

Prática 3 Controlo de fluxo e funções

Objetivos

- Resolução de exercícios em Java recorrendo a instruções de controlo de fluxo
- Decomposição de complexidade através de métodos/funções
- Manipulação de texto – Strings (objetos do tipo *java.lang.String*)

Tópicos

- Vetores
- Instruções de decisão (*if*, *switch*)
- Ciclos (*for*, *while*, *do .. while*)
- Funções (métodos estáticos)

Exercício 3.1

Escreva um programa que leia do teclado um número inteiro positivo e devolva a soma de todos os números primos até esse valor (inclusive). Repare que deve validar o valor de entrada repetindo a leitura se o valor não for válido (positivo).

Deve implementar uma função que devolva se um número é um número primo. Um número natural é um número primo quando tem exatamente dois divisores naturais distintos: o número 1 e ele mesmo.

Exercício 3.2

Um fundo de investimento fornece uma taxa de juros mensal fixa, que acumula com o saldo anterior do investimento (juros rendem juros). Escreva um programa em Java que peça ao utilizador o montante investido (positivo e múltiplo de 1000) e a taxa de juro mensal (entre 0% e 5%). Verifique se os valores são válidos e apresente o valor mensal do fundo nos próximos 12 meses, imprimindo o valor em cada mês.

Exercício 3.3

O jogo *AltoBaixo* consiste em tentar adivinhar um número (inteiro) entre 1 e 100. O programa escolhe um número aleatoriamente. Depois, o utilizador insere uma tentativa e o programa indica se é demasiado alta, ou demasiado baixa. Isto é repetido até o utilizador acertar no número. O jogo deve indicar quantas tentativas foram feitas e de seguida perguntar: “Pretende continuar? Prima (S)im”. O programa termina caso a resposta seja diferente de “S” ou “Sim”.

Sugestão: para ler uma palavra utilize o método next: String resposta = sc.next();

Exercício 3.4

Implemente um programa que leia do teclado as notas da componente prática (*notaP*) e teórica (*notaT*) de uma turma.

Defina uma função para calcular a nota final de um aluno a partir das notas das componentes prática e teórica, da seguinte forma:

- 66 (reprovado por nota mínima), se tiver obtido menos do que 7.0 em pelo menos uma das componentes;
- $0.4 * notaT + 0.6 * notaP$ (arredondada a unidades), nos restantes casos.

O programa deve começar por perguntar o número de alunos na turma. De seguida, crie um vetor bidimensional para armazenar as notas das componentes teórica e prática de todos os alunos e preencha-o com valores aleatórios (mas válidos, i.e. as notas podem variar de 0.0 a 20.0; arredonde os valores às décimas).

Processe as notas e imprima os resultados com o formato seguinte:

NotaT	NotaP	Pauta
11.3	9.3	10
16.7	5.1	66
7.8	18.9	14
10.6	15.9	14
16.9	5.9	66
1.9	12.7	66
17.6	4.8	66
0.7	12.1	66
8.7	8.6	9
19.2	1.4	66
17.5	3.4	66
11.6	11.4	11
7.2	8.5	8
1.9	1.4	66
19.3	14.9	17
0	12.1	12

Exercício 3.5

Escreva um programa que lê do teclado uma data no formato *mm/yyyy* (validando os valores) e o dia da semana em que começa esse mês (1 = Segunda, 2 = Terça, 3 = Quarta, 4 = Quinta, 5 = Sexta, 6 = Sábado, 7 = Domingo). O programa deve depois apresentar no monitor o calendário desse mês com o formato que a seguir se apresenta.

Implemente a funcionalidade pretendida com três funções: leitura de valores, com validação, cálculo de dias no mês, considerando os anos bissextos, e impressão de resultados.

```
February 2019
Su Mo Tu We Th Fr Sa
                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
```