

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧЕРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет компьютерных наук

Кафедра цифровых технологий

Программная реализация (на языке JavaScript) алгоритмов
генерации ФОС ЕГЭ и ОГЭ в 2026 году

Курсовая работа

Направление 020301 Математика и компьютерные науки

Зав. кафедрой _____ д.физ.-мат.н., проф. С.Д. Кургалин

Обучающийся _____ Е.Ю. Колесникова

Руководитель _____ к.физ.-мат.н., доц. Н.П. Стадная

Воронеж 2025

Содержание

Введение	3
1 Генерация текстовых задач	5
1.1 Декорации	5
1.2 Склонения существительных	8
1.3 Проверка условий (генерация утверждений)	11
1.4 Разнообразие в текстовых задачах	17
2 Задачи №4 ЕГЭ (преобразование выражений)	24
2.1 Основные используемые команды	24
2.2 Адаптирование команд	26
2.3 Обратные задачи	31
3 Задачи №7 ОГЭ (координатная прямая)	35
3.1 разработка функции	35
3.2 Назначение функции	35
3.3 Интерфейс функции	36
3.4 Алгоритм работы	37
3.5 Пример использования	42
4 Задачи №7 ОГЭ (сравнение чисел)	49
Заключение	58

Введение

Единый государственный экзамен (ЕГЭ) — централизованно проводимый в Российской Федерации экзамен в средних учебных заведениях — школах, лицеях и гимназиях, форма проведения ГИА (Государственной Итоговой Аттестации) по образовательным программам среднего общего образования. Служит одновременно выпускным экзаменом из школы и вступительным экзаменом в вузы.

Но за время обучения в 9, 10 и 11 классе при подготовке к ОГЭ и ЕГЭ школьники сталкиваются с дефицитом заданий по определённым категориям. Так, за последние 5 лет в список заданий ЕГЭ были добавлены новые задания под номером 1 по теме «Округление с недостатком» и «Округление с избытком», так же задания под номером 15 «Проценты и округление», 21 задания «Текстовые задачи», количество которых для прорешивания было мало. Ко всему прочему в задании номер 7 по теме «Числовые неравенства, координатная прямая - числа на прямой» банк заданий расходуется при подготовке с невероятной скоростью: так как это преимущественно графические задания, решение их занимает менее минуты, а их составление вручную занимает несоразмерно много времени. ОГЭ и ЕГЭ является относительно неизменяемым экзаменом, поэтому все материалы, которые уже были выложены в открытый доступ, имеют полные решения, что приводит к списыванию учениками.

При этом существуют задания со вспомогательным чертежом. Чаще всего для целого ряда заданий используется одна и та же иллюстрация, которая не всегда соответствует условиям задачи, а иногда отвлекает от решения. Проект «Час ЕГЭ» позволяет решить все эти проблемы.

«Час ЕГЭ» — компьютерный образовательный OpenSource-проект, предназначенный для помощи учащимся старших классов при подготовке к тестовой части единого государственного экзамена.

Задания в «Час ЕГЭ» генерируются случайным образом по специализированным алгоритмам, называемых шаблонами, каждый из которых охватывает мно-

жество вариантов соответствующей ему задачи. Для пользователей предназначены четыре оболочки (режима работы): «Случайное задание», «Тесты на печать», «Полный тест» и «Мини-интеграция». «Час ЕГЭ» является полностью открытым (код находится под лицензией GNU GPL 3.0) и бесплатным. В настоящее время в проекте полностью реализованы тесты по ЕГЭ по математике профильного уровня с кратким ответом (бывшая «часть В»). [2] Планируется с течением времени включить в проект тесты по другим предметам школьной программы.

Первую главу посвятим обзору шаблонов для номеров 21,15, 1 из ЕГЭ базовой. Во второй главе рассмотрим функцию добавленному для упрощения отрисовки прямых для 7 задания ОГЭ.

1. Генерация текстовых задач

В Первом разделе представлены работы, связанные Текстовыми задачами. Так как в При построении текстовых шаблонов задач важно обеспечить как вариативность формулировок, так и корректность числовых и языковых выражений. Для этого в проекте применяются следующие инструменты.

1.1. Декорации

Чтобы увеличить количество уникальных вариантов задач, сохраняя их математическую суть, используются так называемые ``декорации'' --- элементы окружения, которые можно менять без потери смысла задачи: имена персонажей, место действия, цель, контекст. Для этого создаются массивы строк, а функция `iz()` случайным образом выбирает элемент из массива.

Пример:

```
1 let contract = ['о дружбе', 'во избежание двойного налогообложения',  
2   'о безвизовом режиме', 'об экологической среде',  
3   'по гуманитарным вопросам', 'по вопросам безопасности'].iz();
```

Задача №514913.

```
1 (function () {  
2   'use strict';  
3   retryWhileError(function () {  
4     NAinfo.requireApiVersion(0, 2);  
5  
6     let firstCountries = sl(3, 9, 1);  
7     let secondCountries = sl(3, 9, 1);  
8  
9     let otherCountriesFirst = firstCountries - 1;  
10    let otherCountriesSecond = secondCountries - 1;  
11  
12    let numberWords = ['три', 'четыре', 'пять', 'шесть', 'семь',
```

```

'восемь', 'девять'];
13     let previousNumberWords = ['две', 'три', 'четыре', 'пять', 'шесть',
'семь', 'восемь'];
14
15     let firstCountriesWord = lx[numberWords[firstCountries - 3]].i;
16     let secondCountriesWord = lx[numberWords[secondCountries - 3]].r;
17
18     let firstOtherCountriesWord =
lx[previousNumberWords[firstCountries - 3]].t;
19     let secondOtherCountriesWord =
lx[previousNumberWords[secondCountries - 3]].t;
20
21     let contract = ['о дружбе', 'во избежание двойного налогообложения', 'о
безвизовом режиме', 'об экологической среде', 'по гуманитарным вопросам',
22         'по вопросам безопасности'].iz();
23
24     let result = firstCountries * otherCountriesSecond +
secondCountries * otherCountriesFirst;
25
26     genAssert(result.kратно(2), "количество подписей должно быть кратно
двум");
27
28     NAtask.setTask({
29         text: 'Из ' + '$' + (firstCountries + secondCountries) + '$' +
' стран ' + firstCountriesWord + ' подписали договор ' +
30         contract + ' ровно с ' + secondOtherCountriesWord + '
другими странами, ' +
31         'а каждая из оставшихся ' + secondCountriesWord + ' - ровно с
' + firstOtherCountriesWord + '. ' +
32         'Сколько всего было подписано договоров?',
33         answers: result / 2,
34     });
35 });
36 })();
37 //https://mathb-ege.sdamgia.ru/problem?id=514913

```

```
38 //zer00player
```

Если необходимо выбрать два случайных, но не повторяющихся элемента, используется форма `iz(2)`:

```
1 let tapeName = sklonlxkand(['лента', 'верёвка', 'нитка'].iz(2));
```

Задача №2434.

```
1 (function () {
2     'use strict';
3     retryWhileError(function () {
4         NAinfo.requireApiVersion(0, 2);
5         let tapeName = sklonlxkand(['лента', 'верёвка', 'нитка'].iz());
6         let secondTapeLength = sluchch(5, 200, 1);
7         let firstTapeLength = sluchch(5, 200, 1);
8         let tapeColor = ['красн', 'жёлт', 'зелён', 'фиолетов', 'черн',
9 'оранжев', 'голуб'].iz(2);
10         let typeOfTapes = ['тонкие', 'незаметные', 'полупрозрачные',
11 'маленькие'].iz();
12         let unitsOfLength = sklonlxkand(['сантиметр', 'дециметр', 'метр',
13 'миллиметр', 'дюйм'].iz());
14         NATask.setTask({
15             text: 'На ' + tapeName.de + ' по разные стороны от середины
16 отмечены две ' + typeOfTapes + ' поперечные полоски: ' +
17             tapeColor[0] + 'ая' + ' и ' + tapeColor[1] + 'ая' + '. Если
18 разрезать ' + tapeName.ve + ' по ' + tapeColor[1] + 'ой' + ' полоске, то
19 одна часть ' +
20             ' будет на ' + chislitlx(secondTapeLength, unitsOfLength,
21 '$') + ' длиннее другой. Если разрезать ' + tapeName.ve + ' по ' +
22             tapeColor[0] + 'ой' + ' полоске, ' +
23             ' то одна часть будет на ' + chislitlx(firstTapeLength,
24 unitsOfLength, '$') + ' длиннее другой.' +
25             ' Найдите расстояние ( в ' + unitsOfLength.pm + ' ) между '
26 + tapeColor[1] + 'ой' + ' и ' + tapeColor[0] + 'ой' + ' полосками.',
27             answers: (secondTapeLength + firstTapeLength) / 2,
28         });
29     });
30 })
```

```

19     });
20  })();
21  //https://ege314.ru/20-zadachi-na-smekalku/reshenie-2434/
22  //zer00player

```

Чтобы каждый раз не задавать список декораций, у нас есть несколько заготовленных, которые используются от задачи к задаче. Например `om.meltov=` и они могут быть расписаны даже по нескольким падежам.

Например:

```

1  om.meltov={}
2  om.meltov.ie = ['фонарик', 'флакон шампуня', 'флэшка', 'компактдиск-',
    'сувенир', 'матрёшка', 'магнит на холодильник', 'сборник тестов для подготовки
    к ЕГЭ', 'тетрадь', 'учебник', 'цветочный горшок'];

```

В задаче же благодаря этому можно записать короче:

```

1  let item = sklonlxkand((om.meltov.ie).iz());

```

Задача №314120.

Как вы могли заметить, некоторые декорации являются словосочетанием, например "флакон шампуня", "цветочный горшок", "магнит на холодильник" и так далее. И для более корректной работы `.ie` и других падежей у нас есть словарь `lxsoch.js`. Однако недавно было выявлено, что если рядом со словосочетанием стоит число, то оно должно изменяться. например "3 цветочных горшка" или "4 флакона шампуня". Они очень схожи с родительным падежом множественного числа, но отличаются. Так что мною был добавлен в `lxsoch.js` `.r2`. Это падеж, в котором учтено изменения слова из-за числительного.

1.2. Склонения существительных

В силу особенностей русского языка слова должны корректно изменяться по падежам и числам. Для этого применяется функция `sklonlxkand`, позволяющая генерировать правильные словоформы. После выбора слова к нему можно обра-

таться по падежу и числу. Например, `.ie` означает именительный падеж, единственное число.

