

1	Класс <code>EqualitySystem[T]</code> , представляющий систему равенств вида $a = b$ , где $a$ и $b$ – имена переменных или значения типа $T$ . В классе должно быть реализовано две операции: добавление нового равенства; проверка, существует ли решение системы. В случае, если тип $T$ – числовой, дополнительно должна быть реализована операция, возвращающая значение переменной, возвращающая <code>Option[T]</code> . Значения должны назначаться переменным произвольно, но так, чтобы система равенств не нарушалась.
2	Класс <code>MegaQueue[T]</code> , представляющий неизменяемую очередь с операциями <code>enqueue</code> , <code>dequeue</code> и <code>empty</code> , реализованную через два списка. В случае, если $T$ – числовой тип, для очереди должна быть также доступна операция <code>max</code> , работающая за константное время.
3	Класс <code>Seq[T : Ordering]</code> , представляющий отсортированную последовательность значений типа $T$ с операцией объединения двух последовательностей. В случае, если $T$ – числовой тип, для последовательности должна быть также доступна операции умножения всех элементов на число, работающая за константное время.
4	Класс <code>Permutation[T]</code> , представляющий неизменяемую перестановку подмножества значений типа $T$ с двумя операциями: получение $i$ -го элемента и транспозиция. В случае, если $T$ – строковый тип, для <code>Permutation[T]</code> доступна дополнительная операция <code>superStringSize</code> , возвращающая длину минимальной суперстроки, в которую строки, содержащиеся в перестановке, входят в том порядке, в котором они расположены в перестановке.
5	Класс <code>Formula[T]</code> , представляющий формулу, состоящую из имён переменных, констант типа $T$ и бинарных операций. Ассортимент операций зависит от типа $T$ : если определён <code>Numeric[T]</code> , то должны присутствовать четыре базовых арифметических операций; если $T$ – это <code>string</code> , то должна присутствовать только операция сложения, обозначающая конкатенацию. Для создания объектов класса <code>Formula[T]</code> доступны два конструктора: первый принимает имя переменной и создаёт формулу, состоящую из единственной переменной; второй конструктор принимает значение типа $T$ и создаёт формулу, состоящую из единственной константы. Все остальные формулы возвращаются бинарными операциями. В классе <code>Formula[T]</code> должен быть определён метод, принимающий отображение имён переменных в их значения и считающий значение формулы.

6	Класс <code>Range[T : Ordering]</code> , представляющий интервал на множестве значений типа <code>T</code> с операциями определения вхождения одного интервала в другой, пересечения двух интервалов и определения принадлежности значения интервалу. В случае, если тип <code>T</code> – числовой, дополнительно должна быть реализована операция вычисления длины интервала.
7	Класс <code>SuperStack[T]</code> , представляющий неизменяемый стек с операциями <code>push</code> , <code>pop</code> и <code>empty</code> , реализованный через список. В случае, если <code>T</code> – числовой тип, для стека должны быть также доступны операции <code>min</code> и <code>max</code> , работающие за константное время.
8	Класс <code>QuadraticEquation[T]</code> , представляющий квадратное уравнение с коэффициентами типа <code>T</code> и оперцией <code>solve</code> , возвращающей список найденных корней типа <code>T</code> . Операция <code>solve</code> должна быть доступна в случае, если <code>T</code> – числовой тип из стандартной библиотеки языка <code>Scala</code> . Кроме того, эта операция должна быть реализована для класса комплексных чисел, написанного самостоятельно или найденного в Интернете.
9	Класс <code>Circle[T]</code> , представляющий круг на плоскости, заданный координатами центра и радиусом. Координаты центра и радиус заданы значениями числового типа <code>T</code> . В классе должна быть реализована операция проверки, что один круг целиком помещается внутри другого круга, и, кроме того, в случае, если тип <code>T</code> – не целый, операции вычисления длины окружности круга и площади.
10	Класс <code>MegaStack[T]</code> , представляющий неизменяемый стек с операциями <code>push</code> , <code>pop</code> и <code>empty</code> , реализованный через список. В случае, если <code>T</code> – числовой тип, для стека должна быть также доступна операция <code>average</code> , возвращающая среднее арифметическое элементов стека и работающая за константное время.
11	Класс <code>Fib[T]</code> , представляющий $i$ -тый член последовательности Фибоначчи типа <code>T</code> . Конструктор класса <code>Fib[T]</code> принимает два первых члена последовательности и создаёт объект, соответствующий первому члену. Метод <code>next</code> возвращает новый объект, соответствующий следующему члену. Если тип <code>T</code> – целочисленный, то очередной член последовательности вычисляется как сумма двух предыдущих. Если тип <code>T</code> – строка, то вместо сложения используется конкатенация.
12	Класс <code>SuperNumber[T]</code> , представляющий число произвольной разрядности в системе счисления, цифрами которой выступают значения типа <code>T</code> . Если <code>T</code> – целочисленный тип <code>Scala</code> или <code>Bool</code> , то для <code>SuperNumber[T]</code> должна быть доступна операция сложения. Если <code>T</code> – <code>Bool</code> , то дополнительно должны работать операции поразрядного И и ИЛИ.

