



## LABORATÓRIO DE PROGRAMAÇÃO II - ATIVIDADE DIRIGIDA 1

Comentários:

- a. Todos os programas devem possuir pelo menos uma classe separada para iniciar sua execução (método *main*).
  - b. O trabalho pode ser feito individualmente ou em grupos de até três pessoas.
  - c. Um (a) integrante do grupo poderá ser escolhido (a) para apresentar uma questão durante a aula e a apresentação indicará a nota parcial do grupo.
- 
1. (2 pontos) Ao fazer exercícios físicos, você pode utilizar um monitor de frequência cardíaca para ver se sua frequência permanece dentro de um intervalo seguro sugerido pelos seus treinadores e médicos. Segundo a American Heart Association (AHA) ([www.americanheart.org/presenter.jhtml?identifier=4736](http://www.americanheart.org/presenter.jhtml?identifier=4736)), a fórmula para calcular a frequência cardíaca máxima por minuto é 220 menos a idade em anos. Sua frequência cardíaca alvo é um intervalo entre 50-85% da sua frequência cardíaca máxima. **[Observação: essas fórmulas são estimativas fornecidas pela AHA. As frequências cardíacas máximas e alvo podem variar com base na saúde, capacidade física e sexo da pessoa. Sempre consulte um médico ou profissional de saúde qualificado antes de começar ou modificar um programa de exercícios físicos.]** Crie uma classe chamada **FrequenciaCardiaca**. Os atributos da classe devem incluir o nome, sobrenome e data de nascimento da pessoa (consistindo em atributos separados para mês, dia e ano de nascimento). Sua classe deve ter um construtor que receba esses dados como parâmetros. Para cada atributo forneça métodos set e get. A classe também deve incluir um método que calcule e retorne a idade (em anos), um que calcule e retorne a frequência cardíaca máxima e um que calcule e retorne a frequência cardíaca alvo da pessoa. Escreva um aplicativo Java que solicite as informações da pessoa, instancie um objeto da classe **FrequenciaCardiaca** e imprima as informações a partir desse objeto — incluindo nome, sobrenome e data de nascimento da pessoa — calcule e imprima a idade da pessoa (em anos), seu intervalo de frequência cardíaca máxima e sua frequência cardíaca alvo.
  2. (2 pontos) Uma questão relacionada à assistência médica discutida ultimamente nos veículos de comunicação é a computadorização dos registros de saúde. Essa possibilidade está sendo abordada de maneira cautelosa por causa de preocupações quanto à privacidade e à segurança de dados sigilosos, entre outros motivos. [Iremos discutir essas preocupações em exercícios posteriores.] A computadorização dos registros de saúde pode facilitar que pacientes compartilhem seus perfis e históricos de saúde entre vários profissionais de saúde. Isso talvez aprimore a qualidade da assistência médica, ajude a evitar conflitos e prescrições erradas de medicamentos, reduza custos em ambulatorios e salve vidas. Neste exercício, você projetará uma classe **PerfilDeSaude** “inicial” para uma pessoa. Os atributos da classe devem incluir nome, sobrenome, sexo, data de nascimento (consistindo em atributos separados para mês, dia e ano de nascimento), altura (em metros) e peso (em quilogramas) da pessoa. Sua classe deve ter um construtor que receba esses dados. Para cada atributo, forneça métodos set

e get. A classe também deve incluir métodos que calculem e retornem a idade do usuário em anos, intervalo de frequência cardíaca máxima e frequência cardíaca alvo (veja a Questão 1 acima), além de índice de massa corporal (IMC). Escreva um aplicativo Java que solicite as informações da pessoa, instancie um objeto da classe **PerfilDeSaude** para ela e imprima as informações a partir desse objeto — incluindo nome, sobrenome, sexo, data de nascimento, altura e peso da pessoa —, e então calcule e imprima a idade em anos, IMC, intervalo de frequência cardíaca máxima e frequência cardíaca alvo. Ele também deve exibir o gráfico de valores IMC.

3. (1,5 ponto) O Facebook alcançou um bilhão de usuários em outubro de 2012 (de acordo com pesquisa do CNNMoney.com). Usando a técnica de crescimento composto com a fórmula abaixo, e supondo que a base de usuários cresça a uma taxa de 4% ao mês, quantos meses levarão para que o Facebook aumente sua base de usuários para 1,5 bilhão? Quantos meses serão necessários para que o Facebook expanda sua base de usuários para dois bilhões? Até dezembro de 2021, qual será a estimativa da base de usuários?

**Fórmula:**  $total = p \times (1 + r)^n$ ;

Onde:

**p:** quantidade de usuários;

**r:** taxa de juros anual (para 4%, utilize 0,04);

**n:** número de anos;

**total:** quantidade no n-ésimo ano.

4. (1,5 ponto) Escreva um aplicativo que solicite aos usuários que insiram o tamanho do lado de um quadrado e, então, exibe um quadrado vazio desse tamanho com asteriscos. Seu programa deve trabalhar com quadrados de todos os comprimentos de lado possíveis entre 1 e 38. *Exemplo:*

*Entrada:* 5

*Saída:*

```

* * * * *
*       *
*       *
*       *
*       *
* * * * *
```

5. (0,5 ponto) Calcule e mostre na tela a soma de todos os números ímpares no intervalo a e b. Leia o valor de início (a) e o valor de fim (b) do teclado utilizando o método apropriado da classe *Scanner*.
6. (1,5 ponto) Crie um programa que apresente um menu aos usuários com as seguintes opções de uma calculadora de volumes:

1. Calcular e exibir o volume de um cilindro;
2. Calcular e exibir o volume de um cone;
3. Calcular e exibir o volume de um prisma;
4. Calcular e exibir o volume de uma esfera.
5. Finalizar a aplicação.

