

Trabalho 2 – Qualidade de Software

Resultados dos Testes Automatizados com JUnit

(Jogo de Cartas)

Gabriel Gomes Pereira

Para a realização do trabalho foram escritos um total de 55 métodos de teste, cada classe de teste corresponde aos testes dos métodos de um determinado objeto ou pacote. Por exemplo a classe *TestCard* busca testar os métodos públicos ligados ao objeto *Card*.

Na classe de teste *TestCard* foram escritos 14 métodos de teste verificando se valores de atributos eram inseridos e retornados corretamente. Em todos os testes foi obtido sucesso.

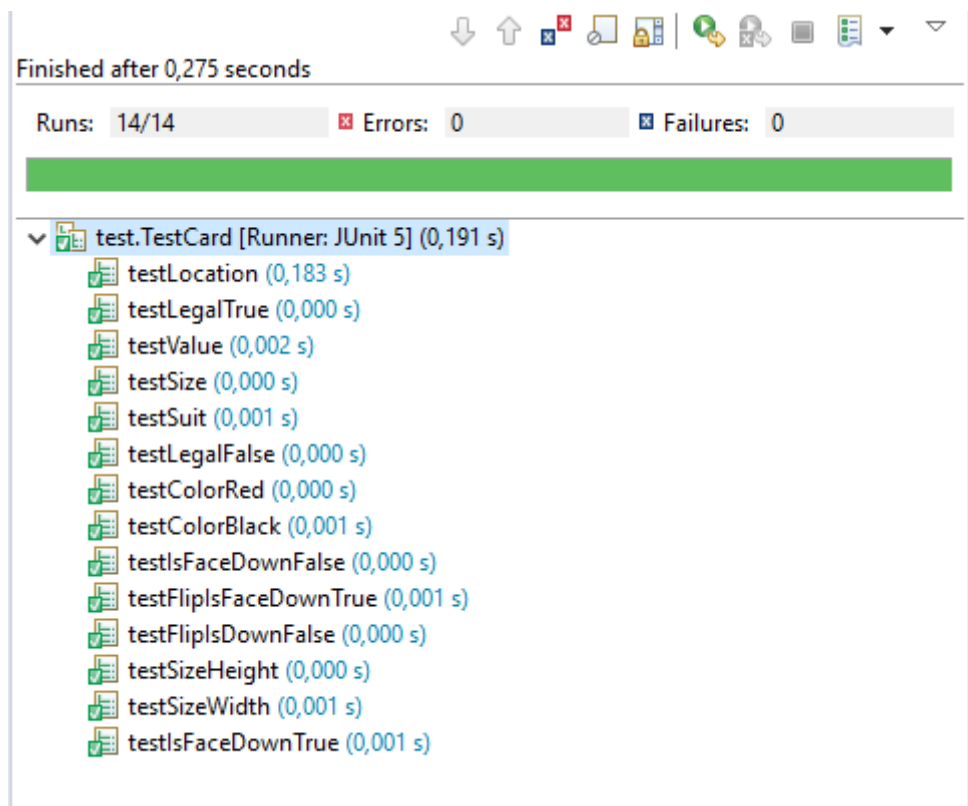


Figura 1 – Resultados TestCard

Também foi criada uma classe de teste chamada *TestStack*, para testar métodos e atributos associados ao objeto *Stack*, que representa uma pilha de cartas no jogo. Os testes visaram analisar se as cartas e demais atributos como *Location* e topo da pilha eram retornados corretamente. Todos os 17 métodos de teste retornaram sucesso.

