	Instituto Federal de Educação, Ciência e Tecnologia do Piauí Campus Teresina Zona Sul Licenciatura em Informática **Disciplina:** Programação Orientada a Objetos

Atividade

Questão 1

Imagine uma lâmpada que possua ter três estados: apagada, acesa e meia-luz. Crie uma classe *LampadaTresEstados* de forma encapsulada e que possua 3 método, um para apagar, outro acender e outro colocar a lâmpada em meia luz. Por padrão, o estado inicial de uma lâmpada deve ser apagado. Teste sua classe criando uma lâmpada, acendendo-a, colocando-a em meia luz e exibindo o estado final da lâmpada. Um exemplo do diagrama da classe *LampadaTresEstados* pode ser visto a seguir.

Lampada
- estado
+ Lampada() + apagar() + aceder() + meiaLuz()

Questão 2

Com base na classe LampadaTresEstados crie agora a classe *LampadaDimerizada* que representa uma lâmpada com controle de luminosidade. Ou seja, a luminosidade da lâmpada pode ser ajustada com qualquer valor entre 0% (apagada) e 100% (acesa). Nesta classe uma lâmpada deve possuir um método de *apagar*, *acender*, *aumentarLuminosidade* e *diminuirLuminosidade*. Os métodos de apagar e acender devem colocar a luminosidade da lâmapada em 0% e 100%. Já os métodos de aumentarLuminosidade e diminuirLuminosidade, deve incrementar e decrementar a luminosidade em 10%.

Teste sua classe criando uma lâmpada, acendendo-a, aumentando e diminuindo a luminosidade dela e exibindo a luminosidade final da lâmpada. Um exemplo do diagrama da classe *LampadaDimerizada* pode ser visto a seguir.

LampadaDimerizada
- luminosidade
+ Lampada() + apagar(): void + aceder(): void + aumentarLuminosidade(): void + diminuirLuminosidade(): void

Questão 3

Uma empresa possui empregados que são pagos ao fim do mês de acordo com quantidade de horas trabalhadas. Para controlar a sua folha de pagamento, esta empresa resolveu solicitar um software, que possui classe uma classe Empregado com campos e métodos que satisfaçam as seguintes exigências:

- Cada empregado possui uma matrícula, o valor da sua hora de trabalho e a quantidade de horas trabalhadas durante o mês, sendo esses dados informando no momento da criação de um novo empregado (menos a quantidade de horas trabalhadas que, por padrão, deve ser iniciada com 0).
- Cada empregado, ao final do dia, deve registrar o ponto, informando a sua hora de chega e de saída (somente as horas, sem minutos). Neste momento o sistema deve calcular a quantidade de horas trabalhada pelo empregado durante o dia e acrescentá-las ao campo de horas trabalhadas no mês. Faça um método *registrarPonto* que realize está ação.
- Ao final de cada mês, o setor de RH calcula o salário que deve ser pago ao empregado (utilizando suas horas trabalhadas no mês e o valor dela) e zerar a quantidade de horas trabalhadas durante o mês. Faça um método *calcularSalario* que realize está ação. **Atenção: esse método deve calcular o salário do empregado e retornar o valor calculado como a saída do método. Não esqueça que ele também deve reiniciar o total de horas trabalhadas no mês.**

Um exemplo do diagrama da classe Empregado pode ser visto a seguir:

Empregado
- matricula - valorHoraTrabalho - totalHorasMes
+ Empregado(matricula, valorHoraTrabalho) + registrarPonto(horaChegada, horaSaida): void + calcularSalario(): double

Para testar a sua classe, crie uma aplicação com um empregado, registre dois pontos para esse empregado e, por fim, execute 2 vezes, exibindo o resultado, o método *calcularSalario* (Na primeira execução, o sistema deve informar o valor de acordo com as horas trabalhadas e, na segunda, esse valor será zero, pois a horas de trabalho foram zeradas).

