



FGA 0238 - Testes de Software – Turma: 01 Semestre: 2022.1

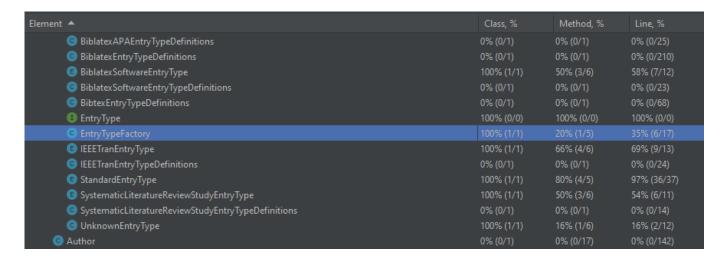
Nome: Ana Carolina Rodrigues Leite Matrícula: 190101792

Equipe: 07/Debugadores

### Atividade 3 – Desenvolver Testes de Unidade

### 1 Cobertura de testes

<Apresentar a cobertura de testes do projeto>





## 2 Código a ser testado

https://github.com/AnaCarolinaRodriguesLeite/jabref

### 3 Classe de Teste

<Apresentar o código dos métodos de testes existentes referente ao método apresentado na seção 2. Pode ser uma captura de tela>

```
package org.jabref.model.entry;

import org.jabref.model.entry.types.EntryType;
import org.jabref.model.entry.types.EntryTypeFactory;
import org.jabref.model.entry.types.IEEETranEntryType;

import org.junit.jupiter.api.Test;

import static org.junit.jupiter.api.Assertions.assertEquals;

* Christoph
public class EntryTypeFactoryTest {

    * Christoph
    @Test
    public void testParseEntryTypePatent() {
        EntryType patent = IEEETranEntryType.Patent;
        assertEquals(patent, EntryTypeFactory.parse( typeName: "patent"));
    }
}
```

https://github.com/AnaCarolinaRodriguesLeite/jabref/blob/main/src/test/java/org/jabref/model/entry/EntryTypeFactoryTest.java

### 4 Tabela de decisões/condições

Tabela 1: Decisões e Condições

| ID  | Decisão<br>(linha) | Condição      | Situação para True | Situação para False |
|-----|--------------------|---------------|--------------------|---------------------|
| CD1 | 25.1               | type1 == null | type1 == null      | type1 != null       |
| CD2 | 25.2               | type2 == null | type2 == null      | type2 != null       |
| CD3 | 27.1               | type1 == null | type1 == null      | type1 != null       |
| CD4 | 27.2               | type2 == null | type2 == null      | type2 != null       |

# 5 Especificação dos Casos de Testes

Tabela 2: Casos de Testes

| ID  | Decisões/Condições                   | Entrada   | Saída Esperada |
|-----|--------------------------------------|-----------|----------------|
| CT1 | (type1 == null)&&<br>(type2 == null) | null,null | True           |
| CT2 | (type1 != null)&&<br>(type2 != null) | a, a      | False          |
| СТ3 | (type1 == null)  <br>(type2 == null) | null,null | False          |
| CT4 | (type1 == null)  <br>(type2 != null) | null, a   | False          |
| CT5 | (type1 != null)  <br>(type2 == null) | a, null   | False          |
| СТ6 | (type1 != null)  <br>(type2 != null  | a,a       | True           |



## 6 Implementação dos Casos de Teste

```
package org.jabref.model.entry;
import java.util.Collections;
import java.util.List;
import java.util.Objects;
import org.jabref.model.entry.field.BibField;
import org.jabref.model.entry.field.FieldPriority;
import org.jabref.model.entry.field.StandardField;
import org.jabref.model.entry.types.EntryType;
import org.jabref.model.entry.types.EntryTypeFactory;
import org.jabref.model.entry.types.IEEETranEntryType;
import org.jabref.model.entry.types.UnknownEntryType;
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.Test;
import static org.junit.jupiter.api.Assertions.assertEquals;
public class MyEntryTypeFactoryTest {
    private static final EntryType CUSTOM_TYPE = new UnknownEntryType( name: "customType");
```

```
new*
@Test
public void twoNull() {
    boolean Result = entryType.isEqualNameAndFieldBased(null, null);
    assertEquals( expected: true, Result);
}

new*
@Test
public void oneNull() {
    boolean Result = entryType.isEqualNameAndFieldBased(null, entry);
    assertEquals( expected: false, Result);
}
```

```
new *
@Test
public void testParseEntryTypePatent() {
    EntryType patent = IEEETranEntryType.Patent;
    assertEquals(patent, EntryTypeFactory.parse( typeName: "patent"));
}

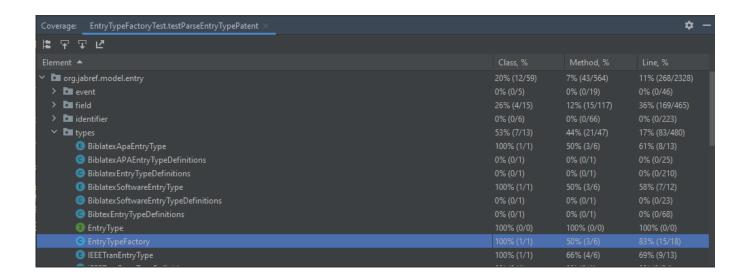
new *
@Test
public void anInvalidNull() {
    boolean Result = entryType.isEqualNameAndFieldBased(entry, null);
    assertEquals( expected: false, Result);
}
```

```
new*
@Test
public void oneNullTrue() {
    boolean Result = entryType.isEqualNameAndFieldBased(entry, entry);
    assertEquals( expected: Objects.equals(entry.getType(), entry.getType())
        && Objects.equals(entry.getRequiredFields(), entry.getRequiredFields())
        && Objects.equals(entry.getOptionalFields(), entry.getOptionalFields())
        && Objects.equals(entry.getSecondaryOptionalFields(), entry.getSecondaryOptionalFields()), Result);
}
```

https://github.com/AnaCarolinaRodriguesLeite/jabref/blob/main/src/test/java/org/jabref/model/entry/MyEntryTypeFactoryTest.java



### 7 Análises e Resultados



Foram cobertos 3 métodos ao todo e a minha implementação não trouxe novos erros, mas não cobriu todos eles.

## 8 Pull Request

<Caso faça um pull request, apresenta aqui uma captura de tela do pull request e o link para o mesmo>

#### 9 Links

Fork do projeto: https://github.com/AnaCarolinaRodriguesLeite/jabref





Implementação no projeto:

https://github.com/AnaCarolinaRodriguesLeite/jabref/blob/main/src/test/java/org/jabref/model/entry/MyEntryTypeFactoryTest.java

Commit da implementação:

https://github.com/Ana Carolina Rodrigues Leite/jabref/commit/9779d11537054360c6c4359d388d0c22249d9899

Pull Request: <link>