

Programação 1

Aula 8

Valeri Skliarov, Prof. Catedrático

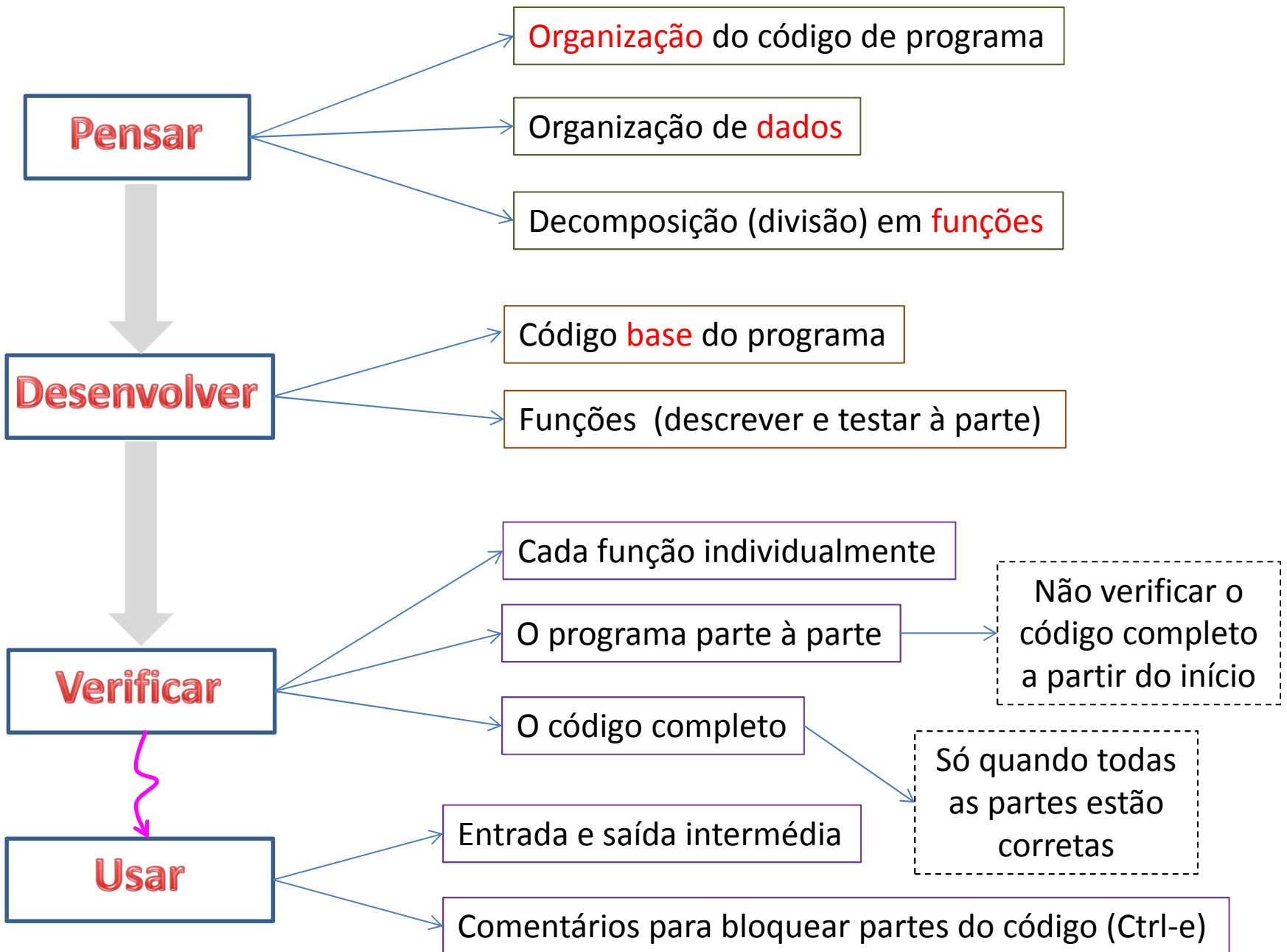
Email: skl@ua.pt

URL: <http://sweet.ua.pt/skl/>

Departamento de Eletrónica, Telecomunicações e Informática
Universidade de Aveiro

<http://elearning.ua.pt/>

Revisão das aulas anteriores



Organização base útil

```
import java.util.*;
```

Parte 1: incluir bibliotecas

```
public class <nome da classe base> {  
    static Scanner sc = new Scanner(System.in);  
    public static void main (String args[]) {
```

Parte 2: classe base

Parte 3: função *main*

```
        public static <tipo retorno> <nome da função>(<tipos e nomes de argumentos separados por virgulas>) {  
            // corpo da função  
        }
```

Parte 5: funções

```
class <nome da classe>{  
    // declaração de dados  
}
```

Parte 4: classes para dados de utilizador

Não esquecer

Exercício 7.1

O programa `TestaHora.java` usa um novo tipo de dados `Hora` que permite guardar uma hora do dia, mas está incompleto. Modifique este programa para que funcione devidamente. Para isso:

- Defina um novo tipo de dados `Hora` que contenha campos inteiros para guardar uma hora do dia no formato hora $[0, 23]$, minutos $[0, 59]$ e segundos $[0, 59]$.
- Escreva uma função para escrita de uma hora no formato `HH:MM:SS`, por exemplo: `09:23:05`.
- Escreva uma função para leitura e validação interativa de uma hora.

```
/**
EXEMPLO do pretendido:
$ java TestaHora
Começou às 09:23:05.
Quando termina?
horas ? 11
minutos ? 72
minutos ? 7
segundos ? 2
Início: 09:23:05 Fim: 11:07:02.
**/
```

```
Começou as                09:23:05.
Quando termina?
horas ? 11
minutos ? 72
minutos ? 7
segundos ? 2
Início:                09:23:05. Fim:                11:07:02.
Press any key to continue . . . _
```

```

import java.util.*;

public class ex7_1 {
    static Scanner sc = new Scanner(System.in);
    public static void main(String[] args) {
        Hora inicio = new Hora();
        inicio.h = 9;
        inicio.m = 23;
        inicio.s = 5;

        Hora fim;

        System.out.print("Começou as ");
        printHora(inicio);
        System.out.println("\nQuando termina?");
        fim = lerHora();
        System.out.print("Inicio:");
        printHora(inicio);
        System.out.print(" Fim: ");
        printHora(fim);
        System.out.println();
    }
}

```

```

public static void printHora(Hora ob)
{
    System.out.printf("\t\t%02d:%02d:%02d.",
ob.h,ob.m,ob.s);
}

public static Hora lerHora()
{
    Hora tmp = new Hora();

    do
    {
        System.out.print("horas ? ");
        tmp.h = sc.nextInt();
    } while(tmp.h < 0 || tmp.h >= 24);

    do
    {
        System.out.print("minutos ? ");
        tmp.m = sc.nextInt();
    } while(tmp.m < 0 || tmp.m >= 60);

    do
    {
        System.out.print("segundos ? ");
        tmp.s = sc.nextInt();
    } while(tmp.s < 0 || tmp.s >= 60);

    return tmp;
}

class Hora
{
    int h,m,s;
}

```

```
import java.util.*; // parte 1
```

```
public class ex7_1 { // parte 2
    static Scanner sc = new Scanner(System.in);
```

```
public static void main(String[] args) { // parte 3
    }
}
```

```
public static void printHora(Hora ob) // parte 5
{
}
```

```
public static Hora lerHora()
{
}
}
```

```
}
```

```
import java.util.*; // parte 1
```

```
public class ex7_1 { // parte 2
    static Scanner sc = new Scanner(System.in);
```

```
public static void main(String[] args) { // parte 3
    }
}
```

```
public static void printHora(Hora ob) // parte 5
{ System.out.printf("\t\t%02d:%02d:%02d.", ob.h,ob.m,ob.s); }
```

```
public static Hora lerHora()
{
}
}
```

```
}
```

```
class Hora // parte 4
{ int h,m,s; }
```

```
class Hora // parte 4
{ int h,m,s; }
```

```
public static void main(String[] args) { // verificação
    Hora inicio = new Hora();
        inicio.h = 9;
        inicio.m = 23;
        inicio.s = 5;
    System.out.print("Começou as ");
    printHora(inicio);
}
```

```
import java.util.*;
public class ex7_1 {
    static Scanner sc = new Scanner(System.in);
    public static void main(String[] args) {
        Hora inicio = new Hora();
        inicio.h = 9;
        inicio.m = 23;
        inicio.s = 5;
        System.out.print("Comecou as ");
        printHora(inicio);
    }
    public static void printHora(Hora ob)
    {
        System.out.printf("\t\t%02d:%02d:%02d.", ob.h,ob.m,ob.s);
    }
    //~ public static Hora lerHora()
    //~ {
    //~ }
}

class Hora
{
    int h,m,s;
}
```

```
09:23:05.Press any key to continue . . .
```



```

public static Hora lerHora()
{
    Hora tmp = new Hora();
    do
    {
        System.out.print("horas ? ");
        tmp.h = sc.nextInt();
    } while(tmp.h < 0 || tmp.h >= 24);

    do
    {
        System.out.print("minutos ? ");
        tmp.m = sc.nextInt();
    } while(tmp.m < 0 || tmp.m >= 60);

    do
    {
        System.out.print("segundos ? ");
        tmp.s = sc.nextInt();
    } while(tmp.s < 0 || tmp.s >= 60);

    return tmp;
}

```

```

Quando termina?
horas ? 3
minutos ? 4
segundos ? 5
Fim:          03:04:05.
Press any key to continue . . .

```

Valeri Skliarov
2015/2016

Verificação

```

import java.util.*;
public class ex7_1 {
    static Scanner sc = new Scanner(System.in);
    public static void main(String[] args) {
        Hora inicio = new Hora();
        inicio.h = 9;
        inicio.m = 23;
        inicio.s = 5;
        Hora fim;
        //~ System.out.print("Começou as ");
        //~ printHora(inicio);
        System.out.println("\nQuando termina?");
        fim = lerHora();
        System.out.print(" Fim: ");
        printHora(fim);
        System.out.println();
    }
    public static void printHora(Hora ob)
    {System.out.printf("\t\t%02d:%02d:%02d.", ob.h,ob.m,ob.s);}
    public static Hora lerHora()
    {
        Hora tmp = new Hora();
        do {
            System.out.print("horas ? ");
            tmp.h = sc.nextInt();
        } while(tmp.h < 0 || tmp.h >= 24);
        do {
            System.out.print("minutos ? ");
            tmp.m = sc.nextInt();
        } while(tmp.m < 0 || tmp.m >= 60);
        do {
            System.out.print("segundos ? ");
            tmp.s = sc.nextInt();
        } while(tmp.s < 0 || tmp.s >= 60);
        return tmp;
    }
}
class Hora
{
    int h,m,s;
}

```

Finalização

```
import java.util.*;

public class ex7_1 {
    static Scanner sc = new Scanner(System.in);
    public static void main(String[] args) {
        Hora inicio = new Hora();
        inicio.h = 9;
        inicio.m = 23;
        inicio.s = 5;

        Hora fim;

        System.out.print("Comecou as ");
        printHora(inicio);
        System.out.println("\nQuando termina?");
        fim = lerHora();
        System.out.print("Inicio:");
        printHora(inicio);
        System.out.print(" Fim: ");
        printHora(fim);
        System.out.println();
    }
}
```

```
public static void printHora(Hora ob)
{
    System.out.printf("\t\t%02d:%02d:%02d.",
ob.h,ob.m,ob.s);
}

public static Hora lerHora()
{
    Hora tmp = new Hora();
    do
    {
        System.out.print("horas ? ");
        tmp.h = sc.nextInt();
    } while(tmp.h < 0 || tmp.h >= 24);
    do
    {
        System.out.print("minutos ? ");
        tmp.m = sc.nextInt();
    } while(tmp.m < 0 || tmp.m >= 60);
    do
    {
        System.out.print("segundos ? ");
        tmp.s = sc.nextInt();
    } while(tmp.s < 0 || tmp.s >= 60);

    return tmp;
}

class Hora
{
    int h,m,s;
}
```

```
Comecou as                09:23:05.
Quando termina?
horas ? 11
minutos ? 72
minutos ? 7
segundos ? 2
Inicio:                09:23:05. Fim:                11:07:02.
Press any key to continue . . . _
```

