



Universidade Federal de Uberlândia
Faculdade de Engenharia Elétrica
Curso de Graduação em Engenharia da Computação

Disciplina: Programação Orientada a Objetos

2º Trabalho – 20/12/2020, até às 23:59

Prof.: Edgard Lamounier

Projeto 01

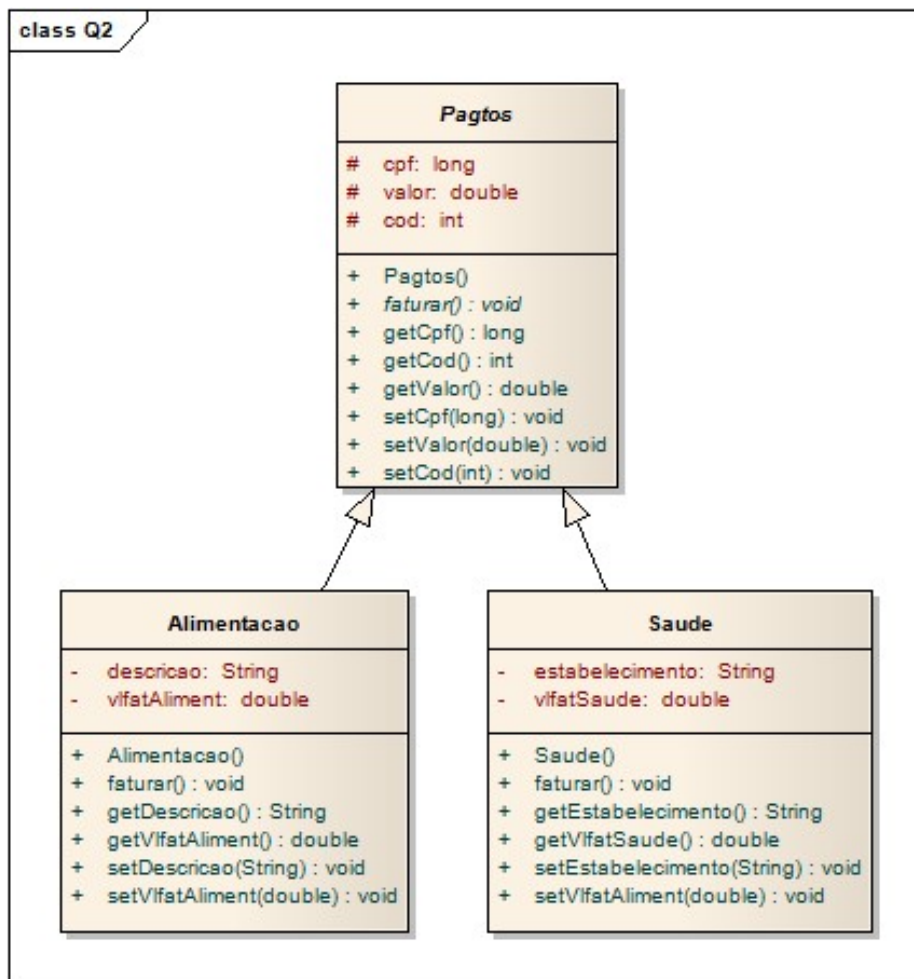
Necessita-se desenvolver um sistema para Controle de Pagamentos. As informações básicas a serem utilizadas serão: **cpf**, **valor** e **codigo**.

Todo pagamento deverá ser faturado e ao efetuar o lançamento de cada pagamento, deve-se respeitar as seguintes regras de negócio:

R1) Quando se tratar de um Pagamento na área de **Saúde**, o valor da fatura deverá ter um acréscimo de 12%, além de ser informado o nome do estabelecimento.

R2) Quando se tratar de um Pagamento na área **Alimentícia**, o valor da fatura deverá ter um acréscimo de 5%, além de se informar a descrição do item adquirido.

Para esta proposta, segue um exemplo do Diagrama UML de Classes e os respectivos métodos. Crie um programa em C# com interface gráfica, de maneira que seja possível faturar até 06 pagamentos (armazenados em um array ou List<>), sendo 03 para Alimentação e 03 para Saúde, registrados de forma aleatória. Exiba um histórico dos pagamentos realizados.



Projeto 02



Uma pequena empresa de logística pretende implementar um porto seco para armazenar containers e precisa de um programa simples para gerenciar seu porto.

Dado o tamanho da empresa, essa possui apenas 4 locais onde pode empilhar os containers e seu equipamento permite empilhar no máximo 3 containers.

Por política da empresa, um novo container deve ser colocado no local com a **pilha** mais baixa disponível (se mais de um local tiver a mesma quantidade de containers, tanto faz em que pilha será adicionado).

Cada container é adicionado a partir de um código informado e o sistema não pode permitir códigos repetidos. Para remoção também é preciso informar o código do container, mas o mesmo somente será removido se estiver no **topo da pilha**.

Desenvolver um programa em Java para atender essa empresa e que mostre as seguintes mensagens:

- **Código inválido!:** quando for fornecido um valor já existe no momento de inserção e quando for fornecido um valor não existente no momento de remoção.
- **Impossível empilhar!:** quando não for mais possível adicionar containers nas pilhas.
- **Impossível desempilhar!:** quando não for possível remover um container, ou por não existir mais containers ou por ele não estar no topo da pilha.

Projeto 03

Usando uma classe Fila, escreva um programa C# que atenda ao seguinte menu:

- [1] Insira um inteiro na Fila;
- [2] Remova um inteiro da Fila;
- [3] Insira um inteiro APÓS um outro a ser escolhido
- [4] Imprima a Fila;
- [5] SAIR