



ALGORITMOS E PROGRAMAÇÃO I

Desafio I

Calculadora



01 |

Implemente um programa em Java com Menu de opções, **para realizar operações matemáticas básicas**. As opções do menu e seus objetivos correspondentes devem ser implementados conforme abaixo.

Opção	Descrição	Objetivo
1	Somar	- Pedir dois valores ao usuário e realizar a soma dos valores. - Apresentar o resultado ao usuário.
2	Subtrair	- Pedir dois valores ao usuário e realizar a subtração dos valores. - Apresentar o resultado ao usuário.
3	Multiplicar	- Pedir dois valores ao usuário e realizar a multiplicação dos valores. - Apresentar o resultado ao usuário.
0	Dividir	- Pedir dois valores ao usuário e realizar a divisão dos valores. - Apresentar o resultado ao usuário.
5	Potência	- Pedir a base e o expoente ao usuário e realizar a potenciação dos valores. - Apresentar o resultado ao usuário.
6	Raiz Quadrada	- Pedir um valor ao usuário e realizar a raiz quadrada do valor. - Apresentar o resultado ao usuário.
0	Sair	Sair do programa.

Pense nas melhores abstrações para seu programa

Função

`exibirMenu() : int`

Objetivo

`Exibir menu e solicitar escolha`

Função

`somar(n1: double, n2: double) : double`

Função

`subtrair(n1: double, n2: double) : double`

Função

`multiplicar(n1: double, n2: double) : double`

Função

`dividir(n1: double, n2: double) : double`

Função

`potencia(num: double, exp: double) : double`

Função

`raizQuadrada(num: double) : double`

Dinâmica do programa

1. Somar
2. Subtrair
3. Multiplicar
4. Dividir
5. Potência
6. Raiz Quadrada
0. Sair

1

Informe um número:

10

Informe outro número:

5

O resultado é 15

1. Somar
2. Subtrair
3. Multiplicar
4. Dividir
5. Potência
6. Raiz Quadrada
0. Sair

4

Informe um número:

10

Informe outro número:

5

O resultado é 2

Início

1. Somar
2. Subtrair
3. Multiplicar
4. Dividir
5. Potência
6. Raiz Quadrada
0. Sair

5

Informe um número:

2

Informe outro número:

3

O resultado é 8

1. Somar
2. Subtrair
3. Multiplicar
4. Dividir
5. Potência
6. Raiz Quadrada
0. Sair

6

Informe um número:

25

O resultado é 5

continuação

1. Somar
2. Subtrair
3. Multiplicar
4. Dividir
5. Potência
6. Raiz Quadrada
0. Sair

0

continuação

Desafio II

Número Sorteado



02 |

Implemente um programa em Java com Menu de opções, que **sorteie um número premiado entre 0 e 1000** em uma competição entre alunos de uma escola onde o objetivo do aluno que participar é descobrir o número sorteado em no máximo **10 tentativas**.

- A cada tentativa, o programa deve informar se o aluno **acertou**, ou em caso negativo, dizer se seu chute era **menor** ou **maior** que o número sorteado.
- Se ao final das tentativas, o aluno não acertar, o programa deve apresentar a mensagem: **"Perdeu"**.

Exemplo da Dinâmica do Programa:

O número já foi sorteado.

Sua escolha: 700
Maior

Sua escolha: 900
Maior

Sua escolha: 915
Menor

Sua escolha: 912
Maior

Sua escolha: 913
Acertou

Utilize o código abaixo para gerar um número aleatório:

```
Random rdn = new Random();  
int n = rdn.nextInt(0, 1000);
```

Pense nas melhores abstrações para seu programa

Função

`sortearNumero() : int`

Objetivo

Gerar um número aleatório entre 0 e 1000

Função

```
ganhou(sorteado: int,  
       numUsuario: int,  
       tentativa: int) : String
```

Objetivo

Se usuário ganhou, retorna "Acertou"
Se número Usuário é maior, retorna "Maior"
Se número Usuário é menor, retorna "Menor"
Se tentativa for igual a 10, retorna "Perdeu"

Desafio III

Votação Representante de Sala



03 |

Implemente um programa em Java com Menu de opções, **para realizar a votação de quem será o representante de turma em uma sala**. O programa deve considerar três alunos como candidatos. Depois, inicia-se a votação sendo que cada aluno é apresentado como uma das opções do menu. A votação continua até que seja escolhida a opção “Encerrar”. Quando isso ocorrer, o programa deve apresentar o novo representante e o vice representante acompanhados do total de votos recebidos. Caso tenha dado qualquer tipo de empate, o programa deve apresentar a mensagem: *“Votação com empate, discutam suas ideias e iniciem uma nova votação”*.