Tarefa

```
import java.util.*;

public class ex7_1 {
    static Scanner sc = new Scanner(System.in);
    public static void main(String[] args) {
        Hora inicio = new Hora();
        inicio.h = 9;
        inicio.m = 23;
        inicio.s = 5;

        Hora fim;

        System.out.print("Comecou as ");
        printHora(inicio);
        System.out.println("\nQuando termina?");
        fim = lerHora();
        System.out.print("Inicio:");
        printHora(inicio);
        System.out.print(" Fim: ");
        printHora(fim);
        System.out.println();
    }
}
```

Tarefa:
transformar
em função

Copiar e colar

```
import java.util.*;

public class ex7_1 {
    static Scanner sc = new Scanner(System.in);
    public static void main(String[] args) {
        Hora inicio = new Hora();
        inicio.h = 9;
        inicio.m = 23;
        inicio.s = 5;

        Hora fim;

        }

        Cabeçalho da função
        {
            1
        }
    }
}
```

```
Comecou as                09:23:05.
Quando termina?
horas ? 11
minutos ? 72
minutos ? 7
segundos ? 2
Inicio:                09:23:05. Fim:                11:07:02.
Press any key to continue . . . _
```

```

import java.util.*;

public class ex7_1 {
    static Scanner sc = new Scanner(System.in);
    public static void main(String[] args) {
        Hora inicio = new Hora();
        inicio.h = 9;
        inicio.m = 23;
        inicio.s = 5;

        Hora fim;
    }
    Cabeçalho da função
    {
        System.out.print("Comecou as ");
        printHora(inicio);
        System.out.println("\nQuando termina?");
        fim = lerHora();
        System.out.print("Inicio:");
        printHora(inicio);
        System.out.print(" Fim: ");
        printHora(fim);
        System.out.println();
    }
}

```

```

import java.util.*;

public class ex7_1 {
    static Scanner sc = new Scanner(System.in);
    public static void main(String[] args) {
        Hora inicio = new Hora();
        inicio.h = 9;
        inicio.m = 23;
        inicio.s = 5;

        Hora fim;
    }
    static void funcao_nova(Hora inicio) // alterar o cabeçalho
    {
        System.out.print("Comecou as ");
        printHora(inicio);
        System.out.println("\nQuando termina?");
        fim = lerHora();
        System.out.print("Inicio:");
        printHora(inicio);
        System.out.print(" Fim: ");
        printHora(fim);
        System.out.println();
    }
}

```

3

```

Comecou as                09:23:05.
Quando termina?
horas ? 11
minutos ? 72
minutos ? 7
segundos ? 2
Inicio:                09:23:05. Fim:                11:07:02.
Press any key to continue . . . _

```

```

import java.util.*;

public class ex7_1 {
    static Scanner sc = new Scanner(System.in);
    public static void main(String[] args) {
        Hora inicio = new Hora();
        inicio.h      = 9;
        inicio.m      = 23;
        inicio.s      = 5;

        // Hora fim;
    }
    static void funcao_nova(Hora inicio) // Declarar variáveis
    {
        Hora fim;
        System.out.print("Comecou as ");
        printHora(inicio);
        System.out.println("\nQuando termina?");
        fim = lerHora();
        System.out.print("Inicio:");
        printHora(inicio);
        System.out.print(" Fim: ");
        printHora(fim);
        System.out.println();
    }
}

```

4

```

import java.util.*;

public class ex7_1 {
    static Scanner sc = new Scanner(System.in);
    public static void main(String[] args) {
        Hora inicio = new Hora();
        inicio.h      = 9;
        inicio.m      = 23;
        inicio.s      = 5;

        // Hora fim;
        funcao_nova(inicio); // Chamar função
    }
    static void funcao_nova(Hora inicio) // Declarar variáveis
    {
        Hora fim;
        System.out.print("Comecou as ");
        printHora(inicio);
        System.out.println("\nQuando termina?");
        fim = lerHora();
        System.out.print("Inicio:");
        printHora(inicio);
        System.out.print(" Fim: ");
        printHora(fim);
        System.out.println();
    }
}

```

5

```

Quando termina?
horas ? 3
minutos ? 4
segundos ? 5
Fim: 03:04:05.
Press any key to continue . . .

```

Verificação

```
import java.util.*;
public class ex7_1 {
    static Scanner sc = new Scanner(System.in);
    public static void main(String[] args) {
        Hora inicio = new Hora();
        inicio.h      = 9;
        inicio.m      = 23;
        inicio.s      = 5;

        funcao_nova(inicio);
    }
    static void funcao_nova(Hora inicio) // alterar o cabeçalho
    {
        Hora fim;
        System.out.print("Começou as ");          printHora(inicio);
        System.out.println("\nQuando termina?");
        fim = lerHora();
        System.out.print("Inicio:");              printHora(inicio);
        System.out.print(" Fim: ");               printHora(fim);
        System.out.println();
    }
    public static void printHora(Hora ob)
    {
        System.out.printf("\t\t%02d:%02d:%02d.", ob.h,ob.m,ob.s);
    }
    public static Hora lerHora()
    {
        Hora tmp = new Hora();
        do {
            System.out.print("horas ? ");
            tmp.h = sc.nextInt();
        } while(tmp.h < 0 || tmp.h >= 24);
        do {
            System.out.print("minutos ? ");
            tmp.m = sc.nextInt();
        } while(tmp.m < 0 || tmp.m >= 60);
        do {
            System.out.print("segundos ? ");
            tmp.s = sc.nextInt();
        } while(tmp.s < 0 || tmp.s >= 60);

        return tmp;
    }
}
class Hora
{
    int h,m,s;
}
```

```
Quando termina?
horas ? 3
minutos ? 4
segundos ? 5
Fim:                03:04:05.
Press any key to continue . . .
```

Exercício 6.3

Pretende-se escrever um programa que leia do teclado uma sequência de números inteiros positivos e que permita detetar um conjunto de características acerca da sequência. A cada operação deverá corresponder uma função. A leitura da sequência termina quando aparecer o número zero como indicador de paragem ou quando tiverem sido lidos 50 números. A interação com o programa deverá ser feita através de um menu, tal como apresentado de seguida:

Análise de uma sequência de números inteiros

- 1 - Ler a sequência
 - 2 - Escrever a sequência
 - 3 - Calcular o máximo da sequência
 - 4 - Calcular o mínimo da sequência
 - 5 - Calcular a média da sequência
 - 6 - Detetar se é uma sequência só constituída por números pares
 - 10 - Terminar o programa
- Opção ->

```
import java.util.*; // parte 1
```

```
public class ex6_3 { // parte 2
    static Scanner read = new Scanner(System.in);
```

```
public static void main(String[] args) { // parte 3
    }
}
```

```
public static int printMenu() { // parte 5
```

```
    int op = 0;
```

```
    return op; }
```

```
public static void readSeq(int[] num_array) {
}
```

```
public static void printSeq(int[] num_array) {
}
```

```
public static void printMax(int[] num_array) {
}
```

```
public static void printMin(int[] num_array) {
}
```

```
public static void printMed(int[] num_array) {
}
```

```
public static void printIfEven(int[] num_array) {
}
```

```
}
```

```
1 - Ler a sequência
2 - Escrever a sequência
3 - Calcular o máximo da sequência
4 - Calcular o mínimo da sequência
5 - Calcular a média da sequência
6 - Detetar se é uma sequência só constituída por números pares
10 - Terminar o programa
Opção ->
```

Não temos a parte 4


```

public static void main (String args[]) {
    int num_array[] = new int[50];
    int op = 0;
    do {
        op = printMenu();
        switch(op) {
            case 1: readSeq(num_array); break;
            case 2: printSeq(num_array); break;
            case 3: printMax(num_array); break;
            case 4: printMin(num_array); break;
            case 5: printMed(num_array); break;
            case 6: printfEven(num_array); break;
        }
        while(op != 10);
        System.out.println("O programa vai terminar...");
    }
}

```

```

public static int printMenu()           { // parte 5
    int op = 0;
    return op;
}
public static void readSeq(int[] num_array) {
}
public static void printSeq(int[] num_array) {
}
public static void printMax(int[] num_array) {
}
public static void printMin(int[] num_array) {
}
public static void printMed(int[] num_array) {
}
public static void printfEven(int[] num_array) {
}

```

```

1 - Ler a sequência
2 - Escrever a sequência
3 - Calcular o máximo da sequência
4 - Calcular o mínimo da sequência
5 - Calcular a média da sequência
6 - Detetar se é uma sequência só constituída por números pares
10 - Terminar o programa
Opção ->

```

Verificar cada função

```
public static void main (String args[]) {  
    int num_array[] = new int[50];  
    int op = 0;  
    do {  
        op = printMenu();  
        switch(op) {  
            //case 1:readSeq(num_array);      break;  
            //case 2:printSeq(num_array);      break;  
            //case 3:printMax(num_array);      break;  
            //case 4:printMin(num_array);      break;  
            //case 5:printMed(num_array);      break;  
            //case 6:printIfEven(num_array);   break;  
        }  
        } while(op != 10);  
    System.out.println("O programa vai terminar...");  
}
```

```
public static int printMenu() {  
    System.out.println("\nAnalise de uma sequencia de numeros inteiros");  
    System.out.println(" 1 - Ler a sequencia");  
    System.out.println(" 3 - Calcular o maximo da sequencia");  
    System.out.println(" 5 - Calcular a media da sequencia");  
    System.out.println(" 6 - Detetar se e uma sequencia so constituida por numeros pares");  
    System.out.println(" 10 - Terminar o programa");  
    int op = 0;  
    do {  
        op = read.nextInt();  
        if(!((op > 0 && op < 7) || op == 10)) System.out.print("Opção inválida!\nOpção -> ");  
    } while(!((op > 0 && op < 7) || op == 10));  
    return op;  
}
```

```
Analise de uma sequencia de numeros inteiros  
1 - Ler a sequencia  
2 - Escrever a sequencia  
3 - Calcular o maximo da sequencia  
4 - Calcular o minimo da sequencia  
5 - Calcular a media da sequencia  
6 - Detetar se e uma sequencia so constituida por numeros pares  
10 - Terminar o programa  
Opcao -> 10  
O programa vai terminar...  
Press any key to continue . . . _
```

```

public static void main (String args[]) {
    int num_array[] = new int[50];
    int op = 0;
    do {
        op = printMenu();
        switch(op) {
            case 1: readSeq(num_array); break;
            //case 2: printSeq(num_array); break;
            //case 3: printMax(num_array); break;
            //case 4: printMin(num_array); break;
            //case 5: printMed(num_array); break;
            //case 6: printIfEven(num_array); break;
        }
    } while(op != 10);
    System.out.println("O programa vai terminar...");
}

