TDD – DESENVOLVIMENTO ORIENTADO A TESTES.

# 1 - GERÊNCIA DE CONFIGURAÇÃO

Passo 1 – Instalar sdk Java 7.

O download da JDK pode ser feito neste link:

<http://www.oracle.com/technetwork/pt/java/javase/downloads/jdk7-downloads-1880260.html>

Passo 2 – Baixar o Eclipse para criarmos o projeto Java.

O download da IDE Eclipse pode ser feito no seguinte link:

<https://www.eclipse.org/downloads/download.php?file=/technology/epp/downloads/release/kepler/SR2/eclipse-standard-kepler-SR2-win32.zip>

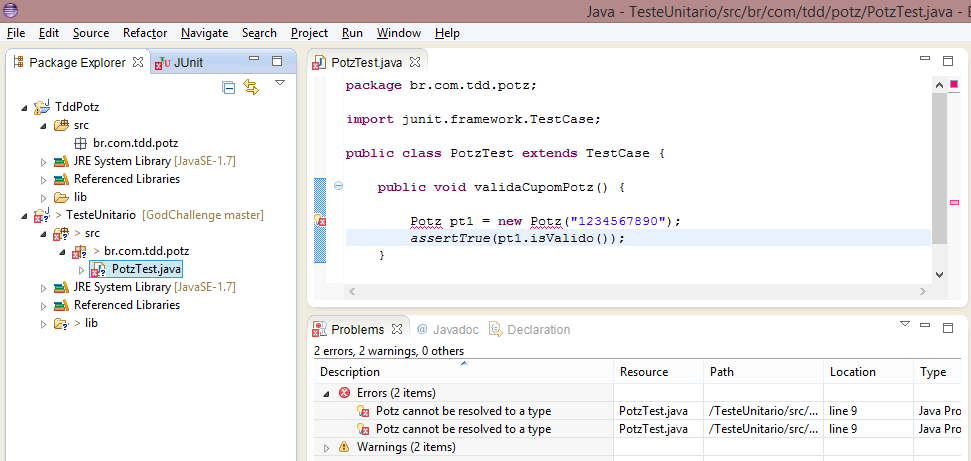
ou

<https://www.eclipse.org/downloads/download.php?file=/technology/epp/downloads/release/kepler/SR2/eclipse-standard-kepler-SR2-win32-x86_64.zip>

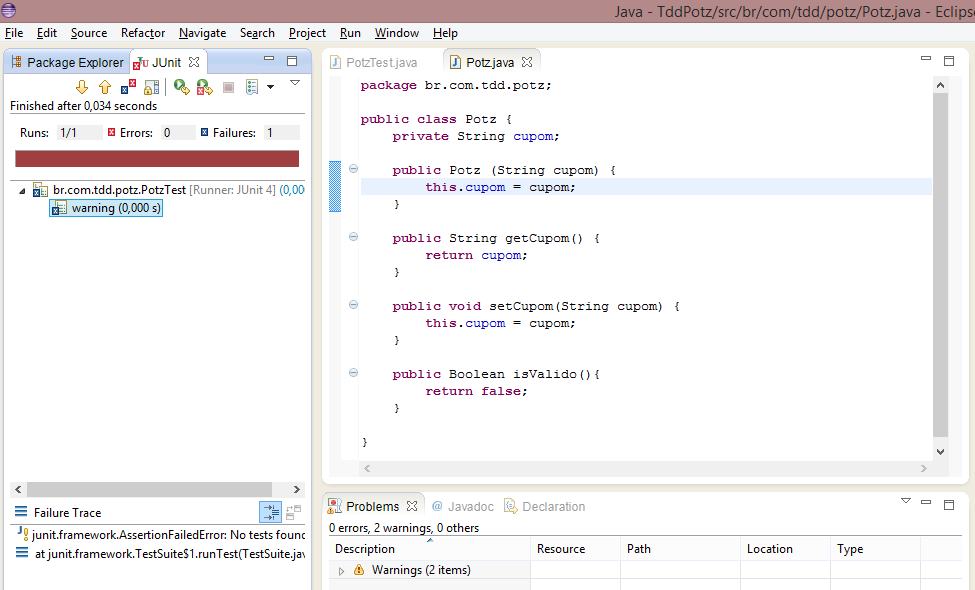
O eclipse não é necessário fazer a instalação, basta descompactar o arquivo zip em um diretório qualquer e executar o arquivo eclipse.exe.

# 2 – CRIAÇÃO DO PROJETO

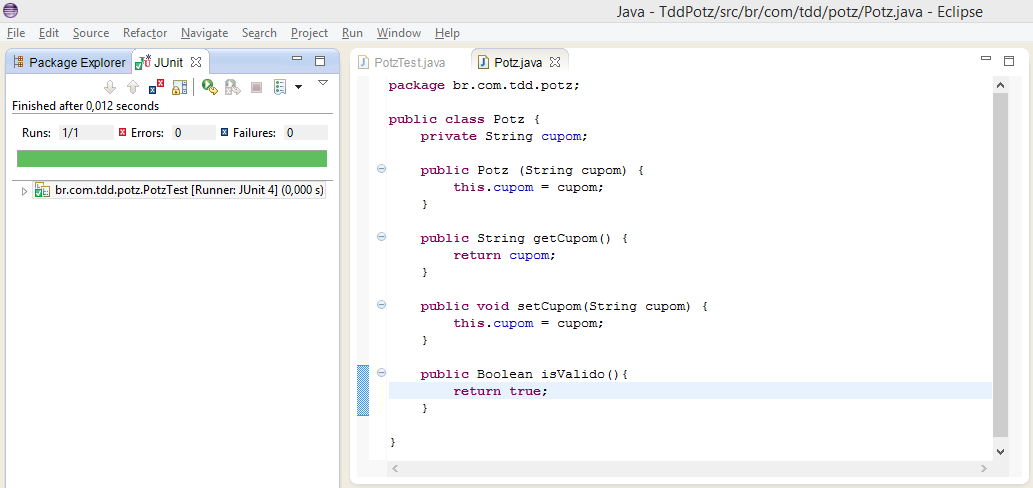
Passo 1 Tdd – Criação do primeiro teste – Erro Classe não existe



