HW2014ITIS2K1S

ПРАВИЛА

1. **Для задач ниже создать один проект в Intellij Idea. Папку этого проекта целиком поместить в репозиторий или создать репозиторий (выполнить git init и проч.) в папке проекта. Не забудьте сделать git pull origin master перед началом работы. Не забудьте поместить в корень репозитория файл .gitignore (есть в моем открытом репозитории). Не забывайте в процессе работы добавлять файлы (можно с помощью git add . (с точкой) - это добавит всю папку), коммитить и пушить**

2. **Для каждого задания создается отдельный файл TaskNNN.java, где NNN - трехзначный номер задачи. Файл TaskNNN содержит метод main. Для вспомогательных классов, используемых в задаче, можно и даже желательно создавать отдельные java.файлы.**

3. **.java код надо подписывать в самом верху следующим образом (привожу пример по себе на примере своей группы** $953a$ **и задачи** 000**):**

```
/**
* @author Mikhail Abramskiy
* 953a
* 000 (для вспомогательного класса указывайте для чего используется,
       например for 001, 002 and 007)
*/
```

001 Hello World.

002 Создание репозитория.

003 Вычислить объем шара радиуса R. R инициализировать прямо в коде. Вывести ответ на экран.

004 Выполнить по действиям $(1 + y) * (2x + y^2 - (x + y)/(y + 1/(x^2 - 4)))$. Инициализировать x, y прямо в коде. Деление — обычное (не целочисленное). Вывести ответ на экран.

005 Выполнить по действиям без использования дополнительных переменных $((x + 2) * y - z)/y + y * z$. Инициализировать x, y, z прямо в коде. Деление — обычное (не целочисленное). Вывести ответ на экран.

006 Вычислить значение многочлена

$$x^5 + 6x^4 + 10x^3 + 25x^2 + 30x + 101$$

в точке x. x задать прямо в коде. Вывести ответ на экран. Вычисление произвести оптимально (подсказка — сх. Гор.)

007 Для целых a и b (заданных прямо в коде) вывести на экран значения следующих операций (строго в указанной форме, не просто значения, а выражения):

```
a + b = ...
a - b = ...
b - a = ...
a * b = ...
a / b = ...
a % b = ...
b / a = ...
b % a = ...
```

008 Для цифры k, задаваемой прямо в коде, вывести таблицу умножения:

```
2 x k = ...
3 x k = ...
.....
9 x k = ...
```

009 Вычислить y для введенного x (обычным if, без двойных неравенств)

$$y = \begin{cases} \frac{x^2-1}{x+2} & , \quad x > 2 \\ (x^2-1)(x+2) & , \quad 0 < x \leq 2 \\ x^2(1+2x) & , \quad else \end{cases}$$

010 Решить предыдущую задачу, используя сокращенный if.

011 Для введенного n подсчитать $n!!$ (двойной факториал n). $n!! = 1 \cdot 3 \cdot 5 \cdot ... \cdot n$, если n — нечетное, и $2 \cdot 4 \cdot 6 \cdot \ldots \cdot n$, если n — четное. (while или for)

012 Вычислить сумму $S_n = 1 - \frac{1}{3^2} + \frac{1}{5^2} - \frac{1}{7^2} + \ldots$ (n слагаемых). n вводится.

013 Для введенного n подсчитать произведение Валлиса:

$$r_n = \frac{2}{1} \cdot \frac{2}{3} \cdot \frac{4}{3} \cdot \frac{4}{5} \cdot \frac{6}{5} \cdot \frac{6}{7} \cdot \ldots \cdot \frac{2n}{2n-1} \cdot \frac{2n}{2n+1}$$

014 Для введенных n и x подсчитать

$$cos(x + cos(x + cos(...cos(x + cos(x))...)...))$$

(n косинусов, считать косинус с помощью Math.cos())

015 Для введенных целого n и вещественного x подсчитать

$$S = 1 + \cfrac{x}{2 + \cfrac{x}{3 + \cfrac{x}{4 + \cfrac{x}{\ldots + \cfrac{x}{n+x}}}}}$$

016 Для введенных целого n и вещественного x подсчитать

$$S = (x+1) + (x+1)(x+2) + (x+1)(x+2)(x+3) + \ldots + (x+1)(x+2)(x+3)\ldots(x+n)$$

017 Для введенного n подсчитать

$$S = \sum_{m=1}^{n} \frac{((m-1)!)^2}{(2m)!}$$

018 Вводится вещественное x, целое n, вечественные $a_n, a_{n-1}, \ldots a_1, a_0$. Вычислить значение

$$a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x + a_0$$

Общее количество переменных, разрешенное для задачи, равно пяти.

