

Universidade Federal de Campina Grande – UFCG

Centro de Engenharia Elétrica e Informática – CEEI

Departamento de Sistemas e Computação – DSC

Disciplina: Laboratório de Programação 2

Período: 2011.1

Professores: Livia Campos (turma1)

Reinaldo Gomes (turma2)

Nazareno Andrade (turma 3)

Laboratório 10

Neste laboratório iremos praticar o uso de interfaces para a criação de classes em Java.

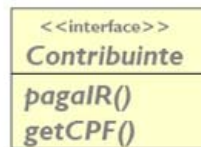
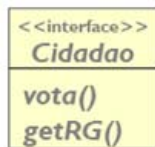
Instruções:

- Data de entrega: 29/04/2011 até 23:59h (Turmas 1, 2 e 3)
- Crie o projeto lab10 no eclipse e programe todos os exercícios dentro do pacote lp2.lab10;
- **Salve o projeto** em um arquivo zip chamado lab10-<seuNome>.zip (ou .tgz) e o envie para [juiz.lp2@gmail.com](mailto:juiiz.lp2@gmail.com) Use como *subject* do email lab10-<seuNome> (turma 1, 2 ou 3). **Nomeações incorretas dos arquivos a serem enviados serão penalizadas na nota do lab;**
- Certifique-se de que seus programas não têm erros de compilação;
- Antes da definição de cada classe escreva o seu nome em comentário (*/*Aluno: <seu nome>*/*);
- Não copie o programa do seu vizinho. Se tiver dúvida converse com o professor ou com um monitor.

Uma empresa está atualizando o cadastro de todos os seus funcionários e para tanto vai informatizar as informações de todo eles. Durante essa informatização, verifica-se que as informações referentes aos dados pessoais de cada funcionários são necessários assim como as informações de contato. Para realizar a implementação desse sistema você deve:

1. Escreva classes e interface que permitirão a criação de uma lista de contatos que armazena as informações de um conjunto de pessoas. Uma pessoa deve possuir informações do nome, RG, CPF, idade, salário e informações de como pode ser realizado o contato com o mesmo (ItemDeContato). Para implementar a classe Pessoa você deve:

- (i) Implementar duas interfaces que representam algumas informações das pessoas: Cidadao e Contribuinte. Essas interfaces devem estar de acordo com a seguinte definição:



- (ii) O método vota() deve retornar um boolean indicando se a pessoa tem idade para votar ou não;
- (iii) O método pagaIR() deve retornar o valor que a pessoa deve pagar de Imposto de renda. Se a pessoa ganhar até R\$ 1500,00 é isento, de R\$ 1500,01 a R\$ 3000,00 pagará 15 % e acima de R\$ 3000,00 deve pagar 25%.
- (iv) Crie um ArrayList para armazenar as informações de contato que cada pessoa pode ter (ItemDeContato).

A informação referente ao contato da pessoa deve ser qualquer objeto que se comporte como um ItemDeContato, que é uma interface contendo os seguintes métodos:

- (v) String exibirContato(), que retorna uma String representando o item de contato;
- (vi) TipoDeItemDeContato tipoDoContato(), que retorna um valor de uma enumeração que pode ser ENDERECO, TELEFONE, EMAIL;
- (vii) boolean equals(Object obj).

Escreva a interface ItemDeContato e escreva as classes Telefone, Endereco e Email que se comportam como itens de contato (implementam a interface).

OBS. E-mails devem ter exatamente um @, pelo menos 1 ponto ("."), e não pode conter caracteres como ", " ; , " ? " & " ! " e " : " (esta é uma verificação simbólica; se fosse na verdade muitas outras checagens seriam necessárias).

OBS. O Telefone deve ser representado por 4 valores inteiros: código internacional, código de área, prefixo e número.

OBS. O endereço deve ser representado por strings para rua/av., complemento, bairro, cidade, estado, CEP e um inteiro para número.

2. Escreva uma classe chamada ListaContatos para representar uma lista de contatos de elementos do tipo Pessoa. Deve ser possível adicionar meios de contatos a pessoas já existentes e também de pessoas que ainda não estão na lista. No primeiro caso, um novo contato para essa pessoa deve ser adicionado, já no segundo, devemos criar um novo elemento que será adicionado na lista. Para implementar essa classe considere as seguintes métodos:

- (i) Adicionar pessoa: não devem ser permitidos que itens de contato iguais sejam adicionados a uma pessoa específica. Se, por exemplo, você já acrescentou uma pessoa com o telefone cuja representação em String é "55 83 33333333" então você não deve permitir que outro item de contato idêntico a este seja associado a esse mesmo contato;

- (ii) Remover item de contato: deve retornar true quando o item existe e é removido com sucesso, e false caso contrário;
- (iii) Verificar se existe um item de contato: deve retornar true quando o item pesquisado está associado ao amigo indicado e false caso contrário;
- (iv) Substituir um item de contato existente por um novo;
- (v) Obter contatos de uma pessoa: este método deve retornar uma lista de itens de contato associada a uma determinada pessoa;
- (vi) Obter nomes das pessoas cadastradas: retorna uma lista com o nome das pessoas já cadastradas.

OBS1. Pense sobre que tipos de coleção usar à luz do que aprendemos sobre coleções em sala de aula.

OBS2. Seu sistema **NÃO deve ser** case-sensitive, assim os nomes João e joão representarão o mesmo contato.

3. Escreva um método main para permitir a manipulação dos contatos através de um menu que utilize os métodos disponíveis na classe.

Observações:

1. Usem a criatividade para escreverem as classes e incluam métodos que acharem necessários. Lembrem-se sempre de ter bom senso quando forem criar novos métodos.
2. Inclua javadoc e testes de unidade para as classes implementadas.