

Trabalho Computacional

Sistema de Gestão de Reservas com Validação Inteligente

Contexto

Um campus dispõe de salas que podem ser reservadas por professores e alunos. Cada reserva deve seguir critérios de horário, limite de tempo e disponibilidade. O sistema deve evitar conflitos, registrar os dados em arquivos e permitir a consulta e cancelamento eficiente usando Stream API.

Especificações

1. Crie uma interface `Reservavel` com os seguintes métodos:
 - `boolean reservar(Reserva r)`
 - `boolean cancelar(String idReserva)`
 - `List<Reserva> listarReservas()`
2. Crie a classe abstrata `Usuario` com os atributos `nome`, `id`, e `email`. Inclua:
 - Construtor com validações (ex: regex para email)
 - Métodos `get/set`, `toString` e `equals`
3. Implemente duas classes que estendem `Usuario`: `Professor` e `Aluno`. Cada uma implementa método específico:
 - `Professor` pode reservar com antecedência maior que aluno
 - `Aluno` só pode reservar no mesmo dia

4. Crie a classe **Reserva** com atributos:
`id`, `dataHoraInicio`, `dataHoraFim`, `idUsuario`, `idSala`. Implemente:
 - Validações de sobreposição de horários
 - Cálculo de tempo total reservado
 - Comparador por duração para ordenação
5. Crie a classe **Sala** que implementa **Reservavel**. Ela deve:
 - Armazenar lista de **Reserva**
 - Implementar métodos da interface garantindo não-conflito de horários
 - Permitir uso de Stream API para buscar reservas futuras ou canceladas
6. Crie a classe **SistemaDeReservas** com funcionalidades:
 - Carregar usuários e salas de arquivos
 - Listar todas as reservas de um usuário
 - Buscar reservas por período usando lambda
 - Cancelar reserva com verificação de permissão (somente o dono ou professor pode cancelar)
 - Gerar relatório (ordenado por tempo de uso) e salvar em arquivo
7. Crie exceções personalizadas:
 - **ReservaInvalidaException**
 - **UsuarioNaoAutorizadoException**
8. Crie o programa principal com:
 - a. Simulação de login (entrada de ID)
 - b. Listagem de salas disponíveis
 - c. Tentativa de reserva com feedback (válida/inválida)
 - d. Cancelamento e geração de relatório final

Obs.: Comente todas as partes do código e explique as escolhas de design. Avalie a reusabilidade, coesão e acoplamento das classes desenvolvidas.