

## Extraindo superclass

Agora que finalizamos todo o processo de refatoração na parte de registro de pagamentos, faremos também esse processo na classe `LeitorPagamento` que é ótima candidata a esse processo.

Note que atualmente essa classe lida apenas com arquivos CSV, ou seja, se quisermos criar um leitor de XML que tenha um comportamento similar, como é o caso de pegar todos os arquivos dentro de recursos, ou até mesmo gerar o backup, teremos que fazer um *copy and paste* ou criar um [God Object](https://en.wikipedia.org/wiki/God_object) ([https://en.wikipedia.org/wiki/God\\_object](https://en.wikipedia.org/wiki/God_object)) que não é algo desejável.

Em outras palavras, vamos refatorar o nosso código para permitir tal flexibilidade.

Para que seja fácil permitir a criação de novos leitores distintos de CSV, uma das abordagens que podemos aplicar é a extração de uma superclass, isto é, uma classe que vai conter um comportamento comum de qualquer leitor que entre no nosso sistema, e então, todo leitor novo fará extensão a essa classe.

### Extraindo superclass

Para fazermos isso, vá até a classe `LeitorPagamento` e utilize o **Refactor This** (**Ctrl + Shift + Alt + T** / **Ctrl + T**) > **Superclass....** Então, coloque o nome da classe como `Leitor` e deixe no mesmo pacote, ela ainda fará parte de `io`.

Em seguida, marque todos os membros que faz sentido enviar para essa classe, no caso são os seguintes membros:

- `geraBackup()`
- `devolveArquivo()`
- `lista()`
- `RECURSOS`
- `BACKUP`

Por fim, execute novamente o projeto e veja se tudo funciona como anteriormente.