```

```

public static void readSeq(int[] num_array) {
    int n = 0, tmp;
    do {
        System.out.println("Insira valores:");
        System.out.printf("Valor[%d]: ", n);
        tmp = read.nextInt();
        if(tmp > 0)
            num_array[n++] = tmp;
        else if(tmp < 0)
            System.out.println("Valor inválido!");
    } while(tmp != 0);
}

```

```

Analise de uma sequencia de numeros inteiros
1 - Ler a sequencia
2 - Escrever a sequencia
3 - Calcular o maximo da sequencia
4 - Calcular o minimo da sequencia
5 - Calcular a media da sequencia
6 - Detetar se e uma sequencia so constituida por numeros pares
10 - Terminar o programa
Opcao -> 7
Opcao invalida!
Opcao -> 1
Insira valores:
Valor[0]: 2
Insira valores:
Valor[1]: 5
Insira valores:
Valor[2]: -1
Valor invalido!
Insira valores:
Valor[2]: 8
Insira valores:
Valor[3]: 0

Analise de uma sequencia de numeros inteiros
1 - Ler a sequencia
2 - Escrever a sequencia
3 - Calcular o maximo da sequencia
4 - Calcular o minimo da sequencia
5 - Calcular a media da sequencia
6 - Detetar se e uma sequencia so constituida por numeros pares
10 - Terminar o programa
Opcao -> 10
O programa vai terminar...
Press any key to continue . . . _

```

```

public static void main (String args[]) {
    int num_array[] = new int[50];
    int op = 0;
    do {
        op = printMenu();
        switch(op) {
            case 1: readSeq(num_array);
            case 2: printSeq(num_array);
            //case 3: printMax(num_array);
            //case 4: printMin(num_array);
            //case 5: printMed(num_array);
            //case 6: printIfEven(num_array);
        }
        } while(op != 10);
    System.out.println("O programa vai terminar...");
}

```

```

public static void printSeq(int[] num_array) {
    int n = 0;
    do {
        if(num_array[n] > 0)
            System.out.print(" " + num_array[n++] + " ");
        } while(num_array[n] != 0);
}

```

```

Analise de uma sequencia de numeros inteiros
1 - Ler a sequencia
2 - Escrever a sequencia
3 - Calcular o maximo da sequencia
4 - Calcular o minimo da sequencia
5 - Calcular a media da sequencia
6 - Detetar se e uma sequencia so constituida por numeros pares
10 - Terminar o programa
Opcao -> 1
Insira valores:
Valor[0]: 2
Insira valores:
Valor[1]: 3
Insira valores:
Valor[2]: 4
Insira valores:
Valor[3]: 0

Analise de uma sequencia de numeros inteiros
1 - Ler a sequencia
2 - Escrever a sequencia
3 - Calcular o maximo da sequencia
4 - Calcular o minimo da sequencia
5 - Calcular a media da sequencia
6 - Detetar se e uma sequencia so constituida por numeros pares
10 - Terminar o programa
Opcao -> 2
2 3 4

Analise de uma sequencia de numeros inteiros
1 - Ler a sequencia
2 - Escrever a sequencia
3 - Calcular o maximo da sequencia
4 - Calcular o minimo da sequencia
5 - Calcular a media da sequencia
6 - Detetar se e uma sequencia so constituida por numeros pares
10 - Terminar o programa
Opcao ->

```

```

public static void main (String args[]) {
    int num_array[] = new int[50];
    int op = 0;
    do {
        op = printMenu();
        switch(op) {
            case 1: readSeq(num_array); break;
            case 2: printSeq(num_array); break;
            case 3: printMax(num_array); break;
            case 4: printMin(num_array); break;
            case 5: printMed(num_array); break;
            case 6: printIfEven(num_array); break;
        }
    } while(op != 10);
    System.out.println("O programa vai terminar...");
}

```

```

public static void printIfEven(int[] num_array) {
    int n=0;
    boolean onlyEven = true;
    do {
        if(num_array[n++] % 2 != 0) onlyEven = false;
    } while(num_array[n] != 0);
    if(onlyEven) System.out.println("São todos pares");
    else System.out.println("Existem impares");
}

```

```

public static void printMax(int[] num_array) {
    int n=0, tmp=0;
    do { if(num_array[n] > tmp) tmp = num_array[n];
        n++;
    } while(num_array[n] != 0);
    System.out.printf("Máximo: %d", tmp);
}

```

```

public static void printMin(int[] num_array) {
    int n=0, tmp=Integer.MAX_VALUE;
    do { if(num_array[n] < tmp) tmp = num_array[n];
        n++;
    } while(num_array[n] != 0);
    System.out.printf("Mínimo: %d", tmp);
}

```

```

public static void printMed(int[] num_array) {
    int n=0, tmp=0;
    do { if(num_array[n] > 0) tmp += num_array[n++];
    } while(num_array[n] != 0);
    tmp /= n;
    System.out.printf("Média: %d", tmp);
}

```

```

Analise de uma sequencia de numeros inteiros
1 - Ler a sequencia
2 - Escrever a sequencia
3 - Calcular o maximo da sequencia
4 - Calcular o minimo da sequencia
5 - Calcular a media da sequencia
6 - Detetar se e uma sequencia so constituida por numeros pares
10 - Terminar o programa

```

```

Opcao -> 1
Insira valores:
Valor[0]: 2
Insira valores:
Valor[1]: 88
Insira valores:
Valor[2]: 24
Insira valores:
Valor[3]: 0

```

```

Analise de uma sequencia de numeros inteiros
1 - Ler a sequencia
2 - Escrever a sequencia
3 - Calcular o maximo da sequencia
4 - Calcular o minimo da sequencia
5 - Calcular a media da sequencia
6 - Detetar se e uma sequencia so constituida por numeros pares
10 - Terminar o programa

```

```

Opcao -> 3
Maximo: 88

```

```

Analise de uma sequencia de numeros inteiros
1 - Ler a sequencia
2 - Escrever a sequencia
3 - Calcular o maximo da sequencia
4 - Calcular o minimo da sequencia
5 - Calcular a media da sequencia
6 - Detetar se e uma sequencia so constituida por numeros pares
10 - Terminar o programa

```

```

Opcao -> 4
Minimo: 2

```

```

Analise de uma sequencia de numeros inteiros
1 - Ler a sequencia
2 - Escrever a sequencia
3 - Calcular o maximo da sequencia
4 - Calcular o minimo da sequencia
5 - Calcular a media da sequencia
6 - Detetar se e uma sequencia so constituida por numeros pares
10 - Terminar o programa

```

```

Opcao -> 5
Media: 38

```

```

Analise de uma sequencia de numeros inteiros
1 - Ler a sequencia
2 - Escrever a sequencia
3 - Calcular o maximo da sequencia
4 - Calcular o minimo da sequencia
5 - Calcular a media da sequencia
6 - Detetar se e uma sequencia so constituida por numeros pares
10 - Terminar o programa

```

```

Opcao -> 6

```

Sao todos pares

```

Analise de uma sequencia de numeros inteiros
1 - Ler a sequencia
2 - Escrever a sequencia
3 - Calcular o maximo da sequencia
4 - Calcular o minimo da sequencia
5 - Calcular a media da sequencia
6 - Detetar se e uma sequencia so constituida por numeros pares
10 - Terminar o programa

```

```

Opcao -> 10

```

O programa vai terminar...
Press any key to continue . . .

Verificar o programa

Tarefa

1. Gravar valores máximo, mínimo e média de array num objeto da classe *dados*:

```
class dados    {  
                int    maximo, minimo;  
                double  media;  
            }
```

2. Implementar a função *imprimirDados* que permite imprimir dados gravados em objeto *dados*

Passo 1: Alterar o código do programa

```
public static void main (String args[]) {
    int num_array[] = new int[50];
    int op = 0;
    do      { op = printMenu();
    switch(op)
    {
        case 1:  readSeq(num_array);      break;
        case 2:  printSeq(num_array);     break;
        case 3:
            System.out.printf("Maximo: %d", printMax(num_array));
                                   break;

        case 4:
            System.out.printf("Minimo: %d", printMin(num_array));
                                   break;

        case 5:
            System.out.printf("Media: %f", printMed(num_array) );
                                   break;

        case 6:  printIfEven(num_array);   break;
    }

    } while(op != 10);
    dados my;
    my = gravarDados(num_array);
    imprimirDados(my);
    System.out.println("O programa vai terminar...");
}
```

```
public static int printMax(int[] num_array)    {
    int n=0, tmp=0;
    do      {
        if(num_array[n] > tmp) tmp = num_array[n];
        n++;
    } while(num_array[n] != 0);
    //System.out.printf("Maximo: %d", tmp);
    return tmp;
}

public static int printMin(int[] num_array)    {
    int n=0, tmp=Integer.MAX_VALUE;
    do      {
        if(num_array[n] < tmp) tmp = num_array[n];
        n++;
    } while(num_array[n] != 0);
    //System.out.printf("Minimo: %d", tmp);
    return tmp;
}

public static double printMed(int[] num_array) {
    int n=0;
    double tmp = 0;
    do      {
        if(num_array[n] > 0)
            tmp += (double)num_array[n++];
    } while(num_array[n] != 0);

    tmp /= n;
    //System.out.printf("Media: %f", tmp);
    return tmp;
}
```


Passo 2: Adicionar a classe *dados* e implementar funções novas

```
dados my;  
my = gravarDados(num_array);  
imprimirDados(my);
```

```
static dados gravarDados(int[] ar) {  
    dados d = new dados();  
    d.maximo = printMax(ar);  
    d.minimo = printMin(ar);  
    d.media = printMed(ar);  
    return d;  
}  
  
static void imprimirDados(dados dd) {  
    System.out.printf("Maximo: %d\n", dd.maximo);  
    System.out.printf("Minimo: %d\n", dd.minimo);  
    System.out.printf("Media: %f\n", dd.media);  
}
```

```
class dados {  
    int      maximo, minimo;  
    double  media;  
}
```

```
dados my;  
my = gravarDados(num_array);  
imprimirDados(my);
```



```
imprimirDados(gravarDados(num_array));
```

```

import java.util.*;
public class e6_3 {
    static Scanner read = new Scanner(System.in);
    public static void main (String args[]) {
        int num_array[] = new int[50]; int op = 0;
        do{ op = printMenu();
        switch(op) {
            case 1: readSeq(num_array); break;
            case 2: printSeq(num_array); break;
            case 3:
                System.out.printf("Maximo: %d", printMax(num_array));
                break;
            case 4:
                System.out.printf("Minimo: %d", printMin(num_array));
                break;
            case 5:
                System.out.printf("Media: %f", printMed(num_array));
                break;
            case 6: printIfEven(num_array); break;
        }
        } while(op != 10);

        imprimirDados(gravarDados(num_array));
        System.out.println("O programa vai terminar...");
    }

    public static int printMenu() {
        System.out.println("\nAnalise de uma sequencia de numeros inteiros");
        System.out.println(" 1 - Ler a sequencia");
        System.out.println(" 2 - Escrever a sequencia");
        System.out.println(" 3 - Calcular o maximo da sequencia");
        System.out.println(" 4 - Calcular o minimo da sequencia");
        System.out.println(" 5 - Calcular a media da sequencia");
        System.out.println(" 6 - Detetar se e uma sequencia so"+
            "constituída por numeros pares");
        System.out.println(" 10 - Terminar o programa");
        System.out.print("Opcao -> ");
        int op = 0;
        do
        { op = read.nextInt();
        if(!((op > 0 && op < 7) || op == 10))
            System.out.print("Opcao invalida!\nOpcao -> ");
        } while(!((op > 0 && op < 7) || op == 10));

        return op;
    }
}

