

Universidade Católica de Pernambuco

Professor: Augusto César Oliveira

Disciplina: Programação III / POO

Aluno(a): _____ data: ____/____/____

Aula 10 - Herança e polimorfismo

1. Considere que você foi contratado para desenvolver um programa que auxilia a organização de uma empresa de logística. O sistema simula uma empilhadeira de caixas. As caixas podem ser **fortes** ou **frágeis**. Caso sejam empilhadas, a caixa frágil não deve suportar uma outra caixa que seja mais pesada que ela. Isso pode vir a danificar o produto que está dentro da caixa frágil. Por exemplo, uma caixa frágil que suporta até 20 kg, não pode ter sobre ela uma outra caixa de 22 kg. Uma vez que 22 kg ultrapassa em 2 kg o peso suportado pela caixa frágil. Além disso, o programa deve garantir que a pilha de caixas não ultrapasse uma altura máxima pré-definida, a fim de evitar que a pilha tombe e as danifique. Admita que as caixas todas têm a mesma dimensão na base.



- a. Crie a classe ``Caixa``, que possui como atributos privados um ``identificador``, um ``conteúdo``, um ``peso (kg)`` e uma ``altura (cm)``. Crie o construtor e os métodos de acesso (gets e sets) para cada um dos atributos. O peso e a altura devem ser números reais maiores que zero. Crie o método ``exibirEtiqueta``, para exibir todos os atributos da caixa.
- b. Crie a classe ``CaixaFrágil``, que tem como atributo privado o ``peso máximo suportado sobre ela``. Crie o construtor e os métodos de acesso. O peso máximo suportado deve ser um número real maior que zero. Utilize polimorfismo para exibir também o peso máximo suportado pela caixa frágil na etiqueta.
- c. Crie a classe ``PilhaDeCaixas``, que será utilizada para organizar as caixas uma sobre a outra. Essa classe tem como atributo a ``altura máxima permitida`` para esta pilha. Essa altura máxima deve ser recebida como parâmetro no construtor da classe e os seus respectivos métodos de acesso devem ser criados. Essa classe possui também como atributo a própria pilha, inicialmente vazia. Ela pode ser implementada utilizando as estruturas de dados prontas da linguagem.
- d. Na classe ``PilhaDeCaixas``, crie o método `empilhar`, que recebe uma caixa como parâmetro e retorna um booleano (**true**, se empilhou, e **false**, caso contrário). Este método deve

empilhar a caixa apenas se a altura máxima não for atingida. Além disso, o peso máximo suportado por alguma caixa frágil que já está na pilha não pode ser ultrapassado.

- e. Na classe `PilhaDeCaixas`, crie o método `desempilhar`, que remove a caixa que está no topo e a retorna. Exiba uma mensagem caso a pilha esteja vazia e não haja mais caixas para desempilhar.
- f. Na classe `PilhaDeCaixas`, crie um método que retorna a altura atual da pilha.
- g. Na classe `PilhaDeCaixas`, crie um método que retorna o peso total atual da pilha.
- h. Na classe `PilhaDeCaixas`, crie um método para exibir os dados de todas as caixas empilhadas, na ordem de cima para baixo, bem como a altura atingida pela pilha e o seu peso total.
- i. Crie exemplos de teste para avaliar se as classes estão funcionando de acordo com a descrição.

2. Com base na questão anterior, responda:

- a. Identifique como foram utilizados os conceitos de herança e encapsulamento de orientação a objetos no projeto da primeira questão. Justifique sua resposta com as suas palavras.
- b. Na letra 1-b, foi pedido para aplicar polimorfismo. Explique com suas palavras o que é polimorfismo e como isso foi aplicado no projeto da primeira questão.