**Università Degli Studi Di Salerno**

**Dipartimento di Informatica**

**Corso di Laurea Magistrale in Softaware Engeneering and IT Mangament**

****

INGEGNERIA, GESTIONE ED EVOLUZIONE DEL

SOFTWARE

ANNO ACCADEMICO 2021/22

**Evoluzione cASpER**

**CR\_01-CR\_02 Test Execution**

**Unit Test Execution**

|  |  |
| --- | --- |
| Fail | Il test non ha trovato nessun fault |
| Pass | Il test ha trovato un fault |

|  |  |
| --- | --- |
| **Test Case ID** | testParserExtractClassImplementate1 |
| **Tester** | Angelo Afeltra |
| **Evironment** | Intellij IDEA |
| **Precondizione** | Lo sviluppatore Lancia il plugin su un progetto fittizio |
| **Input** | Project= testParserExtractClassImplementate1 |
| **Flow** | 1. Viene lanciato il plugin su un progetto fittizio contenente una classe che non implementa nessuna interfaccia 2. Si estraggono le classi dal progetto 3. Si convertono gli oggetti PsiClass, PsiMethod, PsiPackage in ClassBean, MethodBean, PackageBean |
| **Oracolo** | Il numero di interfacce implementate dalla classe è uguale a 0 |
| **Result** | Fail |

|  |  |
| --- | --- |
| **Test Case ID** | testParserExtractClassImplementate2 |
| **Tester** | Angelo Afeltra |
| **Evironment** | Intellij IDEA |
| **Precondizione** | Lo sviluppatore Lancia il plugin su un progetto fittizio |
| **Input** | Project= testParserExtractClassImplementate2 |
| **Flow** | 1. Viene lanciato il plugin su un progetto fittizio contenente una classe che implementa una interfaccia 2. Si estraggono le classi dal progetto 3. Si convertono gli oggetti PsiClass, PsiMethod, PsiPackage in ClassBean, MethodBean, PackageBean |
| **Oracolo** | Il numero di interfacce implementate dalla classe è uguale a 1 |
| **Result** | Fail |

|  |  |
| --- | --- |
| **Test Case ID** | testParserExtractClassImplementate3 |
| **Tester** | Angelo Afeltra |
| **Evironment** | Intellij IDEA |
| **Precondizione** | Lo sviluppatore Lancia il plugin su un progetto fittizio |
| **Input** | Project= testParserExtractClassImplementate3 |
| **Flow** | 1. Viene lanciato il plugin su un progetto fittizio contenente una classe che implementa più interfacce 2. Si estraggono le classi dal progetto 3. Si convertono gli oggetti PsiClass, PsiMethod, PsiPackage in ClassBean, MethodBean, PackageBean |
| **Oracolo** | Il numero di interfacce implementate dalla classe è maggiore di 1 |
| **Result** | Fail |

|  |  |
| --- | --- |
| **Test Case ID** | testParserExtractClasseEstesa1 |
| **Tester** | Angelo Afeltra |
| **Evironment** | Intellij IDEA |
| **Precondizione** | Lo sviluppatore Lancia il plugin su un progetto fittizio |
| **Input** | Project= testParserExtractClasseEstesa1 |
| **Flow** | 1. Viene lanciato il plugin su un progetto fittizio contenente una classe che non usa l’ereditarietà 2. Si estraggono le classi dal progetto 3. Si convertono gli oggetti PsiClass, PsiMethod, PsiPackage in ClassBean, MethodBean, PackageBean |
| **Oracolo** | Il numero di classi estese dalla classe è uguale a 0 |
| **Result** | Fail |

|  |  |
| --- | --- |
| **Test Case ID** | testParserExtractClasseEstesa2 |
| **Tester** | Angelo Afeltra |
| **Evironment** | Intellij IDEA |
| **Precondizione** | Lo sviluppatore Lancia il plugin su un progetto fittizio |
| **Input** | Project= testParserExtractClasseEstesa2 |
| **Flow** | 1. Viene lanciato il plugin su un progetto fittizio contenente una classe che sfrutta l’ereditarietà 2. Si estraggono le classi dal progetto 3. Si convertono gli oggetti PsiClass, PsiMethod, PsiPackage in ClassBean, MethodBean, PackageBean |
| **Oracolo** | Il numero di classi estese dalla classe è uguale a 1 |
| **Result** | Fail |