```

```

public static void readSeq(int[] num_array)
{
    int n = 0, tmp;
    do{ System.out.println("Insira valores:");
        System.out.printf("Valor[%d]: ", n);
        tmp = read.nextInt();
        if(tmp > 0) num_array[n++] = tmp;
        else if(tmp < 0) System.out.println("Valor invalido!");
    } while(tmp != 0);
}

public static void printSeq(int[] num_array)
{
    int n = 0;
    do{ if(num_array[n] > 0)
        System.out.print(" " + num_array[n++] + " ");
    } while(num_array[n] != 0);
}

public static int printMax(int[] num_array)
{
    int n=0, tmp=0;
    do{ if(num_array[n] > tmp) tmp = num_array[n];
        n++;
    } while(num_array[n] != 0);
    return tmp;
}

public static int printMin(int[] num_array)
{
    int n=0, tmp=Integer.MAX_VALUE;
    do{ if(num_array[n] < tmp) tmp = num_array[n];
        n++;
    } while(num_array[n] != 0);
    return tmp;
}

public static double printMed(int[] num_array) {
    int n=0; double tmp = 0;
    do{
        if(num_array[n] > 0) tmp += (double)num_array[n++];
    } while(num_array[n] != 0);
    tmp /= n;
    return tmp;
}

public static void printIfEven(int[] num_array)
{
    int n=0; boolean onlyEven = true;
    do{
        if(num_array[n++] % 2 != 0) onlyEven = false;
    } while(num_array[n] != 0);
    if(onlyEven) System.out.println("Sao todos pares");
    else System.out.println("Existem impares");
}

```

Passo 3: Implementar o código completo

```

static dados gravarDados(int[] ar) {
    dados d = new dados();
    d.maximo = printMax(ar);
    d.minimo = printMin(ar);
    d.media = printMed(ar);
    return d;
}

static void imprimirDados(dados dd) {
    System.out.printf("Maximo: %d\n", dd.maximo);
    System.out.printf("Minimo: %d\n", dd.minimo);
    System.out.printf("Media: %f\n", dd.media);
}

class dados {
    int maximo, minimo;
    double media;
}

```

Passo 4: Verificação

Análise de uma sequência de números inteiros

- 1 - Ler a sequência
- 2 - Escrever a sequência
- 3 - Calcular o máximo da sequência
- 4 - Calcular o mínimo da sequência
- 5 - Calcular a média da sequência
- 6 - Detetar se é uma sequência só constituída por números pares
- 10 - Terminar o programa

Opção -> 1

Insira valores:

Valor[0]: 234

Insira valores:

Valor[1]: 321

Insira valores:

Valor[2]: 265

Insira valores:

Valor[3]: 45

Insira valores:

Valor[4]: 0

Análise de uma sequência de números inteiros

- 1 - Ler a sequência
- 2 - Escrever a sequência
- 3 - Calcular o máximo da sequência
- 4 - Calcular o mínimo da sequência
- 5 - Calcular a média da sequência
- 6 - Detetar se é uma sequência só constituída por números pares
- 10 - Terminar o programa

Opção -> 5

Media: 216.250000

Análise de uma sequência de números inteiros

- 1 - Ler a sequência
- 2 - Escrever a sequência
- 3 - Calcular o máximo da sequência
- 4 - Calcular o mínimo da sequência
- 5 - Calcular a média da sequência
- 6 - Detetar se é uma sequência só constituída por números pares
- 10 - Terminar o programa

Opção -> 10

Máximo: 321

Mínimo: 45

Media: 216.250000

O programa vai terminar...

Press any key to continue . . . _

Outras tarefas possíveis:

1. Criar a classe e preencher os campos da classe.
2. Imprimir dados dum objeto da classe.
3. Imprimir histograma de valores.
4. Executar alguma operação sobre *array* unidimensional.
5. Calcular o desvio padrão.
6. Encontrar o divisor máximo comum entre valores.
7. Imprimir dados formatados.

Não esqueça:

1. Reservar memória para objetos (variáveis) numa classe.
2. Reservar memória para *arrays*.
3. Cuidado com expressões lógicas. Às vezes é melhor separar expressões em partes.
4. Cuidado com formatação, por exemplo, %d para inteiros, %f para reais.
5. Cuidado com ponto e vírgula (;). Por exemplo: **for(int i = 0; i<10; i++)**; - provavelmente erro.

O código da aula anterior não é muito bom, porque:

1. Memória reservada é fixa e contém 50 elementos (`int num_array[] = new int[50];`). Por exemplo, não podemos utilizar a instrução `num_array.length`;
2. Para *arrays* sem elementos não podemos executar opções exceto 1.
3. Memória foi reservada na função *main* que não preencha esta memória. É melhor reservar memória na função *readSeq* que é responsável pela leitura de dados e preenchimento da memória.

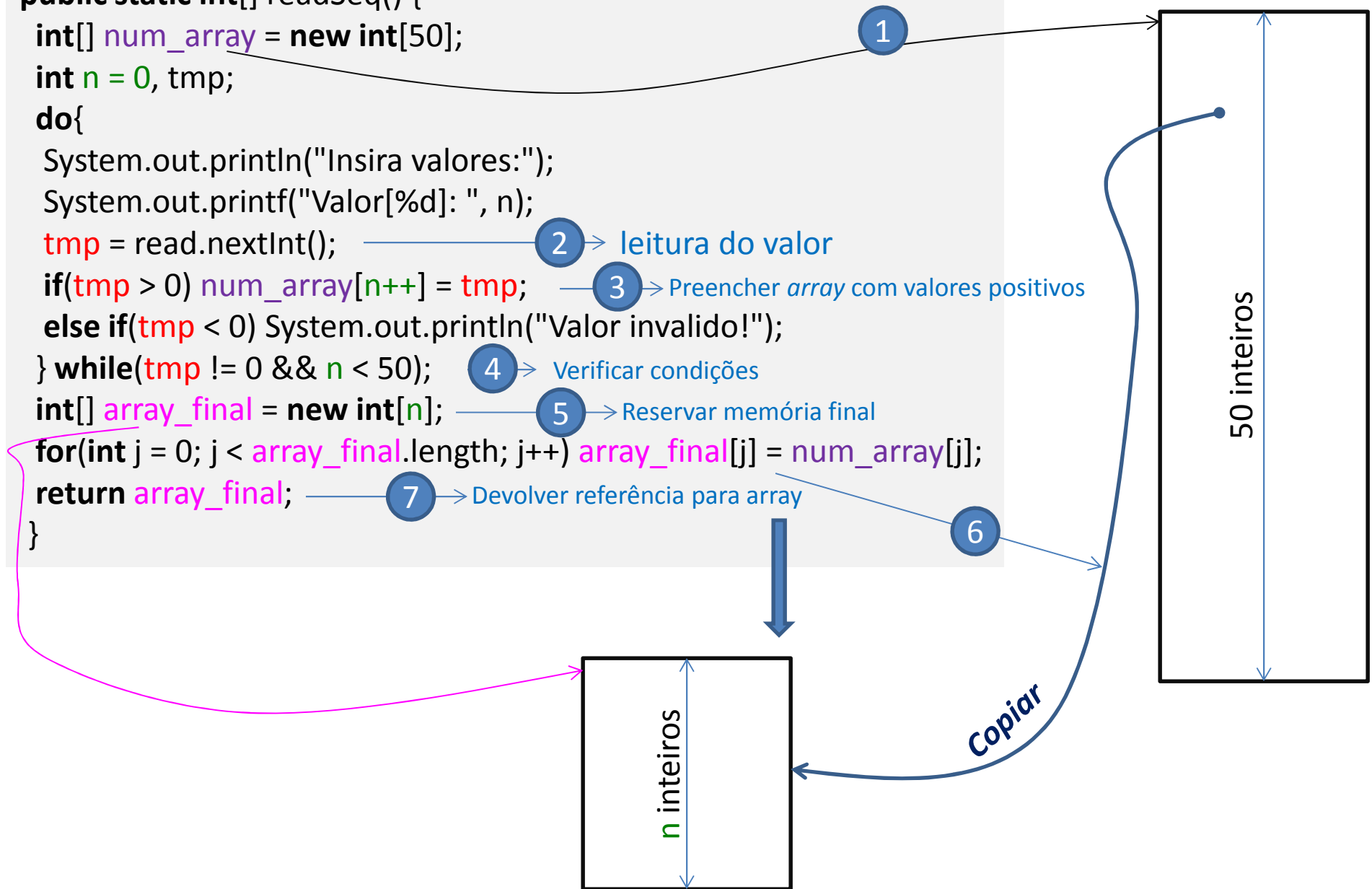
Proposta para alteração do programa anterior:

1. Reservar memória (por exemplo, `num_array`) na função *readSeq* com tamanho inicial 50 elementos (que é o tamanho máximo): `int[] num_array = new int[50];`.
2. Ler `n` dados (até 50 elementos, `n = 50`, ou valor de dado 0) na função *readSeq*.
3. Reservar memória novamente para `n` elementos, onde `n` é o número de dados real (ver ponto 2): `int[] array_final = new int[n];`.
4. Copiar `n` elementos da memória `num_array` para a memória `array_final` : `for(int j = 0; j < array_final.length; j++) array_final[j] = num_array[j];`.
5. Devolver a referência para o *array* novo: `return array_final;`.
6. Na função alterar o valor da referência `num_array`: `num_array = readSeq();`.
7. Utilizar o tamanho do *array* (`num_array.length`) no corpo das funções e não chamar funções para opções 2-6, antes de chamada da opção 1 (*Ler a sequencia*).

```

public static int[] readSeq() {
    int[] num_array = new int[50];
    int n = 0, tmp;
    do{
        System.out.println("Insira valores:");
        System.out.printf("Valor[%d]: ", n);
        tmp = read.nextInt();
        if(tmp > 0) num_array[n++] = tmp;
        else if(tmp < 0) System.out.println("Valor invalido!");
    } while(tmp != 0 && n < 50);
    int[] array_final = new int[n];
    for(int j = 0; j < array_final.length; j++) array_final[j] = num_array[j];
    return array_final;
}

```



Alteração das funções:

```
public static int printMenu() {  
    System.out.println("\nAnalise de uma sequencia de numeros inteiros");  
    System.out.println(" 1 - Ler a sequencia");  
    System.out.println(" 2 - Escrever a sequencia");  
    System.out.println(" 3 - Calcular o maximo da sequencia");  
    System.out.println(" 4 - Calcular o minimo da sequencia");  
    System.out.println(" 5 - Calcular a media da sequencia");  
    System.out.println(" 6 - Detetar se e uma sequencia so constituida por numeros pares");  
    System.out.println(" 7 - Imprimir sumario");  
    System.out.println("10 - Terminar o programa");  
    System.out.print("Opcao -> ");  
    int op = 0;  
    do {  
        op = read.nextInt();  
        if(!((op > 0 && op < 8) || op == 10)) System.out.print("Opcao invalida!\nOpcao -> ");  
    } while(!((op > 0 && op < 8) || op == 10));  
    return op;  
}  
  
// opção 7 adicional permite imprimir o sumário (chamada da função imprimirDados(gravarDados(num_array));)
```

Alteração das funções:

```
public static void printSeq(int[] num_array)      {  
    for(int j = 0; j < num_array.length; j++) System.out.print(" " + num_array[j] + " ");  
}  
  
public static int printMax(int[] num_array)      {  
    int tmp=0;  
    for(int j = 0; j < num_array.length; j++)      if(num_array[j] > tmp) tmp = num_array[j];  
    return tmp;  
}  
  
public static int printMin(int[] num_array)      {  
    int tmp=num_array[0];  
    for(int j = 0; j < num_array.length; j++)      if(num_array[j] < tmp) tmp = num_array[j];  
    return tmp;  
}  
  
public static double printMed(int[] num_array) {  
    int j;  
    double tmp = 0;  
    for(j = 0; j < num_array.length; j++)      if(num_array[j] > 0) tmp += (double)num_array[j];  
    return tmp/j;  
}
```


