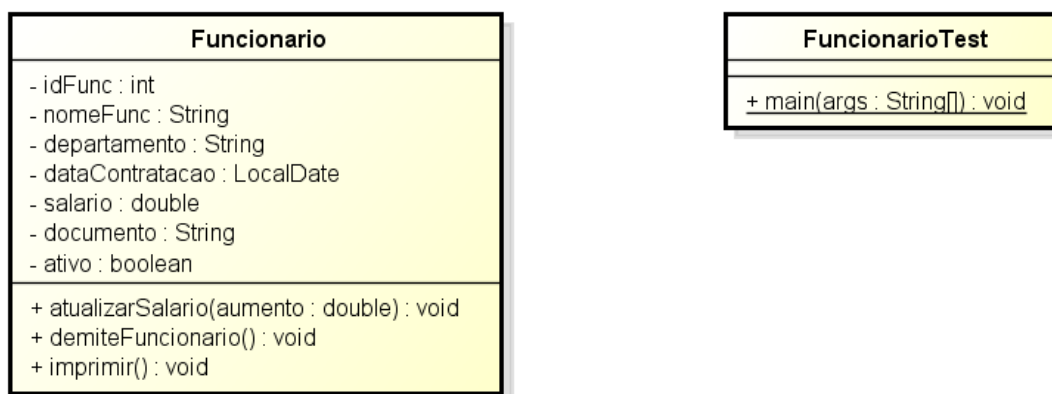


## **Conceitos para Estudo**

Classes, objetos, atributos, métodos, atributos estáticos e não estáticos, métodos estáticos e não estáticos, métodos gets e sets, encapsulamento, visibilidades de atributos e métodos, associação e cardinalidade, tipos primitivos, tipos referências, passagem de parâmetros por valor, passagem de parâmetros por referência, comando pacote (package), comando import, classe Scanner, Math, Random, vetores de tipos primitivos e de tipos referência.

## **Exercícios para a prática:**

1) Dado o diagrama de classe visto na Figura 1, execute as atividades como pedidas. Dê o nome para a aplicação de **RevisaoExercicio1**



**Figura 1 - Diagrama de Classes - Exercício 1**

### **Modele a Classe Funcionario como se pede:**

- Declare os campos membros da Classe funcionário como públicos.
- O campo membro **dataContratacao** deve ser do tipo **java.time.LocalDate**. Importe essa biblioteca.
- O campo ativo é do tipo booleano. Servirá para indicar se o funcionário é ativo ou não.
- Crie os modificadores gets/sets para todos os atributos no padrão da linguagem Java.
- O método **atualizaSalario** recebe como argumento um valor double que se refere ao valor dado como aumento, deve atualizar o campo salário. Importante validar a entrada, ou seja, aumento deve ser positivo.
- O método **demiteFuncionario** altera o campo ativo para false.
- O método **imprimir** deve exibir todos os dados de um objeto instanciado.

### **Modelo a Classe FuncionarioTest como se pede abaixo:**

- No Método **main()** deve-se instanciar um objeto chamado **func** do tipo **Funcionario**.

- b) Utilizando a classe Scanner, atribuir valores para cada campo de dado.  
Nota: para definir uma data, estude os conceitos vistos neste material:  
<http://blog.caelum.com.br/conheca-a-nova-api-de-datas-do-java-8/> Assunto: LocalDate.
- c) Invocar o método aumentarSalario passando o valor digitado pelo usuário. Experimente passar na primeira vez um valor negativo. Depois repita a operação passando um valor positivo. Verifique o tratamento dado.
- d) Imprimir os dados, invocando o método imprimir.
- e) Inativar o cliente usando o método demiteFuncionario.
- f) Imprimir novamente os dados.

## Programas simples para a prática

2) Execute os exercícios pedidos abaixo:

### Programa: RevisaoExercicio2

**Classe:** Pessoa

**Atributos:** nome, idade.

**Método:** void fazAniversario()

Crie uma pessoa, coloque seu nome e idade iniciais, faça alguns aniversários (aumentando a idade) e imprima seu nome e sua idade.

## Próximo programa

### Programa: RevisaoExercicio3

**Classe:** Porta

**Atributos:** aberta, cor, dimensaoX, dimensaoY, dimensaoZ

**Métodos:** void abre(), void fecha(), void pinta(String s), boolean estaAberta()

Crie uma porta, abra e feche a mesma, pinte-a de diversas cores, altere suas dimensões e use o método estaAberta para verificar se ela está aberta.

## Próximo programa

### Programa: RevisaoExercicio4

**Classe:** Casa

**Atributos:** cor, porta1, porta2, porta3

**Métodos:** void pinta(String s), int quantasPortasEstaoAbertas()

Crie uma casa e pinte-a. Crie três portas e coloque-as na casa; abra e feche as mesmas como desejar.

Utilize o método `quantasPortasEstaoAbertas` para imprimir o número de portas abertas.

## Próximo programa

### Programa: RevisaoExercicio5

Considerando que existe uma grande variedade de regras de **Bingo**, com diferentes padrões de sorteio, marcação e possibilidades de vitória. Ler as instruções e entender o formato da disputa é algo indispensável para os jogadores que queiram se divertir e vencer. Como se sabe, o conhecimento teórico é essencial para quem tem vontade de melhorar e aprender. A regra do nosso Bingo é simples, de fácil entendimento, São eles:

1. Cada jogador pode usar de 1 cartela de 24 números aleatórios de 1 a 75.
2. A cada rodada um número é sorteado e o jogador verifica se ele está na sua cartela.
3. O jogador completa sua(s) cartela(s) marcando os números sorteados.
4. O objetivo é completar linhas, colunas ou diagonais, de acordo com o padrão da fase.

Sua missão, elaborar um programa em Java que faça o sorteio e apresente como resultado o número de rodadas sorteadas (um número por vez) para que a cartela inicial também aleatória seja totalmente preenchida.