

Universidade Federal de Alagoas
Instituto de Computação
Programação 2

Relatório do Projeto Folha de Pagamento

Carlos Aleandro Lopes Pessoa

Introdução

O projeto consiste em ter uma classe principal composta por uma série de funções que vão de criar um empregado a gerar uma folha de pagamento com detalhamentos acerca de todos os funcionários que possuem pagamento semanal, quinzenal e mensal. A estrutura do código acaba sendo simples, composta por uma classe principal que se chama Facade, um padrão de projeto que fornece uma interface unificada para um conjunto de interfaces em um subsistema. Fora a facade, fo

Principais Componentes e suas Principais Interações

Os principais componentes consistem na classe Facade, com suas funções que visam manipular outras classes, tais como Empregado, Sindicalizado, Comissionado e Horista. Portanto, Facade não mais do que seria uma fachada para os serviços relacionados a empregados em um sistema de Folha de Pagamento.

Então, partindo para os componentes de Empregados. Foi feita a manipulação da criação dos mesmos pela Facade, onde pode-se remover, obter o empregado por nome, obter atributos, verificar se parâmetros estão corretos, entre outras ações.

Essa classe empregado, é usada para herdar alguns componentes que nela se encontram para a construção e manipulação nas outras classes, Sindicalizado, Comissionado e Horista.

Padrões de Projeto

Como já mencionado, Facade trata-se de um padrão de projeto de tipo fachada, logo, os serviços nela servem como principal interface do sistema. Com isso, temos principais funções na Facade:

- Roda folha: que seria justamente para imprimir a folha de pagamento em um arquivo txt;
- Total Folha: que foi manipulada para que pegasse o total dos pagamentos semanal, quinzenal e mensal.
- Temos funções que lidam a taxa de serviço para empregados sindicalizados, assim como lançamento de horas para quem é horista.
- Há também a possibilidade de alterar componentes de empregados, zerar o sistema dos mesmos e encerrar o programa.

Logo, a fachada seria mais para fornecer uma interface simplificada para um conjunto de funcionalidades mais complexas, bem como o uso de herança e polimorfismo com as classes de empregados.

Tratando da questão de herança e polimorfismo, vale ressaltar que nesse projeto consiste num Padrão de Hierarquia de Classes(ou Herança). Como trata-se de um dos princípios de POO(Programação Orientada a Objetos), temos uma classe que é usada como base para outras, no caso é a classe 'Empregado', definindo os comportamentos e características comuns a todas as subclasses(Sindicalizado, Comissionado e Horista). O que acaba sendo usado dessa herança, são variáveis simples e a possibilidade de implementar uma flexibilidade do uso do Hash Map de lista ligada, que associa chaves e valores de um tipo declarado e mantém salvos na memória de forma ordenada ao que foi inserido.

Conclusão e Aplicação no Projeto

O projeto consistiu em lidar com a manipulação de empregados, dos quais mais tarde poderiam virar comissionados, horistas e sindicalizados, portanto, visando essa possibilidade, o uso do Padrão de Projeto de Hierarquia de Classes foi indispensável para a manipulação dos tipos de empregado, já que todos tinham dados básicos e que poderiam ser armazenados na estrutura Hash Map de tipo Empregado.

Logo, Facade poderia ser construída de maneira simples e como desejada, cumprindo com as principais funções que eram apresentadas nos testes do programa. Como o lançamento de horas para o tipo horista, que tratava de converter o empregado de tipo classe 'Empregado' para 'Horista', quando não, 'Comissionado' quando se tratava de lançar vendas para este tipo de empregado.

Nota-se que ainda existem maneiras de deixar o código visivelmente melhor, porém, como Facade é a base, um Padrão de Projeto que visa simplificar a interface, se conseguiu cumprir com esse propósito, deixando as principais funcionalidades até o momento trabalhado funcionando conforme o desejado. Então, visando explorar e deixar claro essas principais funcionalidades, foi criada uma classe de tipo interface para melhor visualização das funções e cumprir com o que se era solicitado.