|  |  |
| --- | --- |
| **Test Case ID** | StructuralSpaghettiCodeStrategyTest.testNotIsSmelly |
| **Tester** | Angelo Afeltra, Antonio Trovato |
| **Evironment** | Intellij IDEA |
| **Precondizione** | ClassBean creato utilizzando la libreria LightJavaCodeInsightFixtureTestCase |
| **Input** | ClasseJava= classNotIsSmelly |
| **Flow** | 1. Si crea un oggetto ClassBean non affetto da spaghetti code 2. Si passa l’oggetto al detector |
| **Oracolo** | “False” la classe non è affetta dallo smell spaghetti code |
| **Result** | Fail |

|  |  |
| --- | --- |
| **Test Case ID** | StructuralSpaghettiCodeStrategyTest.testIsSmelly |
| **Tester** | Angelo Afeltra, Antonio Trovato |
| **Evironment** | Intellij IDEA |
| **Precondizione** | ClassBean creato utilizzando la libreria LightJavaCodeInsightFixtureTestCase |
| **Input** | ClasseJava= classIsSmelly |
| **Flow** | 1. Si crea un oggetto ClassBean affetto da spaghetti code 2. Si passa l’oggetto al detector |
| **Oracolo** | “True” la classe è affetta dallo smell spaghetti code |
| **Result** | Fail |
| **Test Case ID** | StructuralSwissArmyKnifeStrategyTest .testNotIsSmelly |
| **Tester** | Angelo Afeltra, Walter D’Ambrosio |
| **Evironment** | Intellij IDEA |
| **Precondizione** | ClassBean creato utilizzando la libreria LightJavaCodeInsightFixtureTestCase |
| **Input** | ClasseJava= classNotIsSmelly |
| **Flow** | 1. Si crea un oggetto ClassBean non affetto da swiss army knife 2. Si passa l’oggetto al detector |
| **Oracolo** | “False” la classe non è affetta dallo smell swiss army knife |
| **Result** | Fail |

|  |  |
| --- | --- |
| **Test Case ID** | StructuralSwissArmyKnifeStrategyTest .testIsSmelly |
| **Tester** | Angelo Afeltra, Walter D’Ambrosio |
| **Evironment** | Intellij IDEA |
| **Precondizione** | ClassBean creato utilizzando la libreria LightJavaCodeInsightFixtureTestCase |
| **Input** | ClasseJava= classIsSmelly |
| **Flow** | 1. Si crea un oggetto ClassBean affetto da swiss army knife 2. Si passa l’oggetto al detector |
| **Oracolo** | “True” la classe è affetta dallo smell swiss army knife |
| **Result** | Fail |

**Integration Test Execution**

|  |  |
| --- | --- |
| Fail | Il test non ha trovato nessun fault |
| Pass | Il test ha trovato un fault |

|  |  |
| --- | --- |
| **Test Case ID** | testIntegrationSpaghettiCodeIsSmelly |
| **Tester** | Angelo Afeltra, Antonio Trovato |
| **Evironment** | Intellij IDEA |
| **Precondizione** | Lo sviluppatore Lancia il plugin su un progetto fittizio |
| **Input** | Progetto= testIntegrationSpaghettiCodeIsSmelly  ClasseJava= classIsSmelly |
| **Flow** | 1. Si crea un progetto fittizio con una classe affetta da spaghetti code 2. Il parse estrare le classi del progetto 3. Il parse avvia l’analisi delle classi estratte |
| **Oracolo** | “True” la classe è affetta dallo smell spaghetti code |
| **Result** | Fail |

