

## PROGRAMAÇÃO DE COMPUTADORES II

### Atividade 8 – Agenda



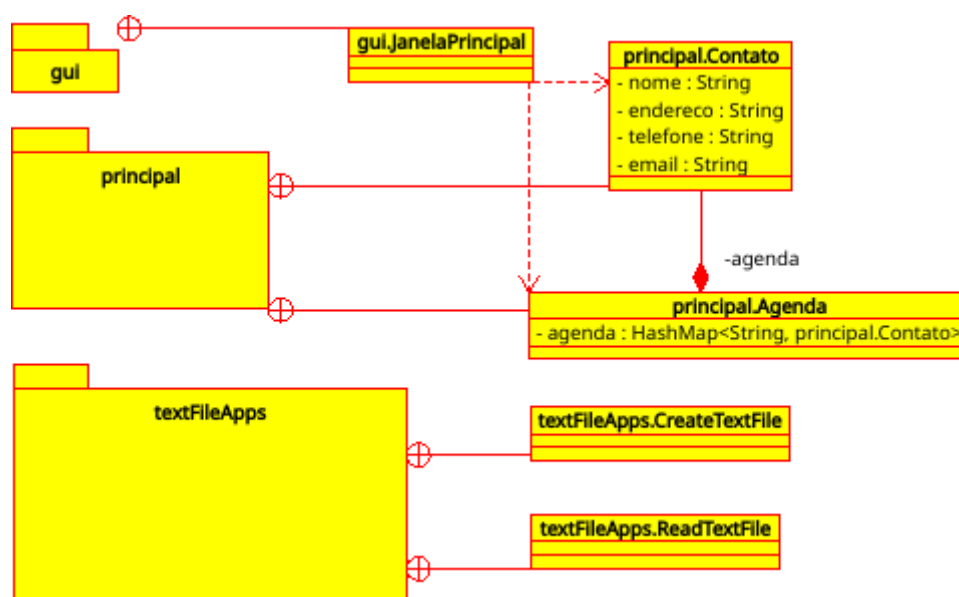
Na aula de hoje você deverá implementar uma agenda pessoal que mantenha detalhes de contato de pessoas, tais como: nome, endereço, número de telefone e e-mail – para um número arbitrário de pessoas. Os detalhes de contato serão indexados na agenda tanto pelo nome quanto pelo número do telefone, isto é, deverá ser possível fazer uma pesquisa por nome ou por número do telefone (ou parte do número). Resumindo, sua agenda terá as funções de incluir, alterar, pesquisar e listar contatos. Você deverá implementar uma classe adicional no seu projeto para realizar a entrada e saída de dados de sua agenda pessoal. Esta classe terá a responsabilidade de gravar/ler os dados em um arquivo texto no próprio diretório do seu projeto com o nome “agendaPessoal.txt”.

Sabendo que há 3 etapas que fazem parte do armazenamento de dados em um arquivo, a saber:

1. O arquivo é aberto
2. Os dados são gravados
3. O arquivo é fechado

Certifique-se de tratar os erros que podem ocorrer nestas etapas, para tornar sua aplicação mais tolerante a falhas.

Diagrama de classes (sugerido):



#### Dicas:

- Verifique o método `public boolean startsWith(String prefix)` da classe `String`



## PROGRAMAÇÃO DE COMPUTADORES II

### Lembretes:

- Somente a decisão de lançar uma exceção evitará ativamente que o programador do cliente ignore as consequências da falha do método.
- É comum que métodos que retornam referências de objeto utilizem o valor null como uma indicação de falha ou de erro.
- Claramente, essa abordagem não pode ser utilizada onde todos os valores do tipo de retorno já têm significados válidos para o cliente. Nesses casos, normalmente será necessário recorrer a uma técnica alternativa de lançar uma exceção.
- Um detalhe importante sem exceções: não há como exigir que o cliente verifique o valor de retorno para suas propriedades diagnósticas.
- Em muitos casos uma solicitação malsucedida não representará um erro de programação lógica, enquanto uma modificação incorreta quase certamente representará.
- As subclasses de Error são normalmente reservadas para erros de sistema em tempo de execução, em vez de erros sobre os quais o programador tem controle.
- Exceções verificadas devem ser utilizadas para situações de falha que estão fora do controle do programador, como um disco cheio ao tentar gravar um arquivo.
- O compilador indicará um erro se quaisquer instruções forem escritas após uma instrução throws porque elas nunca poderiam ser executadas.
- Considere a seguinte linha de comando:  
`AddressDetails details = address.getDetails(null)`  
Caso ocorra uma exceção lançada pelo método getDetails, haverá uma interrupção dessa primeira instrução e nenhuma atribuição ocorrerá na variável details.
- Duas questões complementares que acompanham um informe de erro: recuperação após erro e prevenção do erro.
- Há 3 etapas que fazem parte do armazenamento de dados em um arquivo:
  - O arquivo é aberto
  - Os dados são gravados
  - O arquivo é fechado