Exemplo de menu para os alunos **João, Maria e José**:

Opção	Descrição
1	João
2	Maria
3	José
0	Encerrar

Função para Limpar a Tela. *Esse código roda no REPLIT.

```
public static void limparTela() {  
    System.out.print("\n\npressione <enter> para voltar ao menu.\n");  
    new Scanner(System.in).nextLine();  
  
    // Ativa ANSI.SYS para executar comando que  
    // reposiciona o cursor e limpa o terminal  
    System.out.print("\033[H\033[2J");  
    System.out.flush();  
}
```

Pense nas melhores abstrações para seu programa

Função

```
exibirMenu(aluno1: String,  
           aluno2: String,  
           aluno3: String) : int
```

Objetivo

Exibir menu e solicitar escolha

Função

```
avaliarResultado(  
    votosAluno1: int,  
    votosAluno2: int,  
    votosAluno3: int) : void
```

Objetivo

Exibe o representante e vice representante, ou no caso de empate, exibe a necessidade de uma nova votação

Dinâmica do programa

Informe o primeiro aluno:

João

Informe o segundo aluno:

Maria

Informe o terceiro aluno:

José

1. João
2. Maria
3. José
0. Encerrar
1

(limpar tela)

1. João
2. Maria
3. José
0. Encerrar
1

(limpar tela)

1. João
2. Maria
3. José
0. Encerrar
1

(limpar tela)

Início

1. João
2. Maria
3. José
0. Encerrar
1

(limpar tela)

1. João
2. Maria
3. José
0. Encerrar
2

(limpar tela)

1. João
2. Maria
3. José
0. Encerrar
2

(limpar tela)

1. João
2. Maria
3. José
0. Encerrar
2

(limpar tela)

continuação

1. João
2. Maria
3. José
0. Encerrar
3

(limpar tela)

1. João
2. Maria
3. José
0. Encerrar
3

(limpar tela)

1. João
2. Maria
3. José
0. Encerrar
0

O resultado da votação foi:
Representante: João, 4 votos
Vice-representando: Maria, 3 votos

continuação

Desafio IV

Teste de Digitação



04 |

Implemente um programa em Java com Menu de opções, que **represente um teste de digitação**. Nesse jogo, o usuário escolherá quantas palavras ele deve digitar e o programa gerará palavras *fakes* aleatórias para que ele digite. Ao final, o programa deve apresentar as estatísticas do usuário como ***Precisão de acerto (%)*** e ***Tempo gasto (seg)***.

Exemplo da Dinâmica do Programa:

Quantas palavras você quer digitar?

4

muleui

muleui

ribunola

ribunola

bumivero

bumivero

pelidero

pelidaaa

Seu teste terminou!

Precisão: 75% - Tempo: 25.5seg

Pense nas melhores abstrações para seu programa

Função

gerarPalavra() : String

Objetivo

Gera uma palavra aleatória.

Função

iniciarTempo() : long

Objetivo

Retorna o tempo atual em milisegundo

Função

acertou(palavra: String,
palavraDigitada: String) : boolean

Objetivo

Identifica se o que foi digitado está igual a palavra gerada

Função

tempoGasto(tempoInicial: long) : String

Objetivo

Retorna a diferença do tempo atual com o inicial em segundos. Ex: "2.2seg"

Função

precisao(acertos: int,
totalPalavras: int) : String

Objetivo

Retorna a precisão em (%). Ex: "75%"



04 |

Considere a função abaixo para gerar uma palavra de 8 caracteres que varia entre consoante e vogal.

```
import java.util.Random;

public static String gerarPalavra() {
    String consoantes = "bcdfghjklmnpqrstvwxyz";
    String vogais = "aeiou";

    Random rdn = new Random();

    String novaPalavra = "";
    for (int i = 0; i < 8; i++) {
        char letra = ' ';
        if (i % 2 == 0) {
            letra = consoantes.charAt(rdn.nextInt(0, 20));
        } else {
            letra = vogais.charAt(rdn.nextInt(0, 4));
        }
        novaPalavra += letra;
    }
    return novaPalavra;
}
```