|  |  |
| --- | --- |
| **Test Case ID** | testIntegrationSpaghettiCodeNotIsSmelly |
| **Tester** | Angelo Afeltra, Antonio Trovato |
| **Evironment** | Intellij IDEA |
| **Precondizione** | Lo sviluppatore Lancia il plugin su un progetto fittizio |
| **Input** | Progetto= testIntegrationSpaghettiCodeIsSmelly  ClasseJava= classNotIsSmelly |
| **Flow** | 1. Si crea un progetto fittizio con una classe non affetta da spaghetti code 2. Il parse estrare le classi del progetto 3. Il parse avvia l’analisi delle classi estratte |
| **Oracolo** | “False” la classe non è affetta dallo smell spaghetti code |
| **Result** | Fail |

|  |  |
| --- | --- |
| **Test Case ID** | testIntegrationSwissArmyKnifeIsSmelly |
| **Tester** | Angelo Afeltra, Walter D’ambrosio |
| **Evironment** | Intellij IDEA |
| **Precondizione** | Lo sviluppatore Lancia il plugin su un progetto fittizio |
| **Input** | Progetto= testIntegrationSwissArmyKnifeIsSmelly  ClasseJava= classNotIsSmelly |
| **Flow** | 1. Si crea un progetto fittizio con una classe affetta da swiss army knife 2. Il parse estrare le classi del progetto 3. Il parse avvia l’analisi delle classi estratte |
| **Oracolo** | “True” la classe è affetta dallo smell swiss army knife |
| **Result** | Fail |

|  |  |
| --- | --- |
| **Test Case ID** | testIntegrationSwissArmyKnifeNotIsSmelly |
| **Tester** | Angelo Afeltra, Walter D’ambrosio |
| **Evironment** | Intellij IDEA |
| **Precondizione** | Lo sviluppatore Lancia il plugin su un progetto fittizio |
| **Input** | Progetto= testIntegrationSwissArmyKnifeIsSmelly  ClasseJava= classNotIsSmelly |
| **Flow** | 1. Si crea un progetto fittizio con una classe non affetta da swiss army knife 2. Il parse estrare le classi del progetto 3. Il parse avvia l’analisi delle classi estratte |
| **Oracolo** | “False” la classe non è affetta dallo smell swiss army knife |
| **Result** | Fail |

**System Test Execution**

|  |  |
| --- | --- |
| Fail | Il test non ha trovato nessun fault |
| Pass | Il test ha trovato un fault |

|  |  |
| --- | --- |
| **Test Case ID** | SystemTestSPC\_1 |
| **Tester** | Angelo Afeltra, Antonio Trovato |
| **Evironment** | Intellij IDEA |
| **Precondizione** | Lo sviluppatore Lancia il plugin su un progetto reale |
| **Input** | Progetto = TestCasper |
| **Flow** | 1. Si avvia il plugin su un progetto reale |
| **Oracolo** | Nella lista degli smell presenti nel progetto analizzato è presente lo smell Spaghetti Code |
| **Result** | Fail |

|  |  |
| --- | --- |
| **Test Case ID** | SystemTestSPC\_2 |
| **Tester** | Angelo Afeltra, Antonio Trovato |
| **Evironment** | Intellij IDEA |
| **Precondizione** | Lo sviluppatore Lancia il plugin su un progetto reale |
| **Input** | Progetto = AppDesktopMedQueue |
| **Flow** | 1. Si avvia il plugin su un progetto reale |
| **Oracolo** | Nella lista degli smell presenti nel progetto analizzato non è presente lo smell Spaghetti Code |
| **Result** | Fail |

|  |  |
| --- | --- |
| **Test Case ID** | SystemTest\_SWISS\_1 |
| **Tester** | Angelo Afeltra, Walter D’Ambrosio |
| **Evironment** | Intellij IDEA |
| **Precondizione** | Lo sviluppatore Lancia il plugin su un progetto reale |
| **Input** | Progetto = TestCasper |
| **Flow** | 1. Si avvia il plugin su un progetto reale |
| **Oracolo** | Nella lista degli smell presenti nel progetto analizzato è presente lo smell Swiss Army Knife |
| **Result** | Fail |

