

Exclusão Mútua – locks explícitos e hierarquias de objectos

Grupo de Sistemas Distribuídos
Universidade do Minho

1 Objectivos

Uso das facilidades de biblioteca em Java relativas a locks para resolver problemas de exclusão mútua envolvendo hierarquias de objectos.

2 Mecanismos

Mecanismo de exclusão mútua em `java.util.concurrent.locks`:

- Interface `Lock` e classe `ReentrantLock`;
métodos `lock`, `unlock`.
- Interface `ReadWriteLock` e classe `ReentrantReadWriteLock`;
métodos `readLock`, `writeLock`.

3 Exercícios propostos

1. Modifique o exercício do banco de modo a que, em vez de existir um número fixo de contas, seja possível criar contas e fechar contas, como na seguinte interface:

```
interface Bank {  
    int createAccount(float initialBalance);  
    float closeAccount(int id) throws InvalidAccount;  
    void deposit(int id, int val) throws InvalidAccount;  
    void withdraw(int id, int val) throws InvalidAccount;  
    void transfer(int from, int to, float amount) throws InvalidAccount, NotEnoughFunds;  
    float totalBalance(int accounts[]) throws InvalidAccount;  
}
```

em que criar uma conta devolve um identificador de conta, para ser usado em outras operações e fechar uma conta devolve o saldo desta; deverá ainda ser possível obter a soma do saldo de um conjunto de contas. Algumas operações deverão poder lançar exceções se o identificador de conta não existir ou não houver saldo suficiente.

Apesar de permitir concorrência, garanta que os resultados sejam equivalentes a ter acontecido uma operação de cada vez. Por exemplo, ao somar os saldos de um conjunto de contas, não permita que sejam usados montantes a meio de uma transferência (e.g., depois de retirar da conta origem e antes de somar à conta destino).