Пример использования:

```
1 let typeOfFlowerInVases =
2   sklonlxkand(['роза', 'гвоздника', 'ромашка', 'лилия',
3               'мак', 'ирис', 'лаванда', 'мимоза'].iz());
4 NAtask.setTask({
5   text: 'На прилавке цветочного магазина стоят три вазы с '
6         + typeOfFlowerInVases.tm + ': ' + vaseColor[0] + 'ая, '
7         + vaseColor[1] + 'ая, ' + vaseColor[2] + 'ая.'
8         + ' Слева от ' + vaseColor[2] + 'ой вазы '
9         + chislitlx(leftOfThirdVase, typeOfFlowerInVases, '$')
10        + ', справа от ' + vaseColor[0] + 'ой вазы '
11        + chislitlx(rightOfFirstVase, typeOfFlowerInVases, '$') + '. '
12        + 'Всего в вазах '
13        + chislitlx(allFlowerInVases, typeOfFlowerInVases, '$') + '. '
14        + 'Сколько ' + typeOfFlowerInVases.rm + ' в '
15        + vaseColor[1] + 'ой вазе?',
16   answers: secondVaseCountFlower,
17 });
```

Задача №515842.

```
1 (function () {
2   'use strict';
3   retryWhileError(function () {
4     NAinfo.requireApiVersion(0, 2);
5     let vaseColor = ['красн', 'жёлт', 'зелён', 'фиолетов', 'черн',
6                     'оранжев', 'голуб', 'изумруд', 'бежев', 'бирюзов', 'ал', 'лилов',
7                     'голуб'].iz(3);
8     let typeOfFlowerInVases = sklonlxkand(['роза', 'гвоздника',
9     'ромашка', 'лилия', 'мак', 'ирис', 'лаванда', 'мимоза'].iz());
10
11     let firstVaseCountFlower = sluchch(1, 30, 1);
12     let secondVaseCountFlower = slKrome([firstVaseCountFlower], 1, 30,
```

```

1);
10     let thirdVaseCountFlower = slKrome([firstVaseCountFlower,
secondVaseCountFlower], 1, 30, 1);
11
12     let leftOfThirdVase = firstVaseCountFlower + secondVaseCountFlower;
13     let rightOfFirstVase = thirdVaseCountFlower +
secondVaseCountFlower;
14     let allFlowerInVases = thirdVaseCountFlower +
secondVaseCountFlower + firstVaseCountFlower;
15
16     NATask.setTask({
17         text:
18             'На прилавке цветочного магазина стоят три вазы с ' +
typeOfFlowerInVases.tm + ': ' + vaseColor[0] + 'ая, ' +
19             vaseColor[1] + 'ая, ' + vaseColor[2] + 'ая.' + ' Слева от '
+ vaseColor[2] + 'ой ' + ' вазы ' + chislitlx(leftOfThirdVase,
typeOfFlowerInVases, '$') +
20             ', справа от ' + vaseColor[0] + 'ой ' + ' вазы ' +
chislitlx(rightOfFirstVase, typeOfFlowerInVases, '$') + '. ' +
21             'Всего в вазах ' + chislitlx(allFlowerInVases,
typeOfFlowerInVases, '$') + '. ' +
22             'Сколько ' + typeOfFlowerInVases.rm + ' в ' + vaseColor[1]
+ 'ой ' + 'вазе?',
23             answers: secondVaseCountFlower,
24         });
25
26     }, 100);
27 })();
28 //https://mathb-ege.sdamgia.ru/test?likes=515842
29 //zer00player

```

Для корректного согласования числительных с существительными используется функция `chislitlx`. Она автоматически выбирает нужную форму слова в зависимости от числа. Пример:

```

1 let numberOfApartamentPerFloor = sluchch(3, 12, 1);
2 NAtask.setTask({
3   text: 'В доме, в котором живет ' + nameOfPerson + ', ' +
4     '$' + floorNumber + '$ этажей и несколько подъездов. На каждом этаже
5     находится по ' +
6     chislitlx(numberOfApartamentPerFloor, 'квартира', '$') + '. '
7     + nameOfPerson + ' живет в квартире №' + '$' + apartamentNumber + '$' +
8     ' '
9     + 'В каком подъезде живет ' + nameOfPerson + '? ',
10  answers: '$' + (apartamentNumber /
11    (floorNumber * numberOfApartamentPerFloor)).ceil() + '$',
12 });

```

Задача №77351.

1.3. Проверка условий (генерация утверждений)

Помни во всех шаблонах вы так или иначе увидите окружение `retryWhileError`, которое ограничивает количество перезапусков, плюсом ещё и фиксирует какие проверки не были пройдены и выводить их на экран. Само собой ошибки видны только разработчикам Во время отладки. `function retryWhileError(theFunction, maxIterations, maxCollectedErrors)`

Пример:

```

1 (function () {
2   'use strict';
3   retryWhileError(function () {
4     NAinfo.requireApiVersion(0, 2);
5
6     let placeOfRest = ['пансионате', 'санатории', 'здравнице', 'курортном
7     доме', 'отеле', 'гостинице', 'доме отдыха'].iz();
8     let whoIsRest = ['дети', 'взрослые', 'пенсионеры', 'подростки'].iz();
9     let rand = sl(0, 6);

```

```

9      let theShareOfVacationers = ['Половина', 'Четверть', 'Одна пятая',
    'Десятая часть', 'Двадцатая часть', 'Одна двадцать пятая', 'Пятидесятая'][rand]
10     let theShareOfVacationersInNumbers = [0.5, 0.25, 0.2, 0.1, 0.05,
    0.04, 0.02][rand]
11
12     NATask.setTask({
13         text:
14             theShareOfVacationers + ' всех отдыхающих в ' + placeOfRest
    + ' - ' + whoIsRest + '. Какой процент от всех отдыхающих составляют ' +
    whoIsRest + '?',
15         answers: 100 * theShareOfVacationersInNumbers,
16     });
17 }, 100);
18 })();
19 //https://mathb-ege.sdamgia.ru/test?likes=509768
20 //zer00player

```

Для контроля корректности задачи применяются функции `genAssert` и её вариации. Принцип работы:

- если условие не выполнено --- фиксируется ошибка;
- шаблон перезапускается;
- если достигнуто максимальное количество перезапусков, выводятся накопившиеся ошибки с указанием количества.

Пример использования:

```

1 let students = sl(200, 10000, 10);
2 let percent = sl(10, 90, 1);
3 genAssert(students.kratno(100 / percent),
4     "количество учащихся не кратно процент100/");

```

Задача №77340.

Также часто используется функция `function genAssertZ1000(number, message)`, проверяющая точность чисел: если у числа более трёх знаков после запятой, шаб-

лон перезапускается. `function genAssertIrreducible(numerator, denominator, message)`

- проверка на несократимость дроби.

Пример:

```
1 let prise = sl(100, 2000, 10);
2 let percentSecondMonth = sl(10, 90, 1);
3 let percentThirdMonth = sl(10, 90, 1);
4
5 let middlePrise = prise * (1 + [1, -1][randFirst] * 0.01 *
   percentSecondMonth);
6 let finalePrise = middlePrise * (1 + [1, -1][randSecond] * 0.01 *
   percentThirdMonth);
7
8 genAssertZ1000(finalePrise / 10,
9   'Число имеет более 2 знаков после запятой');
```

Задача №77349.

```
1 (function () {
2   'use strict';
3   retryWhileError(function () {
4     NAinfo.requireApiVersion(0, 2);
5
6     let months = om.months;
7     let firstIndex = sl(0, months.length - 1, 1);
8     let secondIndex = (firstIndex + 1) % months.length;
9     let thirdIndex = (firstIndex + 2) % months.length;
10    let firstMonth = sklonlxkand(months[firstIndex]);
11    let secondMonth = sklonlxkand(months[secondIndex]);
12    let thirdMonth = sklonlxkand(months[thirdIndex]);
13
14    let food = sklonlxkand(['голубика', 'клубника', 'черника',
   'земляника', 'морозка', 'шелковица', 'брусника', 'малина', 'клюква', 'вишня',
   'черешня'], .iz());
15  });
16 }
```

```

16     let randFirst = sl1();
17     let randSecond = sl1();
18
19     let moreOrLessSecondMonth = ['дороже', 'дешевле'][randFirst];
20     let moreOrLessThirdMonth = ['дороже', 'дешевле'][randSecond];
21
22     let prise = sl(100, 2000, 10);
23     let percentSecondMonth = sl(10, 90, 1);
24     let percentThirdMonth = sl(10, 90, 1);
25     let middlePrise = prise * (1 + [1, -1][randFirst] * 0.01 *
percentSecondMonth);
26     let finalePrise = middlePrise * (1 + [1, -1][randSecond] * 0.01 *
percentThirdMonth);
27
28     genAssertZ1000(finalePrise / 10, 'Число имеет более 2 знаков после
запятой');
29
30
31
32     NATask.setTask({
33         text:
34             'В ' + firstMonth.pe,
35         questions: [
36             {
37                 text: ' один килограмм ' + food.re + ' стоил ' + prise +
' р., ' +
38                     'в ' + secondMonth.pe + ' ' + food.ie + ' стала '
+ moreOrLessSecondMonth + ' на ' + percentSecondMonth + '%, а в ' +
thirdMonth.pe +
39                     ' - ' + moreOrLessThirdMonth + ' на ' +
percentThirdMonth +
40                     '%. Сколько рублей стоил 1 кг ' + food.re + ' после
того как стал ' + moreOrLessThirdMonth + ' в ' + thirdMonth.pe,
41                 answers: finalePrise,
42             },

```

```

43         {
44             text: ' начали продавать на рынке ' + food.ve + '. В ' +
secondMonth.pe + ' ' + food.ie + ' стала ' + moreOrLessSecondMonth + '
на ' +
45             percentSecondMonth + '%, а в ' + thirdMonth.pe + '
- ' + moreOrLessThirdMonth + ' на ' + percentThirdMonth + '%. Цена за 1
кг ' + food.re +
46             ' в ' + thirdMonth.pe + ' стала составлять ' +
finalePrise + ' р. Сколько рублей стоил 1 кг ' + food.re + ' в ' +
firstMonth.pe,
47             answers: prise,
48         },
49     ],
50     postquestion: '?',
51 });
52     NATask.modifiers.allDecimalsToStandard();
53 }, 1000);
54 })();
55 //https://mathb-ege.sdamgia.ru/test?likes=77349
56 //zer00player

```

В некоторых случаях может применяться `slKrome`, он работает как `sluchch` - создает случайное число. Однако с условием что оно отличное от первого параметра (первый параметр может быть числом, строкой, массивом или функцией), второй и третий параметры - это диапазон генерации, а четвертый параметр - шаг (по умолчанию это 1). Пример:

```

1 let mlnRuble = slKrome([100], 10, 200, 1);

```

Задача №506346.

