

Programação Modular - Noite
Exercícios de Fundamentos de P00

Aluno: Frederico Martins Rodrigues

1) Faça o projeto de uma classe “Retangulo” para representar um retângulo que será desenhado na tela do Console. Este retângulo terá uma altura, uma largura, um caractere de borda e um deslocamento na tela definidos pelo usuário. O deslocamento indica quantas colunas de tela devem ser saltadas desde a borda esquerda da tela até a borda esquerda do retângulo. Somente o caractere de borda e o deslocamento podem ser modificados após a criação de um retângulo. Implemente sua classe e a utilize num código principal.

```
import java.util.Scanner;

public class Retangulo {
    private final int altura;
    private final int largura;
    private char borda;
    private int deslocamento;

    public Retangulo(int altura, int largura, char borda, int deslocamento) {
        this.altura = altura;
        this.largura = largura;
        this.borda = borda;
        this.deslocamento = deslocamento;
    }

    public void setBorda(char borda) {
        this.borda = borda;
    }

    public void setDeslocamento(int deslocamento) {
        this.deslocamento = deslocamento;
    }

    public void desenhar() {
        for (int i = 0; i < altura; i++) {
            System.out.print(" ".repeat(deslocamento));

            for (int j = 0; j < largura; j++) {
                if (i == 0 || i == altura - 1 || j == 0 || j == largura - 1) {
                    System.out.print(borda);
                } else {
                    System.out.print(" ");
                }
            }

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

```

}

public static int solicitarInteiro(String mensagem, Scanner scanner) {
    System.out.print(mensagem);
    while (!scanner.hasNextInt()) {
        System.out.print("Entrada inválida. " + mensagem);
        scanner.next();
    }
    return scanner.nextInt();
}

public static char solicitarCaractere(String mensagem, Scanner scanner) {
    System.out.print(mensagem);
    return scanner.next().charAt(0);
}

@SuppressWarnings("ConvertToTryWithResources")
public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);

    int altura = solicitarInteiro("Informe a altura do retângulo: ", scanner);
    int largura = solicitarInteiro("Informe a largura do retângulo: ", scanner);
    char borda = solicitarCaractere("Informe o caractere para a borda: ", scanner);
    int deslocamento = solicitarInteiro("Informe o deslocamento do retângulo: ",
scanner);

    Retangulo retangulo = new Retangulo(altura, largura, borda, deslocamento);

    OUTER:
    while (true) {
        retangulo.desenhar();
        System.out.println("\nOpções:");
        System.out.println("1 - Alterar caractere de borda");
        System.out.println("2 - Alterar deslocamento");
        System.out.println("3 - Sair");
        System.out.print("Escolha uma opção: ");
        int opcao = solicitarInteiro("", scanner);
        switch (opcao) {
            case 1 -> {
                borda = solicitarCaractere("Informe o novo caractere de borda: ",
scanner);
                retangulo.setBorda(borda);
            }
            case 2 -> {
                deslocamento = solicitarInteiro("Informe o novo deslocamento: ",
scanner);
                retangulo.setDeslocamento(deslocamento);
            }
        }
    }
}

```

```

        case 3 -> {
            System.out.println("Programa encerrado.");
            break OUTER;
        }
        default -> System.out.println("Opção inválida.");
    }
}

scanner.close();
}
}

```

2) Use um diagrama de classes UML para modelar uma calculadora que consiga fazer as quatro operações matemáticas básicas. Implemente sua classe.

Calculadora
+ somar(a, b)
+ subtrair(a, b)
+ multiplicar(a, b)
+ dividir(a, b)

```

import java.util.Scanner;

public class Calculadora {

    public double somar(double a, double b) {
        return a + b;
    }

    public double subtrair(double a, double b) {
        return a - b;
    }

    public double multiplicar(double a, double b) {
        return a * b;
    }

    public double dividir(double a, double b) {
        if (b == 0) {
            throw new ArithmeticException("Divisão por zero não permitida.");
        }
        return a / b;
    }

    public static double solicitarDouble(String mensagem, Scanner scanner) {

```

```

        System.out.print(mensagem);
        while (!scanner.hasNextDouble()) {
            System.out.print("Entrada inválida. " + mensagem);
            scanner.next();
        }
        return scanner.nextDouble();
    }

