Федеральное государственное автономное

образовательное учреждение

высшего образования

«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт Космических и информационных технологий

институт

Кафедра «Информатика»

кафедра

**ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ №2**

Тестирование программных блоков

тема

Вариант 4

|  |  |  |
| --- | --- | --- |
| Преподаватель |  | А. С. Кузнецов |
|  | подпись, дата |  |
| Студент |  | А. М. Сотниченко |
|  | подпись, дата |  |

Красноярск 2021

# Цель работы

Изучить процесс блочного тестирования программного обеспечения.

# Общая постановка задачи

Продемонстрировать понимание и владение навыками создания:

* простых блочных тестов;
* фикстуры тестирования;
* параметризованных тестов;
* документации разработчика.

## Задание, соответствующее варианту

Для варианта 4 Рациональное число (дробь из двух целых чисел), реализовать и протестировать следующие функции:

* сравнение;
* сложение:
* вычитание;
* умножение;
* деление;

# Ход работы

Для работы был выбран язык TypeScript (надмножество над JavaScript добавляющее статическую типизацию), среда выполнения NodeJS и фреймворк для тестирования Jest.

Напишем класс для работы с рациональными числами.

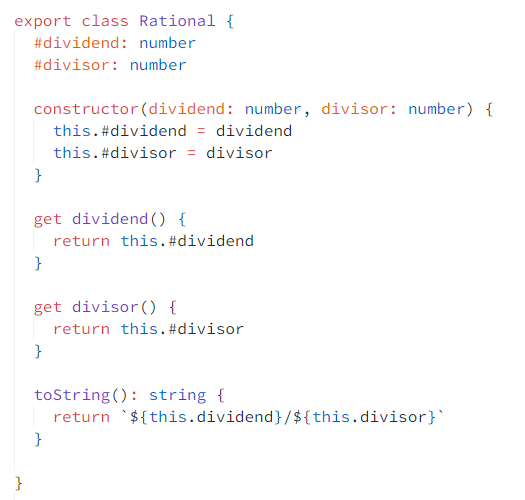


Рисунок 1 – Класс Rational

Напишем методы реализующие требуемую функциональность.

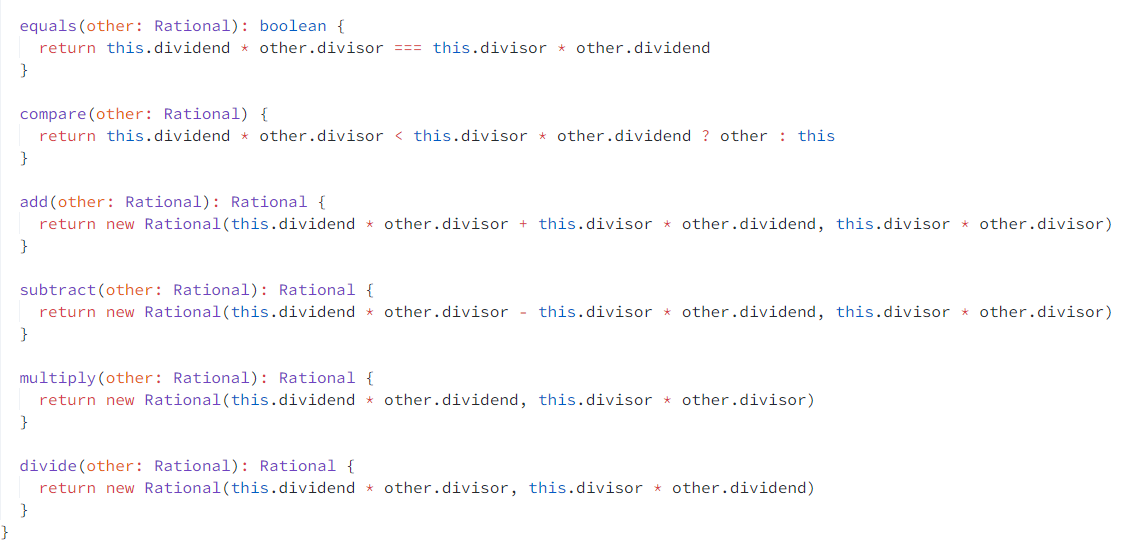


Рисунок 2 – Методы Rational

Теперь задокументируем класс.



