

## PRÁTICA SOBRE COLEÇÕES - FILA

Implemente em Java um programa que realize a simulação da rotina de atendimento em uma agência bancária. O objetivo principal desse simulador é permitir o dimensionamento da quantidade de atendentes do banco para garantir que todos os clientes possam ser atendidos num determinado intervalo de tempo.

Para isso, além da classe **Principal** (que conterá o método **main**), crie as seguintes classes: **Atendente**, **Cliente** e **Simulador**.

A classe **Atendente** representa a entidade responsável pelo atendimento aos clientes que aguardam em uma fila de espera. O único atributo dessa classe caracteriza o momento (valor inteiro) em que o atendente estará disponível para realizar o próximo atendimento. Além disso, implemente nessa classe os seguintes métodos:

- **atenderCliente**: método que recebe por parâmetro a duração do atendimento de um cliente e, com esse valor, atualiza o atributo que define o momento em que ele (atendente) estará disponível para realizar o atendimento do próximo cliente.
- **estaDisponivel**: método que recebe por parâmetro o tempo atual da simulação e, por meio de comparação com o momento em que o atendente estará disponível para realizar o atendimento do próximo cliente, retorna se o atendente está disponível para atendimento no tempo atual.

A classe **Cliente**, que representa o cliente da agência bancária, contará somente com os seguintes atributos: **tempo de chegada** (valor inteiro que define o momento de chegada do cliente na agência) e a **duração do atendimento** (valor inteiro que corresponde ao tempo que será gasto pelo atendente no seu atendimento). Ao criar um objeto do tipo **Cliente**, passe por parâmetro os valores desses dois atributos. Para definir o tempo de chegada de cada cliente, considere que o tempo entre a chegada de clientes na agência segue uma distribuição uniforme discreta com valor mínimo igual a 0 e máximo igual a 3. De modo semelhante, a duração do atendimento segue uma distribuição uniforme discreta com valor mínimo igual a 1 e máximo igual a 7. Utilize o método **nextInt** da classe **Random** para gerar os valores de acordo com uma distribuição uniforme discreta.

Por fim, a classe **Simulador**, que é a responsável pela realização de toda a simulação, é caracterizada pelos seguintes atributos:

- Fila de Clientes (tipo **Queue**): que representa a fila formada pelos clientes que chegaram na agência e ainda não foram atendidos.
- Lista de Atendentes (tipo **ArrayList**): representa a lista de atendentes que realizarão o atendimento aos clientes.
- Número de atendentes (inteiro): define a quantidade de atendentes na agência bancária.
- Duração da simulação (inteiro): define a quantidade de tempo que durará a simulação.

O construtor da classe **Simulador** deve ser responsável por instanciar a fila de clientes e a lista de atendentes, além de inicializar os atributos **número de atendentes** e **duração da simulação**. Além disso, no construtor você deve chamar um método que seja responsável por **criar** toda a **fila de clientes** da simulação e outro para **criar** a **lista de atendentes**.

Além desses métodos, a classe **Simulador** deverá conter um método denominado **simular**, onde efetivamente será colocado o código da simulação de atendimento aos clientes. Esse método conterá uma estrutura de repetição (**for**) para iterar ao longo do tempo até alcançar a duração da simulação. Nesse laço, para cada instante de tempo  $t$ , a seguinte tarefa deve ser executada: para cada atendente na Lista de Atendentes, verificar se o mesmo está disponível para atendimento e, em caso positivo, checar a Fila de Clientes com objetivo de realizar o atendimento de um cliente, ou seja, de retirá-lo da fila de espera. Para que seja possível acompanhar o progresso da simulação, para cada instante de tempo  $t$ , imprima na tela o tamanho da fila de clientes.

No método `main` da classe `Principal` instancie um objeto do tipo `Simulador` passando por parâmetro o número de atendentes da agência bancária e a duração da simulação. Em seguida, chame o método **`simular`** implementado na classe **`Simulador`** para que a simulação seja executada.