

Resolva os seguintes exercícios e apresente os testes com os quais validou a correção da implementação de cada exercício. A entrega deve ser feita através da criação da tag **0.2.0** no repositório de cada grupo.

1. Considere a classe **UnsafeSuccession**, que tem por objectivo representar uma sequência de elementos, consumidos de forma sequencial através do método **next**. Assuma que os elementos do *array* não são alterados após a construção da instância da sequência.

```
class UnsafeSuccession<T>(  
    private val items: Array<T>  
) {  
    private var index = 0  
    fun next(): T? =  
        if (index < items.size) {  
            items[index++]  
        } else {  
            null  
        }  
}
```

Realize uma versão *thread-safe* desta classe, sem recurso ao uso de *locks*.

2. Considere a classe **UnsafeResourceManager**, que tem por objectivo a chamada do método **close** de um **AutoCloseable** quando não existirem mais utilizações desse objecto.

```
class UnsafeResourceManager(private val obj: AutoCloseable, usages: Int) {  
    private var currentUsages = usages  
    fun release() {  
        if (currentUsages == 0) {  
            throw IllegalStateException("usage count is already zero")  
        }  
        currentUsages -= 1  
        if (currentUsages == 0) {  
            obj.close()  
        }  
    }  
}
```

Realize uma versão *thread-safe* desta classe, sem recurso ao uso de *locks*.

3. Implemente a seguinte função

```
fun <T> scopedAny(  
    futures: List<CompletableFuture<T>>,  
    onSuccess: (T)->Unit  
) : CompletableFuture<T>
```

que, dado uma lista não vazia de *futures*, retorna imediatamente um *future* que se completa:

- Apenas quando todos os *futures* passados como argumento se completarem.
- Com sucesso, quando pelo menos um *future* da lista se completou com sucesso. O valor do *future* retornado deve ser o valor do *future* da lista que se completou em primeiro lugar.
- Com exceção, quando todos os *futures* da lista se completarem com exceção. A exceção do *future* retornado deve agregar as exceções de todos os *futures* da lista.

A função **onSuccess** deve ser chamada no máximo uma vez, quando o primeiro *future* se completar com sucesso.

Valoriza-se a minimização da aquisição de *locks* na implementação desta função.

Data limite de entrega: 4 de maio de 2024

ISEL, 15 de abril de 2024