Questão 4

Depois de muitas reivindicações, o sindicato dos empregados conseguiu fazer com que a empresa pagasse horas extras para seus funcionários que trabalham mais de 8h por dia. Para isso ser possível, o sistema deve ser adaptado e, durante o registro do ponto, cada hora que exceder as 8 horas diárias deve ser contabilizada como hora extra. Por exemplo, se um

empregado registrar um ponto com a hora inicial 7 e a hora final 17, ele trabalhou 10 horas no dia, sendo 8 horas contadas como horas normais e 2 horas contadas como horas extras.

No momento de calcular o salário, as horas extras devem valer 50% a mais que a hora de trabalho normal. Altere a classe Empregado para que as novas condições sejam satisfeitas (nenhum **MÉTODO** adicional é necessário para satisfazer essas condições). Por fim, para testar a nova classe, crie um empregado, registre dois pontos que excedam as 8h horas diárias, calcule e exiba o salário do empregado no final do mês. Repita a rotina de calcular salário para ter certeza de que as horas de trabalho e horas extras foram zeradas. Um exemplo do diagrama da classe Empregado pode ser visto a seguir.

Empregado
- matricula - valorHoraTrabalho - totalHorasMes - totalHorasExtrasMes
+ Empregado(matricula, valorHoraTrabalho) + registrarPonto(horaChegada, horaSaida): void + calcularSalario(): double

Questão 4

Na empresa em questão, é comum a prática de venda/transferência de horas entre empregados. Nela, empregados que possuem excesso de horas trabalhadas vendem/transferem suas horas para empregados que não bateram a meta mínima mensal. Por exemplo, um empregado A que possui 210 horas trabalhadas pode vender/transferir suas horas para um empregado B que possui somente 100 horas. Caso essa transferência seja de 10 horas, o empregado A ficará com o total de 200 horas e o empregado B com o total de 110 horas.

Sabendo disso, altere a classe Empregado de forma que a nova exigência possa ser atendida. Para isso, você deve acrescentar o método *transferirHoras* que recebe o total de horas a serem transferidas e o destino das horas (empregado que receberá as horas).

Para facilitar essa operação alguns pontos devem ser levados em consideração:

- O sistema não deve se preocupar com a quantidade de horas mínimas exigidas pela empresa. Estas foram comentadas na questão apenas para uma contextualização do problema.
- Devido à observação anterior, não será necessário verificar se os empregados envolvidos na transferência de horas irão ficar com a quantidade de horas mínimas ou não, podendo inclusive o empregado cedente das horas ficar até com horas negativas. O que poderá gerar um salário negativo também.
- Serão transferidas somente horas normais de trabalho (não horas extras) e em momento algum a transferência deve gerar horas extras para o beneficiado.
- Para ser possível essa transferência, a classe Empregado deve possuir mais 2 métodos: o método *adicionarHoras*, que recebe uma quantidade de horas e acrescenta ao total

de horas mensais do empregado; e o método *retirarHoras* que retira horas da quantidade total de horas mensal do empregado.

Um exemplo do novo diagrama da classe *Empregado* pode ser visto a seguir. Nele, observe que o método de transferir horas recebe a quantidade de horas e um objeto do tipo Empregado. Desse modo, ele deve retirar horas do objeto empregado que executa o método e acrescentar horas no objeto empregado passado com argumento para o método.

Empregado
- matricula - valorHoraTrabalho - totalHorasMes - totalHorasExtrasMes
+ Empregado(matricula: int, valorHoraTrabalho: double) + registrarPonto(horaChegada: int, horaSaida: int): void + calcularSalario(): double + adicionarHoras(horas: int): void + retirarHoras(horas: int): void + transferirHoras(horas: int, empregado: Empregado):void

Para testa sua classe, crie 2 empregados, registre os pontos dos 2 empregados e transfira um valor pequeno de horas do emprega com a maior quantidade de horas para o com menor. Por fim, exiba o salário do 2 empregados.