Alteração das funções:

```
public static void printIfEven(int[] num_array) {
    int j=0;
    boolean onlyEven = true;
    for(j = 0; j < num_array.length; j++)
        if(num_array[j]%2 != 0) onlyEven = false;
    if(onlyEven)      System.out.println("Sao todos pares");
    else              System.out.println("Existem impares");
}

static dados gravarDados(int[] ar) {
    dados d = new dados();
    d.maximo = printMax(ar);
    d.minimo = printMin(ar);
    d.media = printMed(ar);
    return d;
}

static void imprimirDados(dados dd) {
    System.out.printf("Maximo: %d\n", dd.maximo);
    System.out.printf("Minimo: %d\n", dd.minimo);
    System.out.printf("Media: %f\n", dd.media);
}
```

Alteração da função *main*:

```
import java.util.*;
public class e6_3 {
    static Scanner read = new Scanner(System.in);
    public static void main (String args[]) {
        int op = 0, num_array[] = null;
do        {
    op = printMenu();
    if (num_array == null && op != 1) { System.out.println("Array esta vasia"); continue; }
    switch(op) {
        case 1:
            num_array = readSeq(); System.out.printf("Tamanho de array = %d", num_array.length);
        case 2:
            printSeq(num_array);
        case 3:
            System.out.printf("Maximo: %d", printMax(num_array));
        case 4:
            System.out.printf("Minimo: %d", printMin(num_array));
        case 5:
            System.out.printf("Media: %f", printMed(num_array) );
        case 6:
            printIfEven(num_array);
        case 7:
            imprimirDados(gravarDados(num_array));
    }
} while(op != 10);
    System.out.println("O programa vai terminar...");
// as funções
}

class dados {
    int      maximo, minimo;
    double   media;
}
```

Para *array* nulo
podemos avançar
só para opção 1



break;
break;
break;
break;
break;
break;
break;

Os resultados:

```
Analise de uma sequencia de numeros inteiros
1 - Ler a sequencia
2 - Escrever a sequencia
3 - Calcular o maximo da sequencia
4 - Calcular o minimo da sequencia
5 - Calcular a media da sequencia
6 - Detetar se e uma sequencia so constituida por numeros pares
7 - Imprimir sumario
10 - Terminar o programa
Opcao -> 8
Opcao invalida!
Opcao -> 2
Array esta vasia

Analise de uma sequencia de numeros inteiros
1 - Ler a sequencia
2 - Escrever a sequencia
3 - Calcular o maximo da sequencia
4 - Calcular o minimo da sequencia
5 - Calcular a media da sequencia
6 - Detetar se e uma sequencia so constituida por numeros pares
7 - Imprimir sumario
10 - Terminar o programa
Opcao -> 1
Insira valores:
Valor[0]: 345
Insira valores:
Valor[1]: 234
Insira valores:
Valor[2]: 672
Insira valores:
Valor[3]: 34
Insira valores:
Valor[4]: 0
tamanho de array = 4
Analise de uma sequencia de numeros inteiros
1 - Ler a sequencia
2 - Escrever a sequencia
3 - Calcular o maximo da sequencia
4 - Calcular o minimo da sequencia
5 - Calcular a media da sequencia
6 - Detetar se e uma sequencia so constituida por numeros pares
7 - Imprimir sumario
10 - Terminar o programa
Opcao -> 2
345 234 672 34
Analise de uma sequencia de numeros inteiros
1 - Ler a sequencia
2 - Escrever a sequencia
3 - Calcular o maximo da sequencia
4 - Calcular o minimo da sequencia
5 - Calcular a media da sequencia
6 - Detetar se e uma sequencia so constituida por numeros pares
7 - Imprimir sumario
10 - Terminar o programa
Opcao -> 3
Maximo: 672
Analise de uma sequencia de numeros inteiros
1 - Ler a sequencia
2 - Escrever a sequencia
3 - Calcular o maximo da sequencia
4 - Calcular o minimo da sequencia
5 - Calcular a media da sequencia
6 - Detetar se e uma sequencia so constituida por numeros pares
7 - Imprimir sumario
10 - Terminar o programa
Opcao -> 4
```

```
Opcao -> 4
Minimo: 34
Analise de uma sequencia de numeros inteiros
1 - Ler a sequencia
2 - Escrever a sequencia
3 - Calcular o maximo da sequencia
4 - Calcular o minimo da sequencia
5 - Calcular a media da sequencia
6 - Detetar se e uma sequencia so constituida por numeros pares
7 - Imprimir sumario
10 - Terminar o programa
Opcao -> 5
Media: 321.250000
Analise de uma sequencia de numeros inteiros
1 - Ler a sequencia
2 - Escrever a sequencia
3 - Calcular o maximo da sequencia
4 - Calcular o minimo da sequencia
5 - Calcular a media da sequencia
6 - Detetar se e uma sequencia so constituida por numeros pares
7 - Imprimir sumario
10 - Terminar o programa
Opcao -> 6
Existem impares

Analise de uma sequencia de numeros inteiros
1 - Ler a sequencia
2 - Escrever a sequencia
3 - Calcular o maximo da sequencia
4 - Calcular o minimo da sequencia
5 - Calcular a media da sequencia
6 - Detetar se e uma sequencia so constituida por numeros pares
7 - Imprimir sumario
10 - Terminar o programa
Opcao -> 7
Maximo: 672
Minimo: 34
Media: 321.250000

Analise de uma sequencia de numeros inteiros
1 - Ler a sequencia
2 - Escrever a sequencia
3 - Calcular o maximo da sequencia
4 - Calcular o minimo da sequencia
5 - Calcular a media da sequencia
6 - Detetar se e uma sequencia so constituida por numeros pares
7 - Imprimir sumario
10 - Terminar o programa
Opcao -> 10
O programa vai terminar...
Press any key to continue . . .
```

6.2 Exercícios complementares

Exercício 6.5

Escreva um programa que leia uma sequência de N números reais, sendo o valor N pedido ao utilizador antes do início da introdução dos números. O programa deverá calcular a média e o desvio padrão da sequência e imprimir no ecrã os valores superiores à média.

2, 4, 4, 4, 5, 5, 7, 9.

These eight data points have the mean (average) of 5:

$$\frac{2 + 4 + 4 + 4 + 5 + 5 + 7 + 9}{8} = 5.$$

1 Encontrar a média

Exemplo da
wikipedia

First, calculate the deviations of each data point from the mean, and square the result of each:

$$\begin{array}{ll} (2 - 5)^2 = (-3)^2 = 9 & (5 - 5)^2 = 0^2 = 0 \\ (4 - 5)^2 = (-1)^2 = 1 & (5 - 5)^2 = 0^2 = 0 \\ (4 - 5)^2 = (-1)^2 = 1 & (7 - 5)^2 = 2^2 = 4 \\ (4 - 5)^2 = (-1)^2 = 1 & (9 - 5)^2 = 4^2 = 16. \end{array}$$

2 Encontrar *array* de desvios

The variance is the mean of these values:

$$\frac{9 + 1 + 1 + 1 + 0 + 0 + 4 + 16}{8} = 4$$

3 Encontrar a média

and the *population* standard deviation is equal to the square root of the variance:

$$\sqrt{4} = 2.$$






4 Calcular raiz quadrada

A Adicionar função **public static double** printMed(**double**[] num_array)

```
public static double printMed(double[] num_array)    {  
    int j;  
    double tmp = 0;  
    for(j = 0; j < num_array.length; j++)  
        if(num_array[j] > 0) tmp += num_array[j];  
    return tmp/j;  
}
```

pode utilizar para
verificar valores
intermédios

B Calcular Desvio padrão

```
static double standard_deviation(int[] ar)    {  
    double media = printMed(ar);  1 Encontrar a média  
    //System.out.printf("M = %f\n", media);   
  
    double[] ad = new double[ar.length];  
    for (int i = 0; i < ar.length; i++) ad[i] = Math.pow(ar[i] - media,2);  
    double media_ad = printMed(ad);  2 Encontrar array de desvios  
    return Math.sqrt(media_ad);  3 Encontrar a média  
     4 Calcular raiz quadrada  
}
```

Reutilização de funções !!!

```

import java.util.*;
public class e6_3 {
    static Scanner read = new Scanner(System.in);
    public static void main (String args[]) {
        int op = 0, num_array[] = null;
        do{
            op = printMenu();
            if (num_array == null && op != 1) { System.out.println("Array esta vasia"); continue; }
            switch(op) {
                case 1:    num_array = readSeq(); System.out.printf("Tamanho de array = %d", num_array.length); break;
                case 2:    printSeq(num_array); break;
                case 3:    System.out.printf("Maximo: %d", printMax(num_array)); break;
                case 4:    System.out.printf("Minimo: %d", printMin(num_array)); break;
                case 5:    System.out.printf("Media: %f", printMed(num_array) ); break;
                case 6:    printIfEven(num_array); break;
                case 7:    imprimirDados(gravarDados(num_array)); break;
                case 8:    System.out.printf("Desvio padrao: %f", standard_deviation(num_array)); break;
            }
        } while(op != 10);
        System.out.println("O programa vai terminar...");
    }
    public static int printMenu() {
        System.out.println("\nAnalise de uma sequencia de numeros inteiros"); System.out.println(" 1 - Ler a sequencia");
        System.out.println(" 2 - Escrever a sequencia"); System.out.println(" 3 - Calcular o maximo da sequencia");
        System.out.println(" 4 - Calcular o minimo da sequencia"); System.out.println(" 5 - Calcular a media da sequencia");
        System.out.println(" 6 - Detetar se e uma sequencia so constituida por numeros pares");
        System.out.println(" 7 - Imprimir sumario");
        System.out.println(" 8 - Desvio padrao");
        System.out.println(" 10 - Terminar o programa");
        System.out.print("Opcao -> ");
        int op = 0;
        do{
            op = read.nextInt();
            if(!((op > 0 && op < 9) || op == 10)) System.out.print("Opcao invalida!\nOpcao -> ");
        } while(!((op > 0 && op < 9) || op == 10));
        return op;
    }
}

```

// outras funções

```
public static double printMed(double[] num_array)    {
    int j;
    double tmp = 0;
    for(j = 0; j < num_array.length; j++)
        if(num_array[j] > 0) tmp += num_array[j];
    return tmp/j;
}

static double standard_deviation(int[] ar)           {
    double media = printMed(ar);
    //System.out.printf("M = %f\n", media);
    double[] ad = new double[ar.length];
    for (int i = 0; i < ar.length; i++) ad[i] = Math.pow(ar[i] - media,2);
    double media_ad = printMed(ad);
    return Math.sqrt(media_ad);
}

}

class dados    {
    int         maximo, minimo;
    double      media;
}
```

Os resultados:

2, 4, 4, 4, 5, 5, 7, 9.