Passo 2 Tdd – implementada a Classe Potz e o projeto agora compila mas não passa no teste

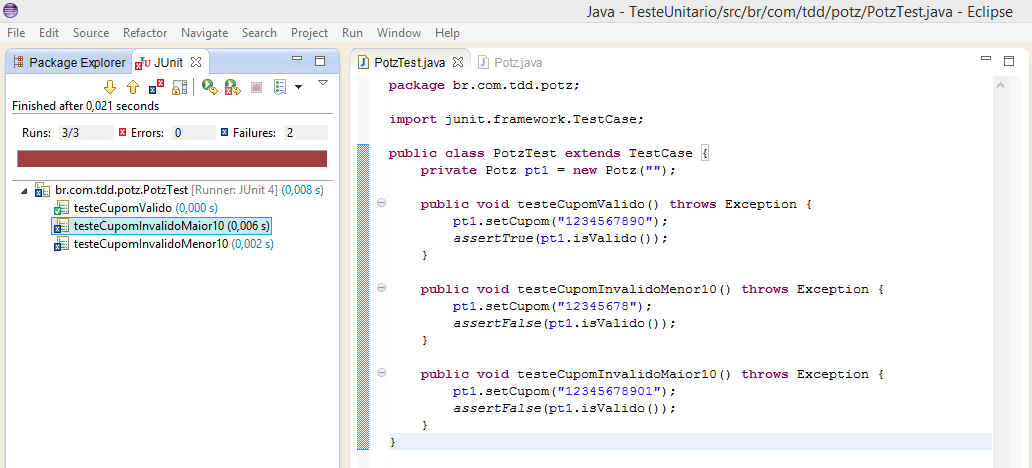


Passo 3 – Fazer o caso de teste passar no teste o mais rápido possível

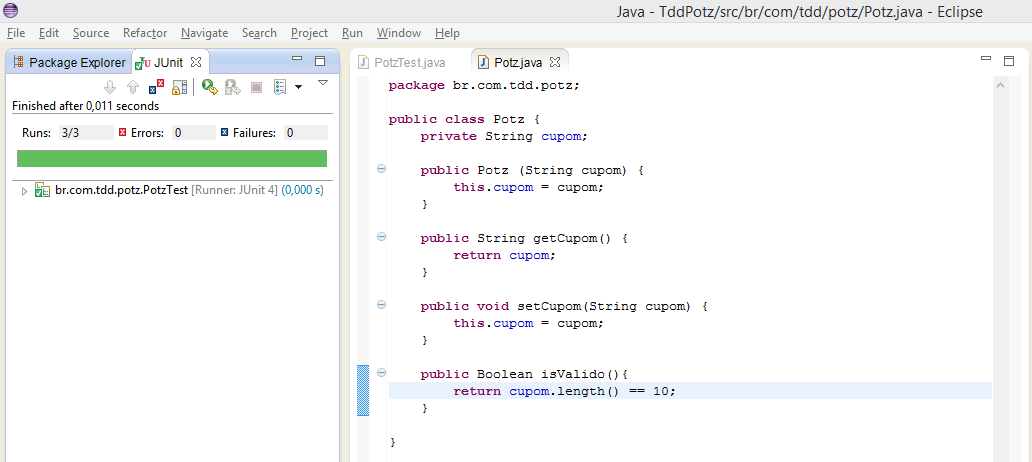


Criação de 2 novos casos de teste para verificar se o cupom é válido (tamnho = 10)

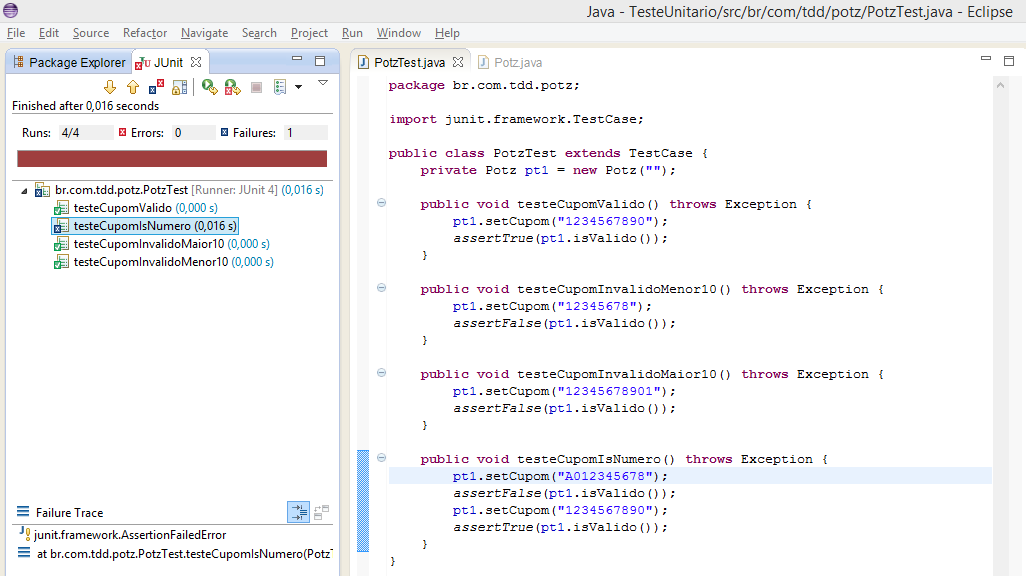
O caso de teste apresenta 2 erros.