Os usuários terão de digitar os dados das dimensões dos objetos os quais o programa solicitará e calculará seus respectivos valores. Os métodos cilindro(), cone(), prisma(), esfera() deverão ser criados para ser chamados na classe.

- i. 1, 2, 3 – votos para candidato (a) 1, candidato (a) 2 e candidato (a) 3 respectivamente;
- ii. 0 – voto em branco;
- iii. 4 – voto nulo.

- O número do (a) candidato (a) vencedor (a);
- O número de votos em branco;
- O número de votos nulos;
- O total de alunos votantes.
- O total de votos para cada candidato (a);

8. (1,5 ponto) Um motorista monitorou vários tanques cheios de gasolina registrando a quilometragem dirigida e a quantidade de combustível em litros utilizados para cada tanque cheio. Escreva um aplicativo Java que receba como entrada os quilômetros dirigidos e os litros de gasolina consumidos (ambos como inteiros) para cada tanque cheio. O programa deve calcular e exibir o consumo em quilômetros/litro para cada tanque cheio e imprimir a quilometragem combinada e a soma total de litros de combustível consumidos até esse ponto. Todos os cálculos de média devem produzir resultados de ponto flutuante. Utilize a classe *Scanner* e repetição controlada por sentinela/bandeira para obter os dados do usuário.

10. (1 ponto) Escreva um aplicativo que aceite como entrada um inteiro contendo somente 0s e 1s (isto é, um inteiro binário) e mostre na tela seu inteiro equivalente decimal.

[illegible]

Maximize sua utilização de repetição (com instruções for aninhadas) e minimize o número de instruções de saída.

12. (1 ponto) Escreva um programa que calcule o valor de PI das séries infinitas:

$$pi = 4 - \frac{4}{3} + \frac{4}{5} - \frac{4}{7} + \frac{4}{9} - \frac{4}{11} + \dots$$

Mostre na tela uma tabela com o valor aproximado de PI utilizando os 200.000 primeiros termos dessa série. Quantos termos você tem de utilizar antes de obter um valor que comece com 3,14159? Mostre também na tela essa informação.

13. (1 ponto) Um varejista on-line vende cinco produtos cujos preços no varejo são como a seguir:

Produto 1, R\$ 2,98; Produto 2, R\$ 4,50; Produto 3, R\$ 9,98; Produto 4, R\$ 4,49 e Produto 5, R\$ 6,87.

Escreva um aplicativo que leia uma série de pares de números como segue:

- a) número de produto;
- b) quantidade vendida.

Seu programa deve utilizar uma instrução *switch* para determinar o preço de varejo de cada produto. Você deve calcular e exibir o valor de varejo total de todos os produtos vendidos. Utilize um laço controlado por sentinela para determinar quando o programa deve parar o laço e exibir os resultados finais.

14. (0,5 ponto) Calcule e mostre na tela a soma  $s = x/1! - x/2! + x/3! - x/4! + \dots - x/10!$ .

O código será validado se utilizar um laço para isso.

Leia o valor de x do teclado (peça aos usuários).

Dica: crie o método *fatorial(valor)* que recebe um inteiro e retorna um double.

15. (1 ponto) Crie uma classe *ContaBancaria*. A classe deve fornecer um construtor *ContaBancaria(double saldoInicial)* que recebe já o valor do saldo inicial da conta. Deve haver um teste para verificar se esse saldo é maior que zero. Forneça o método *depósito* que adiciona uma quantia ao saldo da conta. Forneça um método chamado *saque* que retira dinheiro de uma conta da classe *ContaBancaria*. Assegure que a quantia de débito não exceda o saldo de *ContaBancaria*. Se exceder, o saldo deve ser deixado inalterado e o método deve imprimir uma mensagem que indica "Você não tem saldo suficiente para realizar essa operação". Forneça o método *getSaldo()* que retorna o saldo da conta. Crie a classe *ContaTeste* para testar o método *saque*. Crie dois objetos na classe de teste (*conta1* e *conta2*), com valores diferentes de depósito no construtor. Exiba na tela o saldo de cada conta.

16. (1 ponto) Escreva um aplicativo que receba uma matriz quadrada de ordem n e a seguir contar quantos elementos da matriz são iguais a zero apresentando este resultado na tela.

17. (1 ponto) Crie um método *quadradoDeCaracteres* para formar o quadrado a partir de quaisquer que sejam os caracteres (digitados pelos usuários) contidos nos parâmetros de caracteres *a* e *b*. O parâmetro *a* está associado as diagonais do quadrado. O parâmetro *b*, aos elementos restantes. Escreva um aplicativo de teste chamado *TesteQuadrados* que executa o método. Portanto, se *lado* (digitado pelos usuários) for 5 e *a* for '\*' com *b* '#', o método deve exibir a seguinte saída:

```
* # # # *
# * # * #
# # * # #
# * # * #
* # # # *
```