	-------------------------------------------

Tabela de bloqueio

item	lista
A	R(3), W(1), R(2)
B	R(1), W(3)
C	W(3)

Controle de tempo de espera

transação	tick
T1	1
T2	1
T3	1

T2 não conseguiu executar
Seu contador é incrementado

[illegible]

Transações

T1: read(B); write(A); write(C).	T2: read(A); read(B); read(C).	T3: write(C); read(A); write(B).
-------------------------------------------	-----------------------------------------	-------------------------------------------

Tabela de bloqueio

item	lista
A	R(3), W(1), R(2)
B	R(1), W(3)
C	W(3)

Controle de tempo de espera

transação	tick
T1	1
T2	1
T3	2

T3 não conseguiu executar
Seu contador é incrementado

[illegible]

Transações

T1: read(B); write(A); write(C).	T2: read(A); read(B); read(C).	T3: write(C); read(A); write(B).
-------------------------------------------	-----------------------------------------	-------------------------------------------

Tabela de bloqueio

item	lista
A	W(1), R(2)
B	R(1)
C	

Controle de tempo de espera

transação	tick
T1	1
T2	1
T3	0

O contador de T3 atingiu o seu limite.
T3 aborta e o contador é resetado

T1	T2	T3
R(B)		
	R(A)	
		W(C)
	R(B)	
abort		
		R(A)
R(B)		
	abort	
		abort
W(A)		

Transações

T1: read(B); write(A); write(C).	T2: read(A); read(B); read(C).	T3: write(C); read(A); write(B).
-------------------------------------------	-----------------------------------------	-------------------------------------------

Tabela de bloqueio

item	lista
A	W(1), R(2)
B	R(1)
C	

Controle de tempo de espera

transação	tick
T1	0
T2	1
T3	0

T1 conseguiu executar
Seu contador é resetado

T1	T2	T3
R(B)		
	R(A)	
		W(C)
	R(B)	
abort		
		R(A)
R(B)		
	abort	
		abort
W(A)		

Transações

T1:	T2:	T3:
read(B);	read(A);	write(C);
write(A);	read(B);	read(A);
write(C).	read(C).	write(B).

Tabela de bloqueio

item	lista
A	W(1), R(2)
B	R(1)
C	

Controle de tempo de espera

transação	tick
T1	0
T2	2
T3	0

T2 não conseguiu executar
Seu contador é incrementado

T1	T2	T3
R(B)		
	R(A)	
		W(C)
	R(B)	
abort		
		R(A)
R(B)		
	abort	
		abort
W(A)		
	abort	

Transações

T1:	T2:	T3:
read(B);	read(A);	write(C);
write(A);	read(B);	read(A);
write(C).	read(C).	write(B).

Tabela de bloqueio

item	lista
A	W(1)
B	R(1)
C	

Controle de tempo de espera

transação	tick
T1	0
T2	0
T3	0

O contador de T2 atingiu o seu limite.
T2 aborta e o contador é resetado

T1	T2	T3
R(B)		
	R(A)	
		W(C)
	R(B)	
abort		
		R(A)
R(B)		
	abort	
		abort
W(A)		
	abort	
		W(C)

Transações

T1:	T2:	T3:
read(B);	read(A);	write(C);
write(A);	read(B);	read(A);
write(C).	read(C).	write(B).

Tabela de bloqueio

item	lista
A	W(1)
B	R(1)
C	W(3)

Controle de tempo de espera

transação	tick
T1	0
T2	0
T3	0

T3 conseguiu executar
Seu contador é resetado

T1	T2	T3
R(B)		
	R(A)	
		W(C)
	R(B)	
abort		
		R(A)
R(B)		
	abort	
		abort
W(A)		
	abort	
		W(C)

Transações

T1: read(B); write(A); write(C).	T2: read(A); read(B); read(C).	T3: write(C); read(A); write(B).
-------------------------------------------	-----------------------------------------	-------------------------------------------

Tabela de bloqueio

item	lista
A	W(1)
B	R(1)
C	W(3), W(1)

Controle de tempo de espera

transação	tick
T1	1
T2	0
T3	0

T1 não conseguiu executar
Seu contador é incrementado

T1	T2	T3
R(B)		
	R(A)	
		W(C)
	R(B)	
abort		
		R(A)
R(B)		
	abort	
		abort
W(A)		
	abort	
		W(C)

Transações

T1:	T2:	T3:
read(B);	read(A);	write(C);
write(A);	read(B);	read(A);
write(C).	read(C).	write(B).

