Trabalho 02

Uma empresa deseja desenvolver um sistema de folha de pagamento (*payroll*). A folha de pagamento deve ser emitida mensalmente para todos os funcionários (*employee*) cadastrados na empresa (*company*). O sistema deve guardar o contracheque (*paycheck*) mensal com todas as informações (base e calculadas) de cada funcionário. Cada funcionário é alocado em um departamento (*department*).

O salário líquido (*net salary*) de um funcionário é calculado inicialmente a partir do seu salário base (*base salary*), informado no cadastro, considerando a quantidade de horas trabalhadas (*hours worked*). A quantidade de horas trabalhadas é utilizada para calcular o salário tendo como base uma jornada de trabalho (*working hours*) de 160 horas. Para as horas que ultrapassarem este valor é aplicado um percentual adicional de 50% de hora extra (*overtime*). Se a quantidade de horas trabalhadas for inferior ou igual a 90% da jornada de trabalho, o funcionário deixa de receber o seu salário completo e passa a receber por hora trabalhada.

Caso o funcionário seja um diretor, o salário líquido é calculado inicialmente de forma idêntica ao salário de funcionários regulares. Entretanto, é acrescido um bônus (*bonus*) de 5% para cada ano em este funcionário tem atuado como diretor. O valor do bônus não pode ser maior que 40% do salário líquido (percentual teto para o bônus).

Após o cálculo inicial do salário líquido (independentemente do tipo de funcionário), devem ainda ser aplicados os seguintes descontos:

- Imposto de renda (*income tax*):

Salário líquido	Alíquota (tax rate) %
Até 1.499,15	-
De 1.499,16 até 2.246,75	7,5
De 2.246,76 até 2.995,70	15,0
De 2.995,71 até 3.743,19	22,5
Acima de 3.743,19	27,5

- INSS (social insurance): 11% sobre o salário líquido, com teto máximo de R$ 482,93.

Há ainda, um terceiro tipo de funcionário (funcionário externo). Embora, este funcionário não tenha vínculo empregatício com a empresa, o pagamento de seu salário é realizado por ela. Entretanto, os valores dos salários líquidos são cadastrados diretamente nos dados do funcionário. Estes funcionários não estão alocados em departamentos nem devem ter contracheques armazenados.

1. Faça o desenho UML da sua solução. Você deve utilizar a ferramenta abordada na disciplina (StarUML). Na representação dos relacionamentos, não esqueça de utilizar nomes de papel, cardinalidade (mesmo quando for 1) e navegabilidade.

2. A solução deve, obrigatoriamente, utilizar os conceitos vistos em sala: **herança, polimorfismo e interface**.

3. As classes de entidade (que armazenam informações) e de processamento não devem interagir diretamente com o usuário. Crie classes específicas para isso. Para este trabalho, **utilize o padrão MVC de modo similar ao exemplo do roteiro 02**. Aproveite para organizar seu código em pacotes (não faça tudo em um único pacote).

4. Entidades podem ter identificadores, mas os relacionamentos devem seguir as orientações vistas em sala. Lembre-se que o programa é orientado a objetos.

5. Implemente um programa em Java para o problema apresentado e que demonstre as seguintes funcionalidades:
   - Permitir a consulta, inclusão, edição e exclusão de funcionários.
   - Permitir a impressão de um relatório informando o nome dos funcionários e seus respectivos salários. No rodapé, deve ser apresentado o valor total da folha.
   - Permitir a impressão de um relatório informando o nome do departamento e os respectivos funcionários alocados. No rodapé, devem ser apresentadas a quantidade de departamentos e a quantidade total de funcionários da empresa.
   - Permitir a consulta do funcionário (independentemente do tipo de funcionário) que recebe o maior salário.

6. Você deve utilizar tratamento de exceção quando adequado. Por exemplo, um salário base deve ser maior do que zero e a quantidade de horas trabalhadas não pode ser inferior a zero.

7. Todas as classes e métodos devem ser documentados com notação javadoc. Para classes, você deve colocar uma breve descrição e utilizar o annotation "@author". Para cada método, você deve ter uma ou mais linhas explicando o comportamento e utilizar os annotations "@param" (explicando cada parâmetro) e "@return" (quando a operação for uma função). Veja a documentação disponibilizada no ambiente Material Didático (slides de referência sobre a linguagem Java).

8. As entidades do sistema devem ser **persistentes**, ou seja, devem ser armazenadas/carregadas a partir de arquivos. O roteiro 07 explica como persistir os objetos. **Utilize o padrão Repository**.

9. Utilize o **padrão de projeto (design pattern) Template Method** para a implementação dos relatórios.

10. Para testar o programa, você deve criar um pacote específico chamado "teste (test)". Neste pacote, implemente classes que inicializam os cadastros com objetos (evitando o cadastro manual a cada vez que o programa é executado). Embora o programa permita o armazenamento das entidades, deve ser possível retornar para o estado inicial dos dados (por exemplo, apagando todos os arquivos de repositório).