    @SuppressWarnings("ConvertToTryWithResources")
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        Calculadora calc = new Calculadora();

        while (true) {
            System.out.println("\nEscolha a operação:");
            System.out.println("1 - Somar");
            System.out.println("2 - Subtrair");
            System.out.println("3 - Multiplicar");
            System.out.println("4 - Dividir");
            System.out.println("5 - Sair");
            System.out.print("Opção: ");

            int opcao = scanner.nextInt();

            if (opcao == 5) {
                System.out.println("Calculadora encerrada.");
                break;
            }

            double num1 = solicitarDouble("Digite o primeiro número: ", scanner);
            double num2 = solicitarDouble("Digite o segundo número: ", scanner);

            try {
                switch (opcao) {
                    case 1 -> System.out.println("Resultado: " + calc.somar(num1, num2));
                    case 2 -> System.out.println("Resultado: " + calc.subtrair(num1,
num2));
                    case 3 -> System.out.println("Resultado: " + calc.multiplicar(num1,
num2));
                    case 4 -> System.out.println("Resultado: " + calc.dividir(num1,
num2));
                    default -> System.out.println("Opção inválida.");
                }
            } catch (ArithmeticException e) {
                System.out.println(e.getMessage());
            }
        }
    }

```

```
        scanner.close();
    }
}
```

3) Projete e implemente uma classe “Hora” para ser utilizada em sistemas diversos. A hora será representada até o nível de segundos e deve obedecer às regras básicas:

- Um objeto “Hora” só pode armazenar estados válidos.
- Um objeto “Hora” pode ser incrementado em minutos, segundos ou horas. Esta operação não modifica o valor da “Hora” original, sendo seu objetivo retornar uma nova “Hora” válida.
- Um objeto “Hora” pode ser comparado com outro para verificação de qual horário está mais adiante.

```
import java.util.Scanner;

public class Hora {
    private int horas;
    private int minutos;
    private int segundos;

    public Hora(int horas, int minutos, int segundos) {
        if (horas < 0 || horas > 23 || minutos < 0 || minutos > 59 || segundos < 0 || segundos > 59) {
            throw new IllegalArgumentException("Horário inválido.");
        }
        this.horas = horas;
        this.minutos = minutos;
        this.segundos = segundos;
    }

    public Hora adicionarMinutos(int m) {
        int novaHora = (this.horas + (this.minutos + m) / 60) % 24;
        int novoMinuto = (this.minutos + m) % 60;
        return new Hora(novaHora, novoMinuto, this.segundos);
    }

    @Override
    public String toString() {
        return String.format("%02d:%02d:%02d", horas, minutos, segundos);
    }

    public static int solicitarInteiro(String mensagem, Scanner scanner, int min, int max) {
        int valor;
        do {
```

```

        System.out.print(mensagem);
        while (!scanner.hasNextInt()) {
            System.out.print("Entrada inválida. " + mensagem);
            scanner.next();
        }
        valor = scanner.nextInt();
    } while (valor < min || valor > max);
    return valor;
}

@SuppressWarnings("ConvertToTryWithResources")
public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);

    int horas = solicitarInteiro("Digite a hora (0-23): ", scanner, 0, 23);
    int minutos = solicitarInteiro("Digite os minutos (0-59): ", scanner, 0, 59);
    int segundos = solicitarInteiro("Digite os segundos (0-59): ", scanner, 0, 59);

    Hora horaAtual = new Hora(horas, minutos, segundos);
    System.out.println("Hora definida: " + horaAtual);

    while (true) {
        System.out.println("\n1 - Adicionar minutos");
        System.out.println("2 - Sair");
        int opcao = scanner.nextInt();

        if (opcao == 2) {
            System.out.println("Programa encerrado.");
            break;
        }

        if (opcao == 1) {
            int min = solicitarInteiro("Quantos minutos deseja adicionar? ", scanner,
1, 1440);

            horaAtual = horaAtual.adicionarMinutos(min);
            System.out.println("Nova hora: " + horaAtual);
        } else {
            System.out.println("Opção inválida.");
        }
    }

    scanner.close();
}
}

```

4) Em um curso técnico, que tem uma carga total de 40 aulas, cada aluno é avaliado pelos seguintes critérios: a) Quatro notas de exercícios de 0 a 100, com peso de 20% na nota final.

b) Duas notas de prova de 0 a 100, com peso de 60% na nota final.

c) Um trabalho prático de 20 pontos, a ser somada com as anteriores para calcular a nota final.

Para ser aprovado, o aluno precisa atingir pelo menos 60 pontos na nota final, bem como cumprir no mínimo 75% de frequência às aulas. Projete e implemente uma classe "Aluno" que atenda aos requisitos descritos.