Tabela de bloqueio

item	lista
A	W(1), R(2)
B	R(1)
C	W(3), W(1)

Controle de tempo de espera

transação	tick
T1	1
T2	1
T3	0

T2 não conseguiu executar
Seu contador é incrementado

T1	T2	T3
R(B)		
	R(A)	
		W(C)
	R(B)	
abort		
		R(A)
R(B)		
	abort	
		abort
W(A)		
	abort	
		W(C)

Transações

T1: read(B); write(A); write(C).	T2: read(A); read(B); read(C).	T3: write(C); read(A); write(B).
-------------------------------------------	-----------------------------------------	-------------------------------------------

Tabela de bloqueio

item	lista
A	W(1), R(2), R(3)
B	R(1)
C	W(3), W(1)

Controle de tempo de espera

transação	tick
T1	1
T2	1
T3	1

T3 não conseguiu executar
Seu contador é incrementado

T1	T2	T3
R(B)		
	R(A)	
		W(C)
	R(B)	
abort		
		R(A)
R(B)		
	abort	
		abort
W(A)		
	abort	
		W(C)

Transações

T1: read(B); write(A); write(C).	T2: read(A); read(B); read(C).	T3: write(C); read(A); write(B).
-------------------------------------------	-----------------------------------------	-------------------------------------------

Tabela de bloqueio

item	lista
A	W(1), R(2), R(3)
B	R(1)
C	W(3), W(1)

Controle de tempo de espera

transação	tick
T1	2
T2	1
T3	1

T1 não conseguiu executar
Seu contador é incrementado

T1	T2	T3
R(B)		
	R(A)	
		W(C)
	R(B)	
abort		
		R(A)
R(B)		
	abort	
		abort
W(A)		
	abort	
		W(C)
abort		

Transações

T1: read(B); write(A); write(C).	T2: read(A); read(B); read(C).	T3: write(C); read(A); write(B).
-------------------------------------------	-----------------------------------------	-------------------------------------------

Tabela de bloqueio

item	lista
A	R(2), R(3)
B	
C	W(3)

Controle de tempo de espera

transação	tick
T1	0
T2	1
T3	1

O contador de T1 atingiu o seu limite.
T1 aborta e o contador é resetado

T1	T2	T3
R(B)		
	R(A)	
		W(C)
	R(B)	
abort		
		R(A)
R(B)		
	abort	
		abort
W(A)		
	abort	
		W(C)
abort		
	R(A)	

Transações

T1:	T2:	T3:
read(B);	read(A);	write(C);
write(A);	read(B);	read(A);
write(C).	read(C).	write(B).

Tabela de bloqueio

item	lista
A	R(2), R(3)
B	
C	W(3)

Controle de tempo de espera

transação	tick
T1	0
T2	0
T3	1

T2 conseguiu executar
Seu contador é resetado

T1	T2	T3
R(B)		
	R(A)	
		W(C)
	R(B)	
abort		
		R(A)
R(B)		
	abort	
		abort
W(A)		
	abort	
		W(C)
abort		
	R(A)	
		R(A)

Transações

T1:	T2:	T3:
read(B);	read(A);	write(C);
write(A);	read(B);	read(A);
write(C).	read(C).	write(B).

Tabela de bloqueio

item	lista
A	R(2), R(3)
B	
C	W(3)

Controle de tempo de espera

transação	tick
T1	0
T2	0
T3	0

T3 conseguiu executar
Seu contador é resetado

T1	T2	T3
R(B)		
	R(A)	
		W(C)
	R(B)	
abort		
		R(A)
R(B)		
	abort	
		abort
W(A)		
	abort	
		W(C)
abort		
	R(A)	
		R(A)
R(B)		

Transações

T1: read(B); write(A); write(C).	T2: read(A); read(B); read(C).	T3: write(C); read(A); write(B).
-------------------------------------------	-----------------------------------------	-------------------------------------------

