FUNCIONÁRIO - EMPRESA EXTENSÃO

Dando sequência ao teste anterior, implemente os seguintes requisitos para a classe *Funcionario*:

- Encapsule todos os atributos e crie os respectivos getters e setters;
- Forneça um método *getBonificacao( )* que retorna o valor da bonificação anual do funcionário. A bonificação de um funcionário é igual a 2% do seu salário bruto anual;
- Forneça um método *dataDeAposentadoria* o qual retorna a data de aposentadoria em que o funcionário irá se aposentar. Homens se aposentam com 35 anos de serviço e mulheres com 30 anos de serviço;
- Forneça um método *eMaisVelho* que recebe um objeto funcionário e retorna true se o for mais velho que o objeto corrente (this) e falso, caso contrário.

Implemente uma sub-classe de *Funcionario*: *Programador*. Forneça para a classe *Programador*:

- Forneça os atributos:
  - *linguagem [ ]: String //todas as linguagens em que programa*
- Forneça os seguintes construtores:
  - Construtor sem parâmetros a qual ativa o construtor padrão da classe Funcionario (super) e inicializa o vetor *linguagem* com um *array* de tamanho 5;
  - Construtor a qual inicializa todos os atributos do objeto *Funcionaro/Programador* com parâmetros.
- Forneça um método *programaMesmasLinguagens* o qual recebe um objeto *Programador* e retorna true caso o vetor *linguagem* possua as mesmas linguagens que o objeto corrente (*this*);
- Forneça um método *toString*.

Implemente uma sub-classe de *Funcionario*: *Analista*. Forneça para a classe *Analista*:

- Forneça os atributos:
  - *diagrama [ ]: String //todos os diagramas de análise e projeto de sistemas que domina. Ex: Diagrama de Classes, Diagrama de Sequência, Diagrama de Objetos, Diagrama de Casos de Uso...*

- Forneça os seguintes construtores:
    - Construtor sem parâmetros a qual ativa o construtor padrão da classe Funcionario (*super*) e inicializa o vetor *diagrama* com um *array* de tamanho 5;
    - Construtor a qual inicializa todos os atributos do objeto *Funcionaro/Analista* com parâmetros.
- Forneça um método *dominaMesmaDiagrama* o qual recebe um objeto *Analista* e retorna true caso o vetor *diagrama* possua os mesmos diagramas que o objeto corrente (*this*);
- Sobrescreva o método *getBonificacao* da superclasse *Funcionario* pois um analista recebe 3% de bonificação;
- Forneça um método *toString*.

Implemente uma sub-classe de *Funcionário*: *Gerente*. Forneça para a classe *Gerente*:

- Forneça os atributos:
    - *metodologiaDesenvolvimento [ ]: String //todas as metodologias de desenvolvimento de software que domina.Ex: Scrum, XP...*
- Forneça os seguintes construtores:
    - Construtor sem parâmetros a qual ativa o construtor padrão da classe *Funcionario* (super) e inicializa o vetor *metodologiaDesenvolvimento* com um array de tamanho 5;
    - Construtor a qual inicializa todos os atributos do objeto *Funcionaro*/*Gerente* com parâmetros.
- Forneça um método *dominaMesmasMetodologias* o qual recebe um objeto *Gerente* e retorna true caso o vetor *metodologiaDesenvolvimento* possua as mesmas metodologias que o objeto corrente (*this*);
- Sobrescreva o método *getBonificacao* da superclasse *Funcionario* pois um gerente recebe 5% de bonificação;
- Forneça um método *toString*.

Em seguida, também no pacote *empresa*, incremente a classe *OperacaoFuncionario* onde:

- Implemente um método estático a qual recebe um *array* de *Funcionario* e imprimi a data em que cada *Funcionario* irá se aposentar;

- Implemente um método estático a qual recebe um *array* de *Funcionario* e imprimi a bonificação de cada funcionário e a soma total de bonificações a serem pagas pela empresa.

No *main* da classe *App* do pacote *teste,* crie um vetor de *Funcionario* de tamanho 12 e passe para este vetor 6 programadores, 4 analistas e 2 gerentes. Teste as funcionalidades da classe *OperacaoFuncionario*.