

Programação I

Folha Exercícios 9

António J. R. Neves
João Rodrigues
Osvaldo Pacheco

2017/18

Folha Exercícios 9

Resumo:

- **Strings** (sequências de caracteres)
- Codificação de caracteres: **Unicode** e **ASCII**
- Classe **Character**
- Operações com caracteres
- Propriedades das **Strings**
- Leitura e escrita de **Strings** e caracteres
- Classe **String**
- Strings como argumentos de funções

Existem aplicações informáticas que, para além de necessitarem de processar dados numéricos, também necessitam de processar texto. Em programação, um texto pode ser visto como uma sequência de caracteres. Contudo, uma sequência de caracteres não é simplesmente um conjunto de caracteres pois estes têm particularidades e necessitam de um conjunto de operações específicas para a sua manipulação.

Em Java existe o tipo de dados referência **String** para a manipulação de texto. Este tipo de dados é descrito pela classe **String** que disponibiliza um vasto conjunto de funções para a sua manipulação.

A classe **Character** tem também um papel importante na manipulação de texto, dado que disponibiliza um vasto conjunto de funções para a manipulação de caracteres.

9.1 Problemas para resolver

Exercício 9.1

Pretende-se escrever um programa que leia do teclado uma frase e calcule alguma informação relativamente aos caracteres que a constituem. O programa deve calcular e escrever o número de caracteres minúsculos, o número de caracteres maiúsculos e o número de caracteres numéricos. Pode recorrer aos métodos **Character.toUpperCase()**, **Character.toLowerCase()** e **Character.isDigit()**.

A saída de dados deve obedecer ao formato seguinte:

Análise de uma frase

Frase de entrada -> #####

Número de caracteres minúsculos -> ##

Número de caracteres maiúsculos -> ##

Número de caracteres numéricos -> ##

Crie uma função booleana **isVowel(char c)** para verificar se um carácter *c* é uma vogal e altere o programa anterior para contar também vogais e consoantes. Como sugestão, tenha em consideração que o número de consoantes é igual ao número de caracteres alfabéticos menos o número de vogais.

Número de vogais -> ##

Número de consoantes -> ##

Exercício 9.2

Faça uma função que devolva um acrónimo para um dado nome. Por exemplo:

```
acronym("Organização das Nações Unidas") → "ONU"  
acronym("Universidade de Aveiro") → "UA"
```

(Aqui, $X \rightarrow Y$ significa: *X* deve devolver *Y*.) Sugestão: extraia apenas as maiúsculas.

Implemente depois um programa que vá pedindo texto ao utilizador até que este introduza uma linha vazia e, para cada uma, escreva o seu acrónimo com base na função desenvolvida.

Exercício 9.3

Crie uma função que indique quantas palavras contém uma string. Considere que uma palavra é qualquer texto que não contenha espaços, tabs (`\t`) ou mudanças de linha (`\n`). Por exemplo:

```
countWords("isto é 1 frase ") → 4.
```

Sugestão: pode percorrer a string símbolo a símbolo e manter uma variável que indica se está dentro ou fora de uma palavra. Se estiver “dentro” e aparecer um espaço (ou `\t` ou `\n`), passa a estar “fora”. Se estiver “fora” e aparecer outro carácter, passa a estar “dentro” e conta mais uma palavra.

Exercício 9.4

As matrículas automóveis em Portugal podem ter um de três padrões: AA-00-00, 00-00-AA ou 00-AA-00.

Crie uma função booleana **matchPattern(str, pattern)** que indique se uma dada string **str** corresponde a um dado padrão **pattern**. Um A no padrão corresponde a qualquer letra maiúscula na string, um 0 corresponde a qualquer dígito, e outros caracteres devem coincidir exatamente.

Invocando esta função, escreva um programa que verifique se um texto introduzido pelo utilizador é uma matrícula portuguesa válida.