`kratno` является функцией, провищающее кратно ли данное число другому. Чаще всего используем в `genAssert` для проверки Пример: `let students = sl(200, 10000, 10); let percent = sl(10, 90, 1); genAssert(students.kratno(100 / percent), "количество учащихся не кратно 100/процент");` *Задача №77340.*

`NAtask.modifiers.allDecimalsToStandard` мы используем в случаях работы с десятичными числами и дробями, для более корректного расчёта java script формул. Это помогает избежать излишней точности.

Пример:

```
1 let difference = sl(2, 20, 1);
2     let minAngle = (360 / (2 * difference + 1));
3     let maxAngle = (360 / (difference + 2));
4
5     let result = maxAngle.floor() - minAngle.floor() - Number(maxAngle
6 % 1 == 0 minAngle % 1 == 0);
7
8     NAtask.setTask({
9         text: 'Три луча, выходящие из одной точки, разбивают плоскость на
10 $3$ ' +
11         'разных угла, измеряемых целым числом градусов. Наименьший
12 moreOrLess + 'ый угол в ' +
13         chislitlx(difference, 'раз', 'v$') + ' ' +
14 moreThanLessThan +
15         ' наименьший + lessOrMore + 'ый. Сколько значений может принимать
16 величина среднего угла?',
17         answers: result,
18     });
19     NAtask.modifiers.allDecimalsToStandard();
```

Задача №514918.

```
1 (function () {
2     'use strict';
3     retryWhileError(function () {
4         NAinfo.requireApiVersion(0, 2);
5         let rand = sl1();
6         let moreOrLess = ['бол', 'мен'][rand];
7         let lessOrMore = ['бол', 'мен'][1 - rand];
8         let moreThanLessThan = ['больше', 'меньше'][rand];
9     });
```



```

10     let difference = sl(2, 20, 1);
11     let minAngle = (360 / (2 * difference + 1));
12     let maxAngle = (360 / (difference + 2));
13
14     let result = maxAngle.floor() - minAngle.floor() - Number(maxAngle
% 1 == 0   minAngle % 1 == 0);
15
16     NAtask.setTask({
17         text: 'Три луча, выходящие из одной точки, разбивают плоскость на
$$$ ' +
18             'разных угла, измеряемых целым числом градусов. Наи' +
moreOrLess + 'ьший угол в ' +
19             chislitlx(difference, 'раз', 'v$') + ' ' +
moreThanLessThan +
20             ' наи' + lessOrMore + 'ьшего. Сколько значений может принимать
величина среднего угла?',
21         answers: result,
22     });
23     NAtask.modifiers.allDecimalsToStandard();
24     }, 1000);
25 })();
26 //https://mathb-ege.sdangia.ru/test?likes=514918
27 //zer00player

```

Три луча, выходящие из одной точки, разбивают плоскость на 3 разных угла, измеряемых целым числом градусов. Наибольший угол в 3 раза больше наименьшего. Сколько значений может принимать величина среднего угла?

До: После:

1.4. Разнообразие в текстовых задачах

И самое главное для текстовых задач, возможность их разнообразить. Дело в том что изначально даётся достаочно много условий в задаче,от чего можно написать альтернативные варианты задачи. Что помогает ученикам лучше запонимать не просто последовательность решения, а принцип. Чаше всего от изменений условия задачи на альтернативное сильно меняется и текст, по этому мы используем

questions. Он позволяет разбить на несколько заданий одно и для каждого написать свой ответ. Если же у них общее начало то можно записать его в первый text, а если одинаковое окончание то дописать его в postquestion.

Пример1:

```
1 (function () {
2     'use strict';
3     retryWhileError(function () {
4         NAinfo.requireApiVersion(0, 2);
5
6         let rand = sl1();
7
8         let clotherMR = sklonlxkand(['свитер', 'халат', 'плащ', 'дождевик',
9 'товар'].iz());
10        let clotherWR = sklonlxkand(['рубашка', 'футболка', 'куртка',
11 'кофта'].iz());
12        let clothers = [clotherMR, clotherWR][rand];
13
14        let cost = ['стоил', 'стоила'][rand];
15        let hasBecome = ['стал', 'стала'][rand];
16        let mw = ['', 'a'][rand];
17        let prise = sl(300, 5000, 10);
18        let percent = sl(10, 80, 1);
19
20        let result = (100 - percent) * 0.01 * prise;
21
22        NATask.setTask({
23            text:
24                clothers.ve.toZagl() + ' на распродаже уценили',
25            questions: [
26                {
27                    text: ' на ' + '$' + percent + '$' + '%, при этом он' +
28 mw + ' ' + hasBecome + ' стоить ' + '$' + result + '$' + ' р. ' +
29                    'Сколько рублей ' + cost + ' ' + clothers.ie + ' до
```

```

распродажи',
28         answers: prise,
29     },
30     {
31         text: ', при этом он' + mw + ' ' + hasBecome + ' стоит
' + '$' + result + '$' + ' р. ' + 'До распродажи ' + clothers.ie + ' ' +
cost +
32         ' ' + '$' + prise + '$' + ' р. Сколько процентов
составляет скидка',
33         answers: percent,
34     },
35     {
36         text: ' на ' + '$' + percent + '$' + '%.' + ' До
распродажи ' + clothers.ie + ' ' + cost + ' ' + '$' + prise + '$' +
37         ' р. Сколько рублей стоит ' + clothers.ie + ' во
время распродажи',
38         answers: result,
39     },
40
41 ],
42     postquestion: '?',
43 });
44     NATask.modifiers.allDecimalsToStandard();
45 }, 100);
46 })();
47 //https://mathb-ege.sdamgia.ru/test?likes=506426
48 //zer00player

```

Задача №506426.

Пример2:

```

1 (function () {
2     'use strict';
3     retryWhileError(function () {
4         NAinfo.requireApiVersion(0, 2);
5

```

```

6      let rand = sl1();
7      let anotherRand = sl(0, 4, 1);
8
9      let endPrilag = ['й-', 'ю-'][rand];
10     let anotherEndPrilagFirst = ['ый', 'ую'][rand];
11     let anotherEndPrilagSecond = ['ой', 'ую'][rand];
12     let endChislit = ['х', 'х', 'ти', 'ми', 'ти'][anotherRand];
13     let oneItem = ['один', 'одна'][rand];
14
15     let clotherMR = sklonlxkand(['свитер', 'халат', 'плащ',
'dождевик'].iz());
16     let clotherWR = sklonlxkand(['рубашка', 'футболка', 'куртка',
'кофтра'].iz());
17     let clothers = [clotherMR, clotherWR][rand];
18
19     let prise = sl(300, 5000, 10);
20     let percent = sl(10, 80, 1);
21     let countOfClother = sl(2, 4, 1);
22     let anotherCountOfClother = [2, 4, 6, 8, 10][anotherRand];
23
24     let result = (countOfClother - 1) * prise + (100 - percent) * 0.01
* prise;
25     let anotherResult = (anotherCountOfClother / 2) * prise +
(anotherCountOfClother / 2) * ((100 - percent) * 0.01 * prise);
26
27     NATask.setTask({
28         text:
29         'В спортивном магазине ' + ['любой', 'любая'][rand] + ' ' +
clothers.ie,
30         questions: [
31             {
32                 text: ' стоит ' + '$' + prise + '$' + 'р. ' +
33                 'Сейчас магазин проводит акцию: при покупке ' + '$' +
countOfClother + '$' + '-х ' + clothers.rm + ' - скидка на ' + '$' +
34                 countOfClother + '$' + endPrilag + ' ' +

```

```

clothers.ve + ' ' + '$' + percent + '$' + '%. ' +
35         'Сколько рублей придётся заплатить за покупку ' + '$'
+ countOfClother + '$' + '-x ' + clothers.rm + ' в период действия акции',
36         answers: result,
37     },
38     {
39         text: ' стоит ' + '$' + prise + '$' + 'p. ' +
40         'Сейчас магазин проводит акцию: при покупке ' +
clothers.rm +
41         ' - скидка на кажд' + anotherEndPrilagFirst + '
втор' + anotherEndPrilagSecond + ' составляет ' + '$' + percent + '$' +
'%.' +
42         'Сколько рублей придётся заплатить за покупку ' +
43         '$' + anotherCountOfClother + '$' + '-' +
endChislit + ' ' + clothers.rm + ' в период действия акции',
44         answers: anotherResult,
45     },
46     {
47         text: ' стоит ' + '$' + prise + '$' + 'p. ' +
48         'Сейчас магазин проводит акцию: при покупке ' + '$' +
countOfClother + '$' + '-x ' + clothers.rm + ' - будет скидка на ' + '$' +
49         countOfClother + '$' + endPrilag + ' ' +
clothers.ve + '. ' +
50         'Покупатель оплатил покупку ' + '$' + countOfClother
+ '$' + '-x ' + clothers.rm +
51         ' в период действия акции, что стоило ему ' + '$' +
result + '$' + 'p. ' + 'Сколько процентов составляет скидка',
52         answers: percent,
53     },
54     {
55         text: ' попадает под акцию: при покупке ' + '$' +
countOfClother + '$' + '-x ' + clothers.rm + ' - скидка на ' + '$' +
56         countOfClother + '$' + endPrilag + ' ' +
clothers.ve + ' ' + '$' + percent + '$' + '%. ' +
57         'Покупатель оплатил покупку ' + '$' + countOfClother

```

```

+ '$' + '-x ' + clothers.rm +
58         ' в период действия акции, что стоило ему ' + '$' +
result + '$' + 'p. ' +
59         'Сколько стоит ' + oneItem + ' ' + clothers.ie + '
без акции',
60         answers: prise,
61     },
62     {
63         text: ' стоит ' + '$' + prise + '$' + 'p. ' +
64         'Сейчас магазин проводит акцию: при покупке ' + '$' +
anotherCountOfClother + '$' + '-x ' + clothers.rm +
65         ' - будет скидка на кажд' + anotherEndPrilagFirst +
' втор' + anotherEndPrilagSecond + ' ' + clothers.ve + '. ' +
66         'Покупатель оплатил покупку ' + '$' +
anotherCountOfClother + '$' + '-' + endChislit +
67         ' ' + clothers.rm + ' в период действия акции, что
стоило ему ' + '$' + anotherResult + '$' + 'p. ' +
68         'Сколько процентов составляет скидка',
69         answers: percent,
70     },
71     {
72         text: ' попадает под акцию: при покупке ' + '$' +
anotherCountOfClother + '$' + '-x ' + clothers.rm +
73         ' - скидка на кажд' + anotherEndPrilagFirst + '
втор' + anotherEndPrilagSecond + ' составляет ' + '$' + percent + '$' +
'%. ' +
74         'Покупатель оплатил покупку ' + '$' +
anotherCountOfClother + '$' + '-' + endChislit + ' ' + clothers.rm +
75         ' в период действия акции, что стоило ему ' + '$' +
anotherResult + '$' + 'p. ' +
76         'Сколько стоит ' + oneItem + ' ' + clothers.ie + '
без акции',
77         answers: prise,
78     },
79 ],

```

```
80         postquestion: '?',
81     });
82     }, 100);
83 });
84 //https://mathb-ege.sdamgia.ru/test?likes=522673
85 //zer00player
```

Задача №522673.

