



ПРОГРАММНАЯ ГЕНЕРАЦИЯ ЗАДАНИЙ ОГЭ ПО ИНФОРМАТИКЕ



БЕЗУГЛОВ С.В.

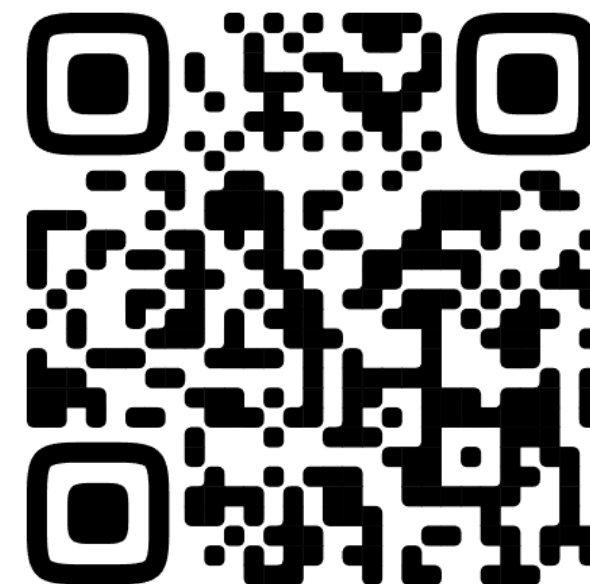
САМАРА

ГБНОУ СО «АКАДЕМИЯ ДЛЯ ОДАРЁННЫХ ДЕТЕЙ (НАЯНОВОЙ)»

ОСНОВНОЙ ГОСУДАРСТВЕННЫЙ ЭКЗАМЕН

ДемOVERсия

<https://fipi.ru/oge/demoversii-specifikacii-kodifikatory#!/tab/173801626-5>
<https://clck.ru/3jXizj>



РЕПОЗИТОРИИ ИСХОДНОГО КОДА

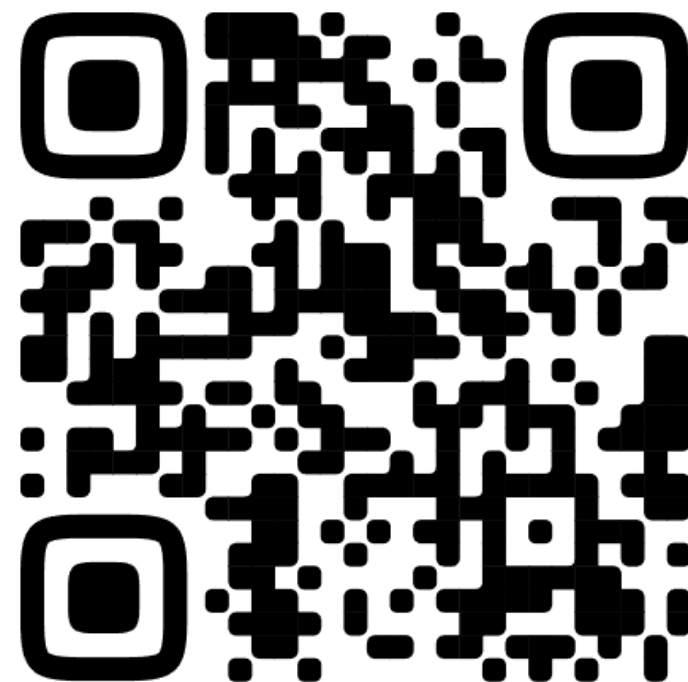
GitFlic

<https://gitflic.ru/project/besuglovs/oge>

GitHub

<https://github.com/BesuglovS/OGE>

<https://clck.ru/3JXjEQ>



ЗАДАНИЕ 0 I - СПЕЦИФИКАЦИЯ

Задание базового уровня сложности.

Задание проверяет умение оценивать объём памяти, необходимый для хранения текстовых данных.

Примерное время выполнения задания – **3 мин.**



ЗАДАНИЕ 01

- Списки
необходимые для
создания
вариантов были
сгенерированы
нейросетями.

```
var enc := [('Unicode', 8), ('Unicode', 16), ('Unicode', 24), ('Unicode', 32),  
('Windows-1251', 8), ('KOI8-R', 8), ('ASCII', 7),  
('UTF-7', 7), ('', 7), ('', 7), ('', 9), ('', 9), ('', 9)];  
  
var names := ['Александр записал', 'Михаил записал', 'Дмитрий записал',  
'Иван записал', 'Алексей записал', 'Андрей записал', 'Артем записал',  
'Никита записал', 'Максим записал', 'Сергей записал', 'Анна записала',  
'Мария записала', 'Анастасия записала', 'Елизавета записала',  
'Екатерина записала', 'Ольга записала', 'Дарья записала', 'Юлия записала',  
'Виктория записала', 'Полина записала'];
```

