

# Relatório Linguagens de Programação Q4

Guilherme Galvão Serra Silva - 520402

Roberto Sérgio Ribeiro Meneses - 520403

Cap 7

6.b)

## **Active Object:**

- Vantagens:
  - A execução simultânea pode ser vantajosa em ambientes multi core, onde operações independentes podem ser conduzidas ao mesmo tempo.
  - Modelagem de objetos ativos pode facilitar a comunicação e sincronização entre as "vítimas".
- Desvantagens:
  - A criação e gerenciamento de objetos ativos podem introduzir algum overhead.
  - A execução concorrente pode adicionar complexidade e exigir sincronização adicional.

O Active Object encapsula objetos em unidades ativas, cada uma com seu próprio thread de execução, permitindo operações simultâneas e facilitando a comunicação entre eles, ideal para ambientes multicore. No entanto, a criação de objetos ativos pode ter algum overhead, e a execução concorrente requer atenção à sincronização para evitar condições de corrida.

## **Data-Driven Concurrent:**

- Vantagens:
  - Abordagem mais direta, operando sobre uma estrutura de dados concorrente.
  - Pode ser mais simples e direta em alguns casos.
- Desvantagens:
  - Menos granularidade em termos de concorrência, pois não utiliza objetos ativos explicitamente.
  - Pode não tirar total proveito de sistemas multicore.

O Data-Driven Concurrent opera sobre uma estrutura de dados concorrente, simplificando a abordagem. No entanto, ao não utilizar objetos ativos explicitamente, possui menor granularidade em termos de concorrência. Embora seja mais direto, pode não explorar totalmente os benefícios de sistemas multicore.

## **Sequential Stateful Model:**

- Vantagens:
  - Simplicidade: A implementação é mais direta e sequencial, o que pode facilitar a compreensão.

- Menos Complexidade: Não há concorrência envolvida, o que pode reduzir a complexidade de sincronização.
- Desvantagens:
  - Não Aproveita Concorrência: Como a implementação é sequencial, ela não aproveita a concorrência para realizar operações independentes simultaneamente.
  - O modelo Sequencial pode ser não tão escalável como o modelo Concorrente.

O modelo Sequencial com Estados destaca-se por sua simplicidade e compreensibilidade, eliminando a complexidade da concorrência. Entretanto, essa abordagem não capitaliza a execução simultânea de operações independentes, o que a torna menos adequada para situações que demandam escalabilidade.

### **No Geral:**

Comparando os três algoritmos, inicialmente o custo computacional das três variações do algoritmo, para valores pequenos, não deve fazer muita diferença em relação à eficiência do tempo de execução, ou seja, a simplicidade de cada algoritmo deve ser a principal característica analisada.

Ao comparar as versões Data-Driven Concurrent e Active Object, a primeira tende a apresentar um desempenho superior, uma vez que incorpora concorrência e threads para otimizar a execução do código. Contudo, em cenários específicos que envolvem interações complexas e sincronizações detalhadas entre as "vítimas", bem como a utilização intensiva de objetos na implementação, a abordagem baseada em classes (Active Object) destaca-se como a escolha preferencial. Nesse contexto, a celeridade permanece como o fator primordial para o Data-Driven Concurrent, enquanto a facilidade de modelagem e as interações entre objetos se mantêm como elementos cruciais para o Active Object.

Porém, quando o número de pessoas na roda aumenta para valores grandes, a versão sequencial de estados é mais eficiente, pois não há necessidade de esperar a execução de computações paralelas, como é o caso do modelo concorrente.

O uso do short-circuit torna o algoritmo mais eficiente pela razão de economizar a quantidade de vezes que a mensagem seria passada, pois se não houvesse o short circuit, eventualmente a mensagem iria passar a maior parte do tempo sendo passada entre objetos mortos.