

## AULAS 4 e 5

### Objectivos

Nestas duas aulas práticas pretende-se que os alunos adquiram competências ao nível da aplicação de ciclos repetitivos para a resolução de problemas. No final da aula 4 os alunos devem compreender e saber utilizar ciclos *while*, *do ... while* e *for*, e saber aplicar o operador *++*. Devem ainda reconhecer a importância da realização de traçagens para a análise e compreensão dos programas exemplo que lhes são propostos. Convém salientar que as ferramentas de depuração de programas (*debugging*) do Dr Java poderão ser um auxiliar valioso no processo de estudo individual.

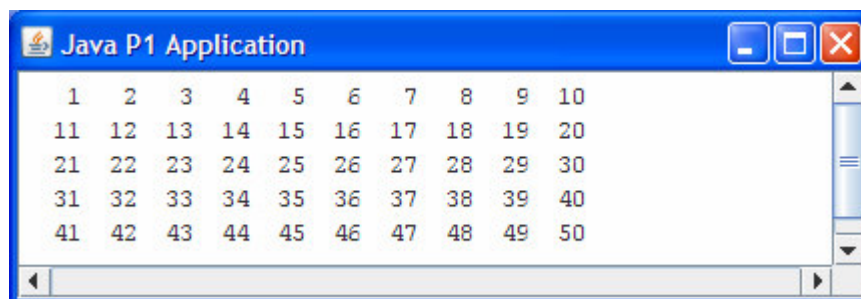
A aula nº 5 permitirá consolidar os conhecimentos e as práticas sobre a aplicação de ciclos repetitivos. Deverão ainda ser incluídas aplicações práticas de utilização de ciclos repetitivos incluídos dentro de outros ciclos repetitivos.

1. Copiar o programa abaixo que escreve no ecrã os números inteiros entre 1 e 50.

```
import pl.*;
class aula04_cicloInteiros extends PlApp {
    static void main(String [] args) {
        int i;

        for(i=1; i<=50; i++)
            printf(4, i);
    }
}
```

- a) Alterar o programa de forma a escrever os números pela ordem inversa (de 50 até 1).
- b) Construir dois novos programas idênticos ao programa inicial listado acima, substituindo o ciclo *for* por ciclos *while* e *do ... while*, respectivamente.
- c) Alterar o programa inicial de forma a escrever apenas 10 números por linha, tal como exemplificado abaixo:



- d) Alterar o programa da alínea anterior de forma a escrever apenas os números pares no intervalo de 1 a 50.
- e) Finalmente, completar o programa da alínea anterior de forma a permitir ao utilizador especificar o limite inferior e o limite superior do intervalo.

2. O programa abaixo deveria escrever os números ímpares entre um limite inferior e um limite superior especificados pelo utilizador.

```
import p1.*;
class aula04_seqImpares extends P1App {
    static void main(String [] args) {
        int i, limInf, limSup;

        limInf = readInt("Limite inferior? ", 0, 998);
        limSup = readInt("Limite superior? ", limInf+1, 999);

        if(limInf%2!=0)    i = limInf + 1;
        else                i = limInf;
        do {
            printf(4, i);
            i = i + 2;
        } while (i < limSup);

        println("Listagem completa!");
    }
}
```

Testar e analisar os resultados obtidos para os casos de teste abaixo e corrigir o programa se necessário.

teste	limite inferior	limite superior	resultado esperado
i.	17	13	Erro, pede novamente o limite superior
ii.	12	20	13 15 17 19 Listagem completa!
iii.	13	21	13 15 17 19 21 Listagem completa!
iv.	15	15	15 Listagem completa!

3. Analisar e descrever o que faz o programa abaixo.

```
import p1.*;
class aula04_serieReais extends P1App {
    static void main(String [] args) {
        double n;

        do {
            n = readDouble("Introduza um n° real... ");
        } while(n < -1 || n > 1);
        println("Fim...");
    }
}
```

Completar o programa de forma a serem lidos no máximo 10 números reais.

4. Completar a traçagem do programa, considerando que o utilizador introduz sucessivamente os valores 5, 8, 10 e 20.

```

import p1.*;
class aula04_tracagem extends P1App {
    static void main(String [] args) {
        int n, ant, conta;

/*P1*/      n = readInt("Nº inteiro: ");
/*P2*/      conta = 1;
        do {
/*P3*/          ant = n;
/*P4*/          n = readInt("Outro nº inteiro: ");
/*P5*/          conta++;
/*P6*/        } while(n != 2*ant);
/*P7*/      println("Nº de valores lidos: " + conta);
        }
    }

```

Traçagem do programa:

passo	n	ant	conta	n != 2*ant ?	Ecrã
P1	5				Nº inteiro
P2			1		
P3		5			
...	...	...	...	...	...