ЗАДАНИЕ 01

```
var items := Dict(  
  'городов' to Dict(  
    4 to ['Омск', 'Тула', 'Сочи'],  
    5 to ['Пенза', 'Курск', 'Тверь'],  
    6 to ['Брянск', 'Выборг', 'Калуга'],  
    7 to ['Воронеж', 'Саратов', 'Барнаул'],  
    8 to ['Тольятти', 'Оренбург', 'Кемерово'],  
    9 to ['Челябинск', 'Волгоград', 'Краснодар'],  
    10 to ['Красноярск', 'Ставрополь', 'Йошкар-Ола'],  
    11 to ['Новосибирск', 'Владивосток', 'Новокузнецк'],  
    12 to ['Екатеринбург', 'Первоуральск', 'Долгопрудный'],  
    13 to ['Багратионовск', 'Дальнереченск', 'Малоярославец'],  
    14 to ['Вышний Волочек', 'Краснотурьинск', 'Новочебоксарск'],  
    15 to ['Малоархангельск', 'Нижний Новгород', 'Санкт-Петербург'],  
    16 to ['Великий Новгород', 'Гусь-Хрустальный', 'Набережные Челны'],  
    17 to ['Каменск-Уральский', 'Ленинск-Кузнецкий', 'Новоалександровск'])),
```

ЗАДАНИЕ 01

```
'стран' to Dict(  
3 to ['ОАЭ', 'США', 'ЦАР'],  
4 to ['Иран', 'Куба', 'Лаос'],  
5 to ['Дания', 'Индия', 'Китай'],  
6 to ['Греция', 'Грузия', 'Италия'],  
7 to ['Армения', 'Венгрия', 'Украина'],  
8 to ['Беларусь', 'Болгария', 'Бразилия'],  
9 to ['Австралия', 'Венесуэла', 'Казахстан'],  
10 to ['Доминикана', 'Коста-Рика', 'Кыргызстан'],  
11 to ['Лихтенштейн', 'Таджикистан', 'Южный Судан'],  
12 to ['Буркина-Фасо', 'Гвинея-Бисау', 'Туркменистан']),
```

ЗАДАНИЕ 01

```
'греческих букв' to Dict(  
2 to ['мю', 'ню', 'пи'],  
3 to ['кси', 'тау', 'пси'],  
4 to ['бета', 'тета', 'йота'],  
5 to ['альфа', 'гамма', 'сигма'],  
6 to ['дельта', 'лямбда'],  
7 to ['эпсилон', 'омикрон', 'ипсилон']),
```

```
'ископаемых' to Dict(  
4 to ['Торф', 'Цинк', 'Медь'],  
5 to ['Битум', 'Олово', 'Алмаз'],  
6 to ['Свинец', 'Золото', 'Янтарь'],  
7 to ['Платина', 'Изумруд', 'Аметист'],  
8 to ['Вольфрам', 'Молибден', 'Палладий'],  
9 to ['Известняк', 'Сапропель', 'Аквамарин'],  
10 to ['Пирофиллит', 'Вермикулит'],  
11 to ['Александрит', 'Борная руда', 'Серная руда']),
```

ЗАДАНИЕ 01

```
'междометий' to Dict(  
2 to ['Ах', 'Ох', 'Эх'],  
3 to ['Ура', 'Ого', 'Увы'],  
4 to ['Тьфу', 'Пока', 'Хось'],  
5 to ['Браво', 'Ладно', 'Мерси'],  
6 to ['Отнюдь', 'Ничуть', 'Пардон'],  
7 to ['Классно', 'Здорово', 'Ну и ну'],  
8 to ['Боже мой', 'Приветик', 'Извините'],  
9 to ['Ну и дела', 'Абсолютно', 'Благодарю'],  
10 to ['Вот это да', 'Ну надо же', 'Безусловно'],  
11 to ['Ничего себе', 'Признателен', 'Спасибо вам'],  
12 to ['Здравствуйте', 'Добрый вечер', 'Вот незадача'],  
13 to ['Какая жалость', 'Вот тебе и на', 'Благодарствую'],  
14 to ['Да что ж такое', 'Прошу прощения', 'Мне очень жаль'],  
15 to ['Вот так сюрприз', 'Вот так история', 'Спасибо большое'])),
```

ЗАДАНИЕ 01

```
'прилагательных' to Dict(  
5 to ['малая', 'новая', 'южная'],  
6 to ['единая', 'родная', 'вторая'],  
7 to ['сборная', 'царская', 'деловая'],  
8 to ['западная', 'северная', 'читающая'],  
9 to ['восточная', 'советская', 'передовая'],  
10 to ['живописная', 'варварская', 'необъятная'],  
11 to ['современная', 'независимая', 'театральная'],  
12 to ['криминальная', 'справедливая', 'литературная'],  
13 to ['дружественная', 'монархическая', 'революционная'],  
14 to ['патриархальная', 'могущественная', 'провинциальная'],  
15 to ['демократическая', 'крепостническая', 'непредсказуемая'],  
16 to ['северо-восточная', 'иллюстрированная', 'социалистическая'],  
17 to ['капиталистическая', 'предреволюционная', 'многонациональная'])),
```