Tabela de bloqueio

item	lista
A	R(2), R(3)
B	R(1)
C	W(3)

Controle de tempo de espera

transação	tick
T1	0
T2	0
T3	0

T1 conseguiu executar
Seu contador é resetado

T1	T2	T3
R(B)		
	R(A)	
		W(C)
	R(B)	
abort		
		R(A)
R(B)		
	abort	
		abort
W(A)		
	abort	
		W(C)
abort		
	R(A)	
		R(A)
R(B)		
	R(B)	

Transações

T1:	T2:	T3:
read(B);	read(A);	write(C);
write(A);	read(B);	read(A);
write(C).	read(C).	write(B).

Tabela de bloqueio

item	lista
A	R(2), R(3)
B	R(1), R(2)
C	W(3)

Controle de tempo de espera

transação	tick
T1	0
T2	0
T3	0

T2 conseguiu executar
Seu contador é resetado

T1	T2	T3
R(B)		
	R(A)	
		W(C)
	R(B)	
abort		
		R(A)
R(B)		
	abort	
		abort
W(A)		
	abort	
		W(C)
abort		
	R(A)	
		R(A)
R(B)		
	R(B)	

Transações

T1:	T2:	T3:
read(B);	read(A);	write(C);
write(A);	read(B);	read(A);
write(C).	read(C).	write(B).

Tabela de bloqueio

item	lista
A	R(2), R(3)
B	R(1), R(2), W(3)
C	W(3)

Controle de tempo de espera

transação	tick
T1	0
T2	0
T3	1

T3 não conseguiu executar
Seu contador é incrementado

T1	T2	T3
R(B)		
	R(A)	
		W(C)
	R(B)	
abort		
		R(A)
R(B)		
	abort	
		abort
W(A)		
	abort	
		W(C)
abort		
	R(A)	
		R(A)
R(B)		
	R(B)	

Transações

T1:	T2:	T3:
read(B);	read(A);	write(C);
write(A);	read(B);	read(A);
write(C).	read(C).	write(B).

Tabela de bloqueio

item	lista
A	R(2), R(3), W(1)
B	R(1), R(2), W(3)
C	W(3)

Controle de tempo de espera

transação	tick
T1	1
T2	0
T3	1

T1 não conseguiu executar
Seu contador é incrementado

T1	T2	T3
R(B)		
	R(A)	
		W(C)
	R(B)	
abort		
		R(A)
R(B)		
	abort	
		abort
W(A)		
	abort	
		W(C)
abort		
	R(A)	
		R(A)
R(B)		
	R(B)	

Transações

T1:	T2:	T3:
read(B);	read(A);	write(C);
write(A);	read(B);	read(A);
write(C).	read(C).	write(B).

Tabela de bloqueio

item	lista
A	R(2), R(3), W(1)
B	R(1), R(2), W(3)
C	W(3), R(2)

Controle de tempo de espera

transação	tick
T1	1
T2	1
T3	1

T2 não conseguiu executar
Seu contador é incrementado

T1	T2	T3
R(B)		
	R(A)	
		W(C)
	R(B)	
abort		
		R(A)
R(B)		
	abort	
		abort
W(A)		
	abort	
		W(C)
abort		
	R(A)	
		R(A)
R(B)		
	R(B)	

Transações

T1:	T2:	T3:
read(B);	read(A);	write(C);
write(A);	read(B);	read(A);
write(C).	read(C).	write(B).

Tabela de bloqueio

item	lista
A	R(2), R(3), W(1)
B	R(1), R(2), W(3)
C	W(3), R(2)

Controle de tempo de espera

transação	tick
T1	1
T2	1
T3	2

T3 não conseguiu executar
Seu contador é incrementado

T1	T2	T3
R(B)		
	R(A)	
		W(C)
	R(B)	
abort		
		R(A)
R(B)		
	abort	
		abort
W(A)		
	abort	
		W(C)
abort		
	R(A)	
		R(A)
R(B)		
	R(B)	
		abort

Transações

T1: read(B); write(A); write(C).	T2: read(A); read(B); read(C).	T3: write(C); read(A); write(B).
-------------------------------------------	-----------------------------------------	-------------------------------------------

