PROGRAMAÇÃO ORIENTADA PELOS OBJECTOS

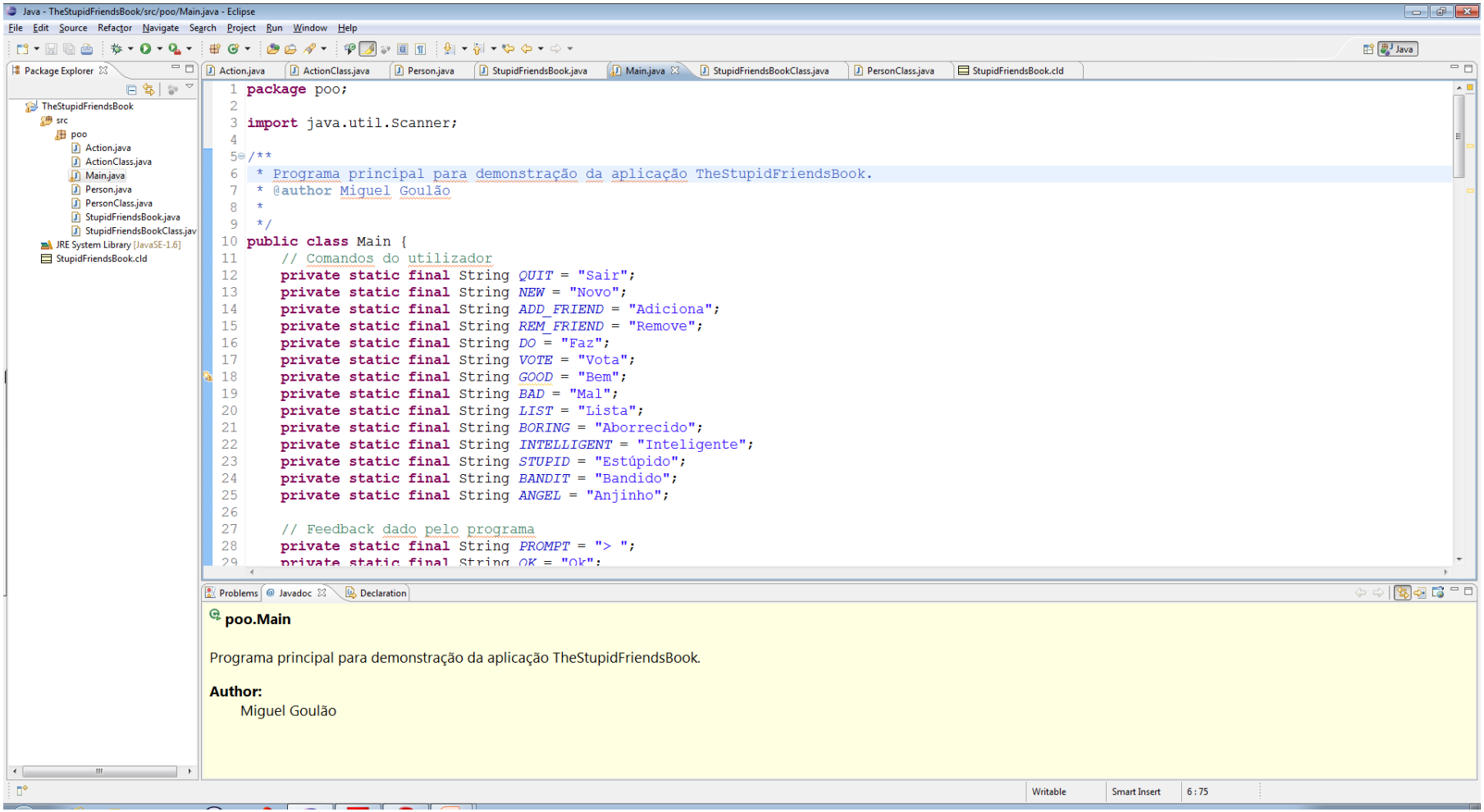
Aula 3

Implementação de interfaces e documentação

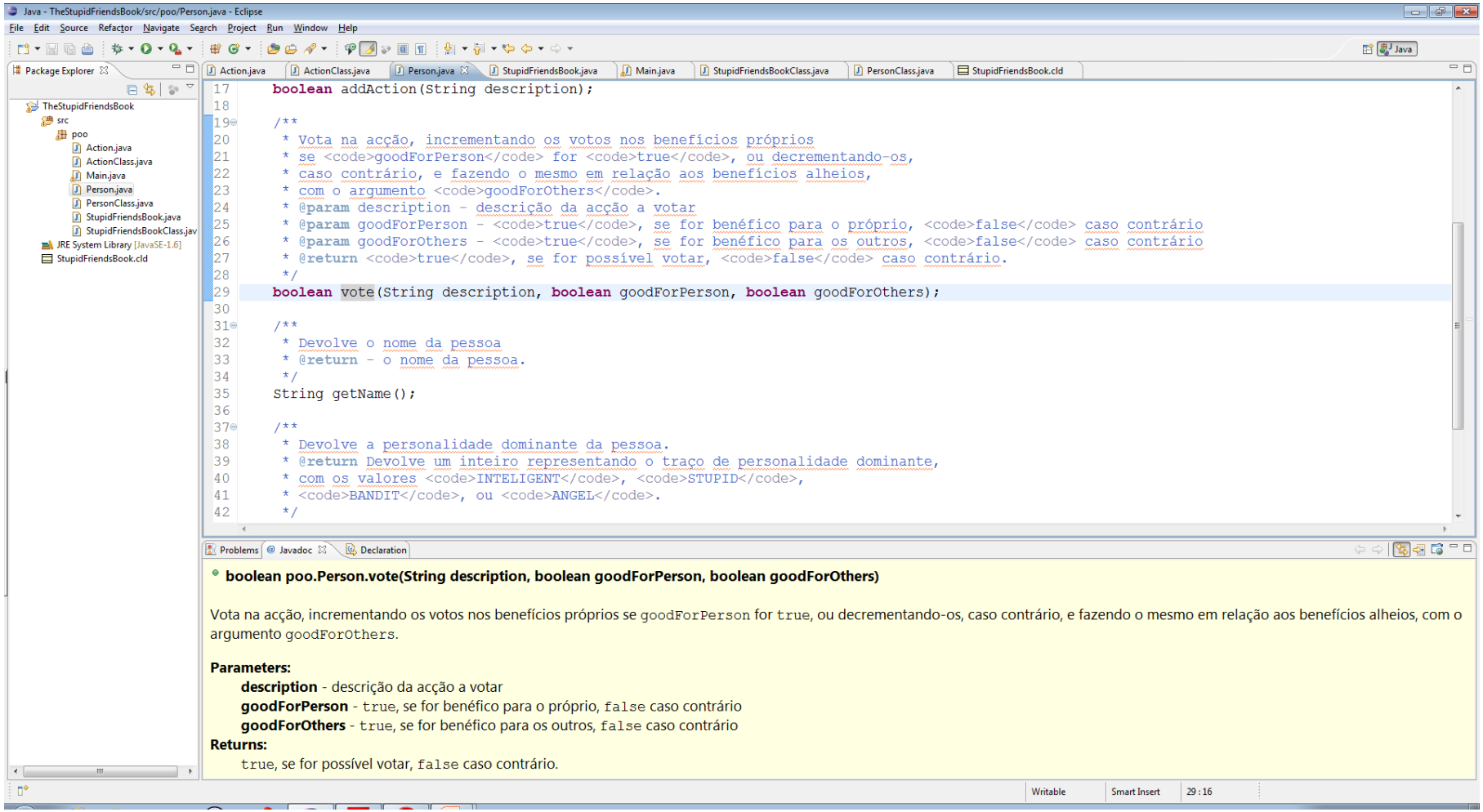
Implemente as suas classes

- Escolha uma ordem que facilite o ensaio incremental da sua aplicação
- Evite deixar "pontas soltas"
- Lembre-se **sempre** da regra da versão estável
- Vá comentando o seu código à medida que o escreve
  - A experiência mostra que comentários deixados para o fim acabam por não ser feitos, ou são feitos à pressa
  - Comentar o cabeçalho dos métodos antes de os desenvolver obriga-o a pensar no que tem de fazer, antes de os começar a programar

Implementação



Documentação



Convenções de documentação

- Escreva a documentação para as pessoas que vão ter de usar o seu código e mantê-lo
  - Documente cuidadosamente a interface pública do seu código para que outros o possam usar de forma correcta e eficiente
  - Documente cuidadosamente a interface privada e os detalhes de implementação internos, para que outros possam manter e melhorar o seu código

Convenções de documentação

- Assuma sempre que alguém completamente estranho ao seu código o vai ter de entender, mais cedo ou mais tarde!
  - Se passar tempo suficiente, até o autor de código mal documentado terá dificuldade em o compreender
- Mantenha o seu código e comentários sincronizados
  - Ambos fazem parte do produto, não menospreze nenhum dos dois
- Seja claro, conciso e preciso