ЗАДАНИЕ 01

```
'произведений' to Dict(  
4 to ['"Ши"', '"Юг"', '"Мы"',  
5 to ['"Нос"', '"Вий"', '"Ася"',  
6 to ['"Бесы"', '"Мать"', '"Нева"',  
7 to ['"Левша"', '"Чайка"', '"Мцыри"',  
8 to ['"Шинель"', '"Пророк"', '"Родина"',  
9 to ['"Обломов"', '"Полтава"', '"Явление"',  
10 to ['"Женитьба"', '"Маскарад"', '"В дороге"',  
11 to ['"Палата №6"', '"Дядя Ваня"', '"Тихий Дон"',  
12 to ['"Три сестры"', '"После бала"', '"Пролетарий"',  
13 to ['"Война и мир"', '"Горе от ума"', '"Отцы и дети"',  
14 to ['"Мертвые души"', '"Тарас Бульба"', '"Темные аллеи"',  
15 to ['"Доктор Живаго"', '"Белая гвардия"', '"Анна Каренина"',  
16 to ['"Собачье сердце"', '"Евгений Онегин"', '"Медный всадник"',  
17 to ['"Золотой теленок"', '"Дама с собачкой"', '"Архипелаг ГУЛАГ"',  
18 to ['"Записки охотника"', '"Руслан и Людмила"', '"Донские рассказы"',  
19 to ['"Братья Карамазовы"', '"Крейцера соната"', '"Капитанская дочка"',  
20 to ['"Мастер и Маргарита"', '"Кавказский пленник"', '"А зори здесь тихие"',  
21 to ['"Записки из подполья"', '"Записки юного врача"', '"Жизнь Клима Самгина"',  
22 to ['"Обыкновенная история"', '"Герой нашего времени"', '"Божественная комедия"']),
```

ЗАДАНИЕ 01

```
var res := new List<(string, string)>();  
for var i := 1 to count do  
begin  
    var encoding := enc[Random(enc.Length)];  
    var name := names[Random(names.Length)];  
    var del_name := name.Replace('записал', 'удалил');  
  
    var task_type := items.Keys.toArray[Random(items.Count)];  
    var answ_vars := items[task_type].Select(x -> x.Value[Random(x.Value.Count)]).toArray;  
    var answ_vars_str := answ_vars.JoinToString(', ');  
    var answ := answ_vars[Random(answ_vars.Length)];  
    var answ_num := answ.Length;  
    var data_size := (answ_num + 2) * encoding[1];  
    var lb := NewLine;
```

ЗАДАНИЕ 01

```
var task := '';

task += $'В одной из кодировок {encoding[0]} каждый символ кодируется {encoding[1]} битами.'
task += $'При подготовке сообщения {name} текст (в нём нет лишних пробелов):' + lb;
task += $'У нас есть список {task_type}: {answ_vars_str}.' + lb;
task += $'Затем {del_name} из списка одно из названий, ' +
    'а также лишнюю запятую и пробел - два пробела не должны идти подряд.' + lb;
if data_size mod 8 = 0 then
    task += $'При этом размер нового сообщения в данной кодировке ' +
        '$оказался на {data_size div 8} байт меньше, чем размер исходного сообщения.' + lb
else
    task += $'При этом размер нового сообщения в данной кодировке ' +
        '$оказался на {data_size} бит меньше, чем размер исходного сообщения.' + lb;
task += $'Напишите в ответе удалённый элемент.';
res.Add((task, answ));
```

ЗАДАНИЕ 02 - СПЕЦИФИКАЦИЯ

Задание базового уровня сложности.

Задание проверяет умение декодировать кодовую последовательность.

Примерное время выполнения задания – **4 мин.**

ЗАДАНИЕ 02

Для этого задания я взял слова из произведения «Война и мир» Льва Николаевича Толстого и отобрал слова из 6, 7, 8 и 9 букв.

Список перемешивается и берётся первые случайные варианты.



ЗАДАНИЕ 02

```
foreach var w in wds do
begin
    var w2 := w.Distinct.ToArray.Aggregate('', (x, y) -> x + y);
    var code_length := Random(w2.Length + 2, w2.Length + 5);
    while w2.Length < code_length do
    begin
        var lst := ('a'..'я').Where(c -> c not in w2).ToArray;
        w2 := w2 + lst[Random(lst.Length)];
    end;
    var Codes := HuffmanEncode(w2);
    var task := 'От разведчика было получено сообщение ';
    task += EncodeString(w, Codes) + '.' + lb;
    task += 'В сообщении зашифрован пароль - последовательность русских букв.' + lb;
    task += 'В коде есть только следующие буквы - ' + CodesToStringAlpha(Codes) + '.' + lb;
    task += 'Расшифруйте пароль и запишите его в ответ.';
    res.add((task, w));
end;
```

ЗАДАНИЕ 02

```
function HuffmanEncode(const S: string): TCodeDictionary;  
begin  
    var FrequencyTable := BuildFrequencyTable(S);  
    var HuffmanTree := BuildHuffmanTree(FrequencyTable);  
    Result := TCodeDictionary.Create;  
    GenerateCodes(HuffmanTree, '', Result);  
end;  
  
function BuildFrequencyTable(s: string) := s.EachCount;
```