2. Задачи №4 ЕГЭ (преобразование выражений)

2.1. Основные используемые команды

Задачи этого вида проверяют умение подставлять значения в формулы, преобразовывать выражения (в том числе со стандартными функциями и корнями), аккуратно работать с единицами измерения и корректно представлять результат. В нашем генераторе текст формулировок размечается *TeX*-фрагментами и рендерится через MathJax. При включённой опции `autoLaTeX` (см. ниже) *TeX*-формулы можно вставлять непосредственно в строку без явного окружения `$...$`.

Пример:

```
1 NAtask.setTask({
2   text:
3     'В фирме «' + name + '» стоимость в( рублях) колодца из железобетонных
   колец рассчитывается по формуле  $C = ' + plus + '+' + multiply +
4     ' \cdot n$ , где  $n$  - число колец, ' +
5     'установленных при рытье колодца. ' +
6     'Пользуясь этой формулой, ' +
7     'рассчитайте стоимость колодца из  $' + number + '$ колец.',
8   answers: cost,
9 });$ 
```

Задача №124.

```
1 (function () {
2   'use strict';
3   retryWhileError(function () {
4     NAinfo.requireApiVersion(0, 2);
5
6     let number = sl(4, 25);
7     let plus = sl(1000, 10000, 100);
8     let multiply = slKrome([plus], 1000, 10000, 100);
9     let cost = plus + multiply * number;
10    let name = ['Родник', 'Исток', 'Ручеёк', 'Веточка', 'Флажок', 'Куточка',
```



```
'КаменьиНожницы--', 'Стальная Красавица', 'Барон Суббота', 'Хозяйка
Малахитовой горы'].iz();
```

```
11
12   NATask.setTask({
13     text:
14       'В фирме «' + name + '» стоимость в( рублях) колодца из железобетонных
колец рассчитывается по формуле $C = ' + plus + '+' + multiply +
15       ' \cdot n$, где $n$ - число колец, ' +
16       'установленных при рытье колодца. ' +
17       'Пользуясь этой формулой, ' +
18       'рассчитайте стоимость колодца из $' + number + '$ колец.',
19     answers: cost,
20   });
21   NATask.modifiers.allDecimalsToStandard();
22 }, 2000);
23 })();
24 //https://oge.sdangia.ru/test?likes=124
25 //zer00player
```

```
1 NATask.setTask({
2
3   text: 'Количество теплоты в( джоулях), полученное однородным телом при
нагревании, ' +
4   'вычисляется по формуле $Q = cm(t_2 - t_1)$, где $c$ - удельная
теплоёмкость в( Дж), ' +
5   ' $m$ - масса тела в( килограммах), $t_1$ - начальная температура тела
в( кельвинах), а $t_2$ ' +
6   ' - конечная температура тела в( кельвинах). Пользуясь этой формулой, ' +
7   the_orderToFind + ' $Q$ в( джоулях), если $t_2 = ' + t_2 + '$ К, $c
= ' + c + '$ $\frac{\mbox{Дж}}{\mbox{кг}} \cdot \mbox{K}$,' +
8   ' $m = ' + m + '$ кг и $t_1 = ' + t_1 + '$ К.',
9   answers: Q,
10
11  });
```

Задача №509609.

`frac`,

`sqrt` и

`sin` самые часто используемые LaTeX команды и задачах вида 4, для отображения дробей, корней и `sin`.

Пример:

```
1 NAtask.setTask({
2     text: 'Радиус окружности, описанной около треугольника, ' +
3         'можно вычислить по формуле  $R = \frac{a}{2\sin{\alpha}}$ , где  $a$ 
- сторона, ' +
4         ' $a$   $\alpha$  - противолежащий ей угол треугольника. ' +
5         'Пользуясь этой формулой, ' + the_orderToFind + '  $R$ ,
a $\sin{\alpha}$  ' + '$' + ['R',
6         ', если  $a =$  ' + a, ' $R =$  ' + R][rand] + '$ и  $\sin{\alpha} =$ 
 $\frac{a}{2R}$  ' + num + '}{ ' + deNum + '}$.',
7         answers: [R, a][rand],
8         preference: preference,
9     });
```

Задача №506300.

2.2. Адаптирование команд

Чтобы уменьшить рукописный TeX и исключить мелкие огрехи, используются вспомогательные методы форматирования. Например `.texfrac()`. Удобная обёртка для дробей: на основе числителя/знаменателя или числового значения возвращает корректную TeX-строку вида $\frac{p}{q}$. Метод избавляет от ручной конкатенации строк и повышает читаемость шаблона. Пример:

```
1 NAtask.setTask({
2
3     text: 'Теорему синусов можно записать в виде  $\frac{a}{\sin{\alpha}}$ 
```

```

\\frac{a}{\\sin{\\alpha}} = \\frac{b}{\\sin{\\beta}} $' +
4     ', где $a$ и $b$ - две стороны треугольника, а $\\alpha$ и $\\beta$ -
углы треугольника, лежащие против них соответственно. ' +
5     ' Пользуясь этой формулой, ' + the_orderToFind + ' ' + '$' +
['\\sin{' + nameSin[0] + '}', nameLetter[0]][rand] + '$' +
6     ', если ' + '$' + [nameLetter[0] + ' =' + a, '\\sin{' + nameSin[0]
+ '}' = ' + sinA.texfrac(1)][rand] + '$' +
7     ', $' + nameLetter[1] + ' =' + b + '$, $\\sin{' + nameSin[1] + '}'
= ' + sinB.texfrac(1) + '$.',
8     answers: [sinA, a][rand],
9     preference: preference,
10
11     });

```

Задача №530329.

```

1 (function () {
2     'use strict';
3     retryWhileError(function () {
4         NAinfo.requireApiVersion(0, 2);
5
6         let key = '530329';
7         let preference = ['findSin', 'findSide'];
8         let rand = getSelectedPreferenceFromList(key, preference);
9
10        let the_orderToFind = decor.orderToFind.iz();
11
12        let a = sl(1, 30);
13        let b = slKrome([a], 1, 30);
14
15        let deNumA = sl(3, 50);
16        let numA = sl(1, deNumA - 1);
17        let sinA = numA / deNumA;
18
19        genAssertZ1000(sinA, 'sinA должно быть не более 3 знаков после запятой');
20

```

```

21 let sinB = (sinA * b) / a;
22
23 genAssertZ1000(sinB, 'sinB должно быть не более 3 знаков после запятой');
24 genAssert(sinB < 1, 'sinB не должен превышать 1');
25
26 let nameLetter = ['a', 'b'].shuffle();
27 let nameSin = ['\\alpha', '\\beta'].shuffle();
28
29 NATask.setTask({
30
31   text: 'Теорему синусов можно записать в виде $
\\frac{a}{\\sin{\\alpha}} = \\frac{b}{\\sin{\\beta}} $' +
32   ', где $a$ и $b$ - две стороны треугольника, а $\\alpha$ и $\\beta$ -
углы треугольника, лежащие против них соответственно. ' +
33   ' Пользуясь этой формулой, ' + the_orderToFind + ' ' + '$' +
['\\sin{' + nameSin[0] + '}', nameLetter[0]][rand] + '$' +
34   ', если ' + '$' + [nameLetter[0] + '=' + a, '\\sin{' + nameSin[0]
+ '}' = ' + sinA.texfrac(1)][rand] + '$' +
35   ', '$' + nameLetter[1] + '=' + b + '$, $\\sin{' + nameSin[1] + '}'
= ' + sinB.texfrac(1) + '$.',
36   answers: [sinA, a][rand],
37   preference: preference,
38
39 });
40 NATask.modifiers.allDecimalsToStandard();
41 }, 20000);
42 })();
43 //zer00player
44 //https://mathb-ege.sdamgia.ru/problem?id=530329

```

`.shuffle()` перемешивает массив (например, буквы параметров) для вариативности формулировок.