Рисунок 3 – Пример документации

Напишем юнит тесты для методов класса.



Рисунок 4 – Пример юнит-тестов

Напишем параметризированные юнит-тесты.

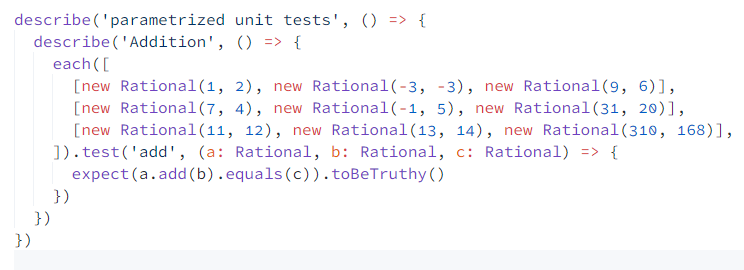


Рисунок 5 – Пример параметризированных юнит-тестов

Теперь настроим тестирование с фиксатурами. Создадим фиксатуры (Рисунок 6, 7), и скрипт, который будет генерировать тесты используя фиксатуры (Рисунок 8). На выходе получаем файл тестов (Рисунок 9).

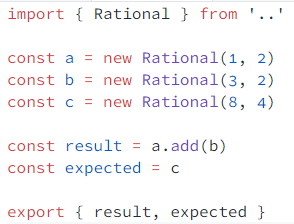


Рисунок 6 – Фиксатура сложения

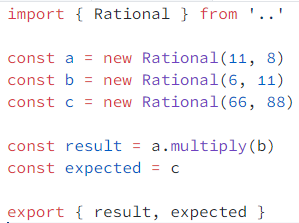


Рисунок 7 – Фиксатура умножения



Рисунок 8 – Скрипт

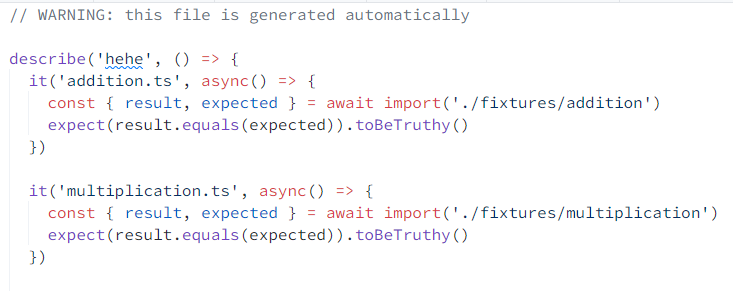


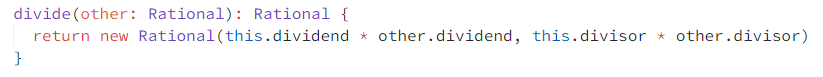
Рисунок 9 – Сгенерированные тесты

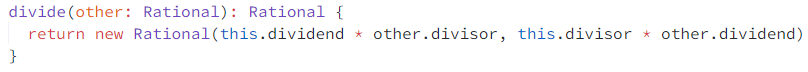
Запустим тесты и посмотрим на результаты.



Рисунок 10 – Результаты тестирования

Увидев результаты тестирования, идем искать ошибку, понимаем, что неправильно переписали формулу с википедии (Рисунок 11), исправляем (Рисунок 12) и снова запускаем тесты (Рисунок 13).

 Рисунок 11 – Неправильная формула

 Рисунок 12 – Исправленная формула

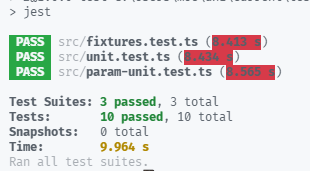


Рисунок 13 – Успешные результаты тестирования

# Вывод

В данной работе мы ознакомились с основами блочного тестирования на языке TypeScript(JavaScript) с применением фреймворка Jest. Были рассмотрены простые и параметризованные блочные тесты. В следующей работе нам предстоит разобраться с контролируемой средой тестирования - фикстурами и фиктивными объектами.