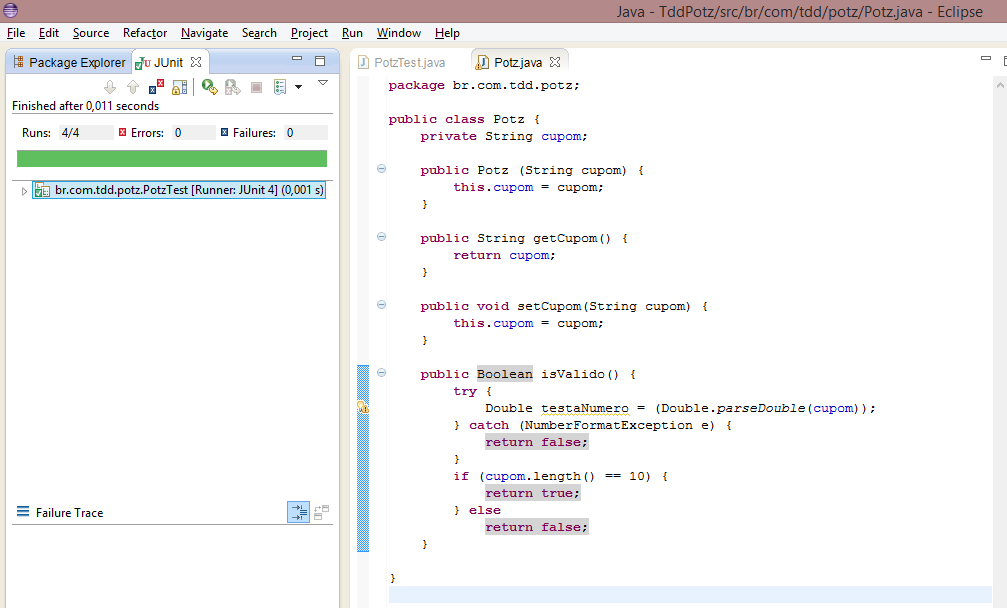
Correção da função is valido. Agora todos os teste passam



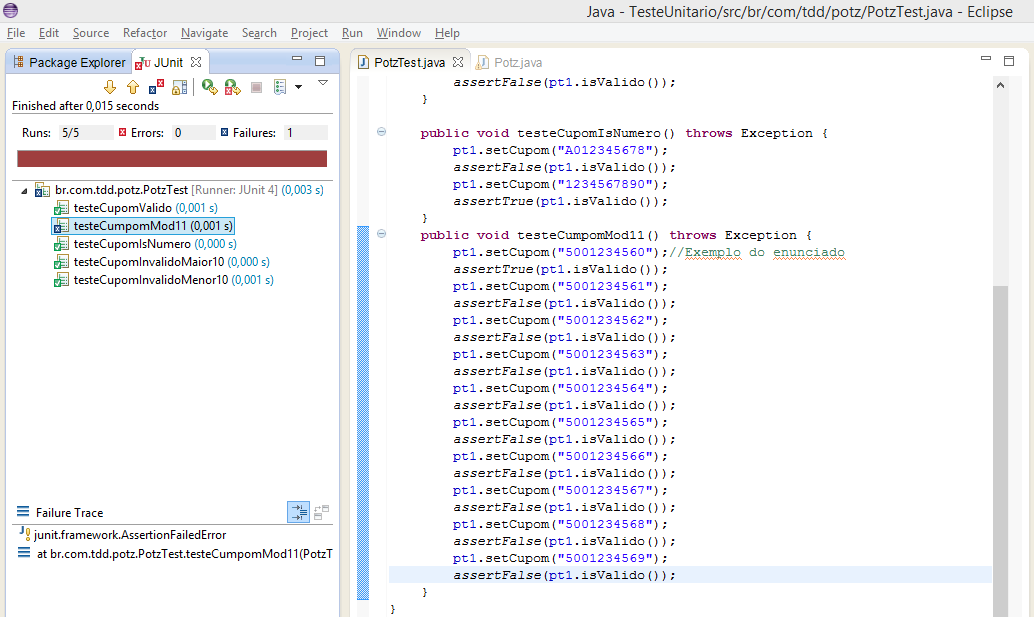
Criação de um novo caso de teste para verificar se o cupom é número: caso de teste não passa



