UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE - UFRN

Instituto Metrópole Digital

IMD0040 - Linguagem de Programação 2

Atividade - Classe Abstrata e Interface

Professor: Emerson Alencar

Exercício 01:

Crie uma interface AreaCalculavel com um método calcularArea() e crie classes de figuras geométricas que implementam este método (como quadrado, circulo e retângulo). Depois crie uma classe com um método main() para exercitar as chamadas aos métodos que calculam a área, utilizando polimorfismo. As classes de figuras geométricas devem conter construtor, que receba como parâmetro os valores para os cálculos da área.
Observação: Na saída do console o resultado da área do circulo deve ter apenas duas casas decimais, para isso formate a String.

Exercício 02:

Crie uma classe ContaBancaria, que representa uma conta bancária genérica e não pode ser instanciada. Esta classe deve ter um atributo saldo (visível apenas para ela e para as suas subclasses) e os métodos depositar(double), sacar(double) e transferir(double, ContaBancaria). Estes métodos devem depositar um valor na conta, sacar um valor da conta e transferir um valor da conta de origem para uma conta de destino, respectivamente.

Além destes, ContaBancaria deve ter um método calcularSaldo(). Este método possui a regra do cálculo do saldo final (que pode ser diferente do saldo armazenado no atributo saldo) e deve ser obrigatoriamente implementado pelas subclasses de ContaBancaria, pois cada classe possui suas próprias regras de cálculo.

Crie duas subclasses de ContaBancaria: ContaCorrente e ContaInvestimento. Cada uma deverá implementar suas regras para calcular o saldo (método calcularSaldo()). No caso de ContaCorrente, o saldo final é o saldo atual subtraído de 10%, referente a impostos que devem ser pagos. Já para a ContaInvestimento, o saldo final é o saldo atual acrescido de 5%, referente aos rendimentos do dinheiro investido.

Crie uma aplicação que instancia uma conta corrente e uma conta investimento (utilizando polimorfismo) e executa as operações de depósito, saque, transferência e cálculo de saldo. Verifique se os resultados obtidos são consistentes com a proposta do modelo e com as regras de cálculo estabelecidas.

Exercício 03:

O Java possui uma interface chamada Cloneable, que pode ser implementada por classes que são capazes de gerar cópias de objetos. Esta interface não possui métodos, mas classes que a implementam devem sobrescrever o método clone(), definido na classe

Object. Dentro deste método é implementada a lógica para criar um novo objeto com base no objeto original.

Com base nisto, crie uma classe Porta que suporta a criação de novos objetos (cópia). Ela deve ter os atributos altura (double), largura (double) e aberta (boolean). Também deve possuir os métodos abrir(), fechar() e os métodos getters correspondentes aos atributos.

A classe deve ter um construtor que recebe a altura e largura da porta. Inicialmente a porta está fechada.

Como uma porta pode criar outras cópias dela mesma, você deve sobrescrever o método clone(), o qual deve criar um novo objeto com os valores dos atributos copiados e retorná-lo.

Dica: O método clone() lança uma exceção (CloneNotSupportedException). Não é preciso se preocupar com ela neste momento, então declare o método main() da sua aplicação da seguinte forma:

```
public static void main(String[] args) throws Exception {


}
```

Crie uma aplicação que cria um objeto porta, executa o método para abrir a porta. Crie um clone dessa porta e escreva no terminal os estados dos dois objetos comparando-os.