13	Класс <code>Polynom[T]</code> , представляющий полином с коэффициентами типа <code>T</code> и операцией, возвращающей степень полинома. В случае, если тип <code>T</code> – числовой, для <code>Polynom[T]</code> также должна быть доступна операция дифференцирования.
14	Класс <code>KadaneStack[T]</code> , представляющий неизменяемый стек с операциями <code>push</code> , <code>pop</code> и <code>empty</code> , реализованный через список. В случае, если <code>T</code> – числовой тип, для стека должна быть также доступна операция <code>maxSum</code> , возвращающая максимальную сумму подряд идущих элементов стека и работающая за константное время.
15	Класс <code>StackMachine[T]</code> , представляющий неизменяемый снимок состояния стековой машины, оперирующей значениями типа <code>T</code> . В случае, если <code>T</code> – числовой тип, в стековой машине должны быть реализованы арифметические операции. В <code>StackMachine[Boolean]</code> должны присутствовать логические операции.
16	Класс <code>Polynom[T]</code> , представляющий полином с коэффициентами типа <code>T</code> и операцией вычисления значения в точке $x$ . Полином должен быть задан списком коэффициентов. В качестве типа <code>T</code> может выступать числовой тип.
17	Класс <code>Line[T]</code> , представляющий отрезок прямой на плоскости, заданный двумя точками с координатами числового типа <code>T</code> . В классе должна быть реализована операция параллельного переноса отрезка, и, кроме того, в случае, если тип <code>T</code> – не целый, операции вычисления длины отрезка и его поворота относительно заданной точки на указанный угол.
18	Класс <code>LinEq[T]</code> , представляющий линейное неравенство вида $a_1x_1 + a_2x_2 + \dots + a_nx_n < b$ с операцией, проверяющей, удовлетворяет ли заданный список значений переменных неравенству. Коэффициенты неравенства, а также значения переменных имеют тип <code>T</code> , который либо является числовым, либо булевым. Для булевого типа умножение – это конъюнкция, сложение – дизъюнкция, и порядок задан как <code>false &lt; true</code> .
19	Класс <code>Shape[T : Ordering]</code> , представляющий бесконечное множество «точек» в пространстве $T \times T$ . Конструктор класса должен принимать две «точки», задающие вершины «прямоугольника» (тем самым конструируется множество, состоящее из всех точек прямоугольника). В классе должны быть реализованы операции: объединение двух множеств, пересечение двух множеств, определение принадлежности «точки» множеству. Кроме того, если тип <code>T</code> – числовой, дополнительно должна быть реализована операция параллельного переноса всех точек множества на заданное расстояние.

20	<p>Класс <code>Bijection[T]</code>, представляющий неизменяемое взаимнооднозначное отображение заданного конечного подмножества значений типа <code>T</code> в себя с операциями применения отображения к значению (возвращает <code>Option[T]</code>) и обмена значений, в которые отображаются два элемента (<code>swap</code>).</p> <p>Конструктор класса должен получать в качестве параметра множество и порождать тождественное отображение. Для <code>Bijection[String]</code> должен быть доступен метод переворачивания всех строк в отображении, работающий за константное время. Для <code>Bijection[Int]</code> должен быть доступен метод прибавления заданного числа к каждому элементу множества, на котором определено отображение (он тоже должен работать за константное время).</p>
21	<p>Класс <code>Triangle[T]</code>, представляющий треугольник на плоскости с вершинами, помеченными значениями типа <code>T</code>. Треугольник должен быть задан координатами вершин и поддерживать операции вычисления периметра и площади. Кроме того, если тип <code>T</code> – числовой, то метки вершин следует понимать как массы расположенных в вершинах материальных точек, и у треугольника должна дополнительно присутствовать операция вычисления координат его центра масс.</p>
22	<p>Класс <code>Powers[T]</code>, представляющий бесконечную последовательность степеней некоторого значения <math>x</math> типа <code>T</code>, для которой реализованы две операции: получение <math>i</math>-того члена последовательности, умножение всех элементов последовательности на <math>x</math>. В качестве типа <code>T</code> может выступать либо числовой тип, либо строка (в случае строки умножением является конкатенация).</p>
23	<p>Класс <code>Vector[T &lt;: Product]</code>, представляющий вектор в двух или трёхмерном пространстве. Если тип <code>T</code> представляет пару или тройку, элементы которой имеют одинаковый числовой тип, то для <code>Vector[T &lt;: Product]</code> должны быть доступны операции сложения и скалярного умножения. Дополнительно, для троек должна быть реализована операция векторного умножения.</p>
24	<p>Класс <code>Matrix[T]</code>, представляющий неизменяемую квадратную матрицу с элементами типа <code>T</code>, для которой реализованы две операции: получение значения элемента на <math>i</math>-той строке в <math>j</math>-том столбце, а также перестановка двух строк матрицы. В случае, если <code>T</code> – числовой тип, для <code>Matrix[T]</code> должна быть доступна дополнительная операция <code>power</code>, возвращающая результат возведения матрицы в указанную степень.</p>