Alterada a função isValido(), agora todos os teste passam

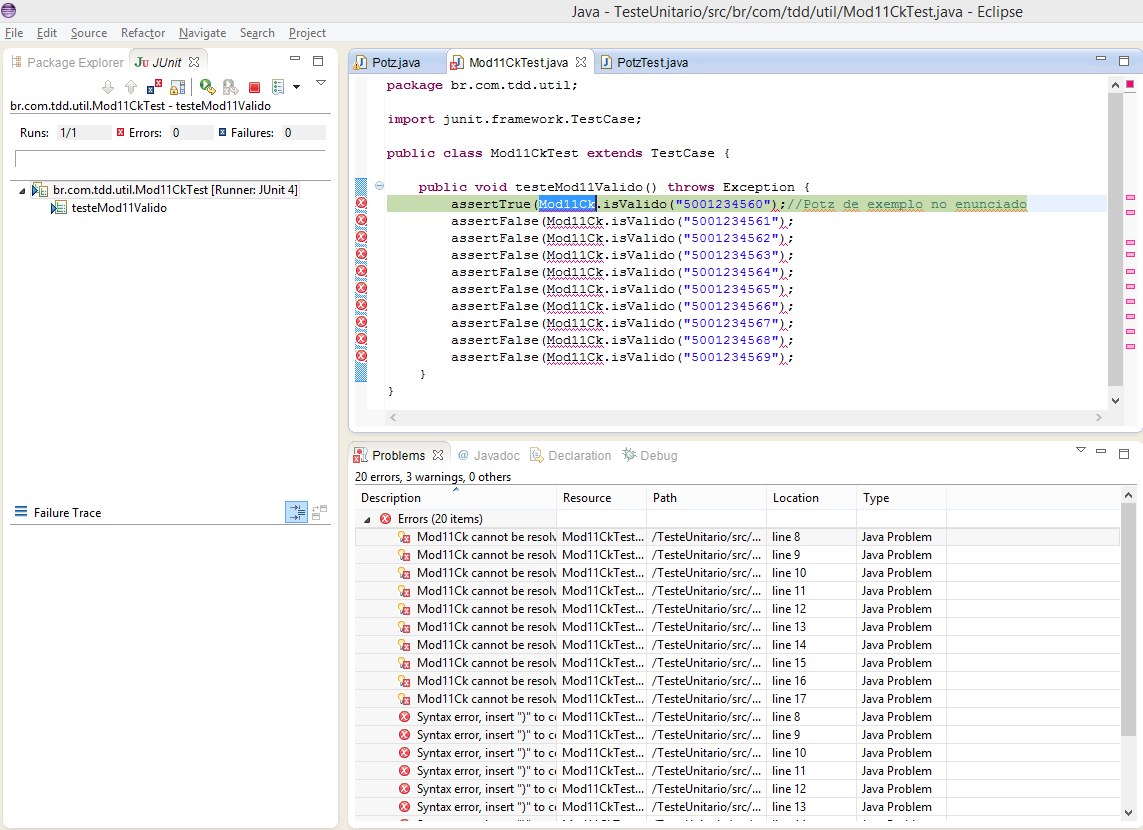


Criado um caso de teste para verificar se o cupom tem digito verificador mod11 válido. Teste não passa

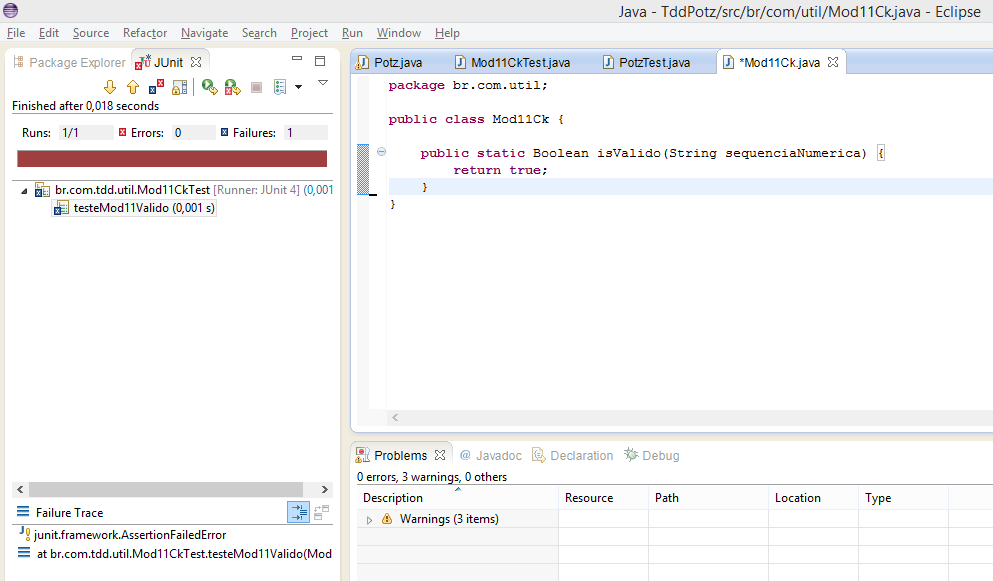


Criado um caso de teste para uma função utilitária que irá verificar o dígito verificador.

Caso de teste não compila pois a classe ainda não foi implementada.