5. Completar os seguintes programas das aulas 2 e 3 acrescentado as funcionalidades indicadas.

a) Exercício 1 – distância em milhas:

- i. Validar a entrada de dados de forma a só serem permitidas distâncias positivas. Se o utilizador introduzir uma distância negativa o programa deve mostrar uma mensagem de erro em português e pedir novos valores até ser introduzido um valor válido.
- ii. Repetir o processamento (pedido com validação de uma distância em milhas e impressão no ecrã do valor correspondente em km) até que o utilizador introduza o valor zero.

b) Exercício 7 – conversão de temperaturas:

- i. Analisar o programa feito anteriormente, verificando se o tipo de temperatura é lido antes do valor da temperatura. Trocar a ordem das instruções de leitura se necessário.

- ii. Validar as entradas de dados, mostrando mensagens de erro adequadas: os tipos de temperatura válidos são 'C', 'c', 'F' ou 'f'; os valores das temperaturas devem ser superiores a  $-273.16^{\circ}\text{C}$  ou  $-459,67^{\circ}\text{F}$ , conforme a opção anterior do utilizador.
  - iii. Repetir o processamento até o utilizador introduzir o valor 'T' ou 't' para o tipo de temperatura
6. Escrever um programa leia uma sequência de valores correspondentes às despesas mensais realizadas por um departamento e que apresente os desvios relativamente ao valor orçamentado. Considerar que o valor orçamentado para as despesas tem o valor fixo de 20.000 € para todos os meses. A leitura deve terminar quando for introduzido o valor -999 ou quando forem introduzidos 12 valores.
  7. Pretende-se escrever um programa que lê do teclado uma série de valores até o utilizador introduzir o valor -1 e apresenta o conjunto de dados estatísticos indicados abaixo. Os valores introduzidos correspondem ao nº de peças produzidas por dia numa linha de montagem de uma fábrica, durante um certo período de tempo. Os dados estatísticos a apresentar são:
    - a) O nº total de peças produzidas;
    - b) O nº médio de peças produzidas por dia;
    - c) O nº de dias em que a produção foi inferior a 250 peças.
  8. Uma empresa recorreu a um empréstimo de 100.000 € para comprar uma nova máquina para a sua linha de produção. A empresa terá que pagar todos os meses uma prestação do valor em dívida, que por razões financeiras, deverá ser um valor fixo igual 5000 €. Por sua vez, o banco aplica uma taxa de juro de 1% sobre o valor em débito no final de cada mês.

Escrever um programa que calcula e apresenta uma simulação mensal com o nº da prestação, juros cobrados pelo banco e valor actualizado da dívida, até a dívida se encontrar totalmente paga. No final, o programa também deve apresentar o valor total dos juros cobrados pelo banco.

Prestação	Juros	Valor em dívida
		100.000 €
1	1.000 €	96.000 €
2	960 €	91.960 €
3	920 €	87.880 €
...	...	...

9. Pretende-se escrever um programa para fazer a conversão de valores em euros para dólares ('D'), libras ('L') ou francos suíços ('F'). O programa deve começar por ler a taxa de conversão para cada uma das moedas indicadas. Em seguida, o programa deve permitir ao utilizador realizar uma série de operações, indicando a moeda e a

quantia em euros a converter. O programa deve terminar quando o utilizador digitar o valor 'T' para terminar. O programa deve funcionar da seguinte forma:

```
Taxa de conversão de euros para dólares (D):      1.2
Taxa de conversão de euros para libras (L):      0.65
Taxa de conversão de euros para francos suíços (F): 1.8
==>
Qual a moeda (D)ólares (L)ibras (F)rancos S (T)erminar) ? D
Qual a quantia em euros ? 500
==> 600 dólares
Qual a moeda (D)ólares (L)ibras (F)rancos S (T)erminar) ? L
Qual a quantia em euros ? 1000
==> 650 libras
Qual a moeda (D)ólares (L)ibras (F)rancos S (T)erminar) ? T
Fim!
```

10. Escrever um programa que leia do teclado um  $n^{\circ} N$ , com  $1 \leq N \leq 10$ , e escreva no ecrã a respectiva tabuada da multiplicação, com o formato apresentado a seguir.

```
-----
| Tabuada dos ## |
-----
| ## x 1 = ### |
| ## x 2 = ### |
|      ...      |
| ## x 10 = ### |
-----
```

11. Escrever um programa que leia um número  $n$  e mostre no ecrã os  $n$  primeiros elementos da série de Fibonacci:

1, 1, 2, 3, 5, 8, 13, ... ( $x_1=1, x_2=1, x_i=x_{i-1} + x_{i-2}$  para  $i>2$ )