These eight data points have the mean (average) of 5:

$$\frac{2 + 4 + 4 + 4 + 5 + 5 + 7 + 9}{8} = 5.$$


First, calculate the deviations of each data point from the mean, and square

$$\begin{aligned} (2 - 5)^2 &= (-3)^2 = 9 & (5 - 5)^2 &= 0^2 = 0 \\ (4 - 5)^2 &= (-1)^2 = 1 & (5 - 5)^2 &= 0^2 = 0 \\ (4 - 5)^2 &= (-1)^2 = 1 & (7 - 5)^2 &= 2^2 = 4 \\ (4 - 5)^2 &= (-1)^2 = 1 & (9 - 5)^2 &= 4^2 = 16. \end{aligned}$$

The variance is the mean of these values:

$$\frac{9 + 1 + 1 + 1 + 0 + 0 + 4 + 16}{8} = 4.$$

and the *population* standard deviation is equal to the square root of the variance

$$\sqrt{4} = 2.$$


```
Analise de uma sequencia de numeros inteiros
1 - Ler a sequencia
2 - Escrever a sequencia
3 - Calcular o maximo da sequencia
4 - Calcular o minimo da sequencia
5 - Calcular a media da sequencia
6 - Detetar se e uma sequencia so constituida por numeros pares
7 - Imprimir sumario
8 - Desvio padrao
10 - Terminar o programa
Opcao -> 1
Insira valores:
Valor[0]: 2
Insira valores:
Valor[1]: 4
Insira valores:
Valor[2]: 4
Insira valores:
Valor[3]: 4
Insira valores:
Valor[4]: 5
Insira valores:
Valor[5]: 5
Insira valores:
Valor[6]: 7
Insira valores:
Valor[7]: 9
Insira valores:
Valor[8]: 0
Tamanho de array = 8
Analise de uma sequencia de numeros inteiros
1 - Ler a sequencia
2 - Escrever a sequencia
3 - Calcular o maximo da sequencia
4 - Calcular o minimo da sequencia
5 - Calcular a media da sequencia
6 - Detetar se e uma sequencia so constituida por numeros pares
7 - Imprimir sumario
8 - Desvio padrao
10 - Terminar o programa
Opcao -> 8
Desvio padrao: 2.000000
Analise de uma sequencia de numeros inteiros
1 - Ler a sequencia
2 - Escrever a sequencia
3 - Calcular o maximo da sequencia
4 - Calcular o minimo da sequencia
5 - Calcular a media da sequencia
6 - Detetar se e uma sequencia so constituida por numeros pares
7 - Imprimir sumario
8 - Desvio padrao
10 - Terminar o programa
Opcao -> 10
O programa vai terminar...
Press any key to continue . . .
```


6.2 Exercícios complementares

Exercício 6.5

Escreva um programa que leia uma sequência de N números reais, sendo o valor N pedido ao utilizador antes do início da introdução dos números. O programa deverá calcular a média e o desvio padrão da sequência e imprimir no ecrã os valores superiores à média.

Exemplo da
wikipedia:

O desvio padrão também pode ser calculado quando não se sabe a média dos

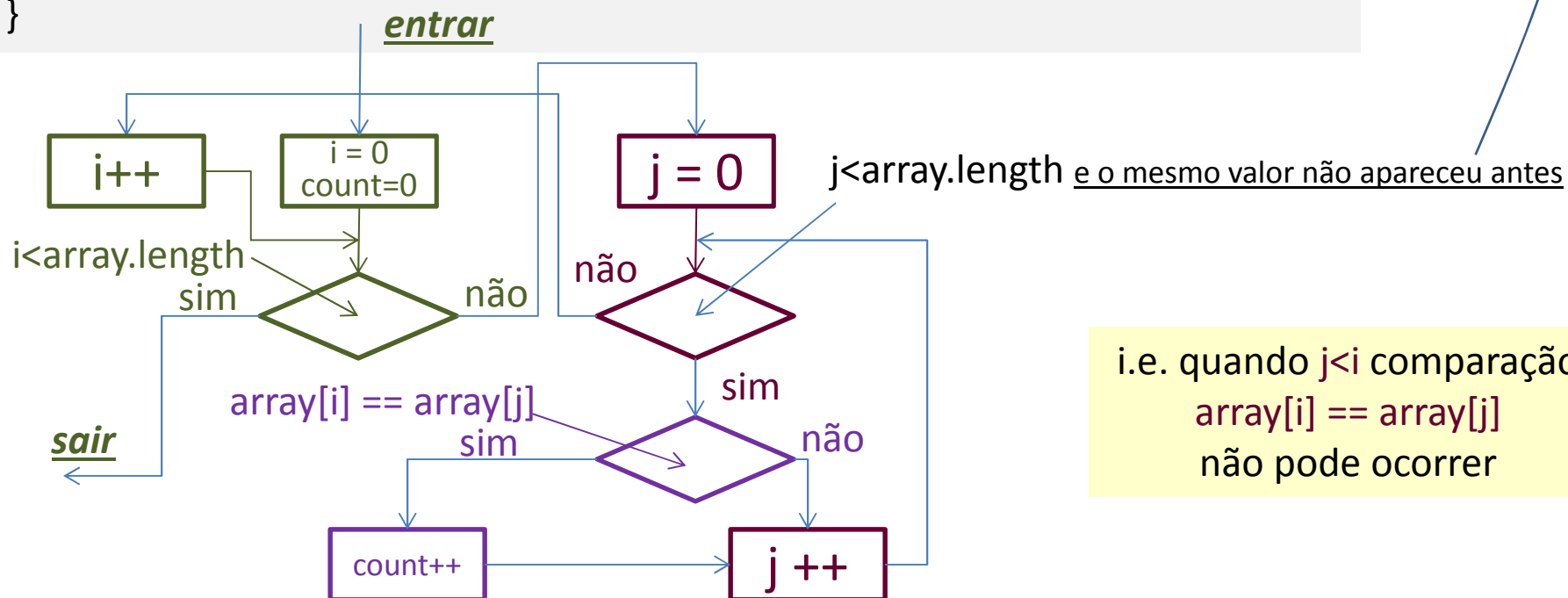
dados. O cálculo é feito conforme a fórmula: $s = \sqrt{\sum x_i^2 - \frac{1}{n}(\sum x_i)^2}$

```
static double desvio_padrao(int[] ar)
{
    double soma_xi2=0, soma_i = 0;
    for (int i = 0; i < ar.length; i++)
    {
        soma_xi2 += Math.pow(ar[i],2);
        soma_i += ar[i];
    }
    return Math.sqrt((soma_xi2-Math.pow(soma_i,2)/ar.length)/ar.length);
}
```

$$\text{soma_xi2-Math.pow(soma_i,2)/ar.length} \longrightarrow \sum x_i^2 - \frac{1}{n} (\sum x_i)^2$$

Escreva um programa que conte o número de vezes (maior que 1) que ocorre cada elemento de sequência.

```
static void ocorre_maior_que_1 (int[] array) {  
    for(int i=0,count=0; i<array.length; i++) {  
        count = 0;  
        for(int j=0; j<array.length; j++) {  
            if (( array[i] == array[j]) && (j<i) ) == true) break;  
            if (array[i] == array[j]) count++;  
        }  
        if (count > 1) System.out.println(array[i]+" ocorre "+count+" vezes");  
    }  
}
```



i.e. quando $j < i$ comparação
 $\text{array}[i] == \text{array}[j]$
não pode ocorrer

```

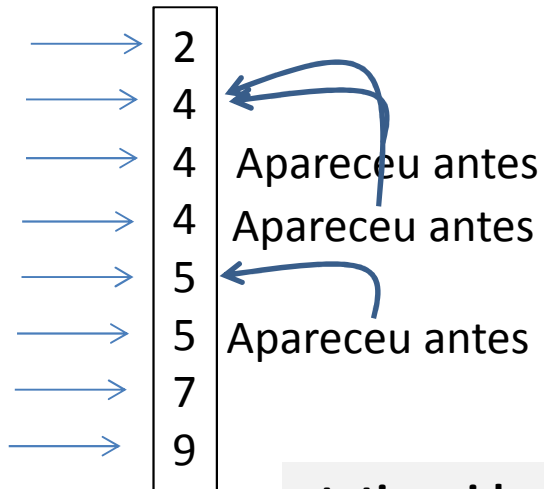
static void ocorre_maior_que_1 (int[] array) {
for(int i=0,count=0; i<array.length; i++)
    {
        count = 0;
        for(int j=0; j<array.length; j++)
            {
                if (( array[i] == array[j]) && (j<i) ) == true) break;
                if (array[i] == array[j]) count++;
            }
        if (count > 1) System.out.println(array[i]+" ocorre "+count+" vezes");
    }
}

```

```

2
4
4
4
5
5
7
9
-1
4 ocorre 3 vezes
5 ocorre 2 vezes
Press any key to continue . . . _

```



if ((array[i] == array[j]) && (j<i)) == true) break;

```

static void ocorre_maior_que_1 (int[] array) {
for(int i=0, count=0; i<array.length; i++)
    {
        for(int j=0; (j<array.length) && (( array[i] == array[j]) && (j<i) ) == false); j++)
            if (array[i] == array[j]) count++;
        if (count > 1) System.out.println(array[i]+" ocorre "+count+" vezes");
    }
}

```

Comparar

```
import java.util.*;
public class e6_7m {
    public static void main (String args[]) {
        int[] array = {2,4,4,4,5,5,7,9 }; ocorre_maior_que_1 (array);
    }
    static void ocorre_maior_que_1 (int[] array) {
        for(int i=0,count=0; i<array.length; i++) {
            count = 0;
            for(int j=0; j<array.length; j++) {
                if (( array[i] == array[j]) && (j<i) ) == true) break;
                if (array[i] == array[j]) count++;
            }
            if (count > 1) System.out.println(array[i]+" ocorre "+count+" vezes");
        }
    }
}
```

```
4 ocorre 3 vezes
5 ocorre 2 vezes
Press any key to continue . . .
```

```
import java.util.*;
public class e6_7m {
    public static void main (String args[]) {
        int[] array = {2,4,4,4,5,5,7,9 }; ocorre_maior_que_1 (array);
    }
    static void ocorre_maior_que_1 (int[] array) {
        for(int i=0,count=0; i<array.length; i++) {
            count = 0;
            for(int j=0; j<array.length; j++) {
                //if (( array[i] == array[j]) && (j<i) ) == true) break;
                if (array[i] == array[j]) count++;
            }
            if (count > 1) System.out.println(array[i]+" ocorre "+count+" vezes");
        }
    }
}
```

```
4 ocorre 3 vezes
4 ocorre 3 vezes
4 ocorre 3 vezes
5 ocorre 2 vezes
5 ocorre 2 vezes
Press any key to continue . . . _
```

Escreva um programa que conte o número de vezes (maior que 1) que ocorre cada elemento de sequência.

```
static void ocorre_maior_que_1 (int[] array) {  
    for(int i=0, count=0; i<array.length; i++)  
        for(int j=0; (j<array.length) && (( array[i] == array[j]) && (j<i) ) == false); j++)  
            if (array[i] == array[j]) count++;  
    if (count > 1) System.out.println(array[i]+" ocorre "+count+" vezes");  
}
```