ЗАДАНИЕ 02

```
function BuildHuffmanTree(FrequencyTable: Dictionary<Char, Integer>): TNode;
begin
    var PriorityQueue := new Queue<TNode>();
    foreach var Key in FrequencyTable.Keys do
    begin
        var Frequency := FrequencyTable[Key];
        PriorityQueue.Enqueue(TNode.Create(Key, Frequency, nil, nil));
    end;

    while PriorityQueue.Count > 1 do
    begin
        var Node1 := PriorityQueue.Dequeue;
        var Node2 := PriorityQueue.Dequeue;
        var NewNode := TNode.Create(#0, Node1.Frequency + Node2.Frequency, Node1, Node2);
        PriorityQueue.Enqueue(NewNode);
    end;

    Result := PriorityQueue.Dequeue;
end;
```

ЗАДАНИЕ 02

```
procedure GenerateCodes(Node: TNode; const Prefix: string; var Codes: TCodeDictionary);
begin
    if Node.Symbol <> #0 then
        Codes.Add(Node.Symbol, Prefix)
    else
        begin
            GenerateCodes(Node.Left, Prefix + '0', Codes);
            GenerateCodes(Node.Right, Prefix + '1', Codes);
        end;
    end;
end;
```

ЗАДАНИЕ 03 - СПЕЦИФИКАЦИЯ

Задание базового уровня сложности.

Задание проверяет умение определять истинность составного высказывания.

Примерное время выполнения задания – **3 мин.**

ЗАДАНИЕ 03

```
function Eval(expression: string; x: integer): string;  
begin  
    expression := expression.Replace('x', x.ToString);  
    var table := new System.Data.Datatable();  
    result := table.Compute(expression, string.Empty).ToString;  
end;
```

```
{$reference System.Data.dll}
```

ЗАДАНИЕ 03

```
function RandomLogical(mode: integer := -1): string;
// mode = 1 - signs
// mode = 2 - %
begin
    if mode = -1 then
        loop 100 do mode := Random(2);
    var signs := ['>', '>=', '<', '<='];
    if mode = 1 then
        result := $('(x {signs[Random(signs.Length)]} {Random(10, 300)})'
    else
        begin
            var mode2 := 0;
            loop 100 do mode2 := Random(2);
            var sign := if mode2 = 0 then '=' else '<>';
            result := $('(x % {Random(2, 20)} {sign} 0)';
        end;
    end;
end;
```

ЗАДАНИЕ

```
function RandExpr(): string;
begin
    var exprs := [RandomLogical(1), RandomLogical, RandomLogical];
    var ts := ['e1 and e2', 'e1 or e2', 'not e1 and e2', 'not e1 or e2',
               'e1 and e2', 'e1 or e2', 'not e1 and e2', 'not e1 or e2',
               'e1 and e2', 'e1 or e2', 'not e1 and e2', 'not e1 or e2',
               'e1 and e2', 'e1 or e2', 'not e1 and e2', 'not e1 or e2',
               'e1 and e2', 'e1 or e2', 'not e1 and e2', 'not e1 or e2'];
    // 'e1 and e2 and e3', 'e1 and e2 or e3', 'e1 or e2 and e3', 'e1 or e2 or e3',
    // 'not e1 and e2 and e3', 'e1 and not e2 and e3', 'e1 and e2 and not e3'];
    // 'not e1 and e2 or e3', 'e1 and not e2 or e3', 'e1 and e2 or not e3',
    // 'not e1 or e2 and e3', 'e1 or not e2 and e3', 'e1 or e2 and not e3',
    // 'not e1 or e2 or e3', 'e1 or not e2 or e3', 'e1 or e2 or not e3'

    result := ts[Random(ts.Length)]
               .Replace('e1', exprs[0]).Replace('e2', exprs[1]).Replace('e3', exprs[2]);
end;
```

ЗАДАНИЕ 03

```
function ReplaceOps(s: string): string;  
Begin  
    s := RegEx.Replace(s, '% (\d+) = 0', 'делится на $1');  
    s := RegEx.Replace(s, '% (\d+) <> 0', 'не делится на $1');  
    s := s.Replace('and', 'и');  
    s := s.Replace('or', 'или');  
    s := s.Replace('not', 'не');  
    result := s;  
end;
```