019 Натуральное число n называется совершенным, если $n = \sum_i a_i$, где $1 \leq a_i < n$ — делители числа n. Вывести на экран все совершенные числа в промежутке от 1 до 1000000.

020 Вводится целое число n. Вывести ромб с горизонтальной диагональю равной $2n + 1$ по следующему образцу (например, для n=5)

```
*****0*****
****000****
***00000***
**0000000**
*000000000*
00000000000
*000000000*
**0000000**
***00000***
****000****
*****0*****
```

021 Для введенного n вывести «трифорс» (пример ниже), n - высота каждого треугольника. Например, для $n = 3$:

```
   *
  ***
 *****

 *   *
***  ***
***** *****
```

022 Для введенного n нарисовать символами круг радиуса n. Например, для $n = 10$:

```
**********0**********
******0000000000******
****00000000000000****
***0000000000000000***
**00000000000000000**
**00000000000000000**
*000000000000000000*
*000000000000000000*
*000000000000000000*
*000000000000000000*
0000000000000000000
*000000000000000000*
*000000000000000000*
*000000000000000000*
*000000000000000000*
**00000000000000000**
**00000000000000000**
***0000000000000000***
****00000000000000****
******0000000000******
**********0**********
```

(выглядит как овал, но только потому, что расстояние между буквами меньше, чем расстояние между строками, а если такое нарисовать, будет именно кружок)

Вычислить сумму с точностью $EPS = 0.000000001$ (x вводится, если нужно)

023

$$\sum_{n=1}^{\infty} \frac{2n+3}{5n^4+1}$$

024

$$\sum_{n=1}^{\infty} \frac{1}{n \cdot 9^n (x-1)^{2n}}$$

025

$$\sum_{n=1}^{\infty} \frac{(-1)^{n+1}}{n^2 + 3n}$$

026

$$\sum_{n=1}^{\infty} \frac{(x-1)^n}{3^n (n^2+3) n!}$$

027 Вводится n целых чисел. Проверить, что среди них существует такое, что делится на 2 и 3 или на 5 и 6.

028 Вводятся целые k, m. Вывести целые числа между k и m, которые делятся на 3.

029 Вводится целое $2 \leq k \leq 9$, затем вводится целое число n, которое можно интерпретировать как число в k-ичной системе счисления. Сконвертировать n в десятичную систему счисления.

030 Вводится n чисел. Проверить, что среди них существует ровно два таких числа, что длина (количество цифр) каждого из них равна 3 или 5, а их цифры либо все четные, либо все нечетные.

031 Найти сумму положительных элементов целочисленного массива размера n, если верно, что каждый третий (ориентироваться не на индекс элемента, а на его фактическую позицию) элемент массива делится на 3. Иначе найти произведение положительных элементов.

032 Вводятся два массива, каждый элемент в них - цифра. Оба массива представляют запись целых чисел m и n. Получить массив, который представлет из себя запись числа $m + n$.

033 Вводятся два целочисленных массива размера n. Считая их n-мерными векторами, найти их скалярное произведение, а также косинус угла между ними.

034 Вводится целочисленный массив размера n. Найти максимум среди сумм каждых трех соседних элементов.

035 Выполнить сортировку введенного массива n любым способом (только не Arrays.sort).

036 Вводится целочисленный массив размера n. Вывести такие элементы массива a_i, для которых верно:

$$a_i > a_0, a_i > a_1, \ldots, a_i > a_{i-1}$$

037 Вводится целочисленная квадратная матрица размера n. Заменить нулями элементы, расположенные на верхнем и нижнем треугольниках, образованных пересечением главной и побочной диагоналей. Правый и левый треугольники не трогать. Пример:

```
Вход:
1 2 3 4 5
2 3 4 1 2
0 3 2 3 1
9 2 3 1 4
3 8 0 8 6
Выход:
1 0 0 0 5
2 3 0 1 2
0 3 2 3 1
9 2 0 1 4
3 0 0 0 6
```

038 Вводится квадратная матрица размера $n \times n$. Привести ее к треугольному виду и вывести на экран в виде таблицы.

