

Exercícios Gerais – Semana 1.3

1. Escreva uma classe para representar uma lâmpada. Desenvolva métodos para ligar e desligar a lâmpada.

2. Crie uma classe para representar uma conta corrente que possui um número, um saldo, um status que informa se ela é especial ou não, e um limite. Desenvolva métodos para realizar saque (verificando se o cliente pode realizar saques), depositar dinheiro, consultar saldo e verificar se o cliente está usando cheque especial ou não. Desenvolva um programa para testar essa classe.

3. Escreva uma classe para representar um Aluno. Adicione atributos relacionados às características de um Aluno, como nome, matrícula, curso que está matriculado, nome de 3 disciplinas que está cursando e as notas dessas 3 disciplinas. Desenvolva um método para verificar se o aluno está aprovado (nota maior ou igual a 7) em uma determinada disciplina. Escreva um programa para testar essa classe, que pede as informações do aluno ao usuário e ao final informa o nome das disciplinas, mostra as notas e mostra se o aluno foi aprovado ou não.

4. Reescreva o exercício 6 da aula 20 (Jogo da Velha). Desenvolva uma classe para representar o Jogo da Velha. Desenvolva uma classe para testar o Jogo.

5. Escreva uma classe chamada Contador, que tem um atributo estático que é incrementado sempre que a classe for instanciada. Crie métodos para zerar, incrementar e retornar o valor do contador. Desenvolva um programa para testar essa classe.

6. Escreva uma classe Calculadora que tenha os seguintes métodos: somar, subtrair, multiplicar, dividir (dois números), elevar à potência n. Desenvolva um programa para testar essa classe.

7. Escreva um método para calcular fatorial de um número na classe Calculadora do exercício anterior.

8. Escreva a classe ConversaoDeUnidadesDeArea com métodos estáticos para conversão das unidades de área segundo a lista abaixo: 1 metro quadrado = 10.76 pés quadrados; 1 pé quadrado = 929 centímetros quadrados; 1 milha quadrada = 640 acres; 1 acre = 43.560 pés quadrados.

9. Escreva a classe ConversaoDeUnidadesDeVolume com métodos estáticos para conversão das unidades de volume segundo a lista abaixo: 1 litro = 1000 centímetros cúbicos; 1 metro cúbico = 1000 litros; 1 metro cúbico = 35.32 pés cúbicos; 1 galão americano = 231 polegadas cúbicas; 1 galão americano = 3.785 litros.

10. Escreva a classe `ConversaoDeUnidadesDeTempo` com métodos estáticos para conversão aproximada das unidades de velocidade segundo a lista abaixo: 1 minuto = 60 segundos; 1 hora = 60 minutos; 1 dia = 24 horas; 1 semana = 7 dias; 1 mês = 30 dias; 1 ano = 365.25 dias.

11. Escreva um método recursivo e estático que calcule e retorne o N-ésimo termo da sequência Fibonacci. Alguns números desta sequência são: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89...

12. Escreva um método recursivo e estático que receba um número inteiro positivo N e calcule o somatório dos números de 1 a N.

13. Escreva uma classe `Agenda`, que contém vários contatos do tipo `Contato`. Cada contato possui nome, telefone e email. A `Agenda` também possui um nome. Crie um programa teste que peça para o usuário entrar com o nome da `Agenda` e em seguida 3 contatos. Crie métodos que retornem uma `String` com a informação de cada contato e também de todos os contatos da agenda.

14. Escreva uma classe para representar um `Curso`, que tem nome e horário. Cada curso tem um `Professor`, que possui nome, departamento e email. Cada `Curso` também pode ter vários alunos (tipo `Aluno`). Cada `Aluno` tem nome, matrícula e 4 notas. Escreva um programa teste que crie um `Curso` com 5 alunos, e que peça para o usuário entrar com as 4 notas de cada aluno. Ao final, imprima a média de cada aluno, se o mesmo está aprovado (média maior ou igual a 7), e qual é a média da turma. Para todos os exercícios, não esqueça de encapsular os atributos com métodos getters e setters e criar os construtores apropriados.