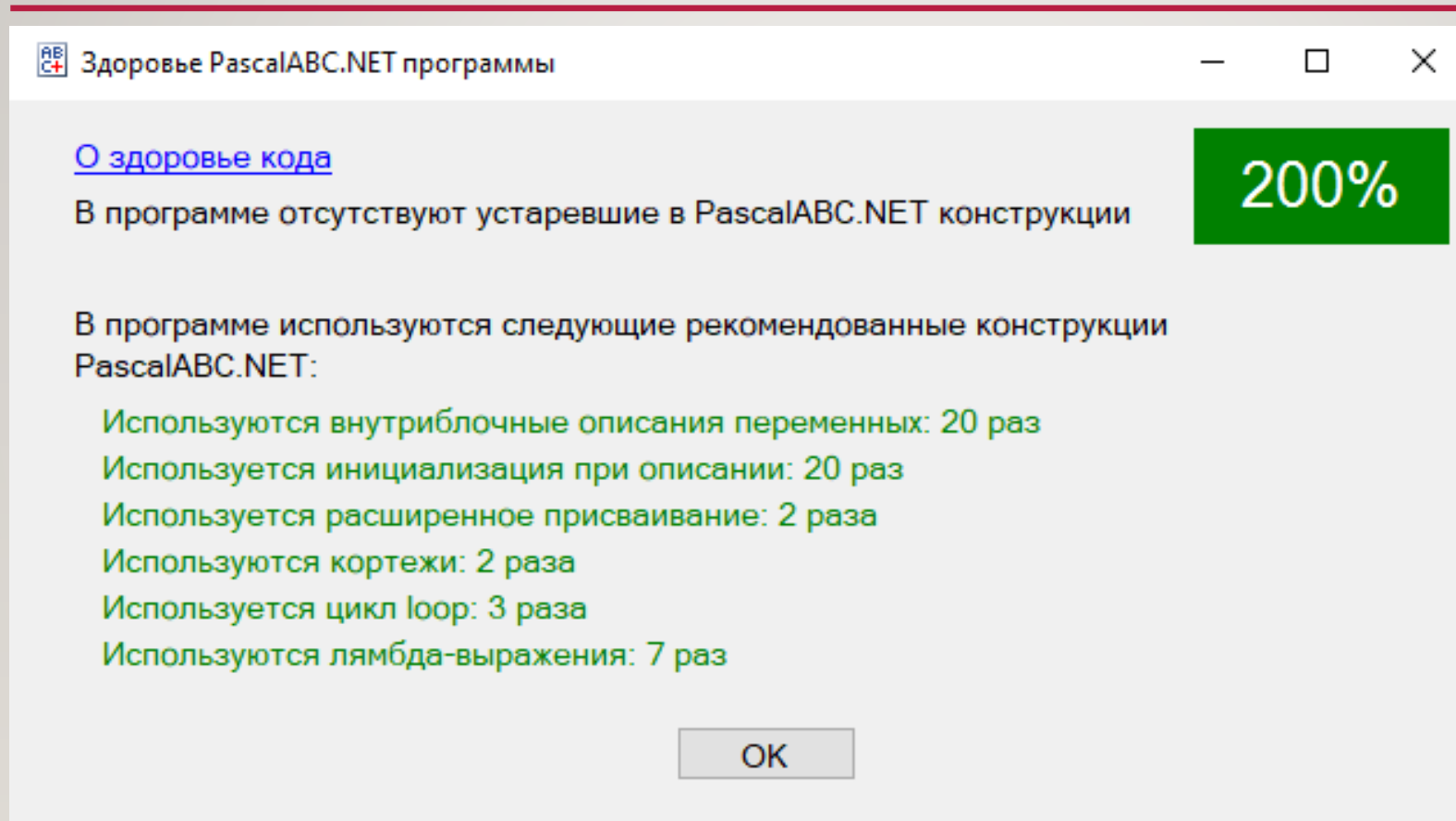
ЗАДАНИЕ 03

```
var k := 0;
while k < count do
begin
  var expr := RandExpr();
  var tf := 0;
  loop 100 do tf := Random(2);
  var orig_expr := expr;
  if tf = 0 then
    expr := $('not ({expr})');
  if Range(1, 300).ToArray.Count(x -> Eval(expr, x) = 'True') > 0 then
  begin
    var tf_str := if tf = 0 then 'ЛОЖНО' else 'ИСТИННО';
    var modes := [0, 1]; // 0 - min; 1 - max
    if Range(300, 1000).ToArray.Count(x -> Eval(expr, x) = 'True') > 10 then
      modes := modes.Where(x -> x <> 1).ToArray;
    if Range(-1000, 0).ToArray.Count(x -> Eval(expr, x) = 'True') > 10 then
      modes := modes.Where(x -> x <> 0).ToArray;
    var mode2 := -1;
    if modes.Contains(1) then
      mode2 := 1
    else if modes.Contains(0) then
      mode2 := 0;
```