Tabela de bloqueio

item	lista
A	R(2), W(1)
B	R(1), R(2)
C	R(2)

Controle de tempo de espera

transação	tick
T1	1
T2	1
T3	0

O contador de T3 atingiu o seu limite.
T3 aborta e o contador é resetado

[illegible]

Transações

T1: read(B); write(A); write(C).	T2: read(A); read(B); read(C).	T3: write(C); read(A); write(B).
-------------------------------------------	-----------------------------------------	-------------------------------------------

Tabela de bloqueio

item	lista
A	R(2), W(1)
B	R(1), R(2)
C	R(2)

Controle de tempo de espera

transação	tick
T1	1
T2	1
T3	0

Continuação do exercício

[illegible]

Transações

T1: read(B); write(A); write(C).	T2: read(A); read(B); read(C).	T3: write(C); read(A); write(B).
--------------------------------------------------	-----------------------------------------	-------------------------------------------

Tabela de bloqueio

item	lista
A	R(2), W(1)
B	R(1), R(2)
C	R(2)

Controle de tempo de espera

transação	tick
T1	2
T2	1
T3	0

T1 não conseguiu executar
Seu contador é incrementado

[illegible]

Transações

T1: read(B); write(A); write(C).	T2: read(A); read(B); read(C).	T3: write(C); read(A); write(B).
--------------------------------------------------	-----------------------------------------	-------------------------------------------

Tabela de bloqueio

item	lista
A	R(2)
B	R(2)
C	R(2)

Controle de tempo de espera

transação	tick
T1	0
T2	1
T3	0

O contador de T1 atingiu o seu limite.
T1 aborta e o contador é resetado

[illegible]

Transações

T1:	T2:	T3:
read(B);	read(A);	write(C);
write(A);	read(B);	read(A);
write(C).	read(C).	write(B).

Tabela de bloqueio

item	lista
A	R(2)
B	R(2)
C	R(2)

Controle de tempo de espera

transação	tick
T1	0
T2	0
T3	0

T2 conseguiu executar
Seu contador é resetado

[illegible]

Transações

T1: read(B); write(A); write(C).	T2: read(A); read(B); read(C).	T3: write(C); read(A); write(B).
-------------------------------------------	-----------------------------------------	-------------------------------------------

Tabela de bloqueio

item	lista
A	R(2)
B	R(2)
C	R(2), W(3)

Controle de tempo de espera

transação	tick
T1	0
T2	0
T3	1

T3 não conseguiu executar
Seu contador é incrementado

[illegible]

Transações

T1: read(B); write(A); write(C).	T2: read(A); read(B); read(C).	T3: write(C); read(A); write(B).
-------------------------------------------	-----------------------------------------	-------------------------------------------

Tabela de bloqueio

item	lista
A	R(2)
B	R(2), R(1)
C	R(2), W(3)

Controle de tempo de espera

transação	tick
T1	0
T2	0
T3	1

T1 conseguiu executar
Seu contador é resetado

[illegible]

Transações

T1: read(B); write(A); write(C).	T2: read(A); read(B); read(C).	T3: write(C); read(A); write(B).
--------------------------------------------------	------------------------------------------------	--------------------------------------------------

Tabela de bloqueio

item	lista
A	
B	R(1)
C	W(3)

Controle de tempo de espera

transação	tick
T1	0
T2	0
T3	1

T2 comita
Seu contador é resetado

[illegible]

Transações

T1: read(B); write(A); write(C).	T2: read(A); read(B); read(C).	T3: write(C); read(A); write(B).
-------------------------------------------	-----------------------------------------	-------------------------------------------

Tabela de bloqueio

item	lista
A	
B	R(1)
C	W(3)

Controle de tempo de espera

transação	tick
T1	0
T2	0
T3	0

T3 conseguiu executar
Seu contador é resetado

T1	T2	T3
	R(A)	
		R(A)
R(B)		
	R(B)	
		abort
abort		
	R(C)	
R(B)		
	commit	
		W(C)
W(A)		

Transações

T1: read(B); write(A); write(C).	T2: read(A); read(B); read(C).	T3: write(C); read(A); write(B).
-------------------------------------------	-----------------------------------------	-------------------------------------------

