

Maurício Carvalho Mucci

199505675

(c) Interface é uma lista de métodos que define o que pode ser feito com um objeto que implementa ela. Não existe construtores dentro de uma interface, pois ela não instancia objetos.

(d) Código 1:

1. Não utilizar o `this` no construtor da classe produto.
2. Uma classe não-abstrata, não pode possuir métodos abstratos.

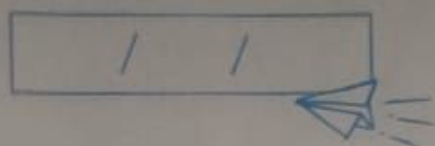
Código 2:

1. Foi implementado somente a função `reseta()` da interface `Resetável`, é necessário adicionar a função `reseta(int origem)` da interface `Modificável`.
2. Função `modifica` possui parâmetro: `int comando`, ou seja, o nome do parâmetro não pode ser alterado para `"tam"`.



MATT GROENING

tilibra



Maurício Carvalho Mucci

19404675



1.(a) Alternativa: [d]

[d] A grande vantagem da herança é a reutilização de código. A palavra-chave que permite isso em Java se chama "super". Super permite acesso aos construtores e aos métodos da classe "mãe".

(b) Alternativa: [c]

[c] Esse argumento é falso, pois caso uma classe abstrata for criada, mas nunca for estendida, ela será ignorada.

2. (a) Uma classe abstrata pode ter métodos não-abstratos, caso, por exemplo, não seja necessário diferenciar o corpo do método nas classes "filhas". Dessa forma ele se torna padrão em todas as subclasses. Logo, só serão abstratos caso seja necessário diferenciá-los em suas subclasses. Não podemos, pois o construtor instancia um objeto, e o abstract não permite a criação de objetos.

(b) Polimorfismo é a capacidade do objeto ser referenciado de diversas maneiras. Exemplos:

Funcionario f1

Gerente g = new Gerente(f1);

Gerente a = new Gerente(f1);

Funcionario b = a;