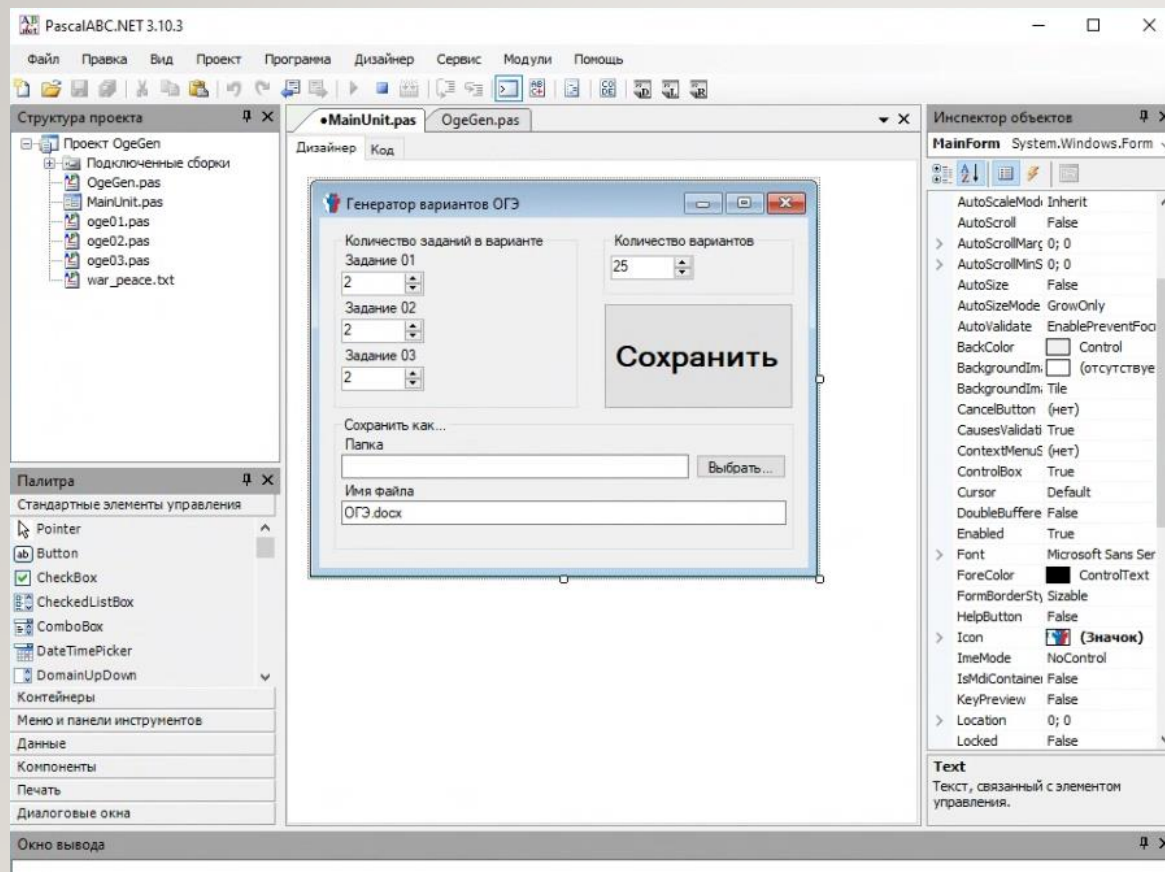
ЗАДАНИЕ 03

```
if mode2 = 0 then
begin
  var data := Range(1, 300).ToArray.Where(x -> Eval(expr, x) = 'True');
  var answ := data.Min;
  orig_expr := ReplaceOps(orig_expr);
  var task := 'Напишите наименьшее натуральное число x, ' +
    '$'для которого {tf_str} высказывание {orig_expr}.';
  res.add((task, answ.ToString));
  k += 1;
end;
if mode2 = 1 then
begin
  var data := Range(1, 300).ToArray.Where(x -> Eval(expr, x) = 'True');
  var answ := data.Max;
  orig_expr := ReplaceOps(orig_expr);
  var task := '$Напишите наибольшее натуральное число x, ' +
    '$'для которого {tf_str} высказывание {orig_expr}.';
  res.add((task, answ.ToString));
  k += 1;
end;
```