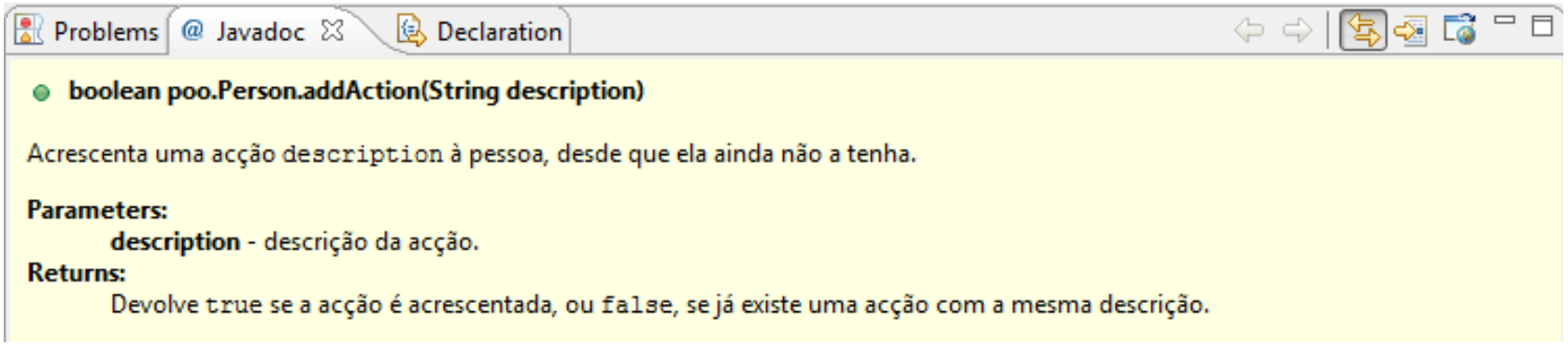
Três tipos de comentário, em Java

- Comentário de documentação
  - Começa em /** e termina em */
- Comentário standard (estilo emprestado do C)
  - Começa em /* e termina em */
- Comentário de uma linha
  - Começa em // e termina no fim da linha

Comentário de documentação

- Exemplo:

```
/**
 * Acrescenta uma acção <code>description</code> à pessoa, desde que ela
 * ainda não a tenha.
 * @param description - descrição da acção.
 * @return Devolve <code>true</code> se a acção é acrescentada,
 * ou <code>false</code>, se já existe uma acção com a mesma descrição.
 */
boolean addAction(String description);
```



Comentário de documentação

- Pode ser usado para comentar qualquer elemento
  - Declaração de interface, classe, método, construtor, campo
- Estes comentários dão a informação necessária para que o Javadoc possa gerar documentação em formato html (ao estilo que costuma ver na documentação da API do Java)
- O Javadoc apenas reconhece estes comentários quando eles aparecem imediatamente antes do elemento a comentar.

Comentário de documentação

- O Javadoc não reconhece estes comentários fora do sítio esperado
  - Não use este formato para comentar detalhes de implementação
- O Javadoc apenas reconhece um bloco de comentários por elemento
- O principal objectivo destes comentários é definir um contrato entre o cliente e o fornecedor de um serviço
  - Não coloque nestes comentários os detalhes de implementação

Comentários de documentação

- Escreva os comentários da interface de programação (API) **antes** de escrever o código
- Documente os membros *públicos*, *protegidos*, de *package* e *privados*, ou seja **todos**
- Escreva uma descrição sumária para cada package
  - É necessário criar um ficheiro de comentários de package especial denominado `package.html` e colocar esse ficheiro dentro do package, para que o Javadoc o encontre
- Escreva uma descrição sumária para cada aplicação, ou grupo de packages
  - É necessário criar um ficheiro de comentários `overview.html` e indicar ao Javadoc a sua localização

Estilo nos comentários de documentação

- Use um formato consistente ao longo de toda a documentação
  - Tipicamente, o comentário começa com um texto, auto-contido, descritivo para a entidade
  - Depois, usam-se várias tags Javadoc, seguidas do respectivo texto, para documentar detalhes como:
    - Parâmetros de operações
    - Valores a retornar pela operação
    - ...
- Envolva palavras chave, identificadores e constantes com as tags **<code>...</code>**.