Tabela de bloqueio

item	lista
A	W(1)
B	R(1)
C	W(3)

Controle de tempo de espera

transação	tick
T1	0
T2	0
T3	0

T1 conseguiu executar
Seu contador é resetado

T1	T2	T3
	R(A)	
		R(A)
R(B)		
	R(B)	
		abort
abort		
	R(C)	
R(B)		
	commit	
		W(C)
W(A)		

Transações

T1:	T2:	T3:
read(B);	read(A);	write(C);
write(A);	read(B);	read(A);
write(C).	read(C).	write(B).

Tabela de bloqueio

item	lista
A	W(1), R(3)
B	R(1)
C	W(3)

Controle de tempo de espera

transação	tick
T1	0
T2	0
T3	1

T3 não conseguiu executar
Seu contador é incrementado

T1	T2	T3
	R(A)	
		R(A)
R(B)		
	R(B)	
		abort
abort		
	R(C)	
R(B)		
	commit	
		W(C)
W(A)		

Transações

T1: read(B); write(A); write(C).	T2: read(A); read(B); read(C).	T3: write(C); read(A); write(B).
--------------------------------------------------	-----------------------------------------	-------------------------------------------

Tabela de bloqueio

item	lista
A	W(1), R(3)
B	R(1)
C	W(3), W(1)

Controle de tempo de espera

transação	tick
T1	1
T2	0
T3	1

T1 não conseguiu executar
Seu contador é incrementado

T1	T2	T3
	R(A)	
		R(A)
R(B)		
	R(B)	
		abort
abort		
	R(C)	
R(B)		
	commit	
		W(C)
W(A)		

Transações

T1:	T2:	T3:
read(B);	read(A);	write(C);
write(A);	read(B);	read(A);
write(C).	read(C).	write(B).

Tabela de bloqueio

item	lista
A	W(1), R(3)
B	R(1)
C	W(3), W(1)

Controle de tempo de espera

transação	tick
T1	1
T2	0
T3	2

T3 não conseguiu executar
Seu contador é incrementado

T1	T2	T3
	R(A)	
		R(A)
R(B)		
	R(B)	
		abort
abort		
	R(C)	
R(B)		
	commit	
		W(C)
W(A)		
		abort

Transações

T1:	T2:	T3:
read(B);	read(A);	write(C);
write(A);	read(B);	read(A);
write(C).	read(C).	write(B).

Tabela de bloqueio

item	lista
A	W(1)
B	R(1)
C	W(1)

Controle de tempo de espera

transação	tick
T1	1
T2	0
T3	0

O contador de T1 atingiu o seu limite.
T1 aborta e o contador é resetado

T1	T2	T3
	R(A)	
		R(A)
R(B)		
	R(B)	
		abort
abort		
	R(C)	
R(B)		
	commit	
		W(C)
W(A)		
		abort
W(C)		

Transações

T1: read(B); write(A); write(C).	T2: read(A); read(B); read(C).	T3: write(C); read(A); write(B).
-------------------------------------------	-----------------------------------------	-------------------------------------------

Tabela de bloqueio

item	lista
A	W(1)
B	R(1)
C	W(1)

Controle de tempo de espera

transação	tick
T1	0
T2	0
T3	0

T1 conseguiu executar
Seu contador é resetado

T1	T2	T3
	R(A)	
		R(A)
R(B)		
	R(B)	
		abort
abort		
	R(C)	
R(B)		
	commit	
		W(C)
W(A)		
		abort
W(C)		

Transações

T1:	T2:	T3:
read(B);	read(A);	write(C);
write(A);	read(B);	read(A);
write(C).	read(C).	write(B).