Пример:

```

1 let letter = ['a', 'b', 'c'].shuffle();

```

Задача №2939.

```
1 (function () {
2   'use strict';
3   retryWhileError(function () {
4     NAinfo.requireApiVersion(0, 2);
5
6     let the_orderToFind = decor.orderToFind.iz();
7
8     let a = sl(2, 50);
9     let b = slKrome([a], 2, 50);
10    let c = slKrome([a, b], 2, 50, 0.5);
11
12    let V = a * b * c;
13    let letter = ['a', 'b', 'c'].shuffle();
14
15    NATask.setTask({
16
17      text: 'Объём прямоугольного параллелепипеда вычисляется по формуле  $V =$ 
18      abc$, где $a$, $b$ и $c$ - длины трёх его рёбер, ' +
19      'выходящих из одной вершины. Пользуясь этой формулой, ' +
20      the_orderToFind + ' $' + letter[0] + '$, если $V = ' + V + '$, ' +
21      ' $' + letter[1] + ' = ' + b + '$ и $' + letter[2] + ' = ' + c +
22      '$.',
23      answers: a,
24
25    });
26    NATask.modifiers.allDecimalsToStandard();
27  }, 2000);
28})();
29//zer00player
30//https://self-edu.ru/ege2021_base_30.php?id=6_4
```

`.sqrt()` `.cbrt()` - утилиты для корней второй и третьей степеней. Удобны как при вычислении ответов, так и при генерации правдоподобных «отвлекающих»

значений. Так же `.rod` - определяет грамматический род существительного: 0 — м.р., 1 — ж.р., 2 — ср.р., 3 — всегда множественное. Это позволяет автоматически согласовать местоимения и прилагательные в задачах. Пример:

```
1 (function () {
2   'use strict';
3   retryWhileError(function () {
4     NAinfo.requireApiVersion(0, 2);
5
6     let rand = sl(0, 8);
7     let the_orderToFind = decor.orderToFind.iz();
8     let planet = sklonlxkand(['Меркурий', 'Венера', 'Земля', 'Луна', 'Марс',
9     'Юпитер', 'Сатурн', 'Уран', 'Нептун'][rand]);
10
11     let m = sl(1, 20, 0.1);
12     let h = slKrome([m], 1, 20, 0.1);
13     let g = [3.7, 8.9, 9.8, 1.6, 3.9, 23.9, 10.4, 8.9, 11][rand];
14     let E = m * h * g;
15
16     NATask.setTask({
17       text: 'Потенциальная энергия тела в( джоулях) в поле тяготения ' +
18       planet.re + ' вблизи ' + ['его', 'её'][planet.rod] + ' поверхности
19       вычисляется по формуле  $E = mgh$ , ' +
20       ' где  $m$  - масса тела в( килограммах),  $g$  - ускорение свободного
21       падения в(  $\text{м/с}^2$ ), а  $h$  - высота в( метрах), ' +
22       ' на которой находится это тело относительно поверхности. ' +
23       'Пользуясь этой формулой, ' + the_orderToFind + '  $m$  в( килограммах),
24       если  $g = ' + g + ' \text{м/с}^2$ ,  $h = ' + h + ' \text{м}$ , а  $E = ' + E +
25       '$ Дж.',
26       answers: m,$ 
```

```

26 } ) ( ) ;
27 //zer00player
28 //https://mathb-ege.sdamgia.ru/test?likes=506737

```

`isValidTriangle()` - из `flatten-shape-geometry 1.8.2`, из 3 переменных проверяет могут ли они составить треугольник. и выдают `true` - если является, и `false` -если нет Пример:

```

1  let a = sl(2, 30);
2    let b = sl(2, 30);
3    let c = sl(2, 30);
4
5    genAssert(isValidTriangle(a, b, c), 'Должно являться треугольником');

```

Задача №506550.

`.isZ()` проверка «число является целым». Удобно для отсеечения случаев, в которых формула даёт «некрасивый» ответ. Чае всего используется вместе с `genAssert`

Пример:

```

1  let second = sl(5, 30);
2    let amperage = sl(5, 30);
3    let voltage = sl(5, 30);
4    let resistance = slKrome([amperage, voltage, second], 5, 30);
5    let answer = [amperage ** 2 * resistance * second, voltage ** 2 *
second / resistance][rand];
6
7    genAssert(answer.isZ(), 'должно быть целым');

```

Задача №523098.

2.3. Обратные задачи

Многие шаблоны №4 имеют *парные* (обратные) постановки: в одном варианте требуется найти, например, R , в другом — a , в третьем — $\sin \alpha$ и т. п. Переключение таких вариантов реализовано через массив предпочтений `preference` и выбор

сценария.

Пример:

```
1 let preference = ['find_R', 'find_a', 'find_sin']; // обратные«» постановки
2 let key = '506300';
3 let rand = getListedPreference(
4   key,
5   preference.map((pref, idx) => ({ preference: pref, preferenceValue: idx
6     })),
7   sl(preference.length - 1)
8 );
9 if (preference[rand] === 'find_R') {
10   // генерируем данные так, чтобы удобно было находить R
11 } else if (preference[rand] === 'find_a') {
12   // обратная постановка: найти a
13 } else {
14   // найти sin(alpha)
15 }
```

Такой механизм позволяет:

- поддерживать несколько «зеркальных» формулировок одной темы;
- управлять частотой появления каждого типа задачи;
- проверять одни и те же навыки в разных направлениях (прямая/обратная задача).

```
1 (function () {
2   'use strict';
3   retryWhileError(function () {
4     NAinfo.requireApiVersion(0, 2);
5
6     let key = '506300';
7     let preference = ['findR', 'findA'];
```



```
8 let rand = getSelectedPreferenceFromList(key, preference);
9
10 let the_orderToFind = decor.orderToFind.iz();
11
12 let a = sl(2, 15);
13 let deNum = sl(3, 20);
14 let num = (1, deNum - 1);
15
16 let R = a / (2 * (num / deNum));
17 genAssertZ1000(R, 'Должно быть не более 3 знаков после запятой');
18
19 NAtask.setTask({
20   text: 'Радиус окружности, описанной около треугольника, ' +
21     'можно вычислить по формуле  $R = \frac{a}{2\sin{\alpha}}$ , где $a$
- сторона, ' +
22     '$a$ - противолежащий ей угол треугольника. ' +
23     'Пользуясь этой формулой, ' + the_orderToFind + ' $' + ['R',
'a'] [rand] + '$' +
24     ', если $' + ['a =' + a, 'R =' + R] [rand] + '$ и  $\sin{\alpha} =$ 
 $\frac{' + num + '}{' + deNum + '}$ $. ',
25     answers: [R, a] [rand],
26     preference: preference,
27 });
28 NAtask.modifiers.allDecimalsToStandard(true);
29 }, 2000);
30 })();
31 //zer00player
32 //https://mathb-ege.sdangia.ru/test?likes=506300
```

На рис. 1 представлены исходная и сгенерированные задачи по шаблону №506300.

Радиус окружности, описанной около треугольника, можно вычислить по формуле $R = \frac{a}{2 \sin \alpha}$, где a — сторона, а α — противолежащий ей угол треугольника. Пользуясь этой формулой, найдите радиус R , если $a = 12$ и $\sin \alpha = \frac{1}{3}$.

а)

Радиус окружности, описанной около треугольника, можно вычислить по формуле $R = \frac{a}{2 \sin \alpha}$, где a — сторона, а α — противолежащий ей угол треугольника. Пользуясь этой формулой, определите R , если $a = 15$ и $\sin \alpha = \frac{6}{7}$.

Радиус окружности, описанной около треугольника, можно вычислить по формуле $R = \frac{a}{2 \sin \alpha}$, где a — сторона, а α — противолежащий ей угол треугольника. Пользуясь этой формулой, вычислите a , если $R = 6$ и $\sin \alpha = \frac{2}{3}$.

б)

Рис. 1: Задачи, соответствующие шаблону 506300:

а) исходная; б) сгенерированные

3. Задачи №7 ОГЭ (координатная прямая)

3.1. разработка функции

Наиболее значимая часть работы — это разработка функций для визуализации координатной прямой и точек на ней. Была создана универсальная функция, позволяющая отображать засечки и подписи в разных режимах.

Одним из ключевых элементов реализации автоматической генерации заданий стала функция *coordAxis_ddrawMarkPoint*. Она предназначена для отрисовки различных типов меток на координатной оси и их подписей.

3.2. Назначение функции

Функция решает задачу визуализации точек и вспомогательных обозначений на оси, что является неотъемлемой частью заданий ОГЭ и ЕГЭ по математике. С помощью данной функции возможно изображать:

- закрашенные точки (``dot``),
- выколотые точки (``emptyDot``),
- засечки (``line``),
- отсутствие метки (``nothing``).

```
1 function coordAxisdrawMarkPoint(ct, coord, text, markForm, textPosition,  
   options = {}) {  
2 //@author : zer00player  
3 //coordAxisdrawMarkPoint функция, которая рисует отметку на координатной оси  
   точку(, выколотую точку, засечку или ничего) с подписью.  
4 /*  
5  * @param {CanvasRenderingContext2D} ct Контекст отрисовки.  
6  * @param {number} coord Координата по оси X, где нужно поставить отметку.  
7  * @param {string} text Подпись
```

```

* @param {'dot' | 'emptyDot' | 'line' | 'nothing'} markForm Форма метки: точка, выколота
точка, засечка, отсутствие метки.* @param 'underAxis' | 'overAxis' | 'onAxis'
textPosition Позиция текста относительно оси.* @param Object [options=]
Дополнительные параметры оформления.*/let font = "16px liberation sans", fillStyle
= om.secondaryBrandColors[0], strokeStyle = fillStyle, lineWidth = 1.5, textOffsetX =
options;/** @param string [options.font="16px liberationsans"] Шрифт подписи.*
@param string [options.fillStyle=om.secondaryBrandColors[0]] Цвет подписи и
заливки точки.* @param string [options.strokeStyle=fillStyle] Цвет засечки (по
умолчанию совпадает с fillStyle).* @param number [options.lineWidth=1.5] Толщина
линии засечки.* @param number [options.textOffsetX] смещение текста по X.*/let
prevFillStyle = ct.fillStyle;let prevStrokeStyle = ct.strokeStyle;let prevFont =
ct.font;let prevLineWidth = ct.lineWidth;/** @private* @type string prevFillStyle
сохраняет текущий цвет перед временной заменой* @type string prevStrokeStyle
сохраняет текущий цвет перед временной заменой* @type string prevFont сохраняет
текущий шрифт перед заменой* @type number prevLineWidth сохраняет текущую толщину
перез заменой*/ct.fillStyle = fillStyle;ct.strokeStyle = strokeStyle;ct.font =
font;ct.lineWidth = lineWidth;/** Устанавливает новые параметры рисования перед
отрисовкой объекта.* @param string fillStyle Цвет заливки (например, для текста
или точки).* @param string strokeStyle Цвет обводки (например, для засечек).*
@param string font Шрифт текста (например, "16px liberationsans").* @param number
lineWidth Толщина линий.*//** @param 'dot' | 'emptyDot' | 'line' | 'nothing' markForm
Форма метки: точка, выколота точка, засечка, отсутствие метки.switch (markForm)
case "dot":ct.drawFilledCircle(coord, 0, 5);break;case
"emptyDot":ct.drawFilledCircle(coord, 0, 5);ct.fillStyle =
"#ffffff";ct.drawFilledCircle(coord, 0, 3);ct.fillStyle = fillStyle;break; case
"line":ct.lineWidth = 1.5;ct.drawLine(coord, -6, coord, 6);break;case
"nothing":break;/** @param 'underAxis' | 'overAxis' | 'onAxis' textPosition Позиция
текста относительно оси.Под осью(для чёрточек), Над осью(для точек) и по линии
оси(после стрелки).switch (textPosition) case "underAxis":ct.fillText(text, coord
- (textOffsetX || 4), 20);break;case "overAxis":ct.fillText(text, coord -
(textOffsetX || 4), -10);break;case "onAxis":ct.fillText(text, coord +
(textOffsetX || 6), 5);break;ct.fillStyle = prevFillStyle;ct.strokeStyle =
prevStrokeStyle;ct.font = prevFont;ct.lineWidth = prevLineWidth;/** Возвращает
исходные параметры рисования после завершения работы функции.* @see prevFillStyle
возвращает исходный цвет* @see prevStrokeStyle возвращает исходный цвет* @see
prevFont возвращает исходный шрифт* @see prevLineWidth возвращает исходную
толщину*/});

```

3.3. Интерфейс функции

Функция имеет следующий набор параметров:

- `ct` --- графический контекст Canvas,
- `coord` --- координата по оси X,
- `text` --- подпись для метки,
- `markForm` --- форма метки: `dot`, `emptyDot`, `line`, `nothing`,
- `textPosition` --- расположение подписи: под осью (`underAxis`), над осью (`overAxis`), на оси (`onAxis`),

- `options` --- дополнительные параметры (шрифт, цвет текста, толщина линии, смещение).

