

Programação - Herança

Nesta atividade você deve desenvolver um programa para cadastrar Clientes e Funcionários. O programa deve ser desenvolvido em Java e deve obrigatoriamente usar hierarquia entre uma classe chamada Pessoa e as classes Cliente e Funcionário. Além dessas três classes, é necessário ter uma classe chamada Util com um método chamado validarEmail. Por fim, deve existir uma classe chamada Menu, onde o usuário realiza o cadastro dos novos clientes e funcionários.

Em resumo, deve existir:

- Classe Pessoa, com os atributos String nome, String sobrenome e Date dataNascimento.
- Classe Cliente que estende Pessoa e possui os atributos Integer idCliente, String email, String nome, String sobrenome e Date dataNascimento.
- Classe Funcionario que estende Pessoa e possui os atributos Integer idFuncionario, String cargo, Float salario, Date dataContratacao, String nome, String sobrenome e Date dataNascimento.
- Classe Util com um método público chamado validarEmail.
- Classe menu, com o método main e as opções para o usuário cadastrar um cliente ou um funcionário.

As classes Pessoa, Cliente e Funcionário devem possuir um método construtor cada. Para todos os atributos das três classes é necessário criar um método get e um set. Atente-se para escolher o modificador de acesso correto para os atributos e para os métodos.

Nas classes Cliente e Funcionário atente-se que Cliente extends Pessoa e Funcionario extends Pessoa.

A classe Menu contém o método main do programa. Nela deve existir uma lista de clientes (`List<Cliente> clientes = new ArrayList<>();`) e uma lista de funcionários (`List<Funcionario> funcionarios = new ArrayList<>();`).

Para realizar a validação do email pesquise sobre o funcionamento de expressões regulares em Java. Você pode utilizar a biblioteca `java.util.regex` para essa tarefa. Você pode usar a expressão `"^[a-zA-Z0-9.!#$%&'*/+=?^_`{|}~-]+@((\\[[0-9]{1,3}\\.[0-9]{1,3}\\.[0-9]{1,3}\\.[0-9]{1,3}\\)|((\\[a-zA-Z\\-0-9+\\.]\\.[a-zA-Z]{2,}))\\.)"` para validar o email.

Em algumas situações é necessário cadastrar datas. É possível converter uma String para um Date através dos comandos:

```
DateFormat formato = new SimpleDateFormat("dd/mm/yy");
formato.parse(nomeString);
```

A imagem a seguir apresentam um exemplo do funcionamento esperado do programa. Note que a entrada do email é chamada duas vezes em caso de erro.

```
Saída - AtividadeCadastroRA3 (run) X
Selecione 1 para cadastrar um Cliente, 2 para cadastrar um Funcionário ou 3 para sair.
1
Cadastro de novo Cliente
Entre com o nome do Cliente:
José
Entre com o sobrenome do Cliente:
Augusto dos Santos
Entre com a data de nascimento do Cliente (dd/mm/aa)
12/02/91
Entre com o id do Cliente:
1
Entre com o email do Cliente:
zeaugusto
Entre com o email do Cliente:
zeaugusto@gmail.com
Cliente adicionado com sucesso.

Selecione 1 para cadastrar um Cliente, 2 para cadastrar um Funcionário ou 3 para sair.
1
Cadastro de novo Cliente
Entre com o nome do Cliente:
Mariana
Entre com o sobrenome do Cliente:
da Silva
Entre com a data de nascimento do Cliente (dd/mm/aa)
04/12/84
Entre com o id do Cliente:
2
Entre com o email do Cliente:
marianas@bol.com
Cliente adicionado com sucesso.

Selecione 1 para cadastrar um Cliente, 2 para cadastrar um Funcionário ou 3 para sair.
2
Cadastro de novo Funcionário
Entre com o nome do Funcionário:
Pedro
Entre com o sobrenome do Funcionário:
Barbosa
Entre com a data de nascimento do Funcionário (dd/mm/aa)
01/08/95
Entre com o Id do Funcionário:
1
Entre com o cargo do Funcionário:
Vendedor
Entre com o salário do Funcionário:
2200.00
Entre com a data de contratação do Funcionário (dd/mm/aa):
03/01/17
Funcionário adicionado com sucesso.
```