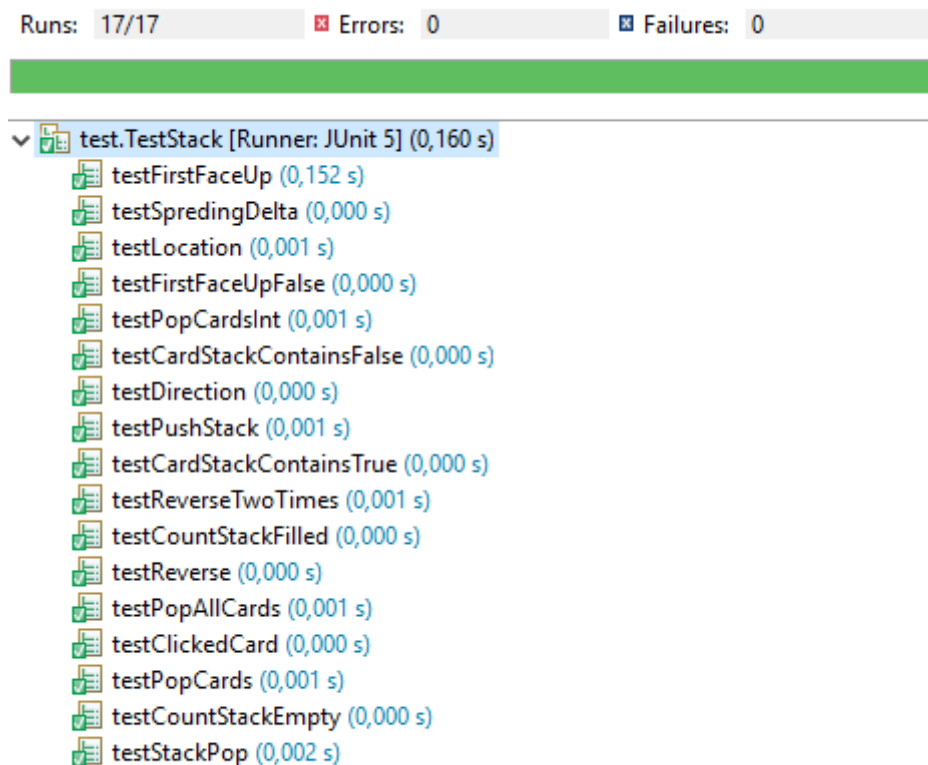


Figura 2- Resultados TestStack

A terceira classe criada foi *TestGameInfo*, cujo objetivo é testar métodos e atributos públicos relacionados a *GameInfo*. Todos os 6 testes apresentaram sucesso.

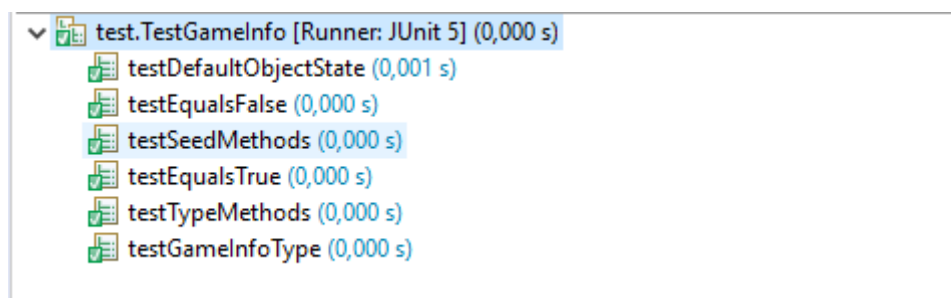


Figura 3 - Resultados TestGameInfo

A quarta classe criada, basicamente testou os métodos relacionados a *Solitaire* que lançavam exceções. O objetivo dos métodos de teste foi forçar os métodos da classe testada a gerarem as exceções descritas, e com isso analisar se exceção correta é gerada. Ao final se obteve sucesso em todos os testes realizados.

