

```

public class App {

    //#region atributos e constantes
    static Scanner teclado;
    static int[] DIASDOMES = {0, 31, 28, 31, 30, 31,
                               30, 31, 31, 30, 31, 30, 31 };
    static String[] DIASDASEMANA = {"sábado", "domingo", "segunda", "terça", "quarta", "quinta", "sexta" };
    static final int ANO_ATUAL = 2022;
    //#endregion

    //#region Úteis

    /**
     * Encapsula uma leitura de teclado, com mensagem personalizada. Retorna uma string que pode ser posteriormente convertida.
     * @param mensagem A mensagem a ser exibida
     * @return String lida do usuário.
     */
    public static String leitura(String mensagem){
        System.out.print("\n"+mensagem+": ");
        return teclado.nextLine();
    }

    /**
     * Imprime todo um vetor de números inteiros. Demonstra o uso do "for" iterável ("foreach")
     * @param vet O vetor a ser impresso
     */
    public static void imprimirVetorDeInteiros(int[] vet){

        for (int i : vet) {
            System.out.print(i + " - ");
        }
    }

    /**
     * Troca dois elementos de posição em um vetor.
     * @param vet O vetor com os dados
     * @param p1 Primeira posição
     * @param p2 Segunda posição
     */
    public static void trocarPosicoes(int[] vet, int p1, int p2){
        int aux = vet[p1];
        vet[p1] = vet[p2];
        vet[p2] = aux;
    }
}

```

```

/**
 * Faz a leitura de um vetor de inteiros já criado.
 * @param vet O vetor para ser preenchido
 */
public static void lerVetorDeInteiros(int[] vet){
    for (int i = 0; i < vet.length; i++) {
        vet[i] = Integer.parseInt(leitura("Digite o " + (i+1) + "º número"));
    }
}

/**
 * Cria um vetor de inteiros a partir de uma linha CSV como 1;2;3;4.
 * @param Linha Linha CSV com valores inteiros no formato valor;valor;valor
 * @return Vetor de inteiros criado a partir da linha
 */
public static int[] criarVetorDeInteiros(String Linha){
    String[] valores = Linha.split(regex: ";");
    int[] vet = new int[valores.length];
    for (int i = 0; i < vet.length; i++) {
        vet[i] = Integer.parseInt(valores[i]);
    }
    return vet;
}

//#endregion

//#region Exercício 1

/**
 * Inverte um vetor de valores inteiros. Calcula o complementar da posição atual para inverter e pára ao chegar na metade do vetor
 * @param vet O vetor a ser invertido.
 */
public static void inverterVetorDeInteiros(int[] vet){
    for (int i = 0; i < (vet.length/2); i++) {
        int complementar = (vet.length-1)-i;
        trocarPosicoes(vet, i, complementar);
    }
}

//#endregion

//#region Exercício 2

/**
 * Soma os elementos vizinhos (0-1, 2-3, 4-5...) em um vetor de inteiros.
 * Cada resultado é armazenado em um novo vetor, o qual terá, portanto a metade do tamanho do vetor original.
 * Em caso de tamanho ímpar, o último elemento é somado consigo mesmo
 * @param vet O vetor com os dados
 * @return O novo vetor com os vizinhos somados
 */
public static int[] somarVizinhosEmVetorDeInteiros(int[] vet){
    int tamanhoNovoVetor = (vet.length/2)+1; //assume que é um vetor de tamanho ímpar
    int ultimoValor = vet[vet.length-1];

    int[] novoVetor = new int[tamanhoNovoVetor];

    for (int i = 0; i < vet.length-1; i+=2) {
        novoVetor[i/2] = vet[i]+vet[i+1];
    }

    if(vet.length%2==1){
        novoVetor[tamanhoNovoVetor-1] = ultimoValor+ultimoValor; //se for tamanho ímpar, guarda a soma do último consigo
    }
    else{
        novoVetor = Arrays.copyOf(novoVetor, tamanhoNovoVetor-1); //senão, descarta a última posição do novo vetor, em caso de tamanho par
    }
    return novoVetor;
}

//#endregion

