

AULA 6

Objectivos

Esta aula prática faz uma introdução à utilização de subprogramas para a organização de programas complexos em unidades mais simples. No final, os alunos devem compreender a importância da decomposição de um programa em subprogramas, a definição e a chamada de subprogramas (métodos | funções | procedimentos), a passagem de parâmetros e o retorno do resultado de subprograma.

Um dos aspectos mais importantes na organização de um programa consiste na identificação das funções relevantes a implementar como subprogramas e na definição da sua assinatura. É particularmente importante a definição correcta dos parâmetros a passar a um subprograma e a definição do valor de retorno. Como, este processo depende fortemente da experiência adquirida através da realização de numerosos programas, uma parte dos exercícios propostos inclui uma sugestão dos subprogramas e respectivas assinaturas a implementar. Em certos casos são dadas apenas algumas pistas sobre os subprogramas a implementar, deixando a definição das assinaturas para os alunos. Nestes casos, recomenda-se fortemente que os alunos discutam as suas propostas com o docente antes de avançar para a implementação. Convém notar que a definição correcta ou incorrecta dos subprogramas, pode simplificar ou aumentar largamente a complexidade dos problemas a resolver.

1. Copiar, analisar e executar o programa abaixo que implementa uma versão simplificada de uma caixa registadora para um posto de abastecimento de combustíveis.

Em seguida, completar o programa acrescentando duas novas opções: “3. Gasolina 95” e “4. Gasolina 98”, atribuindo valores fictícios ao preço por litro dos novos combustíveis.

```
import pl.*;
class aula06_postoAbastecimento extends PlApp {
    static void main(String [] args) {
        int opcao;
        double quantidade, precoLitro;
        do {
            opcao = menu();
            precoLitro = 0;
            switch(opcao) {
                case 1:
                    precoLitro = 0.99;
                    break;
                case 2:
                    precoLitro = 1.09;
                    break;
                case 0:
                    break;
                default: println(" >>> Opção errada!");
            }
            if(precoLitro !=0) {
                quantidade = readDouble(" > quantidade (L): ", 0, 200);
                print(" > valor a pagar: ");
                printfd(5, 2, quantidade * precoLitro); println();
            }
        }while(opcao !=0);
        println("Fecho concluído!");
    }
}
```

// (continua...)

```

static int menu() {
    int op;
    println();
    println("1. Gasóleo normal");
    println("2. Gasóleo super power");
    println("0. Fecho");
    print  (" => Opção: ");

    op = readInt();
    return op;
}
}

```

2. Escrever um programa que simula a caixa registadora da bilheteira de um museu.

- A versão inicial do programa deve incluir um menu com os diversos tipos de bilhetes disponíveis –criança (1€), jovem, (3€) adulto (6€) e idoso (2€)–, permitindo ao utilizador escolher um tipo de bilhetes e a respectiva quantidade. O programa deve calcular e imprimir o valor a pagar correspondente. Deve usar um subprograma para a impressão do menu.
- Em seguida, completar o programa permitindo a cada cliente comprar diversos tipos de bilhetes, por exemplo, 2 bilhetes de adulto e 2 bilhetes de criança. O programa deverá funcionar da seguinte forma:

```

==> Cliente nº 1
[Menu]
opção: 3
quantidade: 2
--> valor a pagar: 12€
[Menu]
opção: 1
quantidade: 2
--> valor a pagar: 14€
[Menu]
opção: 5
--> total = 14 €
--> prima <enter> para
continuar
==> Cliente nº 2
[Menu] ...

```

No exemplo à esquerda, substituir [Menu] por:

```

1. Criança (1€)
2. Jovem (3€)
3. Adulto (6€)
4. Idoso (2€)
5. Total cliente
6. Anular venda
0. Fecho

```

- Finalmente, quando o total a pagar por um cliente for superior a 15 €, o programa deve verificar se a compra inclui bilhetes de criança ou idoso e atribuir gratuitamente dois bilhetes de criança ou um bilhete de idoso, descontando o valor correspondente no total a pagar pelo cliente. Usar um subprograma para determinar o desconto. Este subprograma de ter a seguinte assinatura:

```

static int desconto(int bilhCrianca, int, int bilhIdoso, int
valorTotal)

```

O valor devolvido deve corresponder ao valor do desconto (0, 1 ou 2 euros). No

caso de haver lugar a um desconto, deve aparecer uma mensagem no ecrã, recordando ao funcionário da caixa que deve informar o cliente sobre o desconto atribuído.

3. Considerar novamente o exercício nº 4 das aulas 2 e 3.

O programa deve ler a partir do teclado o valor de um empréstimo bancário em euros, uma duração em anos e uma taxa de juro anual (uma percentagem, ou seja, um valor entre 0 e 1), calcular e escrever no monitor o valor da mensalidade a pagar e o valor total a pagar. Usar as seguintes fórmulas para calcular os valores pedidos:

$$\text{mensalidade} = \frac{\text{juroMes} \times \text{baseMes}^{\text{numMeses}} \times \text{valorEmprestimo}}{\text{baseMes}^{\text{numMeses}} - 1}, \text{ tal que}$$

$$\text{juroMes} = \frac{\text{taxaJuro}}{12.0}, \quad \text{baseMes} = (1.0 + \text{juroMes})$$

$$\text{totalAPagar} = \text{mensalidade} \times \text{numMeses}$$

➔ Reescrever o programa criando um subprograma para calcular o valor da mensalidade. O subprograma deverá ter a seguinte assinatura:

```
static double mensalidade(int valorEmpresimo, int duracaoAnos,
double taxaJuro)
```

4. Considerar novamente o exercício nº 4 das aulas 2 e 3.

Uma fábrica decidiu oferecer aos seus clientes um desconto calculado em função do valor das encomendas e da antiguidade do cliente, determinado pela tabela:

Valor encomenda/antiguidade	≤ 3 anos	> 3 anos
< 2000 €	2,0%	2,5%
[2000 €, 5000 € [4,0%	5,0%
≥ 5000 €	6,0%	7,5%

Escrever um programa que leia do teclado o número de peças encomendadas, o respectivo preço unitário e o nº de anos que decorreram desde o primeiro negócio com o cliente, e calcule e imprima no ecrã o valor a pagar pelo cliente.