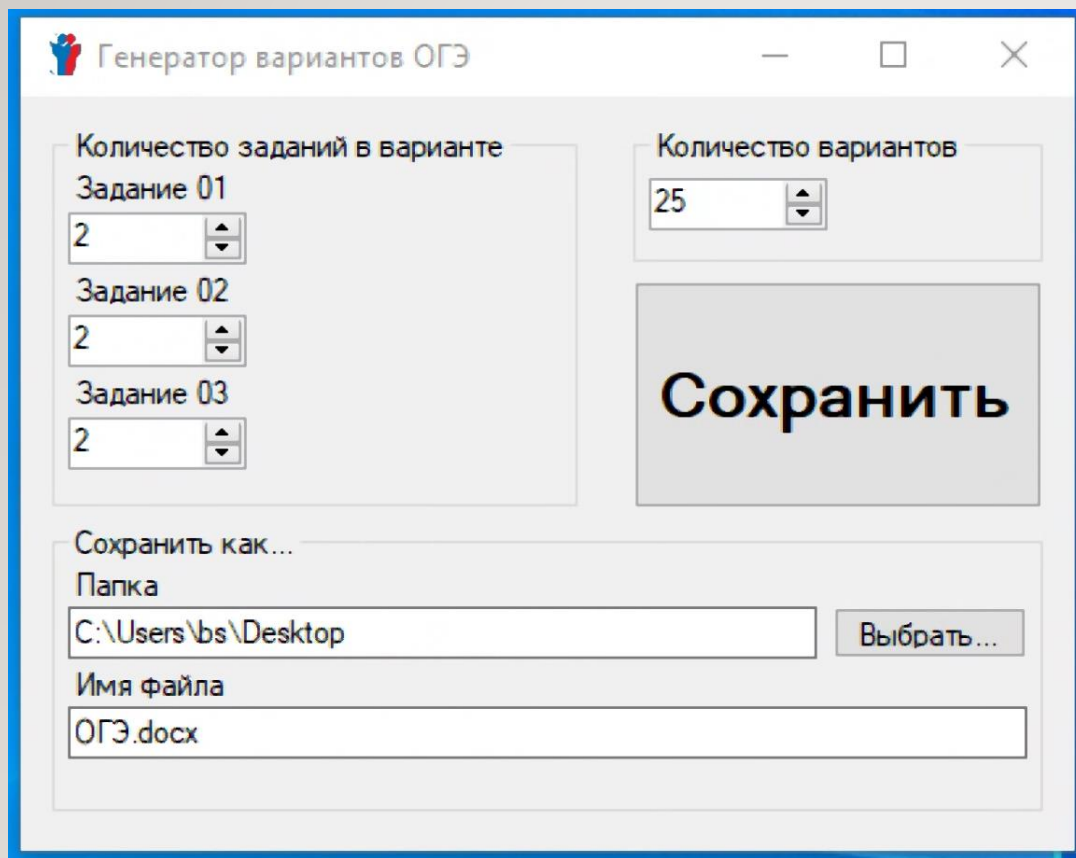
ЗДОРОВЫЙ КОД



ПРОЕКТ ГЕНЕРАТОРА



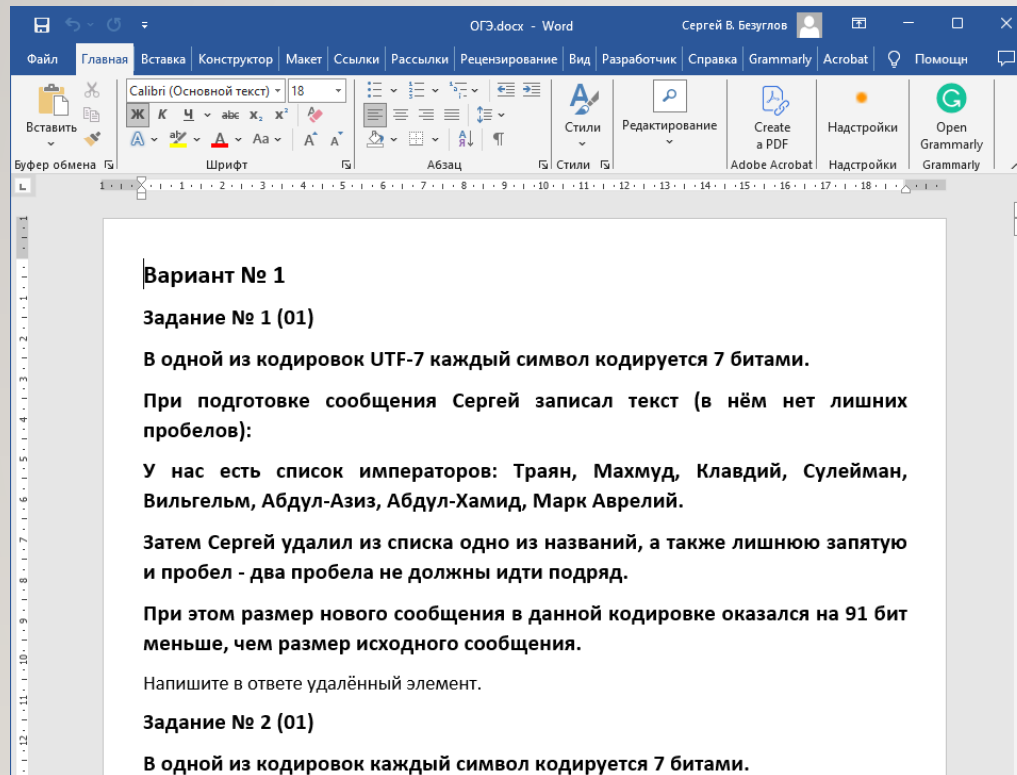
ПРОЕКТ ГЕНЕРАТОРА



The screenshot shows a Windows application window titled "Генератор вариантов ОГЭ" (Generator of OGZ variants). The window has a blue border and standard Windows window controls (minimize, maximize, close). The interface is divided into several sections:

- Left Panel:** Contains three labels: "Количество заданий в варианте" (Number of tasks in the variant), "Задание 01" (Task 01), "Задание 02" (Task 02), and "Задание 03" (Task 03). Each label is followed by a numeric spinner control, all of which are set to the value "2".
- Right Panel:** Contains a label "Количество вариантов" (Number of variants) followed by a numeric spinner control set to "25". Below this is a large, light gray button with the text "Сохранить" (Save).
- Bottom Panel:** Contains a label "Сохранить как..." (Save as...). Below it are two input fields: "Папка" (Folder) with the text "C:\Users\bs\Desktop" and a "Выбрать..." (Choose...) button to its right; and "Имя файла" (File name) with the text "ОГЭ.docx".

ПРОЕКТ ГЕНЕРАТОРА



Задание № 3 (02)

От разведчика было получено сообщение 1100110111101111000001010011.

В сообщении зашифрован пароль - последовательность русских букв.

В коде есть только следующие буквы - и: 1110, м: 010, н: 000, о: 001, т: 100, у: 011, ч: 1111, щ: 1101, ы: 101, я: 1100.

Расшифруйте пароль и запишите его в ответ.