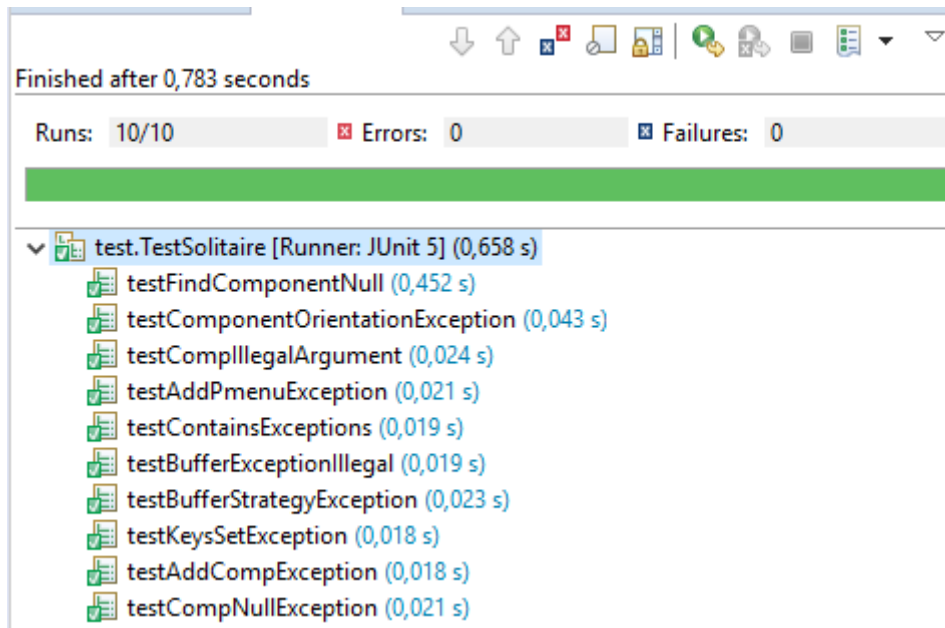


Figura 4 - Resultados TestSolitaire

Nas duas últimas classes de teste *TestSolitaireStack* e *TestSequentialStack* foram realizados testes verificando as operações relacionadas as pilhas de cartas do jogo. Foi testado se os métodos e estruturas condicionais internas aos mesmos são usadas corretamente. Como as classes testadas possuem restrição “protected” foi preciso inserir as classes de teste dentro do mesmo pacote das classes que iriam ser testadas.

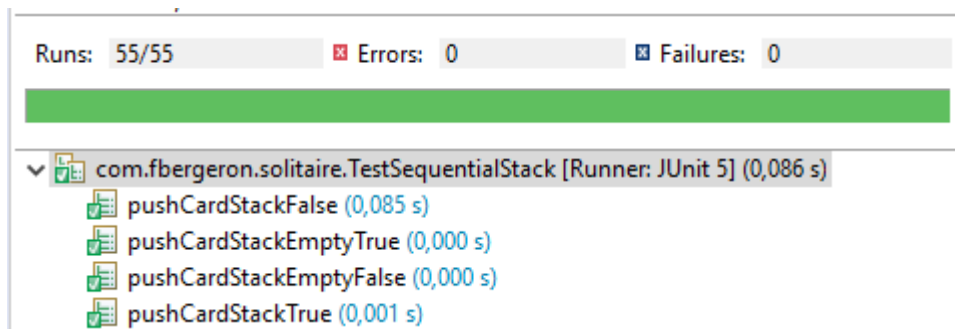


Figura 5- Resultados TestSequentialStack

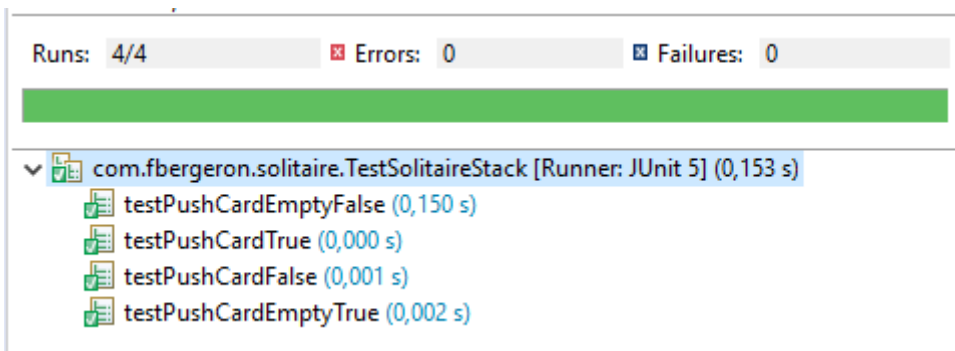


Figura 6- Resultados TestSolitaireStack

Além disso, vale lembrar que os métodos de teste também foram criados levando em consideração métodos e atributos públicos relacionados a estruturas condicionais, loops e sequências de valores. O projeto foi testado usando a IDE Eclipse.