



INSTITUTO FEDERAL DA PARAÍBA
CAMPUS CAMPINA GRANDE
BACHARELADO EM ENGENHARIA DA COMPUTAÇÃO
DISCIPLINA DE POO e LAB. POO
PROF. VICTOR ANDRÉ PINHO DE OLIVEIRA

Atividade Unidade II - 5 - Polimorfismo

Lista Única

Instruções

Responda às questões abaixo. Pode usar este próprio documento. Questões práticas devem ser anexadas separadamente.

As primeiras 8 questões valem 1. As questões 9 a 11 valem 3.

Questões

1. O que é e como se dá o polimorfismo?

R: No polimorfismo podemos processar objetos de classes da mesma hierarquia como se todos fossem da classe principal.

2. Quais os mecanismos básicos para o polimorfismo em C++?

R: São 4: Herança, sobrescrita, métodos virtuais e ponteiros ou referências.

3. Para que servem métodos virtuais?

R: O método virtual irá fazer com que o ponteiro ou referência de uma classe básica para um objeto de classe derivada possa invocar as funcionalidades da classe derivada. Quem irá determinar qual versão do método invocar será o método virtual.

4. Qual a diferença entre método virtual e método virtual puro?

R: Virtual puro: virtual int falar(string) = 0

Virtual: virtual int falar() const;

5. Quando empregar um método virtual puro?

R: Na intenção de criar uma classe abstrata, onde terá pelo menos um método virtual puro.

6. O que é uma classe abstrata? Qual a diferença entre classe abstrata e concreta?

R: É uma classe com um nível de abstração elevado, não podendo instanciar objetos, são classes criadas apenas para serem herdadas.

7. Como podemos criar uma classe abstrata em C++?

R: Deverá ter um método virtual puro.

8. Quando usar destrutores virtuais?

R: É aconselhável utilizar destrutores virtuais, pois tudo deixa de ser feito em tempo de compilação e passa a ser feito em tempo de execução. Se existir a possibilidade de objetos de classes derivadas serem destruídos a partir de ponteiros de classe básica, então o destrutor da classe básica deve ser declarado como virtual.

9. (*Hierarquia Forma*) Aproveite o código Aula11Ex4 desenvolvido em “sala” e faça o que se pede.

- a. Na classe abstrata FormaBidimensional, acrescente os métodos virtuais puros calcularArea e calcularPerimetro.
- b. Em cada uma das classes derivadas, adicione atributos específicos para que se possa calcular a area e o perímetro da forma (circulo: raio; quadrado: lado; e triangulo: base e altura).
- c. Ajuste os construtores das classes derivadas para receber os devidos argumentos.
- d. Implemente (sobrescreva) os métodos virtuais para que cada forma mostre sua própria área e perímetro.
- e. Mostre que suas classes estão funcionando adequadamente.

10. (*Hierarquia Conta*) Um banco trabalha com três tipos de contas: (a) conta corrente comum; (b) conta corrente com limite; e (c) conta poupança. Em todos os casos é necessário guardar o número da conta, o nome do correntista e o saldo.

Para a conta poupança é necessário guardar o dia do aniversário da conta (quando são creditados os juros).

Já para a conta com limite é necessário guardar o valor do limite.

As contas também armazenam uma lista de transações. Uma transação é definida por uma data, valor da transação e descrição. Se o valor for negativo, a transação é considerada um débito (crédito caso contrário).

As operações possíveis são: depósito, retirada e impressão de extrato. Essas operações devem ser definidas numa classe abstrata denominada Conta.

A operação de depósito é igual nos três tipos de conta.

A retirada só é diferente na conta com limite, pois esta admite que o saldo fique negativo até o limite estabelecido.

Finalmente o extrato é diferente para as três: (a) na conta comum exibe o número da conta, nome do cliente, transações e o saldo; (b) na conta limite imprime também o valor do limite; e (c) na conta poupança imprime também o dia do aniversário.

Implemente a hierarquia de classes das contas explorando polimorfismo.

Faça um programa em C++ que permita ao usuário fazer depósitos, retiradas e verificação de extrato nas suas contas a partir do número da conta. Utilize um array (de ponteiros para Conta) para armazenar todos os tipos de contas.

11. (*Agenda de Contatos*) Crie uma agenda de contatos. Nessa agenda, contatos podem ser adicionados, removidos ou pesquisados por nome ou por CPF/CNPJ. Deve ser possível, também, visualizar todos os contatos da agenda. Dois tipos de contatos podem fazer parte da agenda: pessoa física e pessoa jurídica. A pessoa física tem os seguintes atributos: CPF, nome, endereço, data de nascimento, email, estado civil, outros que você achar necessários. A pessoa jurídica possui: CNPJ, nome, endereço, email, inscrição estadual, razão social, outros que você achar necessários. Com base no descrito acima, apresente uma solução Orientada a Objetos em C++ usando os conceitos de Herança e Polimorfismo (e outros que julgar necessários).