2, 6 e 9 só uma vez

```
Analise de uma sequencia de numeros inteiros  
1 - Ler a sequencia  
2 - Escrever a sequencia  
3 - Calcular o maximo da sequencia  
4 - Calcular o minimo da sequencia  
5 - Calcular a media da sequencia  
6 - Detetar se e uma sequencia so constituida por numeros pares  
7 - Imprimir sumario  
8 - Desvio padrao  
9 - Ocorre (<> 1)  
10 - Terminar o programa  
Opcao -> 1  
Insira valores:  
Valor[0]: 2  
Insira valores:  
Valor[1]: 4  
Insira valores:  
Valor[2]: 4  
Insira valores:  
Valor[3]: 4  
Insira valores:  
Valor[4]: 5  
Insira valores:  
Valor[5]: 5  
Insira valores:  
Valor[6]: 6  
Insira valores:  
Valor[7]: 3  
Insira valores:  
Valor[8]: 3  
Insira valores:  
Valor[9]: 3  
Insira valores:  
Valor[10]: 3  
Insira valores:  
Valor[11]: 9  
Insira valores:  
Valor[12]: 0  
Tamanho de array = 12  
Analise de uma sequencia de numeros inteiros  
1 - Ler a sequencia  
2 - Escrever a sequencia  
3 - Calcular o maximo da sequencia  
4 - Calcular o minimo da sequencia  
5 - Calcular a media da sequencia  
6 - Detetar se e uma sequencia so constituida por numeros pares  
7 - Imprimir sumario  
8 - Desvio padrao  
9 - Ocorre (<> 1)  
10 - Terminar o programa  
Opcao -> 9  
4 ocorre 3 vezes  
5 ocorre 2 vezes  
3 ocorre 4 vezes
```

Escreva um programa
que conte o número de
vezes (maior que 1)
que ocorre cada
elemento de
sequência.

```
Análise de uma sequência de números inteiros
1 - Ler a sequência
2 - Escrever a sequência
3 - Calcular o máximo da sequência
4 - Calcular o mínimo da sequência
5 - Calcular a média da sequência
6 - Detetar se é uma sequência só constituída por números pares
7 - Imprimir sumário
8 - Desvio padrão
9 - Ocorre (> 1)
10 - Terminar o programa
Opcao -> 0
Opcao invalida!
Opcao -> 11
Opcao invalida!
Opcao -> 1
Insira valores:
Valor[0]: 2
Insira valores:
Valor[1]: 4
Insira valores:
Valor[2]: 4
Insira valores:
Valor[3]: 4
Insira valores:
Valor[4]: 5
Insira valores:
Valor[5]: 5
Insira valores:
Valor[6]: 7
Insira valores:
Valor[7]: 9
Insira valores:
Valor[8]: 0
Tamanho de array = 8
Análise de uma sequência de números inteiros
1 - Ler a sequência
2 - Escrever a sequência
3 - Calcular o máximo da sequência
4 - Calcular o mínimo da sequência
5 - Calcular a média da sequência
6 - Detetar se é uma sequência só constituída por números pares
7 - Imprimir sumário
8 - Desvio padrão
9 - Ocorre (> 1)
10 - Terminar o programa
Opcao -> 8
Desvio padrão: 2.000000
Análise de uma sequência de números inteiros
1 - Ler a sequência
2 - Escrever a sequência
3 - Calcular o máximo da sequência
4 - Calcular o mínimo da sequência
5 - Calcular a média da sequência
6 - Detetar se é uma sequência só constituída por números pares
7 - Imprimir sumário
8 - Desvio padrão
9 - Ocorre (> 1)
10 - Terminar o programa
Opcao -> 9
4 ocorre 3 vezes
5 ocorre 2 vezes
Análise de uma sequência de números inteiros
1 - Ler a sequência
2 - Escrever a sequência
3 - Calcular o máximo da sequência
4 - Calcular o mínimo da sequência
5 - Calcular a média da sequência
6 - Detetar se é uma sequência só constituída por números pares
7 - Imprimir sumário
8 - Desvio padrão
9 - Ocorre (> 1)
10 - Terminar o programa
```

Exercício 6.6

Escreva um programa que leia uma frase e imprima no monitor quais as letras do alfabeto que apareceram nessa frase. Para a resolução deste problema, sugere-se a utilização de uma sequência de valores booleanos de modo a sinalizar quais os caracteres do alfabeto que apareceram pelo menos uma vez.

```
import java.util.*;
public class e6_6 {
    static Scanner read = new Scanner(System.in);
    public static void main (String args[]) {
        char[] alfa = {'a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t','u','v','x','y','w','z'};
        boolean[] isOne = new boolean[26];
        String s = read.nextLine();
        for(int i = 0; i < s.length(); i++)
            for(int j = 0; j < alfa.length; j++) if(Character.toLowerCase(s.charAt(i)) == alfa[j]) isOne[j] = true;
        System.out.println("As letras inseridas foram:");
        for(int i = 0; i < isOne.length; i++) if(isOne[i]) System.out.print(" " + alfa[i]);
        System.out.println();
    }
}
```

```
universidade de aveiro
As letras inseridas foram:
 a d e i n o r s u v
Press any key to continue . . .
```

```

import
public class e6_6 {
    static Scanner read = new Scanner(System.in);
    public static void main (String args[]) {
        char[] alfa = {'a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t','u','v','x','y','w','z'};
        boolean[] isOne = new boolean[26];
        String s = read.nextLine();
        for(int i = 0; i < s.length(); i++)
            for(int j = 0; j < alfa.length; j++) if(Character.toLowerCase(s.charAt(i)) == alfa[j]) isOne[j] = true;
        System.out.println("As letras inseridas foram:");
        for(int i = 0; i < isOne.length; i++) if(isOne[i]) System.out.print(" " + alfa[i]);
        System.out.println();
    }
}

```

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	x	y	w	z
0	1	2	3	4	5	6	7	8	9	10														
t			t	t				t					t	t			t	t		t	t				

```

universidade de aveiro
As letras inseridas foram:
a d e i n o r s u v
Press any key to continue . . .

```


Exercício 6.4

Tendo como base o problema 6.2, considere agora que, dada uma determinada sequência de notas (valores inteiros de 0 a 20), pretende-se construir e desenhar o respetivo histograma (contagem do número de ocorrências de cada nota). O número de notas a processar deverá ser pedido ao utilizador no início do programa. Implemente esse programa, desenhando o histograma com o seguinte aspeto:

```
Histograma de notas
-----
20 | *****
19 | *
  .
  .
  .
01 | *
00 | **
```

Comece por associar um "*" a cada nota encontrada, fazendo depois a normalização (linear) do gráfico para que o valor máximo do histograma corresponda a 50 asteriscos.

Não imprimir linhas com 0 asteriscos

```

import java.util.*;
public class e6_4 {
    static Scanner read = new Scanner(System.in);
    public static void main (String args[]) {
        int tmp, hist_notas[] = new int[21];
        do {
            System.out.print("Quantas notas pretende introduzir: ");
            tmp = read.nextInt();
        } while (tmp <= 1 || tmp >= 51);
        int n = 0, notas_in[] = new int[tmp];
        do {
            System.out.printf("Nota[%d]: ", n);
            tmp = read.nextInt();
            if(tmp > 0 && tmp <= 20) notas_in[n++] = tmp;
            else
                System.out.println("Valor invalido!");
        } while(n < notas_in.length);
        n = 0;
        do hist_notas[notas_in[n++]]++; while(n < notas_in.length);
        System.out.println(); System.out.println(" Histograma de notas");
        System.out.println("-----");
        n = 20;
        do {
            if (hist_notas[n] != 0) System.out.printf("%2d | ", n);
            tmp = 0;
            do if(tmp++ < hist_notas[n]) System.out.print("*");
            while(tmp < hist_notas[n]);
            if (hist_notas[n--] != 0)
                System.out.println();
        } while(n > 0);
    }
}

```

Valeri Skliarov
2015/2016

```

Quantas notas pretende introduzir: 50
Nota[0]: 20
Nota[1]: 17
Nota[2]: 16
Nota[3]: 15
Nota[4]: 20
Nota[5]: 20
Nota[6]: 20
Nota[7]: 10
Nota[8]: 10
Nota[9]: 11
Nota[10]: 11
Nota[11]: 11
Nota[12]: 11
Nota[13]: 11
Nota[14]: 11
Nota[15]: 11
Nota[16]: 11
Nota[17]: 11
Nota[18]: 11
Nota[19]: 10
Nota[20]: 10
Nota[21]: 10
Nota[22]: 17
Nota[23]: 17
Nota[24]: 18
Nota[25]: 20
Nota[26]: 18
Nota[27]: 19
Nota[28]: 20
Nota[29]: 18
Nota[30]: 19
Nota[31]: 18
Nota[32]: 11
Nota[33]: 11
Nota[34]: 11
Nota[35]: 11
Nota[36]: 11
Nota[37]: 11
Nota[38]: 11
Nota[39]: 11
Nota[40]: 11
Nota[41]: 11
Nota[42]: 11
Nota[43]: 11
Nota[44]: 11
Nota[45]: 11
Nota[46]: 11
Nota[47]: 11
Nota[48]: 11
Nota[49]: 11

```

Histograma de notas

```

-----
20 | *****
19 | **
18 | ****
17 | ***
16 | *
15 | *
11 | *****
10 | *****
Press any key to continue . . . _

```

```
import java.util.*;
public class e6_4 {
    static Scanner read = new Scanner(System.in);
```

```
public static void main (String args[]) {
```

```
    int tmp, hist_notas[] = new int[21];
```

```
    do {        System.out.print("Quantas notas pretende introduzir: ");
                tmp = read.nextInt();
```

```
    } while (tmp <= 1 || tmp >= 51);
```

```
    int n = 0, notas_in[] = new int[tmp];
```

```
    do {        System.out.printf("Nota[%d]: ", n);
                tmp = read.nextInt();
                if(tmp > 0 && tmp <= 20) notas_in[n++] = tmp;
                else        System.out.println("Valor invalido!");
```

```
    } while(n < notas_in.length);
```

```
    n = 0;
```

```
    do hist_notas[notas_in[n++]]++; while(n < notas_in.length);
    System.out.println(); System.out.println(" Histograma de notas");
    System.out.println("-----");
```

```
    n = 20;
```

```
    do {        if (hist_notas[n] != 0) System.out.printf("%2d | ", n);
                tmp = 0;
                do if(tmp++ < hist_notas[n]) System.out.print("*");
                while(tmp < hist_notas[n]);
                if (hist_notas[n--] != 0)        System.out.println();
                } while(n > 0);
```

```
    }
```

```
}
```

Leitura e verificação do
número de notas

Entrada de notas

Preenchimento do *array*
para histograma

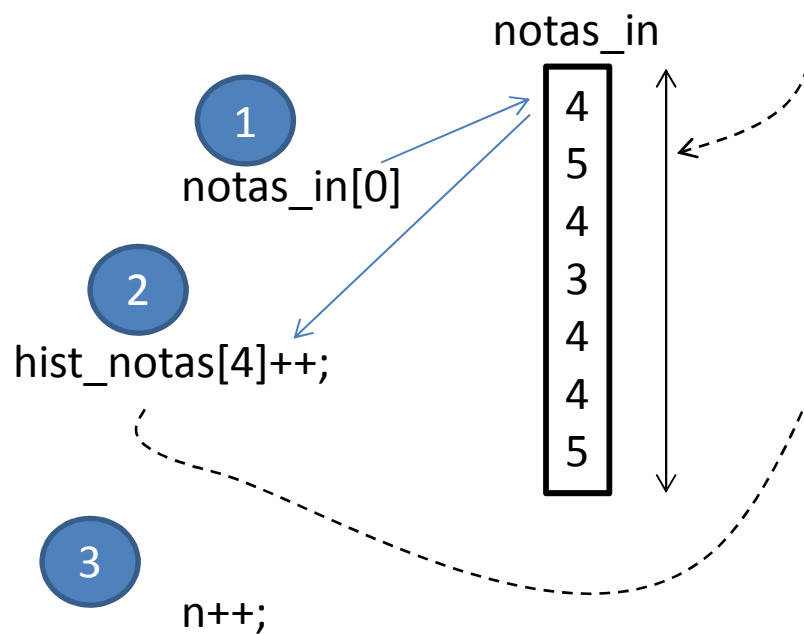
Impressão do
histograma

```
n = 0;  
do hist_notas[notas_in[n++]]++; while(n<notas_in.length);
```

