

Agenda Chip

Gerenciando os contatos do seu chip SIM

David Sena *, Erivelton Facundo. †



1 Instruções Gerais

O objetivo deste trabalho é simular o sistema de um daqueles celulares antigos que guardavam apenas um número de telefone por contato.

2 Parte I

Seu sistema deve ser capaz de:

- adicionar um novo contato dado o nome e o telefone podendo inclusive existir vários contatos com o mesmo nome.
- o número do telefone deve possuir apenas dígitos.
- a memória deve ter espaço para 5 telefones.

*sena.ufc@gmail.com

†eng.erivelton@gmail.com

- deve ser possível alterar um registro da agenda.
- deve ser possível remover um registro da agenda.
- dado um texto, o sistema deve filtrar e retornar os contatos cujo parte do nome ou telefone corresponde ao texto.
- deve ser possível saber quantos registros livres ainda há no telefone.

Esses requisitos estão traduzidos nas interfaces `IAgendaChip` e `IContato` encontrados na figura 1.

Você encontrará junto a este pdf os arquivos `IAgendaChip.java` e `IContato.java`. Deverá fazer suas classes concretas implementarem essas duas interfaces.

3 Parte II

Como segunda parte do trabalho seu sistema deve suportar estes novos requisitos.

- deve ser possível favoritar um contato.
- deve ser possível desfavoritar um contato.
- deve ser possível obter a lista de favoritos.

Os requisitos acima estão traduzidos na interface `IAgendaChipPlus` que estende de `IAgendaChip`.

4 Conclusão

Você também vai receber o arquivo `controller.java` que contém alguns códigos de teste que você poderá usar para testar tando sua implementação concreta de `IAgendaChip` como também de `IAgendaChipPlus`.

Você não tem permissão para alterar as interfaces. Fora isso pode modelar e implementar seu sistema como bem entender.

Você deve entregar o diagrama de classes da sua implementação e um código que passe nos testes.

Se quiser pode implementar uma interface gráfica Swing que opere sobre a interface `IAgendaChipPlus`.

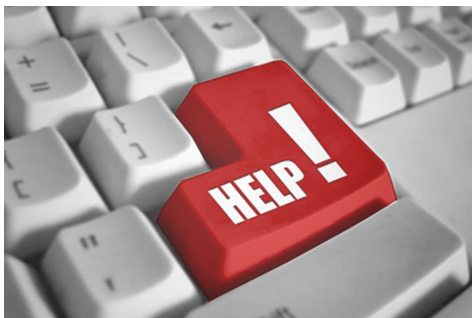
5 Sugestões

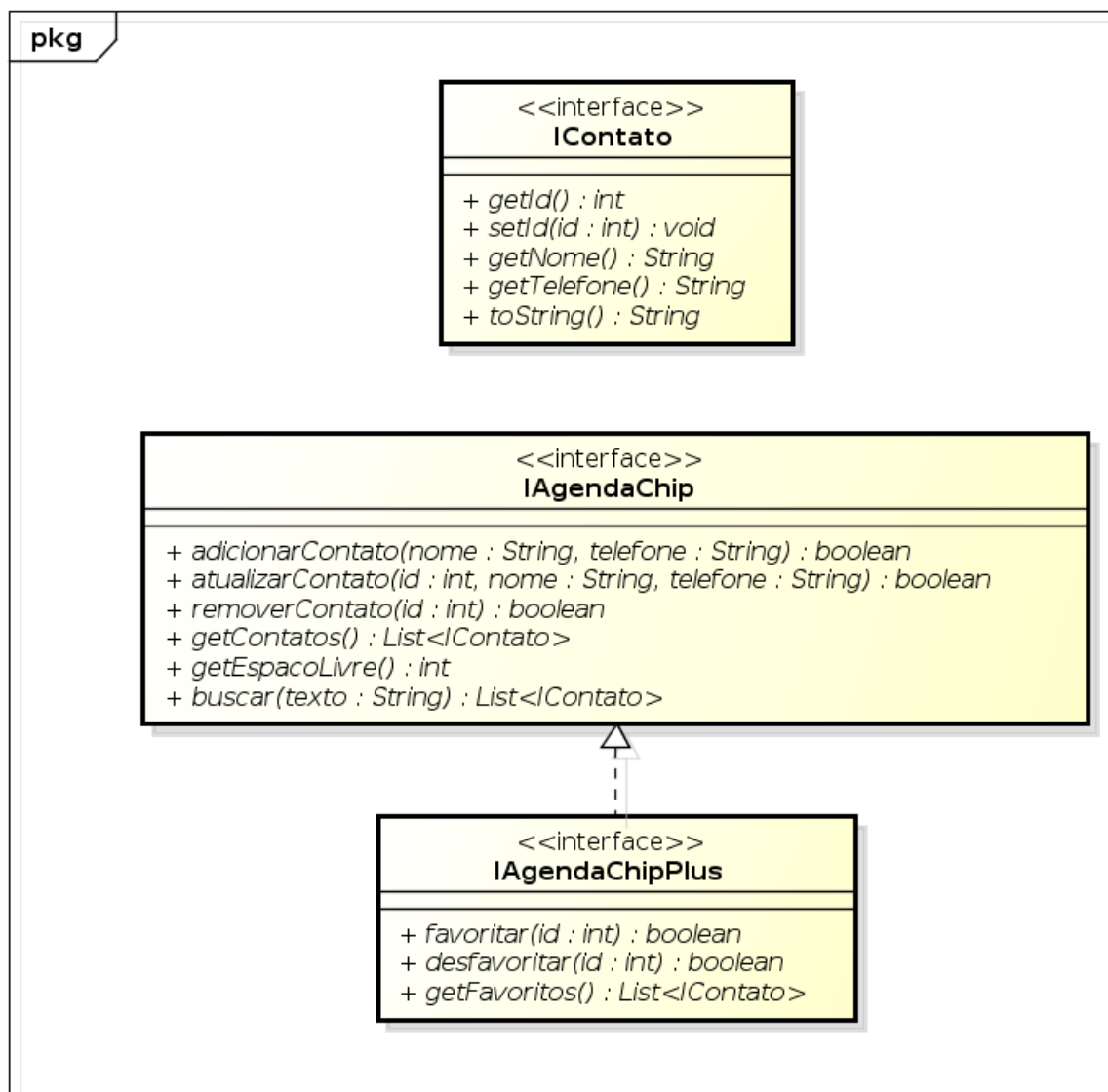
Tão importante quando ter um código que funciona é ter uma boa modelagem OO. Se concentre primeiro em realizar a primeira parte do trabalho e passar nos testes. Após isso pense na modelagem OO da segunda parte e siga adiante.

Observe que os testes não pegam muitos dos possíveis erros do sistema. Eles são um auxílio apenas. Passar nos testes não significa que seu código está completamente livre de erros. Adicione outros testes se tiver interesse.

Atenção, não repita código. Se quando você quiser modificar algo no seu programa, terá que modificar em mais de um lugar no código, provavelmente há alguma coisa errada. Bom trabalho.

Se precisar de ajuda, lembre-se dos professores, bolsistas, monitores e seus amigos. A ajuda pode estar a um botão de distância.





powered by Astah

Figura 1: Interfaces de Serviço