Criada a Classe Mod11CK. Caso de teste agora compila mas não passa



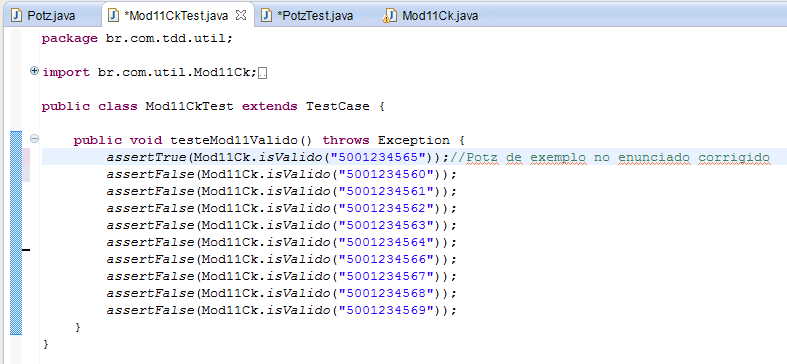
Verificado que o digito verificador do enunciado não é valido

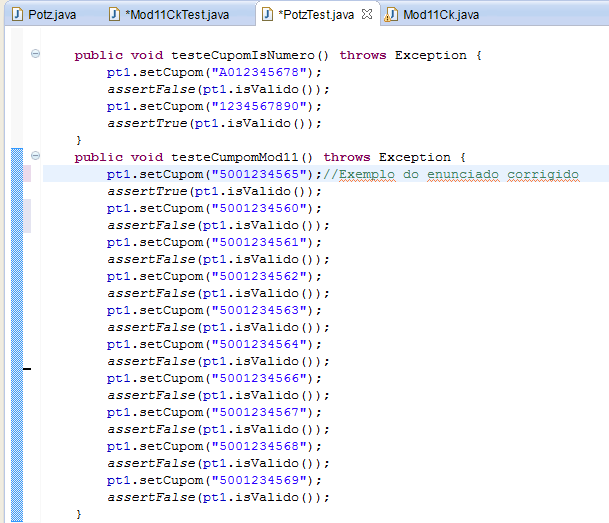
500123456-0

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 5 | 0 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 0 |
| 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 |  |
| 50 | 0 | 0 | 7 | 12 | 15 | 16 | 15 | 12 | 127 |
| 127/11 = 11 sobra 6 | | | | | | | | | |
| 11 - 6 = 5 | | | | | | | | | |

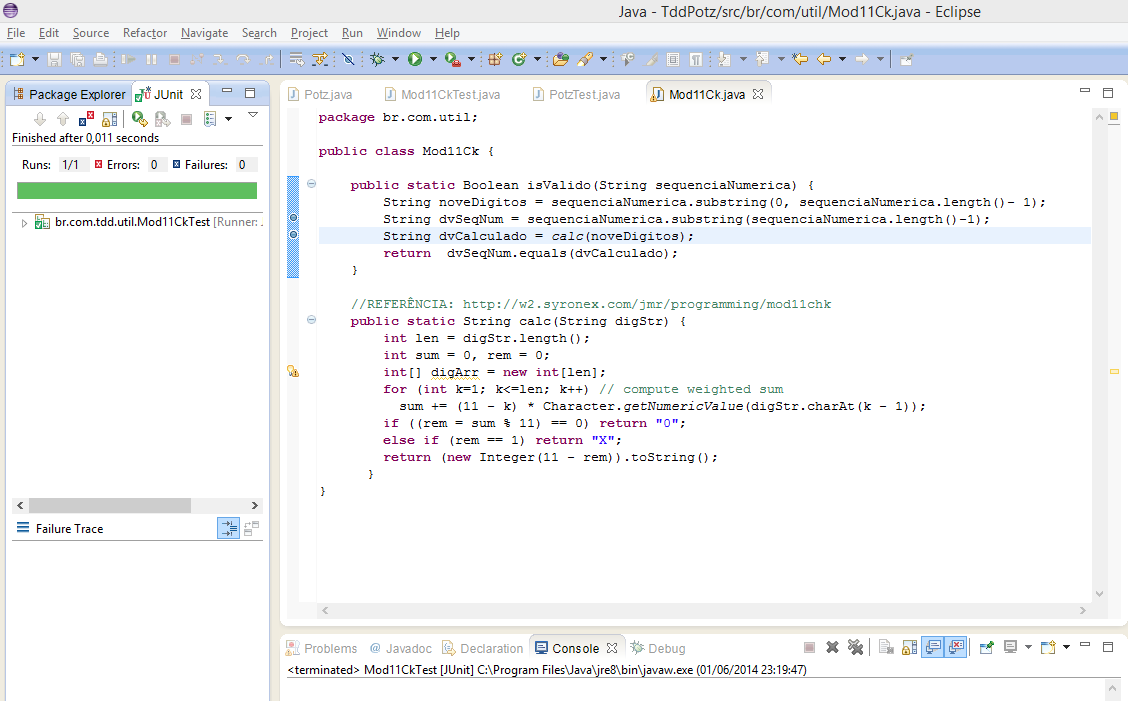
Na verdade deveria ser 5

Alterado os casos de teste com esta nova informação.