|  |  |
| --- | --- |
| **Test Case ID** | SystemTest\_SWISS\_2 |
| **Tester** | Angelo Afeltra, Walter D’Ambrosio |
| **Evironment** | Intellij IDEA |
| **Precondizione** | Lo sviluppatore Lancia il plugin su un progetto reale |
| **Input** | Progetto = AppDesktopMedQueue |
| **Flow** | 1. Si avvia il plugin su un progetto reale |
| **Oracolo** | Nella lista degli smell presenti nel progetto analizzato non è presente lo smell Swiss Army Knife |
| **Result** | Fail |

**Regression Test Execution**

|  |  |
| --- | --- |
| Fail | Il test non ha trovato nessun fault |
| Pass | Il test ha trovato un fault |

|  |  |
| --- | --- |
| **Test Case ID** | System\_Test\_FE\_1 |
| **Tester** | Angelo Afeltra, Antonio Trovato |
| **Evironment** | Intellij IDEA |
| **Precondizione** | Lo sviluppatore Lancia il plugin su un progetto reale |
| **Input** | Progetto = AppDesktopMedQueue |
| **Flow** | 1. Si avvia il plugin su un progetto reale 2. Si selezione una classe affetta da feature envy dalla lista degli smell presenti 3. Si esegue il refactoring |
| **Oracolo** | Nella lista degli smell presenti nel progetto analizzato è presente lo smell feature envy ed il refactoring viene eseguito correttamente |
| **Result** | Fail |

|  |  |
| --- | --- |
| **Test Case ID** | System\_Test\_FE\_2 |
| **Tester** | Angelo Afeltra, Antonio Trovato |
| **Evironment** | Intellij IDEA |
| **Precondizione** | Lo sviluppatore Lancia il plugin su un progetto reale |
| **Input** | Progetto = TestCasper |
| **Flow** | 1. Si avvia il plugin su un progetto reale 2. Si selezione una classe affetta da feature envy dalla lista degli smell presenti 3. Si esegue il refactoring |
| **Oracolo** | Nella lista degli smell presenti nel progetto analizzato non è presente lo smell Feature Envy |
| **Result** | Fail |

|  |  |
| --- | --- |
| **Test Case ID** | System\_Test\_MC\_1 |
| **Tester** | Angelo Afeltra, Antonio Trovato |
| **Evironment** | Intellij IDEA |
| **Precondizione** | Lo sviluppatore Lancia il plugin su un progetto reale |
| **Input** | Progetto = AppDesktopMedQueue |
| **Flow** | 1. Si avvia il plugin su un progetto reale 2. Si selezione una classe affetta da misplaced class dalla lista degli smell presenti 3. Si esegue il refactoring |
| **Oracolo** | Nella lista degli smell presenti nel progetto analizzato è presente lo smell misplaced class ed il refactoring avviene correttamente |
| **Result** | Fail |

|  |  |
| --- | --- |
| **Test Case ID** | System\_Test\_MC\_2 |
| **Tester** | Angelo Afeltra, Antonio Trovato |
| **Evironment** | Intellij IDEA |
| **Precondizione** | Lo sviluppatore Lancia il plugin su un progetto reale |
| **Input** | Progetto = TestCasper |
| **Flow** | 1. Si avvia il plugin su un progetto reale |
| **Oracolo** | Nella lista degli smell presenti nel progetto analizzato non è presente lo smell misplaced class |
| **Result** | Fail |

|  |  |
| --- | --- |
| **Test Case ID** | System\_Test\_BLOB\_1 |
| **Tester** | Angelo Afeltra, Antonio Trovato |
| **Evironment** | Intellij IDEA |
| **Precondizione** | Lo sviluppatore Lancia il plugin su un progetto reale |
| **Input** | Progetto = AppDesktopMedQueue |
| **Flow** | 1. Si avvia il plugin su un progetto reale 2. Si selezione una classe affetta da blob dalla lista degli smell presenti 3. Si esegue il refactoring |
| **Oracolo** | Nella lista degli smell presenti nel progetto analizzato è presente lo smell blob ed il refactoring avviene correttamente |
| **Result** | Pass |