039 Вводится прямоугольная матрица размером $m \times n$ и целочисленный массив размера p. Гарантируется, что прямоугольная матрица содержит только цифры, а целочисленный массив - только положительные числа не более 6 цифр. Для каждого числа из массива проверить, что в матрице есть строчка / столбец / диагональ, в котором по порядку расположены цифры числа. Направление может быть любым из восьми (север, юг, восток, запад, СЗ, СВ, ЮВ, ЮЗ). Вывести либо последовательность индексов (строка:столбец, нумерация с нуля) в том же порядке, в котором цифры расположены в числе, Пример:

```
Вход:
массив: 234 12 3450 17
матрица:
2 3 4 3 4
1 5 2 3 2
3 4 5 0 7
Выход:
234: 0:0 0:1 0:2
12: 1:0 0:0
3450: 2:0 2:1 2:2 2:3
17: нет
```

040 Нарисовать диаграмму автомата, который принимает только двоичные слова, в которых количество нулей четно, а количество единиц делится на 3 (сфотографировать и прислать картинку почтой в формате jpg/png, имя файла 040).

041 Нарисовать диаграмму автомата, который проверяет, что входное двоичное слово начинается на 2 одинаковых символа, а заканчивается на 2 разных (сфотографировать и прислать картинку почтой в формате jpg/png, имя файла 041).

042 Написать таблицу Машины Тьюринга, реализующую функцию f(x) - 1 для двоичных x. Гарантируется, что входное числоы ≥ 1. Таблицу закинуть в репозиторий в файл 042.txt.

043 Написать таблицу Машины Тьюринга, реализующую функцию f(x,y) = x + y для двоичных положительных x и y. Разрешено использовать дополнительные символы на ленте. Таблицу закинуть в репозиторий в файл 042.txt.

044 Вводится строка, состоящая из слов, разделенных пробелами. Найти в ней слова, которые начинаются с заглавной буквы, а все остальные символы в таких словах — строчные буквы. Вывести эти слова.

045 Вводится массив строк размера n, означающий массив футбольных команд. Далее водится число k, а затем k строчек с результатами матчей команд из массива, строки вида:

```
Рубин Мордовия 5:0
```

Необходимо вычислить разницу забитых и пропущенных мячей для каждой команды по всем введенным результатам. Хранить вводимые k строчек с результатами запрещено!

046 Вводятся 2 строки. Выяснить, какая из них находится лексикографически раньше ("раньше по словарю").

047 Перерешайте задачи 008, 024, 033 (2 метода), 038 (вывод в метод не закидывать) с помощью методов. Один класс, один main, много методов, которые проверяются в main-е.

048 Создать класс "Преподаватель". Атрибуты - фио, предмет. Методы: конструктор (с параметрами), все set и get методы, а также метод "оценить студента принимающего в параметры студента, и работающего следующим образом: генерируется случайное число от 2 до 5, выводится строка: "преподаватель ИМЯПРЕПОДАВАТЕЛЯ оценил студента с именем ИМЯСТУДЕНТА по предмету ИМЯПРЕДМЕТА на оценку ОЦЕНКА."Все слова, написанные капслоком, должны быть заменены соответствующими значениями. ОЦЕНКА должна принимать значения "отлично "хорошо "удовлетворительно"или "неудовлетворительно"(в зависимости от значения случайного числа).

Протестировать методы написанного класса в методе main класса TestClass, используя при этом уже написанный на паре класс Student.

049 Класс Vector2D - двумерный вектор. Атрибуты - два вещественных числа (координаты). Далее (здесь и в последующих подобных задачах) указываю методы с типом возвращамых значений, а в скобках пишу только типы параметров. get- и set- методы создавать по необходимости (тоже здесь и далее).

```
Vector2D() - конструктор для нулевого вектора;

Vector2D(double, double) - конструктор вектора с координатами;
в конструкторах устраняйте дублирование кода;

Vector2D add(Vector2D) - сложение вектора с другим вектором,
результат возвращается как новый объект.

void add2(Vector2D) - сложение вектора с другим вектором,
результат сохраняется в том, у кого был вызван этот метод;

Vector2D sub(Vector2D) - вычитание из вектора другого вектора,
результат возвращается как новый объект;

void sub2(Vector2D) - вычитание из вектора другого вектора,
результат сохраняется в том векторе, у кого был вызван этот метод;

Vector2D mult(double) - умножение вектора на вещественное число,
результат возвращается как новый объект;

void mult2(double) - умножение вектора на вещественное число,
результат сохраняется в векторе;

String toString() - строковое представление вектора;

double length() - длина вектора;

double scalarProduct(Vector2D) - скалярное произведение вектора на другой вектор;

double cos(Vector2D) - косинус угла между этим и другим вектором;

boolean equals(Vector2D) - сравнить вектор с другим вектором;
```