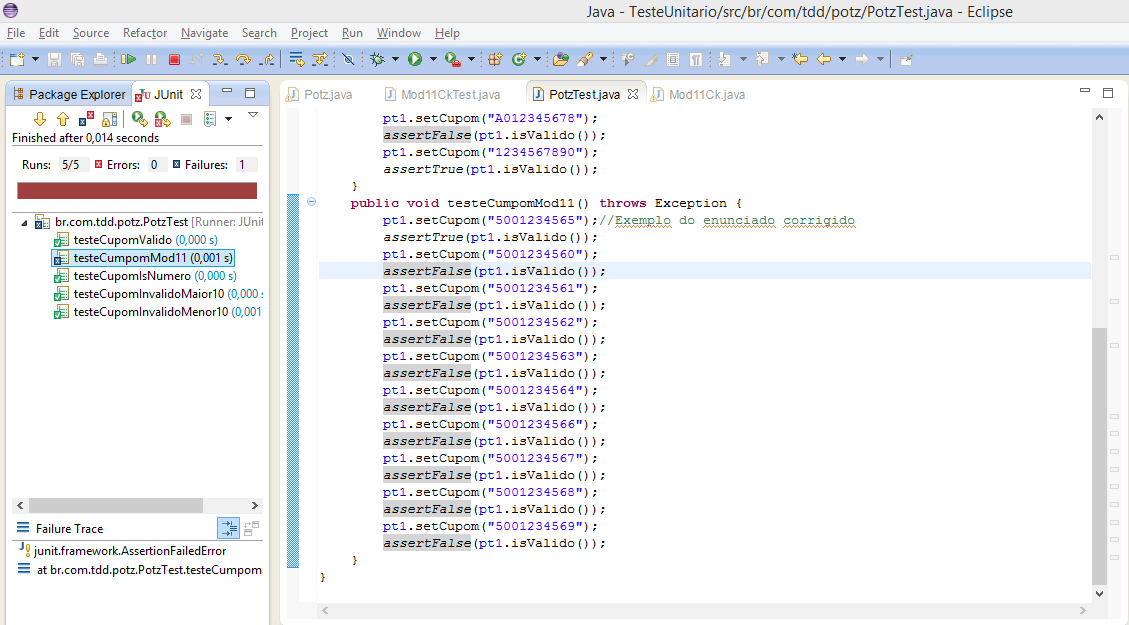




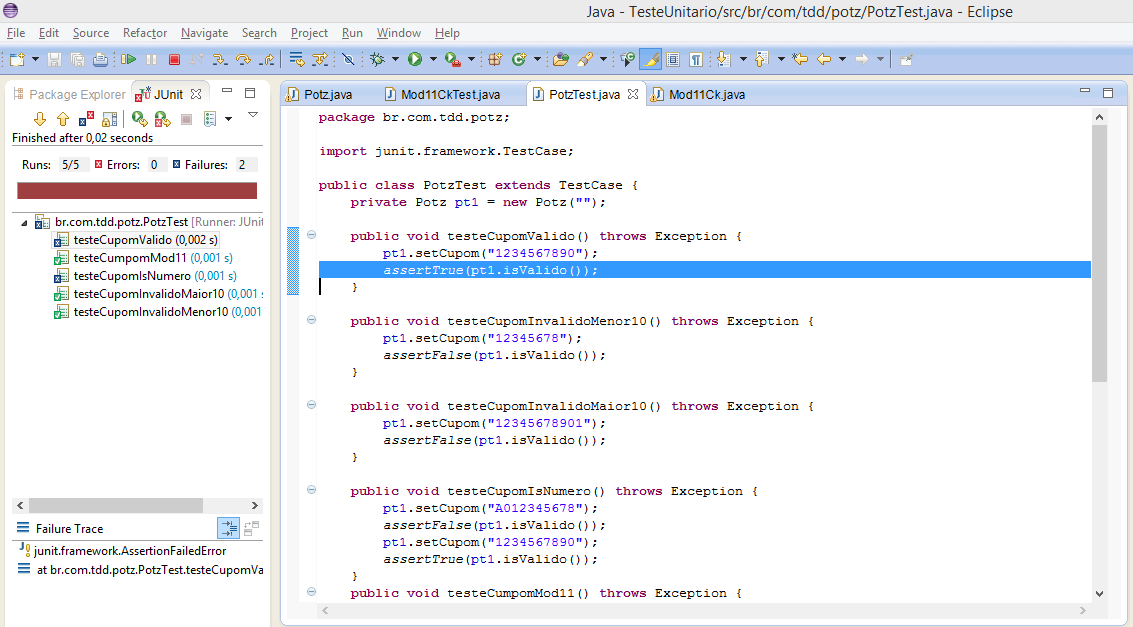
Alterada Função Mod11ck, agora o teste passa.



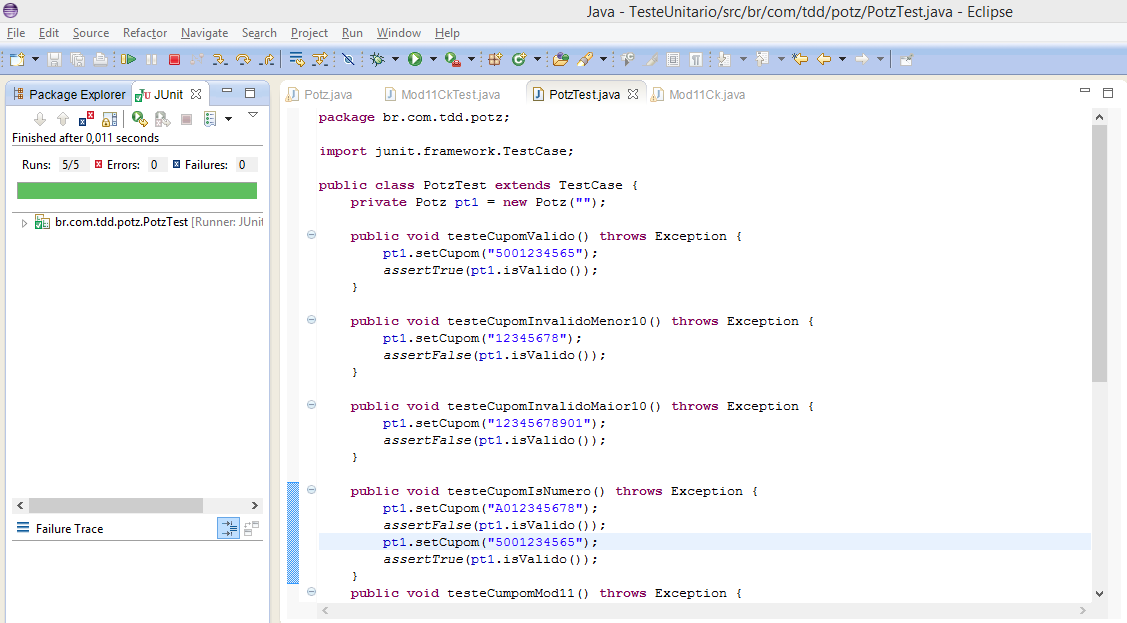
Teste da classe PotzTest continua não passando pois ainda não usa a função de validar digito verificador