Задание № 4 (02)

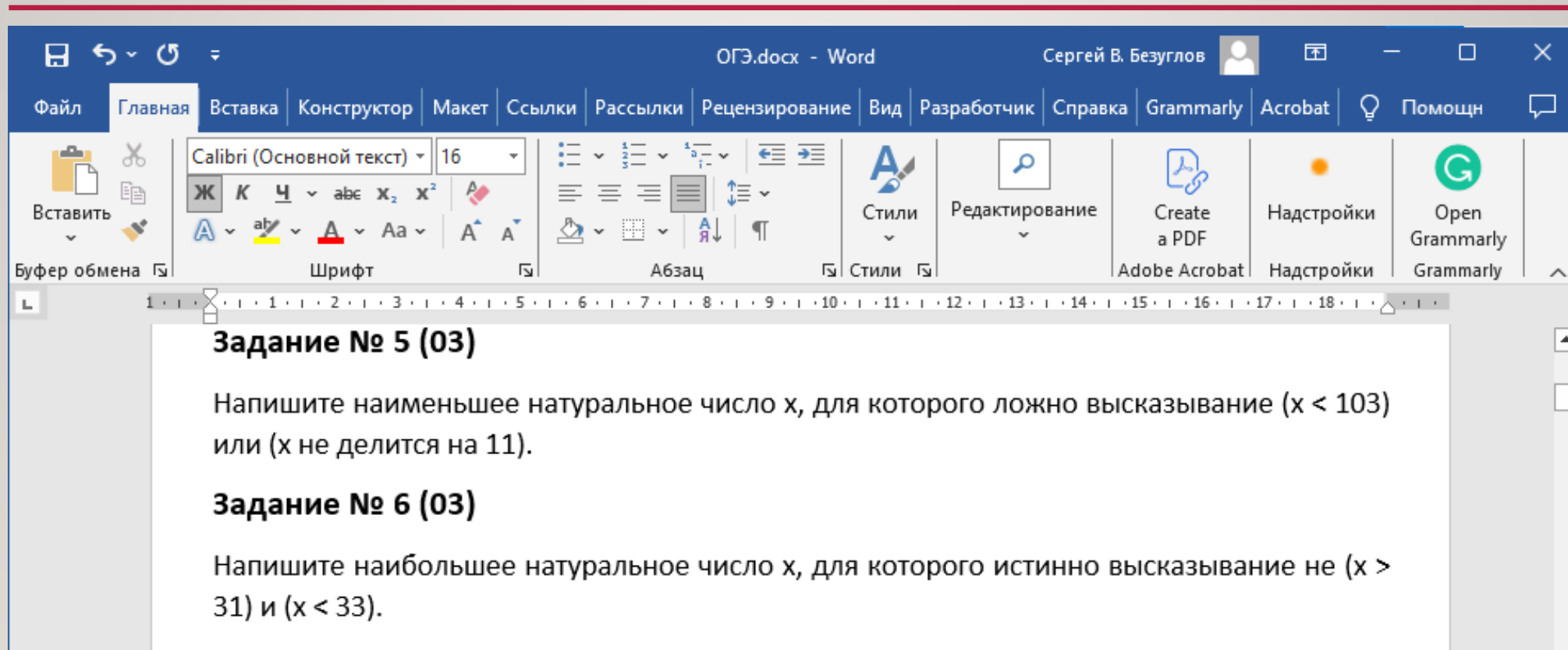
От разведчика было получено сообщение 100010011010100010111100110111101111.

В сообщении зашифрован пароль - последовательность русских букв.

В коде есть только следующие буквы - а: 1010, в: 1110, д: 010, з: 000, и: 1101, й: 001, л: 1100, о: 1111, с: 1000, т: 1011, ч: 1001, ы: 011.

Расшифруйте пароль и запишите его в ответ.

ПРОЕКТ ГЕНЕРАТОРА



ПРОЕКТ ГЕНЕРАТОРА

ОГЭ.docx - Word Сергей В. Безуглов

Файл Главная Вставка Конструктор Макет Ссылки Рассылки Рецензирование Вид Разработчик Справка Grammarly Acrobat Помощь

Вставить Шрифт Абзац Стили Редактирование Create a PDF Настройки Open Grammarly

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18

ОТВЕТЫ

Вариант	№ 1 (01)	№ 2 (01)	№ 3 (02)	№ 4 (02)	№ 5 (03)	№ 6 (03)
1	Абдул-Хамид	мю	ящичному	счастливо	110	31
2	Казахстан	Доминикана	бледнея	пустыми	276	185
3	Мухаммед-шах	Аметист	свидетеля	комиссии	182	148
4	Вышний Волочек	новая	душили	остановке	166	91
5	Амарантовы е	деловая	скучных	доживет	128	119
6	Болгария	Янтарь	мальчиков	осветила	72	287



БЛАГОДАРЮ ЗА ВНИМАНИЕ



ГБНОУ СО «Академия для одарённых детей (Наяновой)»

GitHub

<https://github.com/BesuglovS/OGE>

Безуглов С.В.

Самара

Telegram: @bezuglovs

<https://t.me/bezuglovs>