050 Класс RationalFraction - рациональная дробь. Атрибуты - два целых числа (числитель и знаменатель). Методы:

```
RationalFraction() - конструктор для дроби, равной нулю;
```

```
RationalFraction(int, int) - конструктор дроби
со значениями числителя и знаменателя;
в конструкторах устраняйте дублирование кода;

void reduce() - сокращение дроби;

RationalFraction add(RationalFraction) - сложение дроби с другой дробью,
результат возвращается как новый объект
(не забудьте сократить)

void add2(RationalFraction) - сложение дроби с другой дробью,
результат сохраняется в том, у кого был вызван этот метод
(не забудьте сократить);

RationalFraction sub(RationalFraction) - вычитание из дроби другой дроби,
результат возвращается как новый объект (не забудьте сократить);

void sub2(RationalFraction) - вычитание из дроби другой дроби,
результат сохраняется в том, у кого был вызван этот метод (не забудьте сократить);

RationalFraction mult(RationalFraction) - умножение дроби на другую дробь,
результат возвращается как новый объект (сократить)

void mult2(RationalFraction) - умножение дроби на другую дробь,
результат сохраняется;

RationalFraction div(RationalFraction) - деление дроби на другую дробь,
результат возвращается как новый объект (сократить)

void div2(RationalFraction) - деление дроби на другую дробь,
результат сохраняется;
больше не буду писать "возвращается" или "сохраняется", думаю, уже и так понятно.

String toString() - строковое представление дроби (например, -2/3);

double value() - десятичное значение дроби;

boolean equals(RationalFraction) - сравнить дробь с другой дробью
(не забывайте, что 2/4 и 1/2 - одна и та же дробь)

int numberPart() - целая часть дроби;
```

051 Если вдруг вы не в теме, прочитайте сначала:

```
https://ru.wikipedia.org/wiki/Комплексное_число
```

Создать класс ComplexNumber - комплексное число. Атрибуты - действительная и мнимая части (два числа). Методы:

```
ComplexNumber() - конструктор для нулевого комплексного числа;
```

```
ComplexNumber(double, double) - конструктор комплексного числа
с заданными значениями вещественной и мнимой части;
в конструкторах устраняйте дублирование кода;

ComplexNumber add(ComplexNumber) - сложение комплексного числа
с другим комплексным числом;

void add2(ComplexNumber) - сложение комплексного числа
с другим комплексным числом;

CompexNumber sub(ComplexNumber) - вычитание из комплексного числа
другого комплексного числа;

void sub2(ComplexNumber) - вычитание из комплексного числа
другого комплексного числа;

ComplexNumber multNumber(double) - умножение комплексного числа
на вещественное число;

void multNumber2(double) - умножение комплексного числа
на вещественное число;

ComplexNumber mult(ComplexNumber) - умножение комплексного числа
на другое комплексное число;

void mult2(ComplexNumber) - умножение комплексного числа
на другое комплексное число;

ComplexNumber div(ComplexNumber) - деление комплексного числа
на другое комплексное число;

void div2(ComplexNumber) - деление комплексного числа
на другое комплексное число;

double length() - модуль комплексного числа;

String toString() - строковое представление комплексного числа.
Только без всяких "2 * i + - 3".
Проверяйте знаки, чтобы было красиво: 2 * i - 3.

double arg() - аргумент комплексного числа
(может понадобиться тригонометрическое представление (читайте ссылку)
и арктангенс Math.atan);

ComplexNumber pow(double) - возвести в степень по Формуле Муавра
(иные способы запрещены).
Внимание - разрешено использование Math.pow для возведения аргумента в степень
(т.к. оба аргумента double), также вам понадобятся Math.cos, Math.sin.

boolean equals(ComplexNumber) - сравнить комплексное число
с другим комплексным числом;
```

052 Создать класс Matrix2x2 - двумерная матрица из вещественных чисел. Аргументы - со-