Exercício 9.5

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:

```
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!)

9.2 Exercícios complementares

Exercício 9.6

Na terra do Alberto Alexandre (localmente conhecido por Aubeto Auexande), o dialeto local é semelhante ao português com duas exceções:

- não dizem os R's;
- trocam os L's por U's.

Implemente um tradutor de português para o dialeto do Alberto que permite a introdução de uma frase e apresente a sua tradução. Por exemplo “lar doce lar” deve ser traduzido para “ua doce ua”.

Exercício 9.7

Crie uma função que coloque em maiúsculas a primeira letra de cada palavra de uma string:

```
capitalize("era uma vez") □ "Era Uma Vez".
```

Sugestão: pode usar a mesma “máquina de estados” do problema anterior para detetar o início de cada palavra e usar o método **Character.toUpperCase** para converter em maiúsculas.

Exercício 9.8

Faça uma função inversa da do problema 8.5. Por exemplo, pretende-se:

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

Repare que o primeiro parâmetro é uma string, mas o resultado é um **int**. (Pode converter o char '3' no valor 3 com a expressão '3'-'0'.)

Escreva um programa para testar a função. Lembre-se que um número em base N só pode ter dígitos na gama 0, ..., $N-1$.

9.3 Exercícios de revisão

Exercício 9.9

Implemente e teste uma função **readChar(prompt)** que permita ler um carácter do teclado, utilizando como base a função **nextLine** da classe **Scanner**. A função deve escrever o *prompt* passado como parâmetro e deve garantir que a linha lida contém exatamente um carácter. Caso contrário, volta a pedir.

Exercício 9.10

O Índice de Massa Corporal (IMC) é um dos critérios utilizados para detetar problemas de obesidade nos humanos. Comece por definir uma estrutura de dados que suporte a informação de um indivíduo: nome, idade (anos), sexo (M/F), peso (kg) e altura (m). Construa um programa para o cálculo do IMC, tendo em conta que a idade inserida pelo utilizador não deve ser inferior a 18 anos e o programa não deve aceitar entradas sem sentido (exemplo pesos e alturas negativas, sexo diferente de M/F, etc.). Deve implementar uma função para leitura dos dados, outra para o cálculo do IMC e uma para a impressão do relatório, tal como apresentado de seguida.

Para a determinação da mensagem, utilize a seguinte tabela¹:

IMC Feminino	IMC Masculino	Classificação
< 19,1	< 20,7	Abaixo do peso
19,1 – 25,8	20,7 – 26,4	Peso ideal
25,9 – 27,3	26,5 – 27,8	Pouco acima do peso
27,4 – 32,3	27,9 – 31,1	Acima do peso
> 32,4	31,2	Obesidade

O IMC calcula-se através da expressão: $IMC = \frac{Peso}{Altura^2}$.

O programa deve imprimir um relatório com o seguinte formato:

```

|-----|
| Calculo do Indice de Massa Corporal |
|-----|
| Nome: ##### |
| Sexo: Masculino|Feminino |
| IMC : ##.# (kg/m2) |
| Comentario: [ver tabela] |
|-----|
  
```

<http://indicedemassacorporal.com/index.html>

Exercício 9.11

O responsável por um Clube de Natação pretende construir um programa que o ajude a calcular o valor a pagar por um utente no ato da inscrição. O programa deve pedir ao utilizador a introdução de:

- Qual o nome do indivíduo.
- Qual a modalidade a frequentar (1 para Iniciação e 2 para Aperfeiçoamento).
- O número de horas semanais (de 1 a 10) de prática.
- Se o utente tem familiares a frequentar o clube (0 para indicar que não tem)
- Se é a primeira vez que se inscreve neste clube e o número de mensalidades que pretende pagar.

Comece por definir uma estrutura de dados que suporte a informação de uma nova inscrição. Para o cálculo do valor a pagar é preciso ter em conta que cada hora de Iniciação custa 4 Euros e cada hora de aperfeiçoamento custa 6 Euros. O valor da inscrição é de 50 Euros para quem vem frequentar a piscina pela primeira vez e de 30 Euros para quem já esteve inscrito. Com a inscrição é obrigatório o pagamento de pelo menos a primeira mensalidade. Havendo familiares a frequentar existe um desconto de 10% do valor da mensalidade para o caso de o utente ter um familiar e de 25% para o caso de dois ou mais familiares. Considere que todos os meses têm 4 semanas.

Implemente uma função para a leitura dos dados e outra para a impressão da tabela.

Após pedir todos os dados, o programa deve apresentar os cálculos no seguinte formato:

Nova Inscrição para o Clube de Natação

Nome: #####

Primeira Vez

3 horas de aperfeiçoamento

1 Familiar

Item	Quantidade	Valor
Inscrição	1	50,00
Mensalidade	1	64,80

Total		114,80