|  |  |
| --- | --- |
| **Test Case ID** | System\_Test\_BLOB\_2 |
| **Tester** | Angelo Afeltra, Antonio Trovato |
| **Evironment** | Intellij IDEA |
| **Precondizione** | Lo sviluppatore Lancia il plugin su un progetto reale |
| **Input** | Progetto = TestCasper |
| **Flow** | 1. Si avvia il plugin su un progetto reale |
| **Oracolo** | Nella lista degli smell presenti nel progetto analizzato non è presente lo smell blob |
| **Result** | Fail |

|  |  |
| --- | --- |
| **Test Case ID** | System\_Test\_PP\_1 |
| **Tester** | Angelo Afeltra, Antonio Trovato |
| **Evironment** | Intellij IDEA |
| **Precondizione** | Lo sviluppatore Lancia il plugin su un progetto reale |
| **Input** | Progetto = AppDesktopMedQueue |
| **Flow** | 1. Si avvia il plugin su un progetto reale 2. Si selezione una package affetto da promiscuous package dalla lista degli smell presenti 3. Si esegue il refactoring |
| **Oracolo** | Nella lista degli smell presenti nel progetto analizzato è presente lo smell promiscuous package ed il refactoring avviene correttamente |
| **Result** | Pass |

|  |  |
| --- | --- |
| **Test Case ID** | System\_Test\_PP\_2 |
| **Tester** | Angelo Afeltra, Antonio Trovato |
| **Evironment** | Intellij IDEA |
| **Precondizione** | Lo sviluppatore Lancia il plugin su un progetto reale |
| **Input** | Progetto = TestCasper |
| **Flow** | 1. Si avvia il plugin su un progetto reale 2. Si selezione una classe affetta da feature envy dalla lista degli smell presenti 3. Si esegue il refactoring |
| **Oracolo** | Nella lista degli smell presenti nel progetto analizzato non è presente lo smell Feature Envy |
| **Result** | Fail |

|  |  |
| --- | --- |
| **Test Case ID** | System\_Test\_DC\_1 |
| **Tester** | Angelo Afeltra, Antonio Trovato |
| **Evironment** | Intellij IDEA |
| **Precondizione** | Lo sviluppatore Lancia il plugin su un progetto reale |
| **Input** | Progetto = AppDesktopMedQueue |
| **Flow** | 1. Si avvia il plugin su un progetto reale |
| **Oracolo** | Nella lista degli smell presenti nel progetto analizzato è presente lo smell divergent change ed il refactoring avviene correttamente |
| **Result** | Pass |

|  |  |
| --- | --- |
| **Test Case ID** | System\_Test\_SS\_1 |
| **Tester** | Angelo Afeltra, Antonio Trovato |
| **Evironment** | Intellij IDEA |
| **Precondizione** | Lo sviluppatore Lancia il plugin su un progetto reale |
| **Input** | Progetto = AppDesktopMedQueue |
| **Flow** | 1. Si avvia il plugin su un progetto reale |
| **Oracolo** | Nella lista degli smell presenti nel progetto analizzato è presente lo smell shotgun surgery ed il refactoring avviene correttamente |
| **Result** | Pass |

|  |  |
| --- | --- |
| **Test Case ID** | System\_Test\_PI\_1 |
| **Tester** | Angelo Afeltra, Antonio Trovato |
| **Evironment** | Intellij IDEA |
| **Precondizione** | Lo sviluppatore Lancia il plugin su un progetto reale |
| **Input** | Progetto = AppDesktopMedQueue |
| **Flow** | 1. Si avvia il plugin su un progetto reale |
| **Oracolo** | Nella lista degli smell presenti nel progetto analizzato è presente lo smell parallel inheritnace ed il refactoring avviene correttamente |
| **Result** | Pass |