держимое матрицы (лучше разумеется хранить двумерным массивом, а то замучаетесь). Методы:

```
Matrix2x2() - конструктор для нулевой матрицы;

Matrix2x2(double) - конструктор для матрицы,
у которой каждый элемент равен поданному числу;

Matrix2x2(double [][]) - конструктор для матрицы,
содержимое подается на вход в виде массива;

Matrix2x2(double, double, double, double) - глупый конструктор, но пусть он будет.
Сами знаете, что он делает.
в конструкторах устраняйте дублирование кода;

Matrix2x2 add(Matrix2x2) - сложение матрицы с другой;

void add2(Matrix2x2) - сложение матрицы с другой;

Matrix2x2 sub(Matrix2x2) - вычитание из матрицы другой матрицы;

void sub2(Matrix2x2) - вычитание из матрицы другой матрицы;

Matrix2x2 multNumber(double) - умножение матрицы на вещественное число;

void multNumber2(double) - умножение матрицы на вещественное число;

Matrix2x2 mult(Matrix2x2) - умножение матрицы на другую матрицу;

void mult2(Matrix2x2) - умножение матрицы на другую матрицу;

double det() - определитель матрицы;

void transpon() - транспонировать матрицу;

Matrix2x2 inverseMatrix() - возвратить обратную матрицу для заданной.
Если это невозможно, вывести сообщение об ошибке и вернуть нулевую матрицу
(кто вдруг знает исключения, может их использовать).

Matrix2x2 equivalentDiagonal() - возвратить эквивалентную диагональную матрицу;

Vector2D multVector(Vector2D) - умножить матрицу на двумерный вектор
(считая его столбцом) и возвратить получившийся столбец в виде вектора.
```

А теперь комбинируем! ^^. В дальшейших задачах все методы, которые делают операции (сложение, умножение и т.п.) всегда возращают значения, void среди них нет. Все вспомогательные операции над компонентами должны опираться на операции, написанные в классах из 049-052. Т.е. если надо умножить рациональные дроби, не надо заново эти методы реализовывать, а надо использовать умножение изи задачи 050.

053 Класс RationalVector2D - двумерный вектор, компоненты которого являются рациональными дробями (т.е. объектами класса RationalFraction). Это и есть атрибуты класса. Методы:

```
RationalVector2D() - конструктор для нулевого вектора
(компоненты должны быть равны нулевым рациональным дробям);

RationalVector2D(RationalFraction, RationalFraction) - конструктор вектора
с координатами;
в конструкторах устраняйте дублирование кода;

RationalVector2D add(RationalVector2D) - сложение вектора с другим вектором;

String toString() - строковое представление вектора
(использует строковое представление RationalFraction);

double length() - длина вектора;

RationalFraction scalarProduct(RationalVector2D) - скалярное произведение
вектора на другой вектор;

boolean equals(RationalVector2D) - сравнить вектор с другим вектором
(опираться на equals у RationalFraction);
```

054 Класс ComplexVector2D - двумерный вектор, компоненты которого являются комплексными числами (т.е. объектами класса ComplexNumber). Это и есть атрибуты класса. Уже не буду писать, что надо в операциях опираться на методы класса ComplexNumber. Методы:

```
ComplexVector2D() - конструктор для нулевого вектора
(компоненты должны быть равны нулевым комплексным числам);
ComplexVector2D(ComplexNumber, ComplexNumber) -
конструктор вектора с координатами;
в конструкторах устраняйте дублирование кода;
ComplexVector2D add(ComplexVector2D) - сложение вектора с другим вектором;
String toString() - строковое представление вектора.
ComplexNumber scalarProduct(ComplexVector2D) - скалярное произведение вектора
на другой вектор;
boolean equals(ComplexVector2D) - сравнить вектор с другим вектором;
```

055 Создать класс RationalMatrix2x2 - двумерная матрица из RationalFraction. Аргументы - содержимое матрицы (лучше разумеется хранить двумерным массивом, а то замучаетесь). Методы:

```
RationalMatrix2x2() - конструктор для нулевой матрицы;

RationalMatrix2x2(RationalFration) - конструктор для матрицы,
у которой каждый элемент равен поданному числу;

RationalMatrix2x2(RationalFraction, RationalFraction,
                  RationalFraction, RationalFraction)
- конструктор на 4 дробях.
в конструкторах устраняйте дублирование кода;

RationalMatrix2x2 add(RationalMatrix2x2) - сложение матрицы с другой;

RationalMatrix2x2 mult(RationalMatrix2x2)
- умножение матрицы на другую матрицу;
```

```
RationalFraction det() - определитель матрицы;

RationalVector2D multVector(RationalVector2D)
- умножить матрицу на двумерный вектор (считая его столбцом)
и возвратить получившийся столбец в виде вектора.
```

