

# Ficha Prática 9

António Nestor Ribeiro, Paulo Azevedo, Mário Martins  
{anr,pja,fmm}@di.uminho.pt  
PPIV (LESI)

2005/06

## Objectivos

1. Classes Abstractas.
2. Excepções: classes de excepções e respectivo tratamento.

## Excepções

O mecanismo de excepções em Java, permite que o programador construa as suas próprias classes de excepção, de forma muito simples, por herança da superclasse das excepções, a classe `Exception`.

O código necessário para criar uma classe de excepções toma a seguinte forma:

```
class MinhaExcepcao extends Exception {  
  MinhaExcepcao() {super();}  
  MinhaExcepcao(String s) {super(s);}  
}
```

## Lançar uma excepção

Quando existe uma situação de erro, por exemplo retirar um elemento da colecção quando ele não existe, o programador pode explicitamente sinalizar o envio de um erro que deve ser posteriormente tratado.

Para tal apenas é necessário criar uma instância da excepção pretendida, da forma:

```

...
throw new <nomeClasseExcepcao>(); //construtor vazio
...
...
throw new <nomeClasseExcepcao>("mensagem que queira enviar");
//construtor parametrizado por uma String

```

A única forma que um programador tem de saber que um determinado método pode lançar uma exceção, é isso estar explicitamente reflectido na assinatura do método.

Para tal a assinatura do método deve ser feita de forma a indicar que o método pode "libertar alguma" exceção. Isso deve ser feito de acordo com a seguinte sintaxe:

```

public <tipoDados><nomeMetodo>(<listaParametros>)
throws <listaClassesExcepcao>

```

Um método que invoque outro que pode lançar uma exceção, deve precaver-se e ter disponível código de tratamento para essa exceção. Se não o fizer, então esse método deve explicitamente (na sua cláusula de throws) declarar que também pode libertar uma exceção desse tipo.

## Apanhar uma exceção

Como referido atrás ao invocar um método que pode lançar uma exceção, o programador deve prever a existência de código de tratamento do erro, por forma a que o programa não acabe de forma brusca, devido ao facto de a exceção não ter sido devidamente identificada e tratada.

A construção Java que permite apanhar as exceções e tratá-las tem a seguinte forma:

```

try{
//código que liberta excepções
...
...
}
catch(<tipoExcepcao1>){
//tratamento da excepcao
}
catch(<tipoExcepcao2>){
//tratamento da excepcao

```

```
}  
catch (...){...}  
  
finally{  
//código que pode ser executado após as  
//cláusulas de catch  
  
}
```

Tenha em linha de conta que esta estrutura é muito semelhante a um `switch`, logo após entrar numa zona de tratamento a uma classe de excepção, sai da estrutura `try/catch` e só pode executar o código que estiver em `finally`.

A zona de `finally` é opcional e pode não existir.

## Exercícios

1. Implemente o problema da Biblioteca Multimédia recorrendo aos ensinamentos da Ficha Prática 7. Termine a implementação da classe abstracta `ItemMultimedia` e da classe `BibMultimedia`.
2. Crie as classes de excepção necessárias e codifique as situações em que deve lançar excepções. Sugere-se que valide as pré-condições recorrendo a esta nova técnica de programação, por forma a tornar a classe `BibMultimedia` robusta.