15. Elabore uma classe `ContaBancaria` com os seguintes atributos: `nomeCliente`, `numConta`, `saldo`. E os seguintes métodos: `sacar` (o saldo não pode ficar negativo), `depositar`. Crie também duas subclasses chamadas `ContaPoupanca` e `ContaEspecial`, com as seguintes características: `ContaPoupanca` - atributo `diaRendimento`, método `calcularNovoSaldo` que recebe taxa de rendimento da poupança e atualiza o saldo; `ContaEspecial` - atributo `limite`, sobrescreva o método `sacar` com a nova lógica necessária. Crie uma classe `Teste` que contenha a seguinte lógica: Criar contas, Sacar um valor das contas, Depositar, Mostrar um novo saldo a partir de um rendimento, Mostrar os dados da conta do cliente.

16. A Receita Federal possui um cadastro dos contribuintes. Cada contribuinte possui nome. Escreva um programa para calcular o imposto a ser pago para 5 contribuintes. Os contribuintes podem ser: Pessoa Jurídica - O imposto corresponde a 10% da renda bruta da empresa. Pessoa Física: Renda Bruta 0 a 1400 - Alíquota 0% - Parcela a deduzir R\$ 0; Renda Bruta 1400,01 a 2100 - Alíquota 10% - Parcela a deduzir R\$ 100; Renda Bruta 2100,01 a 2800 - Alíquota 15% - Parcela a deduzir R\$ 270; Renda Bruta 2800,01 a 3600 - Alíquota 25% - Parcela a deduzir R\$ 500; Renda Bruta 3600,01

ou mais - Alíquota 30% - Parcela a deduzir R\$ 700. Escreva um programa para calcular e imprimir na tela o imposto a ser pago de 6 contribuintes, sendo 3 PJ e 3 PF.

17. Escreva uma classe `FiguraGeometrica` com atributos `nome` e `cor`. Escreva uma classe `Figura2D` e uma classe `Figura3D`. Ambas classes `Figura2D` e `Figura3D` são subclasses da classe `FiguraGeometrica`. Crie ainda as classes `Circulo`, `Quadrado` e `Triangulo` que estendem a classe `Figura2D` e crie também as classes `Cubo`, `Cilindro` e `Piramide` que estendem a classe `Figura3D`.

18. Escreva a interface `DimensaoSuperficial` que contém o método `calcularArea`. Escreva a interface `DimensaoVolumetrica` que contém o método `calcularVolume`. Aplique a interface `DimensaoSuperficial` em todas as subclasses de `Figura2D` e `Figura3D` e aplique a interface `DimensaoVolumetrica` nas subclasses de `Figura3D`, implementando seus métodos de acordo com as regras de cada figura geométrica (crie os atributos necessários).

19. Crie pelo menos uma instância de cada classe e adicione em um array do tipo `FiguraGeometrica`. Itere esse array, imprimindo na tela a área e o volume (quando aplicável). Dica: você pode usar o operador `instanceof`.

20. Escreva uma classe `Contato` que contenha `nome`, `telefone` e um `identificador`. Esse `identificador` deve ser gerado automaticamente pela classe através de um contador (cada vez que um `Contato` é criado o contador é incrementado e atribuído ao `identificador`). Crie uma classe `Agenda` que irá funcionar para gerenciar os `Contatos`. Essa classe precisa ter um método para adicionar um `Contato` e também para consultar. Crie uma classe `Teste` que informe um menu ao usuário: 1 - para consultar um contato da agenda e 2 - para adicionar um contato na agenda. Se o contato não existir, o programa deve lançar uma exceção de `ContatoNaoExisteException`. Caso a agenda esteja cheia quando o usuário quiser adicionar um novo contato, o programa deve lançar uma exceção `AgendaCheiaException`. Trate também as exceções de tipo de dados (caso o usuário entre com um tipo de dado inválido).