

Packages

Os *packages* permitem criar um grupo de classes relacionadas (logicamente).

Usam-se para:

- Tornar mais fácil encontrar e usar classes;
- Evitar conflitos de nomes entre classes;
- Controlar o acesso de umas classes às outras.

Um *package* é uma unidade de uma biblioteca (*library*) de classes formada por um dado conjunto de classes.

Conflitos entre Nomes

Os nomes de todos os membros de uma classe estão isolados dos nomes dos membros de outras classes. Mas os nomes de classes podem coincidir.

Para criar um *package* coloca-se uma classe dentro dele.

Para incluir uma classe num *package* coloca-se a instrução *package* no início do ficheiro fonte no qual a classe está definida.

```
package utilitarios;
```

O nome do *package* é implicitamente prefixado a cada nome de classe (tipo) contido no *package*.

Uma instrução *package* deve ser a primeira instrução de um ficheiro (para além dos comentários).

O âmbito de validade de uma instrução *package* estende-se a todo o ficheiro fonte.

Em cada ficheiro só pode existir uma declaração de um *package*.

Se não se usa a instrução *package*, a classe pertence ao *unnamed package*. Todos os tipos que não especificam nome de *package* são tratados como pertencendo ao mesmo *package*, único para a máquina virtual em execução.

Ficheiros Fonte e Ficheiros Classe

Um ficheiro fonte constitui uma unidade de compilação, pode conter mais que uma classe, mas só pode haver uma classe pública, que deve ter o mesmo nome que o ficheiro fonte (incluindo letras maiúsculas, mas excluindo a extensão *.java*).

Depois de compilado obtém-se um ficheiro para cada classe, com o nome da classe e a extensão *.class*.

Se um ficheiro fonte não contém nenhuma classe pública, pode ter qualquer nome.

Os ficheiros classe devem ser colocados num diretório cujo nome reflete o nome do *package*, para que o compilador e o interpretador possam encontrar as classes.

A variável de ambiente *CLASSPATH* (colocada através do sistema operativo ou do ambiente de desenvolvimento) contém um ou mais diretórios que são usados como raízes para pesquisa de ficheiros *.class*.

No ambiente de desenvolvimento Visual J++ a variável *CLASSPATH* é colocada seguindo os menus *Tools, Options, Directories*.

Exemplo:

```
CLASSPATH=.;G:\JAVA\biblio
```

Classes pertencentes ao:	<i>package utilitarios;</i>
devem ser colocadas no diretório	<i>G:\JAVA\biblio\utilitarios</i>
Classes pertencentes ao:	<i>package curso.graficos.trab2;</i>
devem ser colocadas no diretório	<i>G:\JAVA\biblio\curso\graficos\trab2</i>

Para nomes de *packages* usam-se normalmente sempre letras minúsculas.

Acesso a Packages

Ao escrever código fora do *package* que necessite de classes (tipos) declarados dentro de um *package* pode-se:

- Preceder cada nome de um tipo pelo do *package*, ou
- Importar parte ou todo o *package*.

Exemplos:

```
utilitarios.Clasel c1 = new utilitarios.Clasel();
```

```
import utilitarios.Clasel;  
Clasel c1 = new Clasel();
```

```
import utilitarios.*;  
Clasel c1 = new Clasel();
```

Código de um *package* importa outro código do *package* que necessite implicitamente.

A instrução *import* deve ser colocada depois de uma declaração de *package*, mas antes de outro código.

Colisão entre Nomes de Classes

Se duas bibliotecas são importadas e incluem classes com o mesmo nome, mas não é efetuado nenhum acesso a qualquer uma dessas classes com o mesmo nome, não há problema. Se é necessário aceder a algumas dessas classes temos de especificar o nome completo da classe prefixando o nome do *package*.