

## Задание

- Разработать набор тестов с использованием библиотеки TestNG\JUnit для реализации наибольшего общего делителя `int gcd(int x, int y)`
- Набор тестов должен покрывать все требования и все классы чисел, рассмотренные в пункте «Возможные особенности».

## Что сдавать?

Архив содержащий следующее:

1. Полный проект с тестами и необходимыми зависимостями (допускаются указанные в сборочном скрипте, но они должны присутствовать)
2. Файл с описанием ошибок в требованиях и коде (если они были обнаружены). Формат описания проблем приведен в Приложении 2.
3. HTML отчет о покрытии тестами кода метода. При этом желательно разметить отчет полностью в подпапку рядом с содержимым проекта (например, coverage) или в архив. *Стоит помнить, что в отчет обычно входит не один файл, а целый набор связанных ресурсов, нужно добавить их все.*

## Что оценивается?

1. Наличие тестов для каждой категории из описанных ситуаций; (100 процентное покрытие ситуаций)
2. Адекватность тестового набора
  - разбиение на отдельные тесты
  - построение тестов по принципу AAA
  - соответствие тестов решаемым задачам
  - при наличии проблем — наличие тестов на них, возвращающий вердикт за определенное время
3. *Наличие сообщения о проблеме при наличии проблемы в коде;*
4. *Наличие полного (с возможностью переходить по ссылкам, не только index.html) отчета о покрытии со 100 процентным покрытием по строкам \ веткам*

## Приложение 1. Описание 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, точнее:

- 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 формулировка: для значений аргументов  $x$ ,  $y$  и результата  $r$  существуют целые числа  $a$  и  $b$ , такие что  $r = a \cdot x + b \cdot y$ );
- d. результат неотрицателен.

Возможные ситуации (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 1);
- 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  $n$ .

## **Приложение 2. Формат представления сообщений о проблеме**

Для каждой проблемы нужно привести следующие данные

- 1) код до исправления;
- 2) данные, на которых наблюдается некорректное поведение;
- 3) полученное значение, ожидаемое значение;
- 4) код после исправления.

Код до исправления и код после исправления может представлять из себя небольшой фрагмент (место с исправлением и до 1-2 строк вокруг контекста если необходимо).