Considere a função abaixo que recupera o tempo inicial a partir do horário do Sistema Operacional, executa um algoritmo e recupera o tempo final. O tempo é marcado em milissegundos. Ao final, a diferença é convertida em segundos.

```
import java.util.Date;

public static void testeTempo() {

    long inicio = new Date().getTime();

    // início do código
    for (int i = 0; i < 20000; i++) {
        System.out.println(i);
    }
    // fim do código

    long fim = new Date().getTime();
    double tempoEmSeg = (fim - inicio) / 1000.0;

    System.out.println(tempoEmSeg);
}
```

Desafio V
Jogo Blackjack (21)



05 |

Implemente um programa em Java com Menu de opções que **represente o jogo Blackjack (21)**. Nesse jogo, o usuário disputará contra outro jogador, seguindo as regras:

- O jogador 01 deve iniciar comprando as cartas até decidir parar.
- O ato de comprar consiste na geração aleatória de uma carta com número (1-10, J, Q, K) e naipe (♠ ♣ ♥ ♦).
- Cartas de figura (J, Q, K) valem 10.
- Não é necessário garantir que a mesma carta seja comprada apenas uma vez.
- Quando o jogador 01 parar, o programa deve **limpar a tela** para que o jogador 02 não veja seus pontos.
- Em seguida o jogador 02 deve fazer o mesmo.
- Ao final, o jogador que somando suas cartas estiver mais próximo do número 21 é o vencedor.
- Se a soma ultrapassar 21, o jogador é considerado perdedor.
- Se a soma for igual, será considerado empate.

Utilize o código abaixo para gerar um número aleatório:

```
Random rdn = new Random();  
int n = rdn.nextInt(1, 13);
```

Pense nas melhores abstrações para seu programa

Função

`escolherCarta() : String`

Objetivo

Gera uma carta aleatório (1-10, J, Q, K)

Função

`escolherNipe() : String`

Objetivo

Gera um nipe aleatório ♠ ♣ ♥ ♦

Função

`lerValorCarta(carta: String) : int`

Objetivo

Identifica o valor de uma carta.
Ex: "5"=5 "Q"=10

Função

`somarPontos(pontos: int,
 carta: String) : int`

Objetivo

Soma os pontos atuais com o da carta comprada

Dinâmica do programa

Jogador 1, sua vez de começar:

1. Comprar

2. Parar

1

Carta comprada 7 ♣

1. Comprar

2. Parar

1

Carta comprada 5 ♦

1. Comprar

2. Parar

1

Carta comprada K ♥

1. Comprar

2. Parar

2

(limpar tela)

início

Jogador 2, sua vez de começar:

1. Comprar

2. Parar

1

Carta comprada Q ♠

1. Comprar

2. Parar

1

Carta comprada 9 ♠

1. Comprar

2. Parar

2

Resultado: Jogador 1 venceu!

Jogador 1: 22

Jogador 2: 19

continuação

Desafio VI

Compras Supermercado



06 |

Implemente um programa em Java com Menu de opções, **para realizar compras em um Mercado**. As opções do menu e seus objetivos correspondentes devem ser implementados conforme abaixo.

Opção	Descrição	Objetivo
1	Adicionar produto	- Pedir ao usuário o Nome, a Qtd e o Preço do produto. - Atualizar o subtotal da compra e a qtd de itens vendidos.
2	Exibir subtotal	- Exibir o subtotal da compra no formato: (5 itens) - Subtotal: R\$ 135.50 Continuar comprando
3	Finalizar compra	- Exibir a qtd de produtos comprados e o total final no formato: (5 itens) - Total: R\$ 135.50 Compra finalizada Iniciando nova compra.
		- Mostrar o menu novamente, iniciando uma nova compra com os valores zerados
0	Sair	Sair do programa.

Pense nas melhores abstrações para seu programa

Função
`exibirMenu() : int`

Objetivo
Exibir menu e solicitar escolha.

Função
`adicionar(subtotal: double,
 preco: double,
 qtd: int) : double`

Objetivo
Calcular o novo total da compra através do subtotal atual e preço/qtd do novo Produto

Função
`exibirSubtotal(subtotal: double) : void`

Objetivo
Imprime o subtotal ao usuário conforme especificação ao lado.

Função
`exibirTotal(total: double) : void`

Objetivo
Imprime o total ao usuário conforme especificação ao lado.