12. Analisar o programa abaixo sem o executar no Dr Java.

```
import p1.*;
class aula04_ciclosEncadeados extends P1App {
    static void main(String [] arg) {
        int i, j;

/*P1*/        for(i=1; i<=4; i++) {
/*P2*/            for(j=1; j<=3; j++)
/*P3*/                print("(" + i + ", " + j +") ");
/*P4*/                println();
            }
        }
    }
}
```

- a) Fazer a traçagem do programa e indicar qual o resultado apresentado no ecrã (sem executar o programa!).
- b) Qual seria o resultado se a expressão condicional  $j \leq 3$  fosse substituída por  $j \leq i$ .
13. Escrever um programa que mostre a tabuada da multiplicação entre 1 e 10 no formato apresentado abaixo.

	1	2	3	4	5	6	7	8	9	10
1	1	2	3	4	5	6	7	8	9	10
2	2	4	6	8	10	12	14	16	18	20
3	3	6	9	12	15	18	21	24	27	30
4	4	8	12	16	20	24	28	32	36	40
5	5	10	15	20	25	30	35	40	45	50
6	6	12	18	24	30	36	42	48	54	60
7	7	14	21	28	35	42	49	56	63	70
8	8	16	24	32	40	48	56	64	72	80
9	9	18	27	36	45	54	63	72	81	90
10	10	20	30	40	50	60	70	80	90	100

Em seguida, alterar o programa de forma a permitir aos utilizadores indicar qual o intervalo de valores pretendidos (substituindo o intervalo fixo 1 a 10).

14. Escrever um programa que calcule e mostre no ecrã o factorial de um  $n^\circ$  indicado pelo utilizador.
15. Escrever um programa que calcule e mostre no ecrã o factorial de todos os números entre 1 e 8. O formato do resultado deverá ser:
- ```
O factorial de 1 é 1
O factorial de 2 é 2
O factorial de 3 é 3
...
```
16. Escrever um programa que permita determinar se um determinado número introduzido pelo utilizador é um número primo. Um número primo só é divisível por 1 e por si mesmo.
17. Escrever um programa que mostre no ecrã todos os números primos entre um limite inferior e um limite superior indicados pelo utilizador.
18. Uma empresa vai lançar uma campanha de promoção oferecendo um pack surpresa composto por várias unidades de dois tipos de brindes X e Y, para compras superiores a 1500 €. O valor de cada pack surpresa estipulado é de 40 €; o custo de um brinde do tipo X é de 5 € e de um brinde do tipo Y é de 2 €. Escrever m programa que permita

determinar as combinações possíveis para composição dos pack surpresa. O resultado deve ser apresentado no seguinte formato:

```
  X | Y
-----
  0 | 20
  2 | 15
  ...
```

19. Construir uma nova versão do programa para controlo das caixas de venda da Mecopanado (exercício 16 das aulas 2 e 3) considerando que o programa deve permitir registar os pedidos de vários clientes e que em qualquer momento, mediante a selecção de uma opção adequada, o funcionário deve poder saber quantos elementos de cada tipo já vendeu (restringir este registo às bebidas para não tornar o programa demasiado extenso).
20. Escrever um programa para controlo de pagamentos e devolução de troco de uma máquina de distribuição automática de bebidas e alimentos:

```
B1 - Água 50 cl - 0,50 €
B2 - Refrigerante - 0,85 €
A1 - Bolachas - 0,65 €
A2 - Batatas fritas - 0,70 €
A3 - Snack - 0,90 €
```

- a) O programa deve repetir sucessivamente os seguintes passos até a escolha ser o Z9 (este valor só é conhecido pelos operadores responsáveis pela manutenção das máquinas):
- Mostrar a lista de opções.
  - Aceitar a escolha do utilizador (só pode escolher um produto).
  - Pedir a quantia correspondente ao produto escolhido.
  - Simular o pagamento, aceitando uma sequência de valores 2 €, 1€, 0,50 €, 0,20 €, 0,10 € ou 0,05€, até perfazer uma quantia suficiente para pagar o produto escolhido. A operação pode ser cancelada a qualquer momento introduzindo o valor 0.
  - Calcular o troco, indicando quais e quantas moedas devem ser devolvidas.
- b) Acrescentar ao programa a possibilidade de controlar o nº de moedas existentes na máquina. Considerar que inicialmente existem 20 moedas de 1€, 0,50 €, 0,20 €, 0,10 € e 0,05€, e zero moedas de 2 €. O programa deve actualizar o nº de moedas disponíveis sempre que se recebe um pagamento ou se devolve um troco. Se o utilizador escolher a opção Z0 (código conhecido apenas pelo operador de manutenção) o programa deve apresentar no ecrã uma listagem com o nº actualizado de cada tipo de moedas.