Pontifícia Universidade Católica do Paraná - PUCPR Programação Orientada a Objetos

RA1: Codificar classes com membros estáticos em conformidade com o princípio de encapsulamento.

ID1: Codifica classes com membros estáticos definindo atributos de tipos básicos e métodos com parâmetros

ID2: Codifica métodos por meio de variáveis locais, comandos imperativos e invocação de métodos estáticos.

POO - 1

Escreva um código Java composto pelas seguintes classes:

1. Classe **Semaforo**

- a. Atributo: cor, representada com um valor inteiro ==>
 - 0 (verde),
 - 1 (amarelo) ou
 - 2 (vermelho), sendo o valor inicial igual a 2.

b. Métodos:

i.MudaProximaCor:

• sem parâmetros e sem retorno ==> muda a cor atual do semáforo para a próxima cor (se estiver verde, fica amarelo; se estiver amarelo, fica vermelho; se estiver vermelho, fica verde).

ii.MudaCorFixa:

• com parâmetro inteiro, sem retorno ==> muda a cor atual do semáforo para a cor fornecida como parâmetro.

iii.MostraCor:

 sem parâmetro, com retorno inteiro ==> retorna a cor atual do semáforo.

2. Classe Controle

- a. Contém o método main com os seguintes passos:
 - i.Imprime a cor do semáforo
 - ii.Repete 10 vezes as seguintes operações:
 - Muda a cor atual do semáforo para a próxima cor
 - Imprime a cor do semáforo
 - iii. Fixa a cor do semáforo em amarelo
 - iv.Imprime a cor do semáforo

RA1: Codificar classes com membros estáticos em conformidade com o princípio de encapsulamento.

ID3: Aplica proteção de membros (encapsulamento).

ID4: Codifica invariante, pré-condição e pós-condição.

POO - 2

Acrescente as seguintes classes ao projeto estruturado nos pacotes **Matematica** e **Especialista** que contêm, respectivamente, as classes **Circulo** e **Arquiteto**:

1. Classe Retangulo

- a. Contida no pacote Matematica.
- b. Contém os seguintes atributos privados do tipo float:
 - ladoH e
 - ladoV.
- c. Contém os seguintes métodos públicos:
 - i.**definir_lados**: define **ladoH** e **ladoV** de acordo com dois valores fornecidos como parâmetros.
 - ii. area: retorna a área de um retângulo cujas medidas dos lados são dadas por ladoH e ladoV.
 - iii. *perimetro*: retorna o perímetro de um retângulo cujas medidas dos lados são dadas por *ladoH* e *ladoV*.
- d. Possui a seguinte invariante: as medidas dos lados são positivas.

2. Classe EngenheiroCivil

- a. Contida no pacote Especialista.
- b. Contém o método *exibir dados pessoais* de forma análoga à classe Arquiteto.
- c. Contém o método main com os seguintes passos:
 - i.Chama o método exibir dados pessoais.
 - ii.Chama o método *definir_lados* da classe Retangulo com os valores 4.7 e 8.2 como parâmetros.
 - iii.Chama os métodos *area* e *perimetro* da classe Retangulo e imprime os resultados dessas chamadas.

RA1: Codificar programas baseados em interação entre objetos instanciados a partir de classes associadas, em conformidade com o princípio de encapsulamento, com a validação do seu comportamento, de forma sistematizada.

POO - 3

Implemente as seguintes classes:

- 1. Classe ContaCorrente
 - a. Contém os seguintes atributos privados:
 - i.**numero**, do tipo int
 - ii.saldo, do tipo double
 - iii.cliente, do tipo classe Cliente
 - b. Contém os seguintes métodos públicos:
 - i.construtor
 - ii. depositar um valor na conta
 - iii. *retirar* um valor da conta, desde que o saldo não fique negativo; deve retornar o valor efetivamente retirado.
 - iv. verificar o saldo da conta
 - v.imprimir todos os dados da conta corrente e do respectivo cliente
 - c. Possui a seguinte invariante: o saldo nunca é negativo.

2. Classe Cliente

- a. Contém os seguintes atributos privados:
 - i.nome, do tipo String
 - ii.contaCorrente, do tipo classe ContaCorrente
- b. Contém os seguintes métodos públicos:
 - i.construtor (que recebe os parâmetros nome para atribuir ao atributo da classe nome e o número da conta, pois também instancia o respectivo objeto da classe ContaCorrente, no atributo contaCorrente).
 - ii. *operar*, que chama ao menos uma vez os métodos para depositar, retirar, verificar saldo e imprimir os dados da respectiva conta corrente

3. Classe Banco

- a. Contém o método *main*, que realiza os seguintes passos:
 - i.cria três instâncias de **Cliente** (e, consequentemente, as respectivas instâncias de **ContaCorrente**)
 - ii.chama o método *operar* para cada instância de **Cliente**