Preenchimento do *array*
para histograma

||

```
int nota;  
n = 0;  
do  
{  
    nota = notas_in[n];  
    hist_notas[nota]++;  
    n++;  
}  
while(n<notas_in.length);
```



hist_notas

1	0
2	0
3	1
4	4
5	2
6	0
7	0
8	0
9	0
10	0
11	0
12	0
13	0
14	0
15	0
16	0
17	0
18	0
19	0
20	0

if(tmp++ <
hist_notas[n])

```

import java.util.*;
public class e6_4 {
    static Scanner read = new Scanner(System.in);
    public static void main (String args[]) {
        int tmp, hist_notas[] = new int[21];
        do {
            System.out.print("Quantas notas pretende introduzir: ");
            tmp = read.nextInt();
        } while (tmp <= 1 || tmp >= 51);
        int n = 0, notas_in[] = new int[tmp];
        do {
            System.out.printf("Nota[%d]: ", n);
            tmp = read.nextInt();
            if(tmp > 0 && tmp <= 20) notas_in[n++] = tmp;
            else System.out.println("Valor invalido!");
        } while(n < notas_in.length);
        n = 0;
        do hist_notas[notas_in[n++]]++; while(n < notas_in.length);
        System.out.println();
        System.out.println("-----");
        n = 20;
        do {
            if (hist_notas[n] != 0) System.out.printf("%2d | ", n);
            tmp = 0;
            do if(tmp++ < hist_notas[n]) System.out.print("*");
            while(tmp < hist_notas[n]);
            if (hist_notas[n] != 0) System.out.println();
        } while(n > 0);
    }
}

```

```

Quantas notas pretende introduzir: 10
Nota[0]: 11
Nota[1]: 20
Nota[2]: 13
Nota[3]: 12
Nota[4]: 12
Nota[5]: 123
Valor invalido!
Nota[5]: 20
Nota[6]: 11
Nota[7]: 20
Nota[8]: 13
Nota[9]: 13

```

Histograma de notas

```

11 | **
12 | **
13 | ***
20 | ***
Press any key to continue . . .

```

```

for(int i = 0; i < hist_notas.length; i++)
    if (hist_notas[i] != 0) {
        System.out.printf("%2d | ", i);
        for(int j = 0; j < notas_in.length; j++)
            if(j < hist_notas[i]) System.out.print("*");
        System.out.println();
    }
}

```

Exercício 8.5

O número dezassete representa-se em base dez como “17”, em base dois como “10001”, e em base três como “122”. Faça uma função que converta um número inteiro na sua representação numa qualquer base (para bases entre 2 e 10). Por exemplo, pretende-se que:

2 e 9

```
numToBase(17, 10) → "17"  
numToBase(17, 2) → "10001"  
numToBase(17, 3) → "122"
```

Escreva um programa que converte um número dado pelo utilizador nas suas representações em base 2, 3, ..., 10.

Lembre-se que pode converter um número para outra base por divisão sucessiva pela base. O resto de cada divisão dá mais um dígito que deve ser concatenado à esquerda dos anteriores. (Já existe uma função que faz isto na classe Integer, mas às vezes é bom reinventar a roda!)

Exemplo: Representar o número inteiro positivo s (expresso em decimal) num sistema posicional N ($N = \{2,3,4,5,6,7,8,9\}$) e verificar o resultado.

Regras de conversão do valor s :

1. Se $s < N$ a conversão já esta feita.
2. Se $s \geq N$ dividir s por N ($s \neq N$) e gravar o resto da divisão.
3. Se $s < N$, gravar s e a conversão já está pronta.

Exemplo para $s=35$, $N=8$:

1. $35 > 8$, i.e. $s > N$.
2. $s/8 = 4$, $s\%8 = 3$. Gravar 3.
3. $s=4$, $s < N$. Gravar 4

O resultado é 43.

Verificação: $4 \cdot 8^1 + 3 \cdot 8^0 = 32 + 3 = 35$

Exemplo para $s=35$, $N=2$:

1. $35 > 2$, i.e. $s > N$.
2. $s/2 = 17$, $s\%2 = 1$. Gravar 1.
3. $(s=17)/2 = 8$, $s\%2 = 1$. Gravar 1.
4. $(s=8)/2 = 4$, $s\%2 = 0$. Gravar 0.
5. $(s=4)/2 = 2$, $s\%2 = 0$. Gravar 0.
6. $(s=2)/2 = 1$, $s\%2 = 1$. Gravar 0.
7. O resultado é 100011.

Verificação:

$$1 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = \\ = 32 + 0 + 0 + 0 + 2 + 1 = 35$$

Ver dígitos com cores iguais

Decimais	Binários	Octais	Hexadecimais
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000	20	10
17	10001	21	11
18	10010	22	12
19	10011	23	13
20	10100	24	14

Alguns exemplos adicionais

Decimal: $2 \times 10^1 + 0 \times 10^0 = 20 + 0 = 20$

Binário: $1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0$
 $16 + 0 + 4 + 0 + 0 = 20$

Octal: $2 \times 8^1 + 4 \times 8^0 = 16 + 4 = 20$

Hexadecimal: $1 \times 16^1 + 4 \times 16^0 = 16 + 4 = 20$


```

import java.util.*;
public class Ex8_5 {
    static Scanner sc = new Scanner(System.in);
    public static void main(String[] args)
    {
        int i, s_gravado, s, N, tmp=0, indice=0;
        int[] v = new int[32];
        System.out.print("Valor inteiro decimal ? ");
        s = sc.nextInt();
        s_gravado = s;
        do {
            System.out.print("Sistema (2,3,4,5,6,7,8,9) ? ");
            N = sc.nextInt(); } while(N<2 || N>9);
        for(i=0;s>=N;i++) {
            v[i] = s%N;
            tmp += (s%N)*Math.pow(N,indice++);
            s /= N; }

        v[i] = s;
        tmp += (s%N)*Math.pow(N,indice++);
        if (tmp == s_gravado) System.out.println("conversao esta correta");
        else System.out.println("conversao nao esta correta " + tmp);
        System.out.print("O resultado: ");
        for(int j=i;j>=0;j--) System.out.print(v[j]);
        System.out.println();
    }
}

```

$$1 \times 4^2 + 3 \times 4^1 + 3 \times 4^0 = 16 + 12 + 3 = 31_{10}$$


```

Valor inteiro decimal ? 31
Sistema (2,3,4,5,6,7,8,9) ? 4
conversao esta correta
O resultado: 133
Press any key to continue . . . _

```

Um problema com funções da classe Scanner:

```
import java.util.*;
public class nextLine {
    static Scanner sc = new Scanner(System.in);
    public static void main(String[] args)
    {
        int n = sc.nextInt();
        System.out.printf("n = %d\n", n);
        String s = sc.nextLine();
        System.out.printf("s = %s\n", s);
    }
}
```



```
435
n = 435
s =
Press any key to continue . . . _
```

Resolução do problema:

```
import java.util.*;
public class nextLine      {
    static Scanner sc = new Scanner(System.in);
    public static void main(String[] args)
    {
        int n = sc.nextInt();
        System.out.printf("n = %d\n", n);
        String s;
        do s=sc.nextLine(); while (s.length()==0); ←
        System.out.printf("s = %s\n", s);
    }
}
```

```
44
n = 44
University of Aveiro
s = University of Aveiro
Press any key to continue . . . _
```

Conclusão

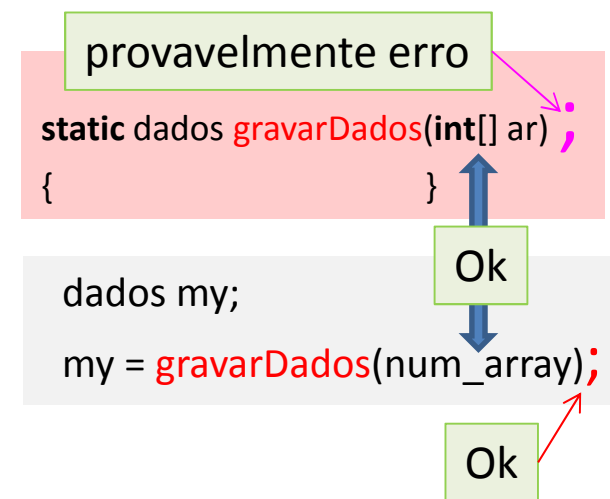
Não esqueça:

1. Reservar memória para objetos (variáveis) duma classe.
2. Reservar memória para *arrays*.
3. Cuidado com expressões lógicas. Às vezes é melhor separar expressões em partes.
4. Cuidado com formatação, por exemplo, %d para inteiros, %f para reais.
5. Cuidado com ponto e virgula (;). Por exemplo: **for(int i = 0; i<10; i++);** - provavelmente erro.
6. Exemplos úteis:

```
dados my;  
my = gravarDados(num_array);  
imprimirDados(my);
```

```
class dados    {  
    int         maximo, minimo;  
    double     media;  
}
```

```
static dados gravarDados(int[] ar)    {  
    dados d = new dados();  
    d.maximo = printMax(ar);  
    d.minimo = printMin(ar);  
    d.media  = printMed(ar);  
    return d;  
}  
  
static void imprimirDados(dados dd)  {  
    System.out.printf("Maximo: %d\n", dd.maximo);  
    System.out.printf("Minimo: %d\n", dd.minimo);  
    System.out.printf("Media: %f\n", dd.media);  
}
```



Conclusão

Erros comuns:

1. Cuidado com ponto e virgula (;). Por exemplo: **for(int i = 0; i<10; i++);** - provavelmente erro.
2. Exemplos úteis:

```
import java.util.*;
public class e_erro {
    static Scanner read = new Scanner(System.in);
    public static void main (String args[]) {
        int indice = 0;
        my array[] = new my[5];
        do {    array[indice] = new my();
                array[indice].a = read.nextInt();
                indice++;
            } while (array[indice].a != 10);
        }
    }
    class my {int a;}
}
```

Aparece erro durante a execução

```
20
Exception in thread "main" java.lang.NullPointerException
    at e_erro.main(e_erro.java:10)
Press any key to continue . . .
```

```
import java.util.*;
public class e_erro {
    static Scanner read = new Scanner(System.in);
    public static void main (String args[]) {
        int indice = 0;
        my array[] = new my[5];
        do {    array[indice] = new my();
                array[indice].a = read.nextInt();
                //indice++;
            } while (array[indice++].a != 10);
        }
    }
    class my {int a;}
}
```

Agora não aparece o mesmo erro durante a execução mas pode aparecer outro erro (tamanho do array)

Conclusão

```
1
2
3
4
5
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 5
    at e_erro.main(e_erro.java:7)
Press any key to continue . . . _
```

Resolução

```
import java.util.*;
public class e_sem_erros {
    static Scanner read = new Scanner(System.in);
    public static void main (String args[]) {
        my[] array = new my[5];
        for(int indice = 0; indice < array.length ;indice++)
        {
            array[indice] = new my();
            array[indice].a = read.nextInt();
            if (array[indice].a == 10) break;
        }
    }
}
class my {int a;}
```

```
import java.util.*;
public class e_erro {
    static Scanner read = new Scanner(System.in);
    public static void main (String args[]) {
        int indice = 0;
        my array[] = new my[5];
        do {
            array[indice] = new my();
            array[indice].a = read.nextInt();
        } while (array[indice++].a != 10);
    }
}
class my {int a;}
```

```
1
2
3
4
5
Press any key to continue . . .
```