

String

A classe `String` representa cadeias de caracteres. Todas as *strings* literais, como por exemplo "abc", são implementadas como instâncias dessa classe.

Os objectos do tipo `String` são constantes, os seus valores não podem ser alteradas depois de serem criados.

```
String str = "abc";
```

representa a cadeia de caracteres criada seguidamente:

```
char data[] = {'a', 'b', 'c'};  
String str = new String(data);
```

Exemplos:

```
String str = new String(); // String vazia
```

ou

```
String str = ""; // String vazia.
```

```
str += "minha String";  
String outraStr = str; // outraStr é uma referência para "minha String"  
str += " imutável"; // str agora é uma referência para "minha String imutável"
```

`str` agora é uma referência para um novo objecto `String` "minha String imutável". A concatenação abandona o objecto antigo e cria um novo objecto (neste caso, "minha String imutável") com a nova string. Assim se a variável `outraString` não referisse a antiga string, esta seria perdida.

Alguns métodos:

`public char charAt(int index)` retorna o caracter correspondente à posição `index`

`public int compareTo(String anotherString)` compara duas *strings* lexicograficamente; retorna um inteiro negativo, zero ou positivo conforme a *string* representada pelo objecto é menor, igual ou Maior que a *string* representada pelo argumento.

`public int compareToIgnoreCase(String str)` compara duas *strings* lexicograficamente, ignorando diferenças entre maiúsculas e minúsculas.

`public boolean equals(Object anObject)` compara as *strings* representadas pelo objecto e pelo argumento.

`public boolean equalsIgnoreCase(String anotherString)` compara as *strings* representadas pelo objecto e pelo argumento, ignorando diferenças entre maiúsculas e minúsculas.

`public int indexOf(int ch)` retorna o índice correspondente à primeira ocorrência de `ch` na *string* ou -1 no caso da *string* não conter o carácter `ch`.

```
public int length() retorna o número de caracteres da string representada.  
public String toLowerCase() retorna a string convertida em minúsculas.  
public String toUpperCase() retorna a string convertida em maiúsculas.
```

StringBuffer e StringBuilder

As classes `StringBuffer` e `StringBuilder` representam *strings* que podem ser alteradas (o comprimento e conteúdo dum objecto pode ser alterado). A diferença entre estas duas classes é que os métodos da classe `StringBuffer` são *synchronized*, (sendo assim uma classe adequada para um ambiente onde se utilizam os recursos de *multi-thread*) e os métodos da classe `StringBuilder` não são (sendo adequada se a *string* for acedida apenas por uma *thread* uma vez que a sua execução é mais rápida).

```
StringBuffer strb = new StringBuffer("minha String buffer");  
StringBuffer outraStrb = strb; // outraStrb é uma referência para "minha String buffer"  
strb.append(" que se altera"); // strb e outraStrb agora referem  
// "minha String que se altera"
```

Alguns métodos:

```
public StringBuffer append(String str) acrescenta a string str.  
public char charAt(int index) retorna o caracter correspondente à posição index.  
public int indexOf(String str) se a string str estiver contida na string representada pelo objecto, retorna o índice do primeiro caracter, caso contrário retorna -1.  
public int length() retorna o número de caracteres da string representada.  
public void setCharAt(int index, char ch) o caracter que estava na posição index é substituído por ch.
```

Exercícios

1. Escreva uma aplicação que implemente o jogo do enforcado.
2. Defina uma classe para representar a seguinte informação acerca de um documento:

Título

Conjunto de autores (versão array de objectos e depois versão `ArrayList` de objectos)

Texto

A classe deve ter as seguintes funcionalidades:

Acrescentar um autor ao documento

Remover um autor

Acrescentar texto

Contar as palavras do texto (as palavras podem estar separadas por mais do que um separador normal e ainda por vírgulas ou pontos finais.

Substituir por letras maiúsculas as primeiras letras das palavras depois de pontos finais

Contar as ocorrências de uma dada palavra.