Dinâmica do programa

1. Adicionar produto
 2. Exibir subtotal
 3. Finalizar compra
 0. Sair
- 1

Informe o nome do Produto:
Coca-cola

Informe a Quantidade:
1

Informe o preço da Unidade:
11.5

Produto adicionado.

1. Adicionar produto
 2. Exibir subtotal
 3. Finalizar compra
 0. Sair
- 1

Informe o nome do Produto:
Barra de Chocolate

Informe a Quantidade:
1

Informe o preço da Unidade:
7.5

Produto adicionado.

Início

1. Adicionar produto
 2. Exibir subtotal
 3. Finalizar compra
 0. Sair
- 2

(2 itens) – Subtotal R\$ 19.00
Continue comprando.

1. Adicionar produto
 2. Exibir subtotal
 3. Finalizar compra
 0. Sair
- 3

(2 itens) – Total R\$ 19.00
Compra finalizada.
Iniciando nova compra.

1. Adicionar produto
 2. Exibir subtotal
 3. Finalizar compra
 0. Sair
- 1

Informe o nome do Produto:
Fanta Laranja

Informe a Quantidade:
1

Informe o preço da Unidade:
10.5

continuação

Produto adicionado.

1. Adicionar produto
 2. Exibir subtotal
 3. Finalizar compra
 0. Sair
- 3

(1 item) – Total R\$ 10.50
Compra finalizada.
Iniciando nova compra.

1. Adicionar produto
 2. Exibir subtotal
 3. Finalizar compra
 0. Sair
- 0

continuação

Desafio VII

Jogo Forca



07 |

Implemente um programa em Java com Menu de opções, que **represente o jogo da forca**. Nesse jogo, o desafiante irá informar a palavra a ser descoberta, em seguida, o programa **limpará a tela** para que o desafiado comece a realizar suas jogadas. As jogadas consistem em duas opções:

Opção	Descrição
1	Tentar letra
2	Já sei a palavra

O jogo deve considerar as regras:

- É possível realizar 12 tentativas de Letra (*opção 1*), antes de perder o jogo.
- Após as 12 tentativas, o desafiado é obrigado a dizer qual a palavra correta.
- Se o desafiado tentar adivinhar a palavra (*opção 2*) antes das 12 tentativas, o programa deve avaliar se ele ganhou ou perdeu e encerrar o jogo.
- Se ao tentar uma Letra (*opção 1*), o usuário acertar e não sobrar letras restantes a serem descobertas, o usuário será considerado vencedor.
- A cada nova tentativa o programa deve informar quantas ocorrências foram encontradas e atualizar, se for o caso, a situação do jogo de forma visual.
- Uma tentativa só pode ser um caractere.
- No caso de uma tentativa possuir dois ou mais caracteres, o programa deve exibir “*Tentativa inválida*”.
- Espaços são considerados caracteres.

Função

exibirMenu() : int

Objetivo

Exibir menu e solicitar escolha

Função

tentarLetra(palavra: String, letra: char, board: String) : String

Objetivo

Retorna a nova situação do Jogo que atualizará o board atual.

Função

ocorrencias(palavra: String, letra: char, board: String) : int

Objetivo

Retorna as ocorrências da tentativa

Função

tentarPalavra(palavra: String, tentativa: String) : boolean

Objetivo

Verifica se a tentativa do usuário é igual a palavra

Função

venceu(board: String) : boolean

Objetivo

Verifica se todas as letras já foram descobertas

Dinâmica do programa

Informe a palavra a ser adivinhada:

batata

(limpar tela)

1. Tentar letra
2. Já sei a palavra
1

Informe uma letra:

z

0 ocorrência(s) encontradas.
Sua força: _ _ _ _ _

1. Tentar letra
2. Já sei a palavra
1

Informe uma letra:
a

3 ocorrência(s) encontradas.
Sua força: _ a _ a _ a

Início

1. Tentar letra
2. Já sei a palavra
1

Informe uma letra:

t

2 ocorrência(s) encontradas.
Sua força: _ a t a t a

1. Tentar letra
2. Já sei a palavra
1

Informe uma letra:

d

0 ocorrência(s) encontradas.
Sua força: _ a t a t a

continuação

1. Tentar letra
2. Já sei a palavra
1

Informe uma letra:

b

1 ocorrência(s) encontradas.
Sua força: b a t a t a

Você venceu!

continuação



Bons estudos!
Bruno de Oliveira