056 Создать класс ComplexMatrix2x2 - двумерная матрица из ComplexNumber. Аргументы - содержимое матрицы (лучше разумеется хранить двумерным массивом, а то замучаетесь). Методы:

```
ComplexMatrix2x2() - конструктор для нулевой матрицы;

ComplexMatrix2x2(ComplexNumber) - конструктор для матрицы,
у которой каждый элемент равен поданному числу;

ComplexMatrix2x2(ComplexNumber, ComplexNumber,
ComplexNumber, ComplexNumber)
- конструктор на 4 дробях.
в конструкторах устраняйте дублирование кода;

ComplexMatrix2x2 add(ComplexMatrix2x2) - сложение матрицы с другой;

ComplexMatrix2x2 mult(ComplexMatrix2x2) - умножение матрицы
на другую матрицу;

ComplexNumber det() - определитель матрицы;

ComplexVector2D multVector(ComplexVector2D) - умножить матрицу
на двумерный комплекснозначный вектор (считая его столбцом)
и возвратить получившийся столбец в виде вектора.
```

057 Создать класс RationalComplexNumber - комплексное число, компонентами которого являются рациональные дроби. Атрибуты - RationalFraction). Методы:

```
RationalComplexNumber() - конструктор для нулевого комплексного числа;

RationalComplexNumber(RationalFraction, RationalFraction)
- конструктор комплексного числа с заданными значениями
вещественной и мнимой части;
в конструкторах устраняйте дублирование кода;

RationalComplexNumber add(RationalComplexNumber)
- сложение с другим таким числом;

RationalComplexNumber sub(RationalComplexNumber) - вычитание;

RationalComplexNumber mult(RationalComplexNumber) - умножение;

String toString() - строковое представление такого комплексного числа.
Только без всяких "2/3 * i + -3/5".
Проверяйте знаки, чтобы было красиво: 2/3 * i - 3/5.
```

058 Класс RationalComplexVector2D - двумерный вектор, компоненты которого являются объектами класса RationalComplexNumber. Это и есть атрибуты класса. Уже не буду писать, что надо в операциях опираться на методы класса RationalComplexNumber. Методы:

```
RationalComplexVector2D() - конструктор для нулевого вектора
(компоненты должны быть равны нулевым рациональным дробям);

RationalComplexVector2D(RationalComplexNumber, RationalComplexNumber) -
конструктор вектора с координатами;
в конструкторах устраняйте дублирование кода;

RationalComplexVector2D add(RationalComplexVector2D)
- сложение вектора с другим вектором;

String toString() - строковое представление такого вектора.

RationalComplexNumber scalarProduct(RationalComplexVector2D)
- скалярное произведение вектора на другой вектор;
```

059 Создать класс RationalComplexMatrix2x2 - двумерная матрица из RationalComplexNumber. Аргументы - содержимое матрицы (лучше разумеется хранить двумерным массивом, а то замучаетесь). Методы:

```
RationalComplexMatrix2x2() - конструктор для нулевой матрицы;

RationalComplexMatrix2x2(RationalComplexNumber)
- конструктор для матрицы, у которой каждый элемент равен поданному числу;

RationalComplexMatrix2x2(RationalComplexNumber, RationalComplexNumber,

RationalComplexNumber, RationalComplexNumber)
- конструктор на 4 дробях.
в конструкторах устраняйте дублирование кода;

RationalComplexMatrix2x2 add(RationalComplexMatrix2x2)
- сложение матрицы с другой;

RationalComplexMatrix2x2 mult(RationalComplexMatrix2x2)
- умножение матрицы на другую матрицу;

RationalComplexNumber det() - определитель матрицы;

RationalComplexVector2D multVector(RationalComplexVector2D)
- умножить матрицу на двумерный комплекснозначный
рациональный вектор (считая его столбцом) и возвратить
получившийся столбец в виде вектора.
```

Все методы из задача 049-059 протестировать в соответствующих main-методах (можно один, можно несколько классов с методами, кто в теме, может main прямо в разрабатываемых классах делать).

ШТРАФНЫЕ ЗАДАЧИ

999 Вычислить (переменные вводятся, в одной строчке не более одной арифметической операции)

$$\frac{\frac{(x+y)(2-x)}{x+100y+687}-\frac{\frac{x+54z}{(y-2)z-365}}{258z}}{\frac{(x+y)(2-x)}{y+100x+607}}-5021z\frac{\frac{x+73z}{(x-2)y-365}}{213z}$$