



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>

Element	Class, %	Method, %	Line, %
BiblatexAPAEntryTypeDefinitions	0% (0/1)	0% (0/1)	0% (0/25)
BiblatexEntryTypeDefinitions	0% (0/1)	0% (0/1)	0% (0/210)
BiblatexSoftwareEntryType	100% (1/1)	50% (3/6)	58% (7/12)
BiblatexSoftwareEntryTypeDefinitions	0% (0/1)	0% (0/1)	0% (0/23)
BibtexEntryTypeDefinitions	0% (0/1)	0% (0/1)	0% (0/68)
EntryType	100% (0/0)	100% (0/0)	100% (0/0)
EntryTypeFactory	100% (1/1)	20% (1/5)	35% (6/17)
IEEETranEntryType	100% (1/1)	66% (4/6)	69% (9/13)
IEEETranEntryTypeDefinitions	0% (0/1)	0% (0/1)	0% (0/24)
StandardEntryType	100% (1/1)	80% (4/5)	97% (36/37)
SystematicLiteratureReviewStudyEntryType	100% (1/1)	50% (3/6)	54% (6/11)
SystematicLiteratureReviewStudyEntryTypeDefinitions	0% (0/1)	0% (0/1)	0% (0/14)
UnknownEntryType	100% (1/1)	16% (1/6)	16% (2/12)
Author	0% (0/1)	0% (0/17)	0% (0/142)

```
Cover: EntryTypeFactoryTest x
[Icons] [Check] [Close] [Zoom In] [Zoom Out] [Full Screen] [Run] [Test] [Debug] [Exit]
Tests passed: 1 of 1 test – 41 ms
EntryTypeFactory 141 ms
testParseEntry 141 ms
C:\Users\Lenovo\.jdk\openjdk-18.0.2\bin\java.exe -ea -javaagent:C:\Users\Lenovo\AppData\Local\J
---- IntelliJ IDEA coverage runner ----
sampling ...
include patterns:
org\.jabref\.model\.entry\.*
exclude patterns:
Class transformation time: 0.0242924s for 1395 classes or 1.741390681003584E-5s per class
Process finished with exit code 0
```

2 Código a ser testado

```
public static boolean isEqualNameAndFieldBased(BibEntryType type1, BibEntryType type2) {  
    if ((type1 == null) && (type2 == null)) {  
        return true;  
    } else if ((type1 == null) || (type2 == null)) {  
        return false;  
    } else {  
        return Objects.equals(type1.getType(), type2.getType())  
            && Objects.equals(type1.getRequiredFields(), type2.getRequiredFields())  
            && Objects.equals(type1.getOptionalFields(), type2.getOptionalFields())  
            && Objects.equals(type1.getSecondaryOptionalFields(), type2.getSecondaryOptionalFields());  
    }  
}
```

<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 (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	<code>(type1 == null)&& (type2 == null)</code>	null,null	True
CT2	<code>(type1 != null)&& (type2 != null)</code>	a, a	False
CT3	<code>(type1 == null) (type2 == null)</code>	null,null	False
CT4	<code>(type1 == null) (type2 != null)</code>	null, a	False
CT5	<code>(type1 != null) (type2 == null)</code>	a, null	False
CT6	<code>(type1 != null) (type2 != null)</code>	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;

new *
public class MyEntryTypeFactoryTest {
    1 usage
    private static final EntryType CUSTOM_TYPE = new UnknownEntryType( name: "customType");
```

```
public class MyEntryTypeFactoryTest {  
    1 usage  
    private static final EntryType CUSTOM_TYPE = new UnknownEntryType( name: "customType");  
  
    5 usages  
    private EntryTypeFactory entryType;  
    13 usages  
    private BibEntryType entry;  
  
    new *  
    @BeforeEach  
    public void setup() {  
        entryType = new EntryTypeFactory();  
        entry = new BibEntryType(CUSTOM_TYPE, List.of(new BibField(StandardField.AUTHOR, FieldPriority.IMPORTANT)),  
            Collections.emptySet());  
    }  
  
    new *
```

```
    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

```
Cover: EntryTypeFactoryTest x
Tests passed: 4 of 4 tests - 57 ms
C:\Users\Lenovo\.jdk\openjdk-18.0.2\bin\java.exe ...
---- IntelliJ IDEA coverage runner ----
sampling ...
include patterns:
org.jabref.model.entry.*
exclude patterns:
Class transformation time: 0.0346212s for 1402 classes or 2.4694151212553493E-5s per class
Process finished with exit code 0
```

Element	Class, %	Method, %	Line, %
org.jabref.model.entry	20% (12/59)	7% (43/564)	11% (268/2328)
event	0% (0/5)	0% (0/19)	0% (0/46)
field	26% (4/15)	12% (15/117)	36% (169/465)
identifier	0% (0/6)	0% (0/66)	0% (0/223)
types	53% (7/13)	44% (21/47)	17% (83/480)
BiblatexApaEntryType	100% (1/1)	50% (3/6)	61% (8/13)
BiblatexAPAEntryTypeDefinitions	0% (0/1)	0% (0/1)	0% (0/25)
BiblatexEntryTypeDefinitions	0% (0/1)	0% (0/1)	0% (0/210)
BiblatexSoftwareEntryType	100% (1/1)	50% (3/6)	58% (7/12)
BiblatexSoftwareEntryTypeDefinitions	0% (0/1)	0% (0/1)	0% (0/23)
BibtexEntryTypeDefinitions	0% (0/1)	0% (0/1)	0% (0/68)
EntryType	100% (0/0)	100% (0/0)	100% (0/0)
EntryTypeFactory	100% (1/1)	50% (3/6)	83% (15/18)
IEEETranEntryType	100% (1/1)	66% (4/6)	69% (9/13)

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/AnaCarolinaRodriguesLeite/jabref/commit/9779d11537054360c6c4359d388d0c2249d9899>

Pull Request: <link>