

**Desenvolva o código conforme pedido abaixo:**

A – Escreva uma classe `Funcionario` com 3 atributos privados: `id` (`String`), `nome` (`String`) e `salario` (`double`). Crie os getters e setters apenas se precisar. Crie em `Funcionario` APENAS UM construtor, que recebe o `id` como argumento.

B – Crie uma classe chamada `Utils`. Implemente um método na classe `Utils` com a seguinte assinatura: `public static boolean existe (List x, Funcionario y)`. Escreva esse método de forma que seja verificada a existência do objeto `Funcionario` representado por `y` na lista representada por `x`, retorne verdadeiro se existir e falso se não existir. Considere que dois objetos `Funcionario` são iguais se possuem o **mesmo id**. Não é permitida qualquer iteração para realizar esse item, ou seja, não use *for*, *iterator*, etc. Implemente o que for necessário para o funcionamento. Dica: você precisa criar uma alteração na classe `Funcionario` para resolver esse item.

C – Considere o cenário que ao utilizar o `System.out.println` em um objeto do tipo `Funcionario`, deve sair no console o `id`, `nome` e `salário` desse funcionário. Ex: 123 – Arthur Matias – R\$ 12000. Modifique a classe `Funcionario` de maneira que isso seja refletido.

D- Crie 2 subclasses da classe `Funcionario`: `AnalistaSistemas` e `Programador`. Faça-as funcionar corretamente.

E – Ainda na classe `Utils`, crie o método `public List retornaDados(String [] array_funcionarios)`;

Considere que o array recebido como argumento (`array_funcionarios`) contém `Strings` no seguinte formato: `id#nome#salario#tipo`. Por exemplo, considere os elementos desse array como as `Strings` (154-3#Simba Silva#3200#A, 553-2#Juju Juba#10000.0#P, etc.). Esses valores representam `id`, `nome`, `salário` e `tipo` do `Funcionario` (`AnalistaSistemas` ou `Programador`). Dessa maneira, implemente o método *retornaDados* de forma que seja retornado uma lista da seguinte forma: os elementos de `array_funcionarios` devem ser percorridos e serão criados objetos do tipo `AnalistaSistemas` ou `Programador`. Resumindo, você irá criar um objeto `AnalistaSistemas` (se o último caractere da `String` for A) ou `Programador` (se o último caractere da `string` for P) e adicionar à lista.

F – Crie uma nova classe, denominada `ProgramaPrincipal`. O programa deverá receber do console 5 `Strings` no seguinte formato: `id#nome#salario#tipo` (i.e., mesmo formato explicado no item E). Utilizando os métodos já criados, exiba no console os nomes e salários dos funcionários.

G – Exiba a média salarial dos analistas e dos programadores separadamente (ou seja, uma média para os analistas e uma média para os programadores). Para isso, crie um novo método em `Utils`.