

Lista de Exercícios 1 – Programação Java (atividade em dupla ou trio)

Objetivos

- Treinar os fundamentos da linguagem Java (tipos primitivos, operadores, estruturas de decisão, estruturas de repetição, arrays, métodos).
- Detectar similaridades com outras linguagens de programação que você já conheça.
- Verificar dificuldades com a linguagem Java.
- Treinar interpretação de texto.

Instruções

- As questões a seguir devem ser resolvidas usando a linguagem Java.
- Para cada questão, escreva o código em um arquivo .java (se você resolver todas as questões, devem existir 10 arquivos .java).
- Coloque a identificação da dupla ou trio – nomes completos e TIAs – no início de cada arquivo, como comentário (use // no começo de cada linha que queira comentar).
- Inclua como comentário quaisquer referências (livros, artigos, sites, vídeos, entre outros) usadas para responder as questões.
- O gabarito das questões estará disponível para quem realizar a entrega no Moodle.
- **Entrega:** via link do Moodle até 05/03/2023 23:59.

Questões

1. Escreva um código Java que leia dois números inteiros informados pelo usuário. Cada número deve ser salvo em variáveis distintas, por exemplo, `valor1` e `valor2`. Após ler os dois números, o código deve trocar o conteúdo das variáveis.

Exemplo: `valor1` armazena o número 30 e `valor2` armazena o número 50. Após o algoritmo ser executado, a variável `valor1` deve armazenar o número 50 e a variável `valor2` deve armazenar o número 30.

Para pensar... e anotar as suas respostas no papel ou no digital (ou como comentário no código da sua solução):

Existe algum tipo de entrada [feita por quem usa o programa] que “quebra” o seu código? Como podemos resolver o problema?

2. Escreva um código Java que leia uma letra e indique se é uma vogal ou consoante.
3. Escreva um código Java que leia 10 números reais e, após a leitura dos números, informe qual é o maior e qual é o menor.
4. Escreva um código Java que leia 10 números inteiros e, em seguida, exiba-os na ordem inversa que foram inseridas pelo usuário.

5. Assumindo a equação de reta $3x + y = 8$, escreva um código que leia os valores para x e y e indique se o ponto (x, y) pertence ou não pertence à reta.
6. Escreva um código Java que exiba a sequência de Fibonacci com N números, sendo N um número informado pelo usuário. A sequência de Fibonacci começa com 1 e 1 e os números subsequentes são obtidos com a soma dos dois números anteriores: 1, 1, 2, 3, 5, 8, 13, ...
7. Escreva um método estático `min()` que recebe dois parâmetros numéricos inteiros e retorna o menor valor entre eles e um método estático `max()` que recebe dois parâmetros numéricos inteiros e retorna o maior valor entre eles.

Inclua um código na `main()` que exemplifica o uso dos dois métodos.

8. Escreva um método estático `mulComAdicao()` que recebe dois números como parâmetros. O primeiro parâmetro é um número real e o segundo parâmetro um número inteiro. A função deve retornar a multiplicação entre os parâmetros, porém, a multiplicação deve ser feita usando apenas a operação de adição.

Inclua um código na `main()` que exemplifica o uso do método `mulComAdicao()`.

9. Escreva um método estático `contarVogais()` que recebe uma string como parâmetro e retorna a quantidade de vogais encontradas na string.

Inclua um código na `main()` que exemplifica o uso do método `contarVogais()`.

Dicas:

- A classe `String` do Java possui um método `charAt(int index)` que retorna um `char` na posição indicada pelo parâmetro `index`.
- A classe `String` do Java possui um método `toCharArray()` que retorna o conteúdo da string como um array de `chars`.

10. Dois círculos se sobrepõem (colidem) se a soma dos seus raios é maior que ou igual a distância entre seus centros. A distância entre dois pontos pode ser calculada usando o teorema de Pitágoras ($c = \sqrt{a^2 + b^2}$). Escreva um método estático `haColisaoEntreCirculos()` que recebe seis parâmetros: a posição $p1(x,y)$ e o raio $r1$ do primeiro círculo e a posição $p2(x,y)$ e o raio $r2$ do segundo círculo. Com os parâmetros informados, a função deve retornar `true` caso os círculos estejam sobrepostos ou `false`, caso contrário.

Inclua um código na `main()` que exemplifica o uso do método `haColisaoEntreCirculos()`.