3.4. Алгоритм работы

1. Сохраняются текущие параметры отрисовки (`fillStyle`, `strokeStyle`, `font`, `lineWidth`).
2. Устанавливаются новые параметры, переданные в `options`.
3. В зависимости от параметра `markForm` рисуется выбранный элемент:
 - точка --- закрашенный круг,
 - выколота точка --- окружность с заливкой белым цветом внутри,
 - засечка --- вертикальная черта,
 - отсутствие --- элемент не отрисовывается.
4. В зависимости от параметра `textPosition` подпись размещается под осью, над осью или на линии оси.
5. Восстанавливаются исходные параметры графического контекста.

`coordAxis_prepare`, что позволяет подготовить область для оси и рисует стрелку. И `coordAxis_drawAuto` она автоматически вычисляет масштаб оси и вызывает `coordAxis_drawMarkPoint` для всех точек.

Функция `coordAxis_prepare` выполняет подготовку холста для отрисовки горизонтальной координатной оси со стрелкой. Она:

- задаёт габариты рабочей области оси (`width`, `height`) и сохраняет их в контексте для последующего использования;
- вертикально центрирует ось (Ох) (смещение системы координат);
- настраивает стили (цвет линии и толщину) и рисует ось со стрелкой;

- бережно восстанавливает исходные графические параметры контекста.

Компонент рассчитан на дальнейшее использование вместе с `coordAxis_drawMarkPoint` и `coordAxis_drawAuto`.

```

1 function coordAxisprepare(ct, { width = 400, height = 100, strokeStyle =
   om.primaryBrandColors[0], lineWidth = 2 } = {}) {
2   /*
3    * coordAxisprepare подготавливает прямую со стрелкой к отрисовке.
4    * @param {CanvasRenderingContext2D} ct Контекст отрисовки.
5    * @param {Object} [params={}] Опции настройки оси.
6    * @param {number} [params.width=400] Ширина области.
7    * @param {number} [params.height=100] Высота области.
8    * @param {string} [params.strokeStyle=om.primaryBrandColors[0]] Цвет
   линии оси.
9    * @param {number} [params.lineWidth=2] Толщина оси.
10  */
11
12  ct.__coordAxisW = width;
13  ct.__coordAxisH = height;
14  /*
15   * @param {number} width Желаемая ширина координатной оси в пикселях.
16   * @param {number} height Желаемая высота координатной оси в пикселях.
17  */
18
19  let prevStroke = ct.strokeStyle;
20  let prevLineWidth = ct.lineWidth;
21  /*
22   * @private
23   * @type {string} prevStrokeStyle сохраняет текущий цвет перед временной
   заменой
24   * @type {number} prevLineWidth сохраняет текущую толщину через замену
25  */
26
27  ct.translate(0, height / 2);
28  /*

```

```

29  * Смещает систему координат: вертикально центрирует ось X.
30  * @param {number} x Горизонтальное смещение в( данном случае 0).
31  * @param {number} y Вертикальное смещение - половина высоты canvas.
32  * @this {CanvasRenderingContext2D}
33  */
34
35  ct.strokeStyle = strokeStyle;
36  ct.lineWidth = lineWidth;
37  /*
38  * @param {string} strokeStyle установка цвет линии оси.
39  * @param {number} lineWidth установка толщина оси.
40  */
41
42  ct.drawArrow(10, 0, width - 10, 0);
43  /*
44  * Рисует горизонтальную ось с направляющей стрелкой.
45  * @param {number} x1 Начальная координата оси.
46  * @param {number} y1 Начальная координата по оси Y (0).
47  * @param {number} x2 Конечная координата оси.
48  * @param {number} y2 Конечная координата по оси Y (0).
49  */
50
51  ct.strokeStyle = prevStroke;
52  ct.lineWidth = prevLineWidth;
53  /*
54  * Возвращает исходные параметры рисования после завершения работы функции.
55  * @see prevStrokeStyle
56  * @see prevLineWidth
57  */
58 };
59
60 function mathToCanvas(x, min, scale, x0) {
61     return x0 + (x - min) * scale;
62 };

```

ct графический контекст `CanvasRenderingContext2D`.

width ширина области оси в пикселях, по умолчанию (400).

height высота области в пикселях, по умолчанию (100).

strokeStyle цвет линии оси (интеграция со стилем проекта через `om.primaryBrandCol`).

lineWidth толщина линии оси, по умолчанию (2).

Функция явно сохраняет текущие значения **strokeStyle** и **lineWidth** в локальные переменные **prevStroke** и **prevLineWidth** и восстанавливает их к концу выполнения. Параметры `ct.__coordAxisW` и `ct.__coordAxisH` записываются в контекст как служебные метаданные — это упрощает доступ к габаритам при последующих рисованиях (например, при авторазметке меток).

Важно, что вызывается `ct.translate(0, height/2)`: система координат сдвигается на половину высоты вниз, чтобы ось (Ох) оказалась по центру холста. Этот сдвиг является *накопительным*; поэтому рекомендуется либо:

- вызывать `coordAxis_prepare` один раз в рамках одного цикла отрисовки, либо
- оборачивать работу в `ct.save() ... ct.restore()`, если требуется многократная подготовка в одном контексте.

`coordAxis_prepare` задаёт «сцену» — габариты, позицию оси и её визуальные атрибуты. Поверх этой сцены функции `coordAxis_drawMarkPoint` и `coordAxis_drawAutoLabel` размещают метки и подписи. Такое разделение обязанностей упрощает поддержку кода: изменения оформления оси не затрагивают логику генерации и размещения меток. Данная функция является частью связки:

Из особенностей можно подчеркнуть что функция поддерживает как закрашенные, так и выколотые точки, что позволяет формировать задания с открытыми и закрытыми интервалами. Так же она восстанавливает исходные параметры,

гарантирует корректную работу при множественной отрисовке. А так же есть возможность смещения текста по оси X , что помогает избежать наложений подписей.

```
1 function coordAxisdrawAuto(ct, {
2   points = [], // [{ value: число, label: строка, mark:
3     'dot' 'line' 'nothing', labelPos: 'overAxis' 'underAxis' 'onAxis' }]
4   margin = 20,
5   width = 400,
6   height = 100
7 } = {}) {
8   coordAxisprepare(ct, { width, height });
9
10  if (points.length === 0) {
11    return;
12  };
13
14  let values = points.map(p => p.value);
15  let min = values.minE();
16  let max = values.maxE();
17
18  if (min === max) {
19    min -= 1;
20    max += 1;
21  } else {
22    let padding = (max - min) * 0.05;
23    min -= padding;
24    max += padding;
25  }
26
27  let range = max - min;
28
29  let x0 = margin;
30  let x1 = width - margin;
31  let scale = (x1 - x0) / range;
32
33  for (let { value, label, mark, labelPos } of points) {
```

```

32     let x = mathToCanvas(value, min, scale, x0);
33     coordAxisdrawMarkPoint(ct, x, label, mark, labelPos);
34 }
35 };

```

3.5. Пример использования

// Отрисовка закрашенной точки A с подписью под осью

```
coordAxis_drawMarkPoint(ct, 100, "A", "dot", "underAxis");
```

// Отрисовка выколотой точки B с подписью над осью

```
coordAxis_drawMarkPoint(ct, 200, "B", "emptyDot", "overAxis");
```

Пример работы:

```

1 (function () {
2     'use strict';
3     retryWhileError(function () {
4         NAinfo.requireApiVersion(0, 2);
5
6         let x = sl(1, 9).pm();
7         // Абсолютные величины x и y должны не совпадать
8         // и быть различимы визуально без труда
9         let y = slKrome([x.abs(), (x+1).abs(), (x-1).abs()], 1, 9).pm();
10
11         let labels = window.smallLatinLetters.iz(2);
12         let label1 = labels[0];
13         let label2 = labels[1];
14
15         let paint1 = function (ct) {
16             coordAxis_drawAuto(ct, { points: [
17                 // Выставляем правый край насильно, чтобы при авторасчёте масштаба не
18                 терялась вариативность
19                 { value: 10, mark: "nothing" },

```

```

19      // Левый тоже
20      { value: -10, mark: "nothing" },
21      // Нолик с чёрточкой!
22      { value: 0, mark: "line", label: "0", labelPos: "underAxis"
23    },
24      // Сами точки
25      { value: x, mark: "dot", label: label1, labelPos: "overAxis"
26    },
27      { value: y, mark: "dot", label: label2, labelPos: "overAxis"
28    },
29    ] });
30  };
31
32  // строки + логическая проверка
33  let forms = [
34    [label1 + " + " + label2 + " > 0", x + y > 0],
35    [label1 + " + " + label2 + " < 0", x + y < 0],
36    [label1 + " - " + label2 + " > 0", x - y > 0],
37    [label1 + " - " + label2 + " < 0", x - y < 0],
38    [label2 + " - " + label1 + " > 0", y - x > 0],
39    [label2 + " - " + label1 + " < 0", y - x < 0],
40    [label1 + " \\cdot " + label2 + " > 0", x * y > 0],
41    [label1 + " \\cdot " + label2 + " < 0", x * y < 0],
42    [label1 + "^2 \\cdot " + label2 + " > 0", (x ** 2) * y > 0],
43    [label1 + "^2 \\cdot " + label2 + " < 0", (x ** 2) * y < 0],
44    [label2 + "^2 \\cdot " + label1 + " > 0", (y ** 2) * x > 0],
45    [label2 + "^2 \\cdot " + label1 + " < 0", (y ** 2) * x < 0]
46  ];
47
48  let rand = sl1();
49  let correctOrNot = ['верно', 'неверно'][rand];
50
51  let trueExprs = forms.filter(f => f[1]);
52  let falseExprs = forms.filter(f => !f[1]);

