Análise e Teste de Software

Testes Unitários e Cobertura - Parte 2

Universidade do Minho

2023/2024

1 Exercícios

Considere o projeto disponibilizado juntamente com esta ficha. Utilizando o IDE IntelliJ Idea:

1. Este projeto tem a classe `Aluno` em falta. Implemente-a e verifique a sua correção executando os testes unitários disponibilizados para essa classe.
2. Execute os testes unitários da classe `Turma`. Irá verificar que alguns testes falham. Neste caso assumimos que os testes estão corretamente implementados e o problema é com o código do projeto. Localize os bugs no código e corrija-os de forma a que todos os testes passem.
3. Analize a cobertura dos testes unitários. Desenvolva mais testes unitários por forma a maximizar a cobertura da test suite.

2 Maven

Maven é uma ferramenta para gestão de projetos de software em Java. Configurando um ficheiro (neste caso de nome `pom.xml`) para descrever todo o projeto, pode-se usar Maven para, entre outros, compilar, executar, testar e gerar documentação para um projeto.

A sua instalação deverá ser trivial na maioria dos sistemas. A titulo de exemplo:

```
sudo apt update
sudo apt install maven
```

Poderá ser necessário definir a environment variable `$JAVA_HOME` para apontar para a instalação do Java disponível na máquina. Pode-se verificar se o Maven está instalado tentando consultar a sua versão:

```
mvn -version
```

Alguns comandos úteis a usar com o Maven:

- `mvn -h` - mostra as opções disponíveis para a utilização do Maven.
- `mvn validate` - valida se a configuração do projeto está bem feita.
- `mvn compile` - compila o código fonte do projeto.
- `mvn test` - corre os testes unitários do projeto.
- `mvn package` - gera um ficheiro executável (p.e. um `.jar`) a partir do código fonte do projeto.

3 Exercícios

Considere o projeto utilizado no exercício anterior.

1. Configure este projeto para poder ser tratado com o Maven. Para isso deverá colocar o código fonte em `$PROJECT_HOME/src/main/java/`, os testes unitários em `$PROJECT_HOME/src/test/java/`, e criar um ficheiro `pom.xml` na raíz do projeto (sugestão: procurar uma template). Confirme que consegue compilar o projeto com `mvn compile`
2. Adicione a dependência do JUnit ao ficheiro `pom.xml`. Execute os testes unitários com `mvn test`.
3. Adicione a dependência do JaCoCo ao ficheiro `pom.xml`. Execute os testes unitários e produza um relatório HTML com os detalhes de cobertura.