Tabela de bloqueio

item	lista
A	W(1)
B	R(1)
C	W(1), R(3)

Controle de tempo de espera

transação	tick
T1	0
T2	0
T3	1

T3 não conseguiu executar
Seu contador é incrementado

T1	T2	T3
	R(A)	
		R(A)
R(B)		
	R(B)	
		abort
abort		
	R(C)	
R(B)		
	commit	
		W(C)
W(A)		
		abort
W(C)		
commit		

Transações

T1: read(B); write(A); write(C).	T2: read(A); read(B); read(C).	T3: write(C); read(A); write(B).
--------------------------------------------------	------------------------------------------------	--------------------------------------------------

Tabela de bloqueio

item	lista
A	
B	
C	R(3)

Controle de tempo de espera

transação	tick
T1	0
T2	0
T3	1

T1 comita
Seu contador é resetado

T1	T2	T3
	R(A)	
		R(A)
R(B)		
	R(B)	
		abort
abort		
	R(C)	
R(B)		
	commit	
		W(C)
W(A)		
		abort
W(C)		
commit		
		W(C)
		R(A)
		W(B)
		commit

Transações

T1:	T2:	T3:
read(B);	read(A);	write(C);
write(A);	read(B);	read(A);
write(C).	read(C).	write(B).

Tabela de bloqueio

item	lista
A	
B	
C	

Controle de tempo de espera

transação	tick
T1	0
T2	0
T3	0

Como só sobrou uma transação ativa, ela obtém todos os recursos que precisa e comita

ConcurrencyManager

- Para resolver o problema, deve-se compreender como a classe **ConcurrencyManager** funciona
- Isso é importante para saber quais funções precisam ser sobrescritas

Gerenciamento de acesso concorrente

- Essas são as principais funções do `ConcurrencyManager`
 - que podem ser sobrescritas

```
public Record processInstruction(Transaction t) throws Exception;  
public void commit(Transaction t) throws Exception;
```

```
public void recoverFromLog() throws Exception;  
public void clearLog() throws Exception;  
public void flushLog() throws Exception;
```

```
protected Record processCurrentInstruction(Transaction t) throws Exception;  
protected boolean shouldAbort(Transaction t);  
protected void abort(Transaction t) throws Exception;
```

```
protected Transaction queueTransaction(Instruction instruction);  
protected Transaction addToQueue(Item item, Instruction instruction);
```

Gerenciamento de acesso concorrente

- Funções públicas chamadas pelo simulador

```
//tenta executar a instrução atual de uma transação  
public Record processInstruction(Transaction t) throws Exception;
```

```
//comita a transação  
public void commit(Transaction t) throws Exception;
```

Gerenciamento de acesso concorrente

- Funções públicas usadas para controle de log
 - São importantes para recuperação em caso de erro

```
public void recoverFromLog() throws Exception;  
public void clearLog() throws Exception;  
public void flushLog() throws Exception;
```


Gerenciamento de acesso concorrente

- Funções chamadas pela função `processInstruction()`

```
//efetivamente processa a instrução atual de uma transação  
protected Record processCurrentInstruction(Transaction t) throws Exception;
```

```
//verifica se é necessário abortar uma transação  
protected boolean shouldAbort(Transaction t);
```

```
//aborta uma transação  
protected void abort(Transaction t) throws Exception;
```

Gerenciamento de acesso concorrente

- Funções de enfileiramento chamadas pela função `processInstruction()`

```
//adiciona uma transação na tabela de locks, se necessário  
protected Transaction queueTransaction(Instruction instruction);
```

```
//efetivamente adiciona uma transação na fila de locks de um item  
protected Transaction addToQueue(Item item, Instruction instruction);
```

Dica

- Importante: ao sobrescrever uma função, pode ser necessário assegurar que o comportamento original siga sendo executado.
 - Isso pode ser atingido usando o comando **super**

```
@Override
protected int foo() {
    int resp_ = super.foo()
    int resp = resp_ + ... ;
    return resp;
}
```

Testes

- Para testar, é necessário trocar o gerenciador de acesso concorrente usado pelo simulador
- Isso é feito dentro da função run() do simulador

```
public void run(int error) throws Exception {  
    ConcurrencyManager manager = new XXXConcurrencyManager(2);  
    ...  
}
```

- Teste com diferentes cenários para garantir que o controle seja realizado de forma consistente

Entrega

- Prazo final de entrega, sem descontos
 - Domingo, 22 de agosto às 22:00
- A cada dia de atraso, a nota é decrementada em 50%.
- O que entregar
 - O código fonte da classe criada (.java)
 - A classe deve pertencer ao pacote `ibd.transaction`