

Programação I

Folha de Exercícios 7

António J. R. Neves

João Rodrigues

Osvaldo Pacheco



2015–2016

Folha de Exercícios 7

Resumo:

- Estruturas de Dados - registos (ou Tipos Compostos)
- Definição de registos (ou classe em Java)
- Criação de novos tipos de dados
- Declaração de variáveis de novos tipos
- Cópia de variáveis tipo referência
- Utilização de registos em funções

Em muitas situações práticas, para caracterizar uma certa entidade, precisamos de registar vários itens de informação, geralmente de diversos tipos. Por exemplo, para criar uma conta de email no GMail, temos de preencher o nome (próprio + apelido) a data de nascimento (dia, mês e ano), o sexo, o país, etc. Cada um destes itens pode ser armazenado numa variável distinta de tipo String, int, char, etc, mas também é possível e preferível armazená-los todos juntos em diferentes campos de um objeto único de um tipo composto.

A maioria das linguagens de programação permitem que o programador defina novos tipos de dados para adequar a representação da informação às condições concretas do problema. Estes tipos de dados são designados normalmente por *registos* (records), *tuplos* (tuples), *tipos compostos* ou *structs* (em língua inglesa). Na linguagem JAVA podemos utilizar classes (**class**) para definir novos tipos de registo. (Na verdade, as classes são uma generalização da noção de registo e também podem servir para outras finalidades como coleções de funções, por exemplo.)

7.1 Problemas para resolver

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.

Exercício 7.2

Escreva um programa para calcular distâncias entre pontos.

- Defina um registo (`class Ponto2D`) para representar um ponto como um par de valores reais (x, y).
- Escreva uma função para ler um ponto.
- Escreva uma função para escrever no monitor um ponto no formato (`#.#, #.#`).
- Escreva uma função para o cálculo da distância entre dois pontos
- Escreva um programa (função `main`) que peça ao utilizador pontos até à introdução do ponto (0, 0). O programa deve contar quantos pontos foram introduzidos e calcular a soma das suas distâncias à origem (ponto (0, 0)). O programa deverá também determinar qual o ponto mais afastado da origem. O modelo de interação do programa será o seguinte (procure respeitar a formatação do exemplo que se segue):

Introduza um ponto:

Coordenada x: 1

Coordenada y: 0

Introduza um ponto:

Coordenada x: 0

Coordenada y: -4

Introduza um ponto:

Coordenada x: -2

Coordenada y: 0

Introduza um ponto:

Coordenada x: 0

Coordenada y: 3

Introduza um ponto:

Coordenada x: 0

Coordenada y: 0

A soma das distâncias dos 4 pontos à origem é 10.0

O ponto mais afastado da origem foi: (0.0, -4.0)

Exercício 7.3

Complete o programa `DoStats.java` para que leia uma lista de números reais até ao *fim-de-ficheiro* (indicado no teclado por `Ctrl-D` no início de uma linha) e mostre um conjunto de estatísticas no final: máximo, mínimo, médian, variância, etc. Para isso falta:

- Definir um tipo de dados que permita guardar medidas estatísticas de valores reais: valor máximo, mínimo, contagem dos valores (n), somatório dos valores ($\sum x_i$) e somatório dos quadrados ($\sum x_i^2$).
- Escreva a função `updateStats` que dado um valor real e um registo de estatísticas, atualiza esse registo para contabilizar o novo valor.
- Escreva as funções `mean` e `variance` que calculam a média ($\mu = \sum x_i/n$) e a variância ($\sigma^2 = \sum x_i^2/n - \mu^2$) a partir dos dados guardadas num registo de estatísticas.

7.2 Exercícios complementares

Exercício 7.4

Pretende-se implementar um programa que permita efetuar aritmética de números complexos.

- Defina um registo (`class Complexo`) para representar um número complexo como um par de valores reais (r, i).
- Escreva uma função para ler um valor complexo.
- Escreva uma função que imprima um número complexo com o seguinte formato: `#.#+/-#. #i` (em que $+/-$ depende do sinal da parte imaginária).
- Escreva uma função para cada uma das operações seguintes:
 - Adição: $a + bi + c + di = (a + c) + (b + d)i$
 - Subtração: $a + bi - c + di = (a - c) + (b - d)i$
 - Multiplicação: $(a + bi) * (c + di) = (ac - bd) + (bc + ad)i$
 - Divisão: $(a + bi)/(c + di) = ((ac + bd)/(c^2 + d^2)) + ((bc - ad)/(c^2 + d^2))i$
- Escreva um programa (função `main`) que peça ao utilizador uma operação ($+$, $-$, $*$, $/$) e dois operandos complexos, e apresente o resultado respetivo. Deve repetir isso enquanto a operação for diferente de `'=`'.

O modelo de interação do programa será o seguinte (procure respeitar a formatação do exemplo):

```
Operação: +
Introduza um número complexo:
Parte real: 1
Parte imag: 1
Introduza um número complexo:
Parte real: 2
Parte imag: 2
1.0+1.0i + 2.0+2.0i = 3.0+3.0i
Operação: x
Operação não permitida!
Operação: -
Introduza um número complexo:
Parte real: 50
Parte imag: 50
Introduza um número complexo:
Parte real: 100
Parte imag: 100
50.0+50.0i - 100.0+100.0i = -50.0-50.0i
Operação: =
O programa vai terminar...
```

Exercício 7.5

Escreva um programa capaz de calcular uma estimativa da taxa de alcoolemia de um indivíduo (TAS).

O programa deverá começar por solicitar ao utilizador, com validação, o nome, sexo, o peso (kg), a quantidade de bebida ingerida (ml), o teor alcoólico da bebida (graduação em % volume) e se foi consumida em jejum. Deverá ser desenvolvida uma função que peça ao utilizador esta informação e devolva um registo preenchido.

O valor aproximado da taxa de alcoolemia deverá ser calculado numa função usando a fórmula:

$$TAS = \frac{DensidadeAlcool \times QuantidadeBebida \times TeorAlcoolico/100}{Peso \times Coeficiente}$$

onde $DensidadeAlcool = 0.8(g/ml)$ e $Coeficiente$ é dado na tabela abaixo.

| Coeficiente (l/kg) | Situação |
|--------------------|----------------------------------|
| 0.6 | Mulheres em jejum |
| 0.7 | Homens em jejum |
| 1.1 | Consumido à refeição (não jejum) |

Deverá desenvolver também uma função para a impressão do resultado calculado.

Exercício 7.6

Escreva um programa que permita manipular os valores de temperatura de uma estação meteorológica numa série de 5 dias consecutivos. O programa deverá ler os valores da temperatura mínima (tMin) e da temperatura máxima (tMax) de cada dia e calcular a amplitude térmica correspondente ($tMax - tMin$).

Deve desenvolver uma função que aceite apenas temperaturas no intervalo $[-20, 50]$ e que garanta um valor de temperatura máxima superior ou igual à temperatura mínima. Esta função deve devolver um registo de temperaturas preenchido. Utilize uma estrutura de dados para armazenar a informação das temperaturas de um dia, bem como a sua amplitude térmica.

No final, o programa desenvolvido deve indicar a amplitude térmica máxima e o dia em que ocorreu.