

IFPB – Campus Campina Grande

Curso de Bacharelado em Engenharia de Computação

Disciplina: **Programação Orientada a Objetos (Teoria e prática)** – Semestre 2021.1

Professor: Leandro de Almeida Melo (leandro.melo@ifpb.edu.br)

Prova Polimorfismo

1. (30) Escreva uma classe template Fila para armazenamento de elementos numéricos(int, float, double...), ou seja, na sua estrutura o primeiro que chegar também será o primeiro a sair (FIFO) – insere no fim e remove do início. A classe genérica Fila recebe como parâmetros o número (tamanho máximo da fila) e o tipo de elementos de um vector(da classe STL). Essa classe oferece operações de inserção, remoção e exibição de todos os elementos. Além disso, a classe Fila possui um sistema de propriedade onde todos os elementos (números) acima de 60 são inseridos no início da fila, na ordem que eles chegaram. Teste o seu programa criando uma instância dessa classe como int e outra como float.

2. (70) Uma empresa de CG te contratou para fazer um programa em C++ para gerenciar os seus funcionários e a folha de pagamento da empresa. A empresa contrata os funcionários através de um regime de trabalho intermitente, ou seja, os funcionários ganham por hora de trabalho mais um adicional dependendo do seu cargo. Os profissionais da empresa são:

- * engenheiros de software
- * Gerente
- * designer

Em todos os funcionários é necessário guardar a matrícula, o valor da sua hora de trabalho (HT) e o tempo de experiência na empresa.

Para o Engenheiro de software (ES) é necessário guardar o seu nível, um texto que pode ser, Júnior, Pleno 1, Pleno 2 e o Sênior. O nível do ES é definido pelo seu tempo de empresa, começando por Júnior e aumentando o nível a cada 3 anos. Já para o Gerente é necessário guardar o número de equipes (NE) que ele gerencia.

Os funcionários também guardam o seu histórico de pagamento mensal (vector ou list). Um pagamento é definida por um mês(int), ano(int), um valor(float) e descrição(string).

As operações possíveis para os tipos de funcionários são: realizarPagamento, getSalario e toString. Essas operações devem ser definidas numa classe abstrata pura denominada Funcionário.

O método realizarPagamento recebe como parâmetro a carga horária (CR - int), mês(int), ano(int) e uma descrição (string - opcional, por padrão o seu valor é “Salário”). Esse método é sem retorno e ele deve imprimir o valor do salario do funcionário. Para calcular o salário do gerente, deve-se: CRxHTxNE. Já o para o engenheiro de software, deve-se: CRxHT + um valor adicional que varia de acordo com o seu nível, o nível Júnior vale R\$ 500 e para os demais níveis esse valor é aumentado em 1000 para cada nível. Por fim, o

designer ganha de acordo com a sua CR, sem nenhum adicional no seu salário. Lembre-se no final desse método, adicionar esse pagamento no histórico de pagamento mensal de cada funcionário.

O cálculo do salário realizado no `getSalario` é semelhante ao `realizarPagamento`, no entanto é feito apenas uma consulta e não o pagamento. Assim, o método `getSalario` recebe apenas CH e retorna o valor que esse funcionário vai receber.

O método `toString` deve retornar as informações de cada funcionário, incluindo o seu histórico de pagamento.

Faça um construtor em todas as classes e inicialize suas variáveis usando esses construtores. Construa métodos getters e setters conforma a sua necessidade.

Por fim, construa uma classe chamada `gerenteFuncionarios`, essa classe guarda um vector (**vector -STL**) de funcionários e cada funcionário pode ser uma instância de engenheiro de software, gerente ou designer. Nessa classe, deve ter os seguintes métodos:

- `void AddFuncionario(Funcionario &f)`: Adiciona um novo funcionário no sistema/ lista.
- `void removerFuncionario(int matricula)`: remove um funcionário do sistema/ lista ou informa que a matrícula não existe.
- `void pagar(int matricula, int CR, int mês, int ano, string descrição= "Salário")`: Busca o funcionário com essa matrícula e chama o método `realizarpagamento`.
- `void consultarSalario(int matricula, int CH)`: Busca o funcionário com essa matrícula e chama o método `getSalario`.
- `void buscarFuncionario(int matricula)`: Imprime o `toString` do funcionário ou informa que a matrícula não existe.
- `void imprimirFuncionarios()`: Imprime o nome e o tempo de empresa de todos os funcionários.

Implemente a hierarquia de classes dessa questão explorando polimorfismo. Em seguida, faça um programa principal (main – driver), crie uma instância de `gerenteFuncionarios` e teste os métodos dessa classe.