```

```

51   genAssert(trueExprs.length >= 3, "Мало истинных выражений");
52   genAssert(falseExprs.length >= 3, "Мало ложных выражений");
53
54   let correct = trueExprs.iz();
55   let wrong = falseExprs.iz();
56
57   NATask.setTask({
58     text: 'На координатной прямой отмечены числа. Какое из приведённых
утверждений для этих чисел ' + correctOrNot + '?',
59     answers: [correct[0], wrong[0]][rand],
60     wrongAnswers: [falseExprs.map(x => x[0]), trueExprs.map(x =>
x[0])][rand]
61   });
62   AtoB(3, {autoLaTeX: true});
63
64   chas2.task.modifiers.addCanvasIllustration({
65     width: 400,
66     height: 100,
67     paint: paint1,
68   });
69   }, 1000);
70 })();
71 //zer00player
72 //https://oge.sdangia.ru/test?likes=314802

```

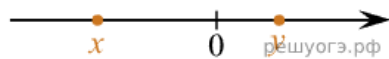
На рис. 2 представлены исходная и сгенерированные задачи по шаблону №314802.

На координатной прямой отмечены числа a и b . Укажите номера неверных утверждений.



- 1) $ab^2 < 0$ 2) $a - b > 0$ 3) $a + b < 0$ 4) $ab < 0$

На координатной прямой отмечены числа x и y .



Какое из приведенных утверждений для этих чисел **неверно**?

- 1) $xy < 0$ 2) $x^2y > 0$ 3) $x + y > 0$ 4) $x - y < 0$

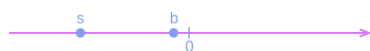
а)



На координатной прямой отмечены числа. Какое из приведённых утверждений для этих чисел верно?

- 1) $c^2 \cdot r < 0$
2) $r - c > 0$
3) $r - c < 0$

- 4) $c \cdot r < 0$



На координатной прямой отмечены числа. Какое из приведённых утверждений для этих чисел верно?

- 1) $b - s < 0$
2) $b \cdot s < 0$
3) $s - b < 0$

- 4) $s^2 \cdot b > 0$



На координатной прямой отмечены числа. Какое из приведённых утверждений для этих чисел неверно?

- 1) $g \cdot h > 0$
2) $h - g < 0$
3) $g^2 \cdot h < 0$

- 4) $h^2 \cdot g < 0$

б)

Рис. 2: Задачи, соответствующие шаблону 314802:

а) исходная; б) сгенерированные

Пример 2:

```

1 (function () {
2   'use strict';
3   retryWhileError(function () {
4     NAinfo.requireApiVersion(0, 2);
5
6     let borders = sl1() ? [-1, 0] : [0, 1];
7     let [left, right] = borders;
8
9     let outsideLeft = left - 0.5;
10    let outsideRight = right + 0.5;
11    let pointA = slKrome((x) => (x-x.round()).abs() < 0.1, outsideLeft,

```

```

outsideRight, 0.01);

12
13   let paint1 = function (ct) {
14       coordAxis_drawAuto(ct, { points: [
15           // Нолик и плюсминус- единичка с чёрточками!
16           { value: borders[0], mark: "line", labelPos: "underAxis", label:
borders[0] },
17           { value: borders[1], mark: "line", labelPos: "underAxis", label:
borders[1] },
18           // Сама точка a
19           { value: pointA, mark: "dot", labelPos: "overAxis", label: "a"
},
20           // И немного разбавляем края для вариативности
21           { value: borders[0] - sl(0.1,0.5,0.01), mark: "nothing" },
22           { value: borders[1] + sl(0.1,0.5,0.01), mark: "nothing" },
23       ] });
24   };
25
26   let pointA2 = pointA ** 2;
27   let pointA3 = pointA ** 3;
28   let pointA4 = pointA ** 4;
29
30   let randA = sl1();
31   let options = [[["$a$", pointA], ["$a^2$", pointA2], ["$a^3$",
pointA3]], [["$a^2$", pointA2], ["$a^3$", pointA3], ["$a^4$",
pointA4]]][randA];
32
33   let isMaxTaskOrMin = sl1();
34   let targetValue = [options.T(x => x[1])[1].maxE(), options.T(x =>
x[1])[1].minE())[isMaxTaskOrMin];
35   let maxOrMin = ['бол', 'мен'][isMaxTaskOrMin];
36
37   let correctOptions = options.filter(x => x[1] === targetValue);
38   let correct = correctOptions.length === 1 ? correctOptions[0][0] :
"нет данных";

```

```

39
40     let allLabels = options.map(x => x[0]);
41     allLabels.pushIf("нет данных", !allLabels.includes("нет данных"));
42
43     NAtask.setTask({
44         text: "На координатной прямой отмечены числа. Какое из перечисленных чисел
наи" + maxOrMin + "ьшее?",
45         answers: correct,
46         wrongAnswers: allLabels.filter(label => label !== correct)
47     });
48
49     AtoB(3, allLabels.indexOf(correct));
50
51     chas2.task.modifiers.addCanvasIllustration({
52         width: 400,
53         height: 100,
54         paint: paint1,
55     });
56     }, 1000);
57 })();
58 //zer00player
59 //https://oge.sdangia.ru/test?likes=337301

```

На рис. 3 представлены исходная и сгенерированные задачи по шаблону №337301.

На координатной прямой отмечено число a .

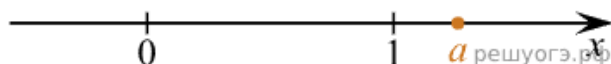


Найдите наибольшее из чисел a^2 , a^3 , a^4 .

В ответе укажите номер правильного варианта.

- 1) a^2
- 2) a^3
- 3) a^4
- 4) не хватает данных для ответа

На координатной прямой отмечено число a .



Найдите наибольшее из чисел a^2 , a^3 , a^4 .

В ответе укажите номер правильного варианта.

- 1) a^2
- 2) a^3
- 3) a^4
- 4) не хватает данных для ответа

а)



4) a^3

На координатной прямой отмечены числа. Какое из перечисленных чисел наибольшее?

- 1) a^2
- 2) нет данных
- 3) a^4



4) a

На координатной прямой отмечены числа. Какое из перечисленных чисел наименьшее?

- 1) a^2
- 2) a^3
- 3) нет данных

б)

Рис. 3: Задачи, соответствующие шаблону 337301:

а) исходная; б) сгенерированные

4. Задачи №7 ОГЭ (сравнение чисел)

В проекте присутствует отдельный шаблон, посвящённый задачам на сравнение действительных чисел. Основная цель подобных заданий --- научить учащегося ориентироваться в числовой прямой, дробях и приближённых значениях корней.

Так как в задачах встречаются как обыкновенные дроби, так и десятичные приближения квадратных корней, важно уметь корректно округлять результаты. Для этого используется метод `toFixed(1)`, позволяющий оставить одно десятичное число. Например:

```
1 let correctVal = ((frac1 + frac2) / 2);
2 let correct = correctVal.toFixed(1).ts();
```

Здесь мы берём среднее значение между двумя дробями, а затем округляем его до одного знака после запятой, чтобы получить корректный ответ, который предлагается ученику. Пример:

```
1 (function () {
2   'use strict';
3   retryWhileError(function () {
4     NAinfo.requireApiVersion(0, 2);
5
6     let labels = window.smallLatinLetters.iz(2);
7     let labelA = labels[0];
8     let labelB = labels[1];
9
10    let variant = sl1();
11    let rand = [variant, 1 - variant].iz();
12    let conditionText = labelA + [' < ', ' > '][variant] + labelB;
13    let reverseconditionText = labelA + [' < ', ' > '][1 - variant] +
14    labelB;
15
16    let minus = sl(1, 99);
```

```

16 let plus = slKrome([minus], 1, 99);
17 let denominatorMinus = slKrome([minus, plus], 1, 99);
18 let denominatorPlus = slKrome([minus, plus, denominatorMinus], 1, 99);
19
20 let wrongForm = '';
21 let correctForms = [
22     labelA + ' - ' + minus + [' < ', ' > '][variant] + labelB + ' - ' +
minus,
23     labelA + ' + ' + plus + [' < ', ' > '][variant] + labelB + ' + ' +
plus,
24     labelA.texfrac(denominatorPlus) + [' < ', ' > '][variant] +
labelB.texfrac(denominatorPlus),
25 ];
26 wrongForm = '-' + labelA.texfrac(denominatorMinus) + [' < -', ' >
-'][variant] + labelB.texfrac(denominatorMinus);
27
28 NAtask.setTask({
29     text: 'Какое из данных утверждений ' + ['не', ''][rand] + 'верно, если
$' + [conditionText, reverseconditionText][rand] + '$?',
30     answers: wrongForm,
31     wrongAnswers: correctForms
32 });
33
34 AtoB(3, { autoLaTeX: true });
35 }, 1000);
36 })();
37
38 //zer00player
39 //https://oge.sdangia.ru/problem?id=205843

```

На рис. 4 представлены исходная и сгенерированные задачи по шаблону №205843.

Для проверки знаний учащегося необходимо не только предъявить правильный ответ, но и сформировать несколько правдоподобных «ловушек». В нашем проекте это реализовано через генерацию трёх ложных ответов. Ложные вариан-

О числах a и c известно, что $a < c$. Какое из следующих неравенств неверно?

- 1) $a - 3 < c - 3$
- 2) $a + 5 < c + 5$
- 3) $\frac{a}{4} < \frac{c}{4}$
- 4) $-\frac{a}{2} < -\frac{c}{2}$

а)

Какое из данных утверждений неверно, если $t > j$?

- 1) $\frac{t}{58} > \frac{j}{58}$
- 2) $t + 56 > j + 56$
- 3) $t - 13 > j - 13$
- 4) $-\frac{t}{21} > -\frac{j}{21}$

Какое из данных утверждений неверно, если $b < l$?