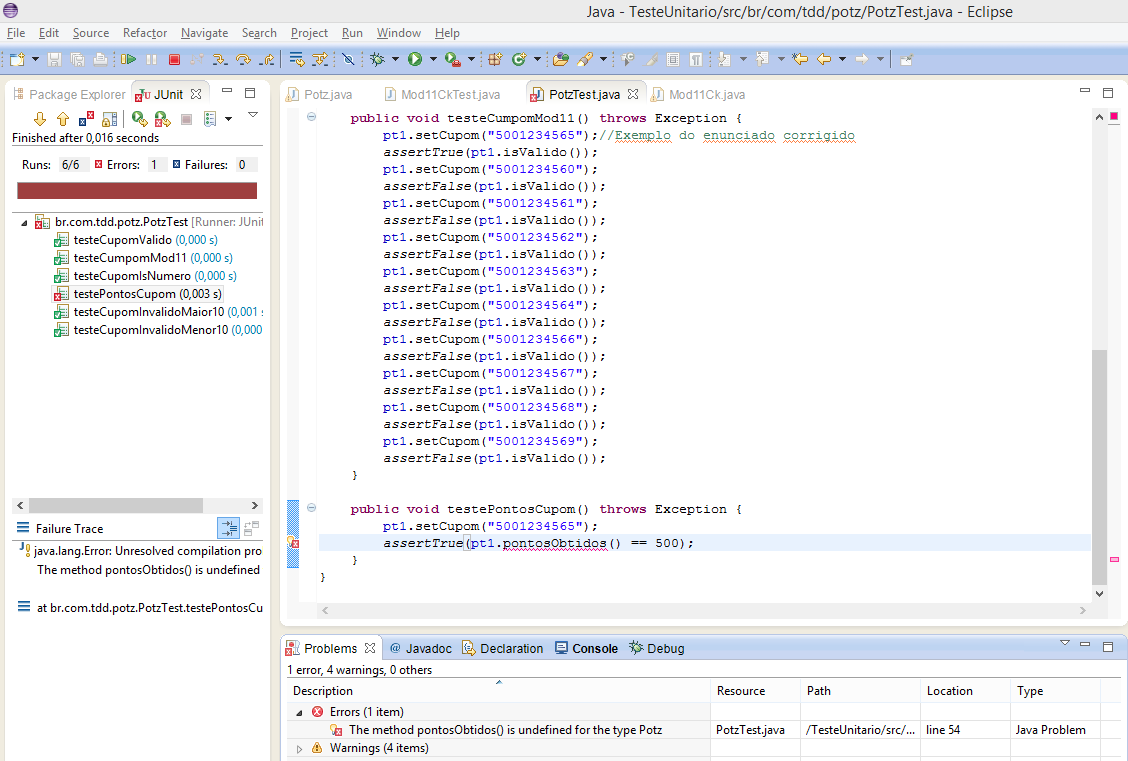
Incluída a validação de digito verificador, 2 outros testes passaram a não falhar, pois antes não validavam digito verificador



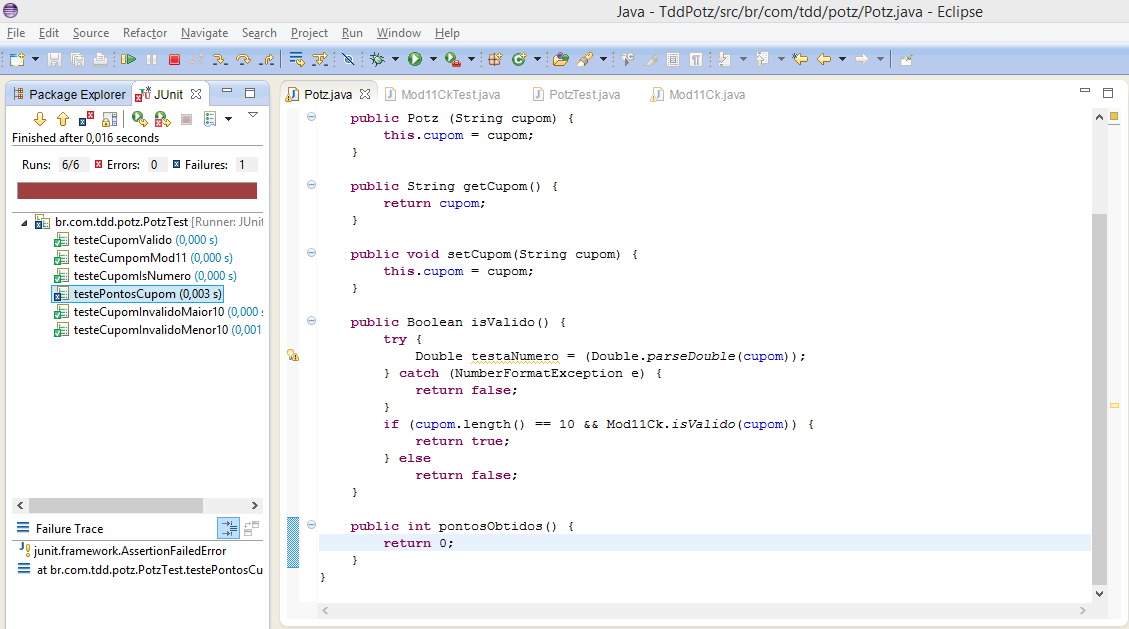
Correção dos casos de teste que falharam do cupom por um cupom com digito verificador correto. Agora todos os teste passam.