```

```

//#region Exercício 3

/**
 * Verifica a validade de uma data no formato DD/MM/AAAA. Retorna uma string informando se a data é válida ou onde ela encontra-se errada.
 * As datas consideradas válidas se iniciam em 01/01/1900.
 * @param data String DD/MM/AAAA com a data a verificar.
 * @return String com informação da validação
 */
public static String dataValida(String data){
    String resposta="Data válida.";

    String[] detalhes = data.split(regex: "/");
    if(detalhes.length!=3 || data.length()!=10)
        return "Data em formato incorreto.";

    int dia = Integer.parseInt(detalhes[0]);
    int mes = Integer.parseInt(detalhes[1]);
    int ano = Integer.parseInt(detalhes[2]);
    int maxDia;

    if(ano<1900)
        resposta = "Ano inválido: só de 1900 em diante.";
    else{
        if (mes < 1 || mes > 12) //mês<1 ou mês>12 --> data inválida
            resposta = "Mês inválido: só de 1 a 12.";
        else {
            maxDia = DIASDOMES[mes];
            if(anoBissexto(ano)&&mes==2) //para fevereiro de bissextos...
                maxDia++;

            if (dia > maxDia) //verifica validade de acordo com o mês atual
                resposta = "Dia inválido: máximo de "+maxDia+" para o mês "+mes+ ".";
        }
    }
    return resposta;
}

/**
 * Verificação de ano bissexto. Retorna verdadeiro ou falso.
 * @param ano Ano a ser verificado.
 * @return true para ano bissexto, false caso contrário.
 */
public static boolean anoBissexto(int ano){
    boolean resposta = false;
    if(ano%400==0)
        resposta = true;
    else if(ano%4==0 && ano%100!=0)
        resposta = true;

    return resposta;
}
//#endregion

```

```

//#region Exercício 4
/**
 * Calcula e retorna o dia da semana para uma data de 2022.
 * @param data Data no formato DD/MM/AAAA. O ano sempre será ajustado para 2022.
 * @return Mensagem contendo a data validada e a informação de seu dia na semana.
 */
public static String diaDaSemana(String data){
    String resposta = "";
    data = data.substring(beginIndex: 0, endIndex: 6)+"2022";
    String validacao = dataValida(data);
    int deslocamento = 0;

    if(!validacao.equals(anObject: "Data válida.")){
        resposta = validacao;
    }
    else{
        String[] detalhes = data.split(regex: "/");
        int dia = Integer.parseInt(detalhes[0]);
        int mes = Integer.parseInt(detalhes[1]);

        int totalDias = 0;
        int mesParaSomar = 1;

        while(mesParaSomar!=mes){ //somamos todos os dias dos meses anteriores ao atual
            totalDias += DIASDOMES[mesParaSomar];
            mesParaSomar++;
        }
        totalDias += (dia-1); //somamos os dias para chegar na data dentro do mês
        deslocamento = totalDias%7; //o deslocamento em dias no vetor contante de dias da semana indica o dia da data.
        resposta = "O dia "+ String.format(format: "%02d", dia)+"/"+String.format(format: "%02d",mes)+"/"+"2022 cairá na(o) "+DIASDASEMANA[deslocamento]+".";
    }
    return resposta;
}
}
//#endregion

//#region Exercícios 5 e 6

/**
 * Faz o mesmo que o exercício 2 (soma de vizinhos), mas com dados de arquivo e, portanto, múltiplas linhas. Imprime o resultado para cada linha lida.
 * @param nomeArquivo O nome do arquivo com os dados
 * @throws Exception Exceção genérica ⚠ para problemas de leitura de arquivos
 */
public static void somarVizinhosInteirosDeArquivo(String nomeArquivo) throws Exception{
    Scanner leitor = new Scanner(new File(nomeArquivo));
    int[] vetor;
    while(leitor.hasNextLine()){
        String linha = leitor.nextLine();
        System.out.println();
        vetor = criarVetorDeInteiros(linha); //usando método utilitário
        vetor = somarVizinhosEmVetorDeInteiros(vetor); //mesma ação do exercício 2
        imprimirVetorDeInteiros(vetor);
    }
    leitor.close();
}

/**
 * Faz o mesmo que os exercício 3 e 4 (data válida e dia da semana), mas com dados de arquivo e, portanto, múltiplas datas.
 * Imprime o resultado para cada linha lida, aproveitando o resultado da data válida para a resposta. Datas inválidas são reportadas.
 * O dia da semana sempre é relativo a 2022, independentemente da data original.
 * @param nomeArquivo O nome do arquivo com os dados;
 * @throws Exception Exceção genérica ⚠ para problemas de leitura de arquivos
 */
public static void operacoesDataComArquivos(String nomeArquivo) throws Exception{
    Scanner leitor = new Scanner(new File(nomeArquivo));
    int quantos = Integer.parseInt(leitor.nextLine());
    String valida="";
    for (int i = 0; i < quantos; i++) {
        String data = leitor.nextLine();
        valida = dataValida(data);
        System.out.print(valida);
        if(!valida.toLowerCase().contains(s: "inváli"))
            System.out.println(" "+diaDaSemana(data));
        else
            System.out.println(" - "+data);
    }
    leitor.close();
}
}
//#endregion

```