- 1) $b + 37 < l + 37$
- 2) $-\frac{b}{94} < -\frac{l}{94}$
- 3) $\frac{b}{76} < \frac{l}{76}$
- 4) $b - 36 < l - 36$

Какое из данных утверждений верно, если $c < d$?

- 1) $\frac{c}{94} > \frac{d}{94}$
- 2) $-\frac{c}{38} > -\frac{d}{38}$
- 3) $c - 59 > d - 59$
- 4) $c + 63 > d + 63$

б)

Рис. 4: Задачи, соответствующие шаблону 205843:

а) исходная; б) сгенерированные

ты создаются с помощью небольших шумов (**noise**), добавляемых к правильному значению.

```
1 let wrong = new Set();
2 while (wrong.size < 3) {
3     let noise = slKrome([0], -7, 7) * 0.1;
4     let candidate = +(correctVal + noise).toFixed(1);
5
6     if (candidate <= 0 || candidate > frac1 && candidate < frac2
7         candidate === +correctVal.toFixed(1)) {
8         continue;
```

```

9     };
10    wrong.add(candidate.ts());
11 }

```

Таким образом, в итоговом задании всегда предлагается **4 варианта ответа**: один правильный и три ложных.

После задания параметров задачи и вариантов ответа вызывается функция `AtoB(3)`. Она отвечает за автоматическую генерацию списка вариантов с правильным ответом, расположенным случайным образом.

```

1 NAtask.setTask({
2   text: 'Какое из следующих чисел заключено между числами  $\${' + text1 + '}'\$$  и
    $\${' + text2 + '}'\$?$ ',
3   answers: correct,
4   wrongAnswers: Array.from(wrong)
5 });
6
7 AtoB(3, { autoLaTeX: true });

```

Чтобы не окружать каждую формулу знаками $\$ \dots \$$, используется параметр `{ autoLaTeX: true }`. Это позволяет сразу включать математические выражения (например, дроби и корни) прямо в текст задачи. В результате формулы корректно отображаются в интерфейсе без дополнительной ручной разметки.

В результате ученик видит задачу: «*Какое из следующих чисел заключено между числами ...?*», к которой автоматически предлагаются четыре варианта ответа. Примеры:

```

1 (function () {
2   'use strict';
3   retryWhileError(function () {
4     NAinfo.requireApiVersion(0, 2);
5
6     let key = "317132";
7     let preference = ['frac', 'square'];
8     let rand = getListedPreference(key, preference.map((pref, index) => ({

```

```

9     preference: pref,
10    preferenceValue: index
11  })), sl(preference.length - 1));
12
13  let countDrob = sl(1, 5);
14  let denominator = sl(2, 25);
15  let numerator = sl(1, denominator - 1);
16
17  let numDrob = countDrob * denominator + numerator;
18  let valueDrob = numDrob / denominator;
19
20  genAssert(!(valueDrob * 100).isAlmostInteger(), "дробь должна иметь 2
или более знака после запятой");
21
22  let exprStrDrob = numDrob.texfrac(denominator);
23
24  let countSqrt = sl(5, 25);
25  let numSqrt = countSqrt * countSqrt + denominator;
26
27  genAssert(!numSqrt.isPolnKvadr(), "корень не должен быть полным
квадратом");
28
29  let valueSqrt = numSqrt.sqrt();
30
31  let value = [valueDrob, valueSqrt][rand];
32  let step = [0.1, 1][rand];
33  let format = rand === 0 ? x => ((x * 10).round() / 10).ts() : x => x;
34
35  let start = Math.floor(value / step) * step;
36  let end = start + step;
37  let correct = `[${format(start)}; ${format(end)}]`;
38
39  let wrongAnswers = new Set();
40  wrongAnswers.add(correct);
41

```

```

42     let tries = 0;
43     while (wrongAnswers.size < 4 && tries < 50) {
44         let offset = slKrome([0], -3, 3);
45         let fakeStart = start + offset * step;
46         let fakeEnd = fakeStart + step;
47         if (fakeStart < 0) continue;
48
49         let fake = `[${format(fakeStart)}; ${format(fakeEnd)}]`;
50         wrongAnswers.add(fake);
51         tries++;
52     }
53     wrongAnswers.delete(correct);
54
55     NATask.setTask({
56         text: 'Какому из данных промежутков принадлежит число $' + [exprStrDrob,
57             '\\sqrt{' + numSqrt + '}'][rand] + '$? В ответе укажите номер правильного
58             варианта.',
59         answers: correct,
60         wrongAnswers: Array.from(wrongAnswers),
61         preference: preference,
62     });
63
64     AtoB(3, { autoLaTeX: true });
65 }, 1000);
66 }());
67 //zer00player
68 //https://oge.sdamgia.ru/test?likes=317132
69 //https://oge.sdamgia.ru/test?likes=317223

```

На рис. 5 представлены исходная и сгенерированные задачи по шаблону №317132.

```

1 (function () {
2     'use strict';
3     retryWhileError(function () {
4         NAinfo.requireApiVersion(0, 2);
5

```

Какому промежутку принадлежит число $\sqrt{27}$?
В ответе укажите номер правильного варианта.

- 1) [4; 5]
- 2) [5; 6]
- 3) [6; 7]
- 4) [7; 8]

Какому из данных промежутков принадлежит число $\frac{7}{11}$?

- 1) [0,4; 0,5]
- 2) [0,5; 0,6]
- 3) [0,6; 0,7]
- 4) [0,7; 0,8]

а)

Какому из данных промежутков принадлежит число $\frac{31}{6}$? В ответе укажите номер правильного варианта.

- 1) [5, 1; 5, 2]
- 2) [5; 5, 1]
- 3) [5, 4; 5, 5]
- 4) [5, 2; 5, 3]

Какому из данных промежутков принадлежит число $\frac{31}{6}$? В ответе укажите номер правильного варианта.

- 1) [5, 1; 5, 2]
- 2) [5; 5, 1]
- 3) [5, 4; 5, 5]
- 4) [5, 2; 5, 3]

б)

Рис. 5: Задачи, соответствующие шаблону 317132:

а) исходная; б) сгенерированные

```

6      let count = sl(0, 3);
7
8      let denominatorFirst = sl(2, 25);
9      let denominatorSecond = slKrome([denominatorFirst], 2, 25);
10     let numeratorFirst = sl(1, denominatorFirst - 1) + count *
denominatorFirst;
11     let numeratorSecond = sl(1, denominatorSecond - 1) + count *
denominatorSecond;
12
13     let frac1 = numeratorFirst / denominatorFirst;
14     let frac2 = numeratorSecond / denominatorSecond;
15
16     genAssert((frac1 - frac2).abs() >= 0.1, "разница между двумя значениями
должна составлять 0,1");
17

```

```

18     let text1 = numeratorFirst.texfrac(denominatorFirst);
19     let text2 = numeratorSecond.texfrac(denominatorSecond);
20
21     if (frac1 > frac2) {
22         [frac1, frac2] = [frac2, frac1];
23         [text1, text2] = [text2, text1];
24     }
25     let correctVal = ((frac1 + frac2) / 2);
26     let correct = correctVal.toFixed(1).ts();
27
28
29     let wrong = new Set();
30     while (wrong.size < 3) {
31         let noise = slKrome([0], -7, 7) * 0.1;
32         let candidate = +(correctVal + noise).toFixed(1);
33
34
35         if (candidate <= 0 || candidate > frac1 && candidate < frac2
candidate === +correctVal.toFixed(1)) {
36             continue
37         };
38
39         wrong.add(candidate.ts());
40     }
41
42     NATask.setTask({
43         text: 'Какое из следующих чисел заключено между числами ${' + text1 + '}$
и ${' + text2 + '}$? В ответе укажите номер правильного варианта.',
44         answers: correct,
45         wrongAnswers: Array.from(wrong)
46     });
47
48     AtoB(3, { autoLaTeX: true });
49 }, 1000);
50 })();

```



```
51  
52 //zer00player  
53 //https://oge.sdamgia.ru/test?likes=311420
```

На рис. 6 представлены исходная и сгенерированные задачи по шаблону №311420.

Какое из следующих чисел заключено между числами $\frac{18}{17}$ и $\frac{17}{15}$.

В ответе укажите номер правильного варианта.

- 1) 0,8
- 2) 0,9
- 3) 1
- 4) 1,1

а)

Какое из следующих чисел заключено между числами $\frac{3}{2}$ и $\frac{18}{11}$? В ответе укажите номер правильного варианта.

- 1) 1,6
- 2) 1,5
- 3) 1,9
- 4) 2,1

Какое из следующих чисел заключено между числами $\frac{1}{13}$ и $\frac{2}{7}$? В ответе укажите номер правильного варианта.

- 1) 0,4
- 2) 0,3
- 3) 0,2
- 4) 0,5

б)

Рис. 6: Задачи, соответствующие шаблону 311420:

а) исходная; б) сгенерированные

Заключение

В ходе выполнения курсовой работы за 3 курс был покрыт открытый банк заданий ФИПИ по темам:

- Текстовые задачи (на смекалку) — 12 шаблонов принято.
- Текстовые задачи (проценты и дроби) — 29 шаблонов принято.
- Преобразования выражений — 29 шаблонов (25 принято 4 на внутреннем рецензировании).
- Задачи с прямыми — 10 шаблонов принято. (3 с рисунком и 7 на сравнение чисел)

В ядро проекта добавлены:

- Функции, упрощающие написание шаблонов по теме «Координатная прямая».
- r_2 - род для числительных значений.

А также сокращён технический долг проекта.

Все добавленные в проект задания можно использовать для составления контрольных работ, проведения текущего контроля знаний учащихся, подготовки к ЕГЭ. [1]

В будущем планируется добавить в проект ещё большее количество заданий и функций для них д.

Список литературы

- [1] Тренажёр "Час ЕГЭ". – URL: <https://math.vsu.ru/chas-ege/sh/katalog.html>
- [2] Федеральный институт педагогических измерений. – URL: <https://fipi.ru/ege/otkrytyy-bank-zadaniy-ege>
- [3] Открытый банк задач ОГЭ по математике. – URL: <https://oge.sdamgia.ru/?Redir=1>
- [4] Открытый банк задач ЕГЭ по математике. Базовый уровень. – URL: <https://mathb-ege.sdamgia.ru>
- [5] Единый государственный экзамен. – URL: https://ru.wikipedia.org/wiki/Единый_государственный_экзамен
- [6] Решу ЕГЭ - Сдам ГИА. – URL: <https://ege.sdamgia.ru/problem?id=27074>