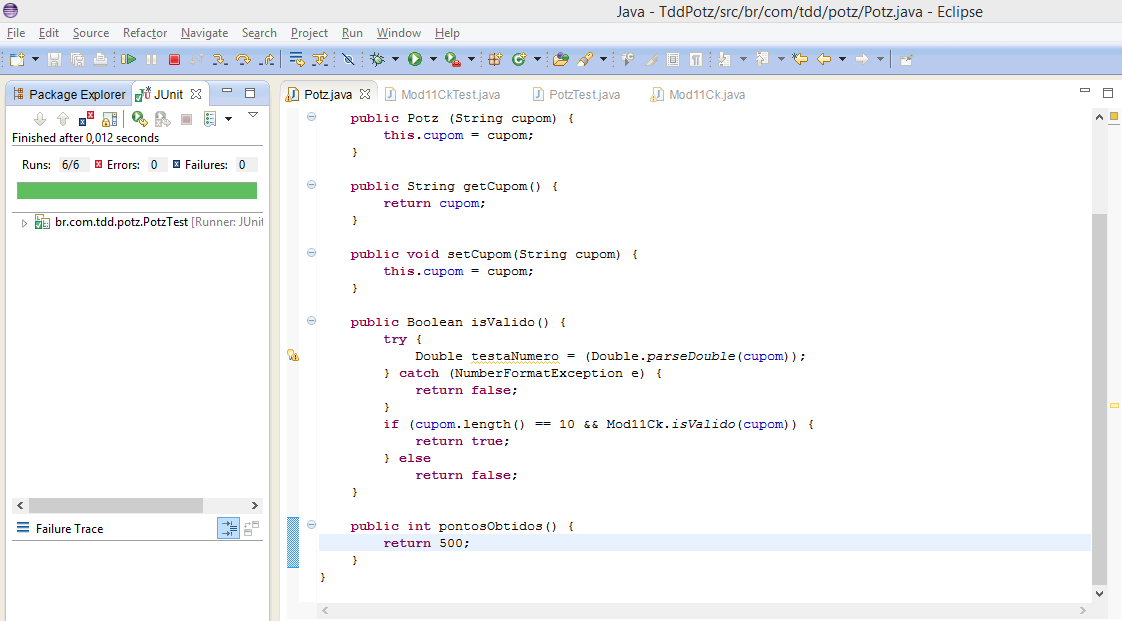
Criação de um caso de teste para verificar pontos obtidos no cupom. Caso de teste não compila pois função que retorna pontos ainda não existe.



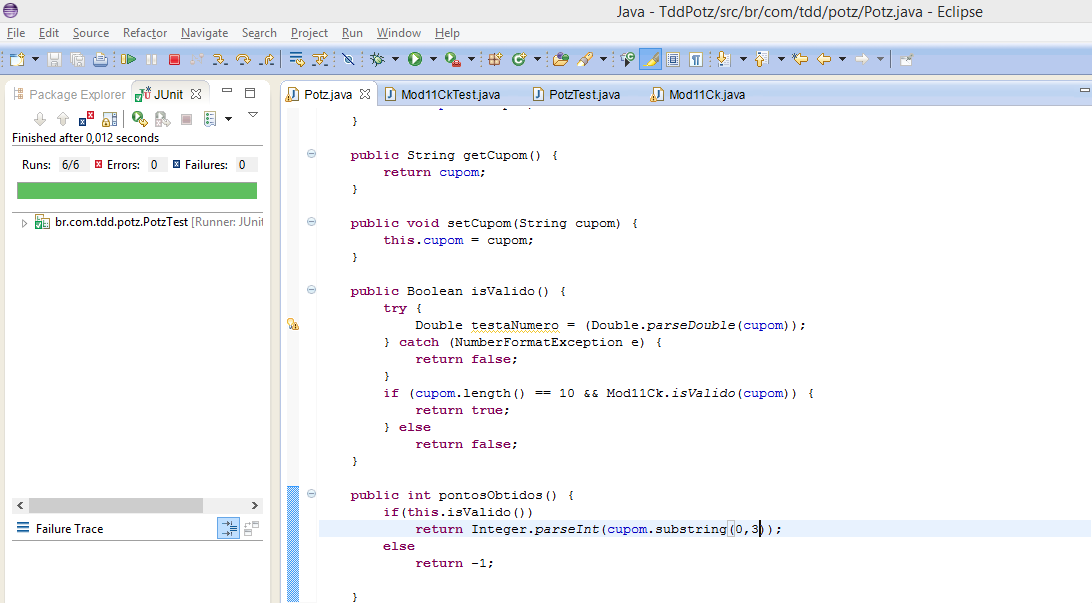
Criada a função. Projeto compila mas da erro.



Alterada a função para passar rapidamente



Implementada a função para o caso geral.



Classe compra para teste