```
import java.util.Scanner;

public class Aluno {
    private final double[] exercicios = new double[4];
    private final double[] provas = new double[2];
    private double trabalho;
    private int frequencia;

    public void inserirNotas(Scanner scanner) {
        for (int i = 0; i < 4; i++) {
            exercicios[i] = solicitarDouble("Nota do exercício " + (i + 1) + ": ",
scanner);
        }
        for (int i = 0; i < 2; i++) {
            provas[i] = solicitarDouble("Nota da prova " + (i + 1) + ": ", scanner);
        }
        trabalho = solicitarDouble("Nota do trabalho prático: ", scanner);
        frequencia = solicitarInteiro("Número de aulas assistidas (0-40): ", scanner, 0,
40);
    }

    public double calcularNotaFinal() {
        double mediaExercicios = (exercicios[0] + exercicios[1] + exercicios[2] +
exercicios[3]) / 4 * 0.2;
        double mediaProvas = (provas[0] + provas[1]) / 2 * 0.6;
        return mediaExercicios + mediaProvas + trabalho;
    }

    public boolean aprovado() {
        return calcularNotaFinal() >= 60 && frequencia >= 30;
    }

    public static double solicitarDouble(String mensagem, Scanner scanner) {
        System.out.print(mensagem);
        while (!scanner.hasNextDouble()) {
            System.out.print("Entrada inválida. " + mensagem);
            scanner.next();
        }
        return scanner.nextDouble();
    }
}
```

```

    }

    public static int solicitarInteiro(String mensagem, Scanner scanner, int min, int
max) {
        int valor;
        do {
            System.out.print(mensagem);
            while (!scanner.hasNextInt()) {
                System.out.print("Entrada inválida. " + mensagem);
                scanner.next();
            }
            valor = scanner.nextInt();
        } while (valor < min || valor > max);
        return valor;
    }

    @SuppressWarnings("ConvertToTryWithResources")
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        Aluno aluno = new Aluno();
        aluno.inserirNotas(scanner);

        System.out.println("Nota final: " + aluno.calcularNotaFinal());
        System.out.println(aluno.aprovado() ? "Aprovado!" : "Reprovado.");

        scanner.close();
    }
}

```

5) Uma partida de basquete é disputada por duas equipes. A partida é dividida em 4 quartos de 10 minutos e será considerada vencedora a equipe que somar mais pontos na soma das pontuações dos quartos. Em caso de empate, serão disputadas quantas prorrogações de 5 minutos forem necessárias, até que uma equipe supere a outra em pontuação. Modele e implemente uma classe “PartidaBasquete” que registre os placares das equipes e, indique quem venceu a partida e quando solicitada, forneça o placar no formato abaixo:

```

Q1 Q2 EQUIPE1 25 19 EQUIPE2 22 25
Q3 Q4 FINAL 26 21 91
28 25 100

```

No caso de prorrogações, devem aparecer no formato acima como “P1”, “P2” etc., após o Q4.

```

import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;

```



```

public class PartidaBasquete {
    private final List<Integer> pontosEquipe1 = new ArrayList<>();
    private final List<Integer> pontosEquipe2 = new ArrayList<>();

    public PartidaBasquete() {
        for (int i = 0; i < 4; i++) {
            pontosEquipe1.add(0);
            pontosEquipe2.add(0);
        }
    }

    public void adicionarPontos(int quarto, int equipe, int pontos) {
        if (equipe == 1) {
            pontosEquipe1.set(quarto, pontosEquipe1.get(quarto) + pontos);
        } else {
            pontosEquipe2.set(quarto, pontosEquipe2.get(quarto) + pontos);
        }
    }

    public void adicionarProrrogaçao(int equipe1, int equipe2) {
        pontosEquipe1.add(equipe1);
        pontosEquipe2.add(equipe2);
    }

    public String obterPlacarFinal() {
        int total1 = pontosEquipe1.stream().mapToInt(Integer::intValue).sum();
        int total2 = pontosEquipe2.stream().mapToInt(Integer::intValue).sum();
        return total1 > total2 ? "Equipe 1 venceu!" : total1 < total2 ? "Equipe 2 venceu!" : "Empate!";
    }

    public void exibirPlacar() {
        System.out.print("\nQ1  Q2  Q3  Q4");
        for (int i = 4; i < pontosEquipe1.size(); i++) {
            System.out.print("  P" + (i - 3));
        }
        System.out.println("  FINAL");

        System.out.print("Equipe1 ");
        pontosEquipe1.forEach(p -> System.out.print(p + "  "));
        System.out.println();

        System.out.print("Equipe2 ");
        pontosEquipe2.forEach(p -> System.out.print(p + "  "));
        System.out.println();
    }
}

```

```

public static int solicitarInteiro(String mensagem, Scanner scanner) {
    System.out.print(mensagem);
    while (!scanner.hasNextInt()) {
        System.out.print("Entrada inválida. " + mensagem);
        scanner.next();
    }
    return scanner.nextInt();
}