➔ Reescrever o programa criando um subprograma para calcular o valor do desconto a atribuir à encomenda. Em seguida, modificar o programa de forma permitir o tratamento de várias encomendas. O programa deve terminar quando for introduzido um nº de peças encomendadas negativo.

5. Escrever um subprograma que permita determinar se um nº é primo. O subprograma deve receber um valor inteiro e devolver um valor do tipo *boolean* (verdadeiro ou falso).

Para testar o subprograma escrever um programa que imprima no ecrã os números primos entre 3 e 100.

6. Escrever um subprograma que permita determinar a distância entre dois pontos $p1=(x1, y1)$ e $p2=(x2, y2)$. A fórmula da distância é $d = \sqrt{(x1 - x2)^2 + (y1 - y2)^2}$. Testar o subprograma.

7. A secção de controlo de qualidade de uma unidade de produção deve realizar o controlo do peso das peças produzidas. Existem dois tipos de peças: as peças do tipo A devem ter um peso de 1200g com um desvio máximo de ± 5 g; as peças do tipo B devem ter um peso de 1800g com um desvio máximo de ± 10 g. As peças com um desvio superior ao admitido são consideradas defeituosas.

Escrever um programa que leia uma sequência de valores, tipo da peça e peso da peça, e indique se a peça é ou não defeituosa. O programa deve terminar quando o utilizador insere o valor 'F' para o tipo de peça. Usar um subprograma para controlar o peso das peças e determinar se são ou não defeituosas.

8. Escrever um programa que lê do teclado uma data composta pelo mês eo ano, e o dia da semana em que começa esse mês (1 = domingo, 2 = segunda, ..., 7 = sábado) e que mostra no ecrã o calendário desse mês no formato apresentado abaixo.

```

-----
|   Março       - 2007   |
-----
|  D  S  T  Q  Q  S  S  |
-----
|                1  2  3  |
|  4  5  6  7  8  9 10  |
| 11 12 13 14 15 16 17  |
| 18 19 20 21 22 23 24  |
| 25 26 27 28 29 30 31  |
-----
    
```

Usar subprogramas para calcular o nº de dias do mês escolhido pelo utilizador (não esquecer de tratar os anos bissextos) e imprimir o calendário no ecrã.

9. Os quadros abaixo apresentam (a) os resultados de experiências desenvolvidas por Mendel, utilizando flores de ervilha (*Pisum Sativum*), e (b) os resultados de uma experiência posterior a Mendel, com flores da espécie *Miravilis Jalapa*.

Genótipo		Fenótipo
Gene 1	Gene 2	
B	B	B
P	B	P
B	P	P
P	P	P

Quadro (a)

Genótipo		Fenótipo
Gene 1	Gene 2	
B	B	B
V	B	R
B	V	R
V	V	V

Quadro (b)

Cores	
B	Branco
P	Púrpura
R	Rosa
V	Vermelho

Legenda

O programa listado no final desta aula lê do teclado as letras correspondentes aos dois genes que compõem o genótipo de uma flor de ervilha e mostra no ecrã a cor do fenótipo correspondente (opção 1 do menu proposto).

- a) Completar o programa de forma a tratar também a opção 2 do Menu proposto, correspondente às experiências realizadas com flores da espécie *Miravilis Jalapa*.
- b) Alterar o programa, criando um novo subprograma para escrever o nome do fenótipo correspondente a uma dada letra de acordo com a legenda acima. A assinatura do subprograma deverá ser `static void escreveFenotipo(char)`.

```
import pl.*;
class aula06_Mendel extends PlApp {
    static void main(String [] args) {
        int opcao;
        char gene1, gene2, fenotipo;

        do {
            opcao = menu();
            switch(opcao) {
                case 1: // flores de ervilheira
                    gene1 = lerGeneErvilheira(1);
                    gene2 = lerGeneErvilheira(2);

                    fenotipo = fenotipoErvilheira(gene1, gene2);

                    print(" > Fenotipo - ");
                    switch(fenotipo) {
                        case 'B': println("Branco"); break;
                        case 'P': println("Púrpura"); break;
                    }
                    break;

                case 2: // flores Miravilis Jalapa
                    break;
                case 0: // terminar
                    break;
                default:
                    println("Opcao errada! Tente novamente.");
            }
            println();
        } while(opcao != 0);
    }

    static int menu() {
        int op;

        println("1. Flores de ervilheira");
        println("2. Flores Miravilis Jalapa");
        println("0. Terminar");

        op = readInt(" => Opção? ");
        return op;
    }

    static char lerGeneErvilheira(int n) {
        char gene;
        do {
            gene = readChar(" > Gene " + n + " (B|P):");
        } while (Character.toLowerCase(gene) != 'b' &&
                Character.toLowerCase(gene) != 'p');
        return gene;
    }

    static char fenotipoErvilheira(char g1, char g2) {
        if(Character.toUpperCase(g1) == 'B' &&
            Character.toUpperCase(g2) == 'B')
            return 'B';
        else
            return 'P';
    }
}
```