

## Trabalho 2 – Programação Orientada a Objetos

### PARTE 1

#### QUADRILATEROS

1. Crie uma classe abstrata que represente um quadrilátero e receba como parâmetros do construtor os quatro valores referentes a cada lado.
2. Estenda a classe criada no exercício 1, a subclasse deve representar um quadrado e portanto receber como parâmetro um único valor referente aos seus lados.
3. Estenda novamente a classe criada no exercício 1, a subclasse deve representar um retângulo e portanto deve receber como parâmetros dois valores diferentes.
4. Crie uma interface denominada *FiguraGeometrica*, adicione os métodos *calcular área* e *calcular perímetro*
5. Modifique a classe abstrata criada no exercício 1 de forma que ela implemente a interface *FiguraGeométrica*.
6. Implemente os métodos definidos na interface *FiguraGeometrica* nas classes *Quadrado* e *Retangulo*.
7. Modifique o construtor da classe *Quadrado* de forma que caso seja recebido um valor igual a zero ou negativo seja “jogada” uma exceção do tipo *IllegalArgumentException* com o seguinte texto: “Valor inválido, o valor esperado é maior que 0 (zero)”.
8. Modifique o construtor da classe *Retangulo* de forma que caso seja recebido como parâmetro um valor igual a zero ou negativo ou ambos os valores positivos idênticos porém idênticos seja “jogada” duas exceções do tipo *IllegalArgumentException* com os seguintes textos respectivamente: “Valor inválido, os valores esperados são maior que 0 (zero)” e “Valor inválido, modifique um dos valores a fim de torná-los diferentes”.
9. Crie um programa que solicite a entrada de 2 parâmetros, crie um *Quadrado* e imprima a área e o perímetro.
10. Modifique o programa desenvolvido no exercício 9. Adicione o tratamento de exceções e caso aconteça uma exceção imprima a *stacktrace*. Execute o programa forçando uma exceção e observe a *stacktrace*.
11. Modifique novamente o programa desenvolvido no exercício 9 de forma que caso sejam passados valores inválidos ele trate a exceção e exiba a mensagem em um diálogo (*JOptionPane*).

### PARTE 2

#### CONTA BANCARIA

12. Crie uma classe abstrata denominada *conta corrente*. Adicione os métodos *sacar*, *depositar* e *obter saldo*, adicione o atributo *saldo total*.
13. Estenda a classe desenvolvida no exercício anterior. Crie a classe *conta corrente especial* que contenha um atributo que represente um limite extra da conta corrente convencional de forma que:
  1. O saque debita primeiro o saldo total, na sequência deve-se debitar o limite do cheque especial;
  2. O depósito primeiro deve creditar o limite do cheque especial, até cobrir, e somente após o saldo total.
14. Crie uma exceção que represente o estouro da conta corrente, modifique a classe desenvolvida no exercício 13 de forma que caso seja extrapolado o limite do extra esta exceção seja lançada.
15. Crie um programa que, utilizando as classes desenvolvidas nos exercícios anteriores (12, 13 e 14), efetue alguns saques e depósitos, tanto sobre o limite extra quanto sobre saldo total, e ainda faça um saque que extrapole todo o valor disponível (limite extra + saldo total).