25	Класс <code>Circle[T : Numeric]</code> , представляющий окружность на плоскости, заданную координатами центра и радиусом. Координаты центра и радиус заданы значениями числового типа <code>T</code> . В классе должны быть реализованы две операции: вычисление площади и длины окружности (всегда возвращают <code>Double</code> ). В случае, если тип <code>T</code> – целый, в классе должна быть доступна операция, возвращающая список точек окружности, имеющих целые координаты.
26	Класс <code>AntidiagonalMatrix[T : Numeric]</code> , представляющий неизменяемую антидиагональную матрицу с элементами числового типа <code>T</code> . (Все элементы антидиагональной матрицы, кроме лежащих на диагонали, идущей от нижнего левого угла до верхнего правого угла, равны нулю.) В классе должны быть реализованы три операции: получение значения элемента на $i$ -той строке в $j$ -том столбце; сложение с другой антидиагональной матрицей; вычисление определителя. В случае, если тип <code>T</code> – <code>Float</code> или <code>Double</code> , в классе должна быть доступна операция «нормализации» матрицы, т.е. деления всех элементов матрицы на её определитель.
27	Класс <code>ArithProgression[T]</code> , представляющий арифметическую прогрессию с элементами типа <code>T</code> и операцией вычисления $i$ -го члена и суммы первых $n$ членов. Арифметическая прогрессия должны быть задана начальным членом и разностью. В качестве типа <code>T</code> может выступать числовой тип или строка. В случае строки сложение означает конкатенацию.
28	Класс <code>Matrix[T]</code> , представляющий неизменяемую квадратную матрицу с элементами типа <code>T</code> , для которой реализованы две операции: получение значения элемента на $i$ -той строке в $j$ -том столбце, а также транспонирование матрицы. В случае, если <code>T</code> – числовой тип, для <code>Matrix[T]</code> должна быть доступна дополнительная операция <code>square</code> , возвращающая результат произведения матрицы на саму себя.
29	Класс <code>Tree[K : Ordering, V]</code> , представляющий неизменяемое дерево поиска с ключами типа <code>K</code> и значениями типа <code>V</code> . Дерево должно поддерживать операции добавления словарной пары и поиска значения по ключу. Кроме того, в случае, если <code>V</code> – булевский тип, у дерева должна дополнительно присутствовать операция замены всех <code>true</code> в вершинах на <code>false</code> , и наоборот, работающая за константное время.
30	Класс <code>Matrix[T]</code> , представляющий матрицу размера $m \times n$ с элементами типа <code>T</code> и операциями сложения и умножения. В качестве типа <code>T</code> может выступать числовой тип или множество строк. В случае множества строк сложение означает объединение множеств, а умножение даёт множество, составленное из всех попарных конкатенаций строк, первая из которых взята из левого операнда, а вторая – из правого.

31	Класс <code>Matrix[T]</code> , представляющий квадратную матрицу размера $n$ с элементами типа <code>T</code> и операцией удаления указанных строки и столбца. Кроме того, если тип <code>T</code> – числовой, дополнительно должна быть реализована операция вычисления определителя матрицы.
32	Класс <code>EquationSystem[T]</code> , представляющий систему из двух линейных уравнений от двух переменных с коэффициентами типа <code>T</code> и оперцией <code>solve</code> , возвращающей решение уравнения в виде <code>Option[(S, S)]</code> , где тип <code>S</code> зависит от типа <code>T</code> . Эта зависимость задаётся следующими правилами: если <code>T</code> – тип с плавающей точкой, то <code>S</code> совпадает с <code>T</code> ; если <code>T</code> – целочисленный тип, то <code>S</code> – дробь, числитель и знаменатель которой имеют тип <code>T</code> ; в остальных случаях операция <code>solve</code> недоступна. Для представления дробей следует создать отдельный класс.