UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
DEPARTAMENTO DE INFORMÁTICA E MATEMÁTICA APLICADA
DIM0124 - PROGRAMAÇÃO CONCORRENTE

Trabalho Prático

Mecanismos de Sincronização

João de Souza Fernandes Vieira
Rafael Fortunato de Paula Pereira

Natal - RN
2022

O Problema

O problema abordado neste trabalho é o “Banheiro Unissex”, segue a descrição que nos serviu de base para o desenvolvimento da solução:

“Um escritório contém um banheiro que pode ser utilizado tanto por homens quanto por mulheres,mas não por ambos ao mesmo tempo. Se um homem estiver no banheiro, outros homens podem entrar, porém eventuais mulheres que desejem utilizar o banheiro devem esperar ele ficar vazio.Se uma mulher estiver no banheiro, outras mulheres podem entrar, porém eventuais homens que desejem utilizar o banheiro devem esperar ele ficar vazio. Cada pessoa (homem ou mulher) pode passar um determinado tempo utilizando o banheiro, que possui uma capacidade limite de pessoas que podem utilizá-lo ao mesmo tempo.”

A Solução

A solução foi criada na linguagem Java, e utilizando suas bibliotecas padrões para programação concorrente. Foram desenvolvidas quatro classes, sendo elas:

- BanheiroUnissex → Representa o banheiro unissex da descrição do problema, possui uma capacidade máxima, definida como argumento na execução do programa, uma *array* de pessoas que estão no banheiro, e o sexo atual das pessoas que o ocupam. Além disso, também possui métodos que utilizam uma *ReentrantLock* para permitir a entrada e saída do banheiro, garantindo a corretude.

- Pessoa → A classe Pessoa representa, nesse contexto, um funcionário ou funcionária do escritório que deseja utilizar o banheiro. Pessoas possuem nome, sexo, uma referência ao BanheiroUnissex e duas flags que definem se a pessoa já utilizou o banheiro e se o tempo necessário para usá-lo já transcorreu.

- Sexo → Um Enum criado para diferenciar Pessoas do sexo masculino e feminino. Também foi necessário criar o “NENHUM” para lidar com referências ao sexo ocupado do banheiro enquanto ele estava vazio (null).

- Main → Classe que possui o método main do programa, responsável por instanciar o BanheiroUnissex, criar as pessoas e iniciar as threads.

Corretude

A corretude do programa é garantida através do uso da implementação Reentrant Lock da interface Lock do Java. A interface provê de métodos com sincronização ao acessar recursos compartilhados, o BanheiroUnissex, no nosso caso. Ela o fez cercando os acessos ao recurso com chamadas dos métodos *lock* e *unlock*, que passam o “cadeado/fechadura” para a thread que o adquirir, e bloqueando o acesso das outras threads. Após acessar o recurso, a thread que detém a Lock abre mão dela, permitindo que uma das outras threads que aguardavam tentem adquiri-la.

Dificuldades

A maior dificuldade foi no momento de definição da solução e dos mecanismos que seriam utilizados na sua escrita. A linguagem Java possui um vasto arsenal de ferramentas disponíveis para se trabalhar com concorrência e *multithreading*, como: Thread, Runnable, Lock, Mutex, Semaphore, Executor, Phaser, e muitas mais.

Compilação e Execução

O projeto foi criado com Maven, e pode ser compilado com o seguinte comando na raíz do projeto:

```
mvn clean package
```

E para executar o .jar gerado, passando a capacidade máxima do banheiro como argumento, basta utilizar o seguinte:

```
java -jar .\target\banheiro-unissex-1.0-SNAPSHOT.jar 10
```

Referências

https://docs.oracle.com/javase/7/docs/api/java/util/concurrent/locks/Lock.html

https://www.geeksforgeeks.org/reentrant-lock-java/

https://www.baeldung.com/java-singleton

https://www.geeksforgeeks.org/singleton-class-java/

https://github.com/julianaabs/bathroom-with-threads/blob/master/src/lockVersion/Bathroom.java

https://github.com/brenov/unisex-bathroom

https://gitlab.com/veronicamars73/banheiro-unissex/-/blob/main/src/Banheiro.java