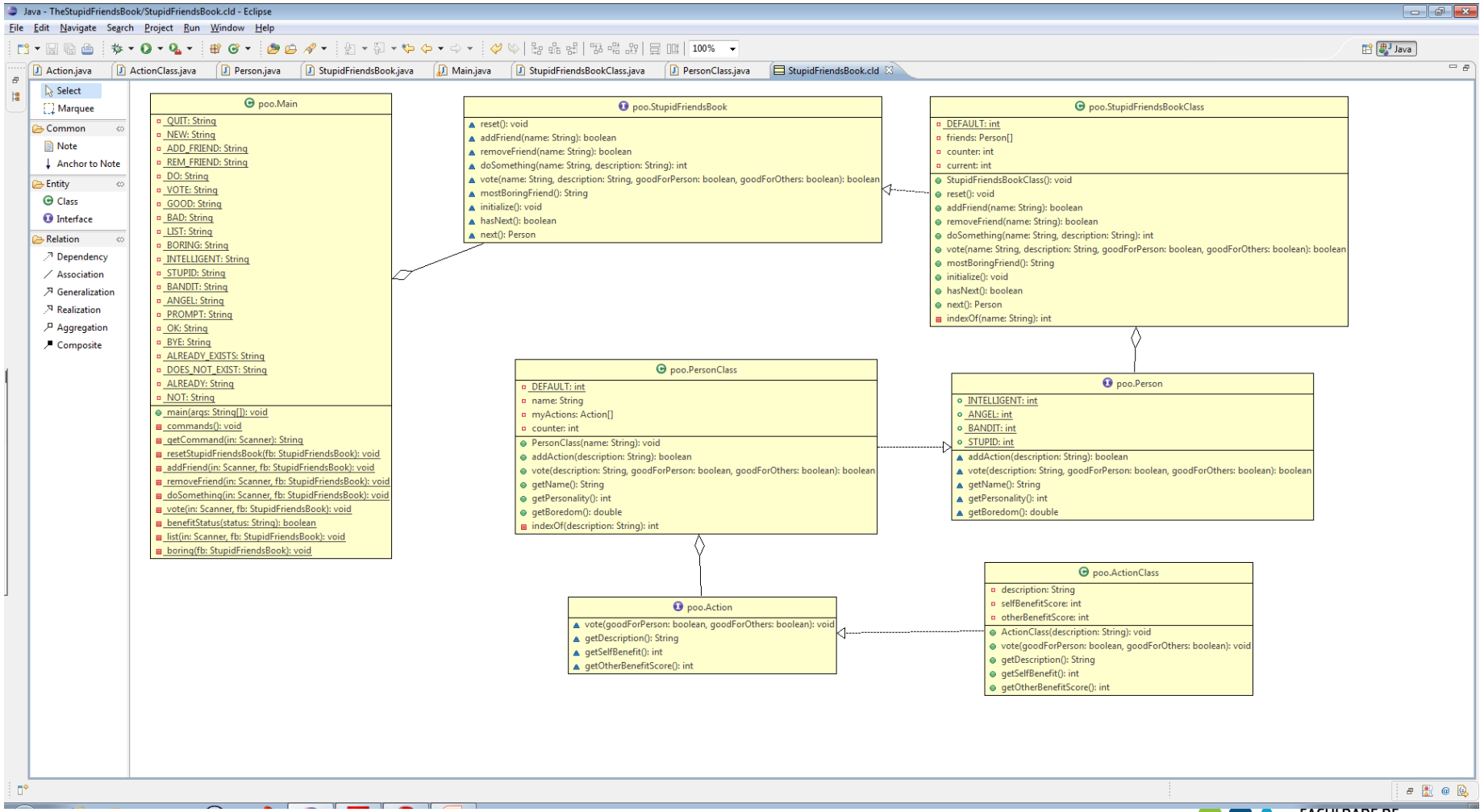
Comentários standard (estilo C)

- Use este estilo de comentário **apenas para esconder temporariamente do compilador código**
  - Por exemplo, esta técnica é usada como passo intermédio, quando estamos a remover código e queremos ter a certeza de que é seguro fazer essa remoção **antes** de o apagar definitivamente

Comentário de uma linha

- Use este tipo de comentário para explicar detalhes de implementação
  - Use um ou mais comentários destes para documentar
    - O objectivo de variáveis ou expressões
    - Decisões ao nível da implementação
    - Informação adicional sobre algoritmos complicados
    - "Remendos" aplicados a defeitos
    - Código a necessitar de melhorias/optimização
    - Problemas conhecidos, limitações, ou deficiências
- Descreva **porque é que o código faz algo**, não o que o código faz
- Use comentários de fim de linha para documentar variáveis locais e chavetas de fecho ("**}**"), em estruturas de controle aninhadas complicadas

Diagrama de classes e interfaces



Entregáveis

- Código fonte do projecto
  - Package **poo**
    - Main.java
    - StupidFriendsBook.java
    - StupidFriendsBookClass.java
    - Person.java
    - PersonClass.java
    - Action.java
    - ActionClass.java
- Documentação em formato Javadoc
- Outra documentação (e.g. Diagramas de classes)