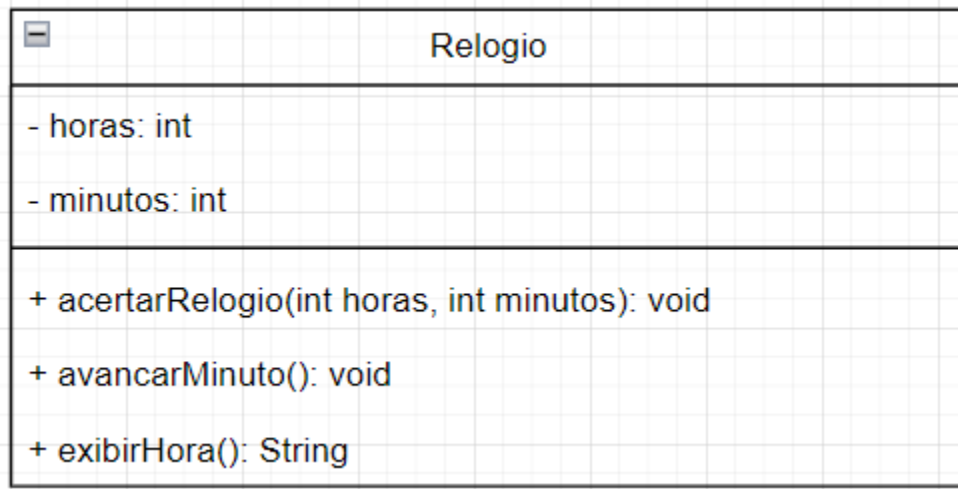


1. Classe **Relógio**



Implemente uma classe **Relógio** que tenha os seguintes atributos e métodos:

- **Atributos:**
 - `horas: int`
 - `minutos: int`
 - **Métodos:**
 - `acertarRelógio(int horas, int minutos): void` – Define a hora e os minutos.
 - `avançarMinuto(): void` – Avança um minuto no relógio, **ajustando também a hora se necessário**.
 - `exibirHora(): String` – Retorna a hora e minutos formatados no estilo "hh".
-

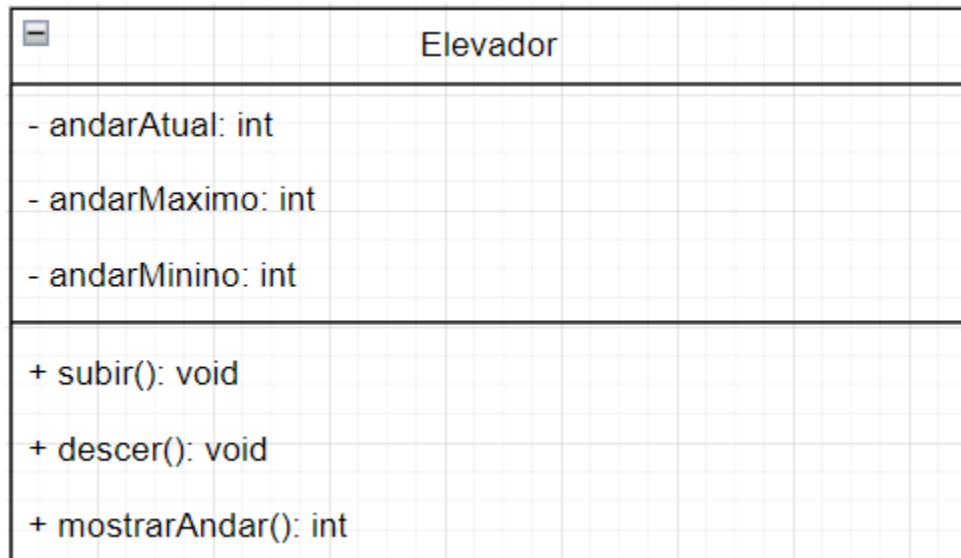
2. Classe **Carro**

Elabore o diagrama de classe e Implemente uma classe **Carro** com:

- **Atributos:**
 - `velocidadeAtual: double`
 - `marca: String`
- **Métodos:**
 - `acelerar(int valor): void` – Aumenta a velocidade atual em `valor`.
 - `frear(int valor): void` – Diminui a velocidade atual em `valor`, sem deixar a velocidade negativa.

- `exibirVelocidade(): String` – Retorna a velocidade atual do carro.
-

3. Classe **Elevador**



Crie uma classe **Elevador** com:

- **Atributos:**
 - `andarAtual: int`
 - `andarMaximo: int`
 - `andarMinimo: int`
 - **Métodos:**
 - `subir(): void` – Sobe um andar, até o andar máximo.
 - `descer(): void` – Desce um andar, até o andar mínimo.
 - `mostrarAndar(): int` – Retorna o andar atual.
-

4. Classe **Cinema**

Um cinema deseja automatizar a venda de ingressos para suas sessões. Desenvolva um sistema que gerencie as informações de uma sessão de cinema e permita realizar as seguintes operações:

- **Atributos:**
 - `filme: String` – Nome do filme.
 - `totalAssentos: int` – Total de assentos disponíveis na sala de cinema.

- `assentosOcupados: int` – Número de assentos já ocupados.
 - `precoIngresso: double` – Valor do ingresso por pessoa.
 - **Métodos:**
 - **Implemente um construtor que defina o total de assentos, o nome do filme e o preço do ingresso.**
 - `venderIngresso(int quantidade): void` – Registra a venda de ingressos e reduz o número de assentos disponíveis.
 - `cancelarIngresso(int quantidade): void` – Cancela uma venda de ingressos, liberando os assentos.
 - `verificarDisponibilidade(): boolean` – Verifica se há assentos disponíveis.
 - `calcularFaturamento(): double` – Calcula o faturamento total com base no número de ingressos vendidos e o preço por ingresso.
-

5. Classe **Turma**

Uma escola deseja automatizar o controle de notas e médias dos alunos. Para isso, desenvolva um sistema que gerencie as informações dos alunos e permita realizar as seguintes operações:

- **Atributos:**
 - `nomeAluno: String` – Nome do aluno.
 - `notas: List<Double>` – Lista das notas obtidas pelo aluno.
- **Métodos:**
 - `adicionarNota(double nota): void` – Adiciona uma nota à lista de notas do aluno.
 - `calcularMedia(): double` – Calcula e retorna a média das notas do aluno.
 - `verificarAprovacao(): String` – Verifica se a média do aluno é igual ou superior a 6.0. Retorna "Aprovado" se for, e "Reprovado" caso contrário.
 - `exibirNotas(): String` – Exibe todas as notas do aluno e a média final.