

Тестирование функции вычисления наибольшего общего делителя

Интерфейс: метод `int gcd(int x, int y)` в классе `root.gcd.GCD`

Требования:

1. Предусловие тривиально, т.е., метод должен работать для всех целочисленных значений своих параметров
2. В качестве результата метод возвращает наибольший общий делитель своих аргументов, более точно
 - a. Первый аргумент делится нацело на результат
 - b. Второй аргумент делится нацело на результат
 - c. Среди всех общих делителей аргументов результат является наибольшим по абсолютной величине.
Эквивалентная переформулировка: для значений аргументов x , y и результата r существуют целые числа a , b , такие что $r = ax + by$.
 - d. Результат неотрицателен

Возможные особенности (которые стоило бы протестировать отдельно)

1. Положительные значения аргументов
2. Отрицательное значение первого, второго, обоих аргументов
3. Нулевое значение первого, второго, обоих аргументов (если один из аргументов равен 0, результат по определению должен совпадать по абсолютной величине со вторым).
4. Неединичные взаимно простые аргументы (наибольший общий делитель равен единице)
5. Равные значения аргументов (результат должен быть равен им по абсолютной величине)
6. Неравные значения аргументов, при которых первый делит второй, второй делит первый (результат должен совпадать с меньшим аргументом по абсолютной величине)
7. Неравные значения аргументов, дающие неединичный наибольший общий делитель
8. Граничные значения аргументов (-2^{31} , $2^{31}-1$)
9. ***Отсутствие этого случая не является поводом для снижения оценки. Алгоритм Евклида работает дольше всего на соседних числах Фибоначчи, поэтому полезным тестом для оценки производительности является пара аргументов, являющихся числами Фибоначчи F_n и F_{n+1} для достаточно большого n .

Задание

1. Разработать набор тестов с использованием библиотеки **Junit(или аналога)** для реализации наибольшего общего делителя `int gcd(int x, int y)`
2. Набор тестов должен покрывать все требования и все классы чисел рассмотренные в пункте Возможные особенности.

Тесты следует присылать в виде проекта с исходным кодом(исправленным если найдены ошибки), тестами и всеми библиотеками.

К архиву с проектом следует приложить список обнаруженных ошибок и исправлений(если были) по следующему формату

1. Код до исправления
2. Данные на которых наблюдается некорректное поведение
3. Полученное значение, ожидаемое значение
4. Код после исправления