@SuppressWarnings("ConvertToTryWithResources")
public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    PartidaBasquete partida = new PartidaBasquete();

    for (int i = 0; i < 4; i++) {
        System.out.println("Digite os pontos do Q" + (i + 1));
        int pontos1 = solicitarInteiro("Equipe 1: ", scanner);
        int pontos2 = solicitarInteiro("Equipe 2: ", scanner);
        partida.adicionarPontos(i, 1, pontos1);
        partida.adicionarPontos(i, 2, pontos2);
    }

    while (partida.obterPlacarFinal().equals("Empate!")) {
        System.out.println("\nEmpate! Digite os pontos da prorrogação:");
        int pontos1 = solicitarInteiro("Equipe 1: ", scanner);
        int pontos2 = solicitarInteiro("Equipe 2: ", scanner);
        partida.adicionarProrrogaçao(pontos1, pontos2);
    }

    partida.exibirPlacar();
    System.out.println("\n" + partida.obterPlacarFinal());

    scanner.close();
}
}

```

6) Uma mensagem criptografada é um tipo de mensagem que cria uma cifra a partir de uma mensagem de texto. A cifra é um texto que esconde a mensagem original e que pode, posteriormente, ser decifrada para mostrá-la. Utilizando seus conhecimentos de POO, implemente uma classe “Cifra” que usa o seguinte algoritmo: a mensagem original é dividida em blocos de N caracteres. A cifra será obtida lendo-se a posição correspondente de cada bloco, sendo inserido ainda um “*” a cada mudança de posição. Por exemplo, se a mensagem original for “mensagem secreta” e o bloco tiver tamanho 5:

mensa
gem s
ecret
a

A cifra seria "mgea*eec *nmr *s e *ast *" (atenção para os espaços em branco). A classe também deve fornecer a mensagem original, mediante um pedido do usuário informando o tamanho do bloco utilizado.

```
import java.util.Scanner;

public class Cifra {

    public static String cifrar(String msg, int bloco) {
        StringBuilder cifrado = new StringBuilder();
        int linhas = (int) Math.ceil((double) msg.length() / bloco);
        char[][] matriz = new char[linhas][bloco];

        for (int i = 0, index = 0; i < linhas; i++) {
            for (int j = 0; j < bloco && index < msg.length(); j++, index++) {
                matriz[i][j] = msg.charAt(index);
            }
        }

        for (int j = 0; j < bloco; j++) {
            for (int i = 0; i < linhas; i++) {
                if (matriz[i][j] != '\0') {
                    cifrado.append(matriz[i][j]);
                }
            }
            cifrado.append("*");
        }

        return cifrado.toString();
    }

    public static String decifrar(String cifra, int bloco) {
        String[] partes = cifra.split("\\*");
        int linhas = partes.length;
        int maxColunas = partes[0].length();

        char[][] matriz = new char[linhas][maxColunas];

        for (int j = 0; j < linhas; j++) {
            char[] caracteres = partes[j].toCharArray();
            System.arraycopy(caracteres, 0, matriz[j], 0, caracteres.length);
        }
    }
}
```

```

        StringBuilder mensagem = new StringBuilder();
        for (int i = 0; i < maxColunas; i++) {
            for (int j = 0; j < linhas; j++) {
                if (matriz[j][i] != '\0') {
                    mensagem.append(matriz[j][i]);
                }
            }
        }

        return mensagem.toString();
    }

    public static int solicitarInteiro(String mensagem, Scanner scanner, int min, int
max) {
        int valor;
        do {
            System.out.print(mensagem);
            while (!scanner.hasNextInt()) {
                System.out.print("Entrada inválida. " + mensagem);
                scanner.next();
            }
            valor = scanner.nextInt();
        } while (valor < min || valor > max);
        return valor;
    }

    @SuppressWarnings("ConvertToTryWithResources")
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        while (true) {
            System.out.println("\n1 - Cifrar mensagem");
            System.out.println("2 - Decifrar mensagem");
            System.out.println("3 - Sair");
            System.out.print("Escolha uma opção: ");
            int opcao = scanner.nextInt();
            scanner.nextLine();

            if (opcao == 3) {
                System.out.println("Programa encerrado.");
                break;
            }

            System.out.print("Digite a mensagem: ");
            String mensagem = scanner.nextLine();

            int bloco = solicitarInteiro("Digite o tamanho do bloco: ", scanner, 1,
mensagem.length());

```

```
        switch (opcao) {
            case 1 -> System.out.println("Mensagem cifrada: " + cifrar(mensagem,
bloco));
            case 2 -> System.out.println("Mensagem decifrada: " + decifrar(mensagem,
bloco));
            default -> System.out.println("Opção inválida.");
        }
    }

    scanner.close();
}
}
```