Exercícios[1]

Estruturas de Controle

Segunda-feira, 20 de março de 2023.

---

1. Escreva um programa que leia um caractere e escreva se o caractere é um(a):

- vogal;
- consoante;
- dígito decimal;
- operador aritmético: + , - , *, /, % ;
- símbolo especial: @, #, $, &, <, >, }, {, \, |, =, £, §, ª, º, °, ¬ , _ , ¢;
- sinal de pontuação: !, ?, : , ; , . , , " , ' ,) , (,] , [;
- sinal de acentuação: ¨ , ^ , ~ , ´ , ` .

O programa deve exibir a categoria e o nome do caractere Unicode. Use os métodos da classe java.lang.Character.

2. Resolva os exercícios abaixo do Capítulo 4 - Instruções de controle: parte 1 - do livro Java Como Programar. 6ª ed. DEITEL, H. M., DEITEL, P. J. São Paulo: Pearson Prentice Hall, 2006.

Exercícios 4.14 a 4.22, 4.25 a 4.27, 4.30, 4.31, 4.34, 4.37 e 4.38.

3. Resolva os exercícios abaixo do Capítulo 5 - Instruções de controle: parte 2 - do livro Java Como Programar. 6ª ed. DEITEL, H. M., DEITEL, P. J. São Paulo: Pearson Prentice Hall, 2006.

Exercícios 5.7 a 5.10, 5.13 e 5.21.

☞ **Nota:** Os exercícios relacionados dos Capítulos 4 e 5 da 6ª edição do livro Java Como Programar também estão disponíveis nas edições 8 e 10, publicados, respectivamente, em 2010 e 2017.

---

4. Escreva um programa para ler uma *string*, calcular e exibir o seu número de caracteres, a quantidade de caracteres que são letras, vogais, consoantes, dígitos, sinais de pontuação, sinais de acentuação, operador aritmético e símbolo especial. Use a relação de caracteres do Exercício 1 e os métodos das classes java.lang.String e java.lang.Character.

[1] Atualizado em 21/03/2023.

5. Faça um programa para calcular o valor de S dado por:

$$S = \frac{1}{N} + \frac{2}{N-1} + \frac{3}{N-2} + \frac{4}{N-3} + ... + \frac{N-1}{2} + \frac{N}{1}$$, onde N é um número inteiro e positivo.

6. O valor aproximado de π pode ser calculado usando-se a série:

$$s = \frac{1}{1^3} \quad \frac{1}{3^3} + \frac{1}{5^3} \quad \frac{1}{7^3} \cdots \frac{1}{N^3}$$, sendo que $\pi = (s \times 32)^{\frac{1}{3}}$

Para um número N, inteiro e positivo, desenvolva um programa que calcule o valor de π. O número π definido pela classe Math é 3.141592653589793, qual o valor de N necessário para o programa calcular o mesmo valor com 15 casas decimais?

7. A série de Fibonacci é formada pela sequência: 1, 1, 2, 3, 5, 8, 13, 21, 34, 55,....., etc. Escreva um programa que gere a série de Fibonacci até o n-ésimo termo (quantidade de números) fornecido pelo usuário.

8. Desenvolva um programa Java GUI que exiba as 24 letras maiúsculas e minúsculas do alfabeto grego, como se pode ver na imagem ao lado.

Use a codificação UTF-16 para representar os caracteres Unicode ao exibir cada letra do alfabeto grego.

O grupo de letras maiúsculas e minúsculas devem ser exibidas em caixas de diálogo separadas, com um texto de identificação adequado na barra de título.

A codificação UNICODE UTF-8, UTF-16 e UTF-32 de cada caractere do alfabeto grego pode ser consultado em https://www.compart.com/en/unicode/block/U+0370.

Ao acessar o endereço acima, observe que a primeira letra do alfabeto, Alpha, na codificação UTF-16, possui, em sua forma maiúscula o código U+0391 e em minúscula o código U+03B1.

Α Alpha	Β Beta	Γ Gamma	Δ Delta	Ε Epsilon	Ζ Zeta
Η Eta	Θ Theta	Ι Iota	Κ Kappa	Λ Lambda	Μ Mu
Ν Nu	Ξ Xi	Ο Omicron	Π Pi	Ρ Rho	Σ Sigma
Τ Tau	Υ Upsilon	Φ Phi	Χ Chi	Ψ Psi	Ω Omega

α Alpha	β Beta	γ Gamma	δ Delta	ε Epsilon	ζ Zeta
η Eta	θ Theta	ι Iota	κ Kappa	λ Lambda	μ Mu
ν Nu	ξ Xi	ο Omicron	π Pi	ρ Rho	σ Sigma
τ Tau	υ Upsilon	ϕ Phi	χ Chi	ψ Psi	ω Omega

Figura 1 - Alfabeto Grego, disponível em https://www.worldhistory.org/image/3426/greek-alphabet/

9. Escreva um programa para calcular o tempo de execução de um *loop* for que faça a concatenação de 100.000 *strings* com um valor inteiro usando a classe String e outro *loop* for que faça a mesma quantidade de união de *strings* usando a classe StringBuilder. O programa deve exibir o tempo inicial e final no formato HH:MM:SS:MS, mostrar a duração em segundos se o tempo de execução for maior do que 1 segundo ou em milissegundos se for menor. Use o leiaute abaixo para exibir a saída do programa com os tempos de execução de cada *loop* for e as classes do pacote java.time para obtê-los.

Concatenação com StringBuilder....

Tempo inicial = 08:38:00.394
Tempo final = 08:38:00.643
Duração = 249 ms

Concatenação com String....

Tempo inicial = 08:38:00.646
Tempo final = 08:38:28.290
Duração = 27 s

A concatenação com foi StringBuilder foi 108 vezes mais rápida.

Prof**. M**árlon **O**liveira **d**a **S**ilva
marlon.silva@ifsudestemg.edu.br