

Рецензия
на материалы методического пособия по информатике
«Методы решения задания № 17 по материалам ЕГЭ по
информатике 2021 и 2022 года»
учителя информатики МБОУ лицей №4 им. профессора Е.А.
Котенко г. Ейска МО Ейский район
Рыжовой Натальи Ивановны

Материалы методического пособия предназначены для учителей информатики, работающих в 9-11 классах, для учащихся 10-11 классов; планирующих сдавать ЕГЭ по предмету в новой компьютерной форме. Главная задача учителя информатики – организовать работу с обучающимися таким образом, чтобы выбор предмета «информатика» на государственной итоговой аттестации был осознанным и правильным, создать условия для обеспечения качественной подготовки обучающихся и успешной сдачи ими ЕГЭ по информатике и ИКТ.

Количество страниц-32.

Цель методического пособия – продемонстрировать основные методы решения задания № 17 «Перебор последовательности целых чисел. Проверка делимости». По спецификации тем ЕГЭ 2022 года задание № 17 относится к заданиям повышенного уровня сложности. Для успешного выполнения этого задания необходимо свободно владеть базовыми навыками программирования, в том числе чтением данных из файлов и обработкой массивов.

Автор акцентирует внимание на том, что ЕГЭ в компьютерной форме проводится только второй год, материалов для качественной подготовки учащихся - недостаточно. Имеющиеся печатные пособия и интернет - публикации не содержат подробных методических рекомендаций для решения и отработки задания № 17. В демоверсии 2022 года данное задание существенно изменили в сторону усложнения. В пособии предлагается два метода решения задания. Первый метод – создание программы на языке программирования PascalABC.NET, второй – альтернативный метод при помощи организации электронной таблицы с использованием мастера функций. Автором сделана попытка представить сравнительную характеристику методов.

Содержание данного пособия обеспечивает формирование умений и навыков создавать небольшие по размеру программы, которые способствуют развитию потенциала учащихся и готовят их к экзамену. Изучая представленные в сборнике материалы, учащиеся, получают возможность познакомиться и самостоятельно научиться решать задачи на обработку числовых последовательностей и формирование условий отбора чисел, а выполняя задания для самостоятельного решения, смогут объективно оценить уровень своей подготовки по данной теме.

Пособие удобно как для работы в классе под руководством учителя, так и для самостоятельной отработки навыков в домашних условиях. После

пояснений решения задачи представлены однотипные задания для самостоятельного решения. Наличие программного кода к предлагаемым заданиям крайне облегчает работу по организации контроля и самоконтроля обучающихся.

Рецензируются материалы методического пособия «Методы решения задания № 17 по материалам ЕГЭ по информатике 2021 и 2022 года» Рыжовой Н.И. актуальны и могут быть рекомендованы к использованию в работе общеобразовательных организаций Ейского района.

Рецензент:
Муниципальный тьютор
по информатике

О.Д.Гусева

Подпись удостоверяю

Директор
МКУ «Информационно - методический центр
системы образования Ейского района»

Г.П.Гришко

29.11.2021 г.



**Муниципальное бюджетное общеобразовательное учреждение лицей № 4
имени профессора Е.А. Котенко г. Ейска МО Ейский район**

МЕТОДИЧЕСКОЕ ПОСОБИЕ

**«Методы решения задания № 17
по материалам ЕГЭ по информатике
2021 и 2022 года»**

Автор - составитель
Учитель информатики
МБОУ Лицей № 4 имени профессора Е.А. Котенко
города Ейска
муниципального образования Ейский район
Рыжова Наталья Ивановна

**КОПИЯ
ВЕРНА**



2021 год

Содержание

• Пояснительная записка.....	3
• Описательная часть.....	5
• Разбор задания номер 17 по материалам демоверсии 2021.....	10
○ Методика создания программного кода на языке PascalABCNET..	10
○ Методика создания электронной таблицы.....	12
○ Примеры решения других заданий по материалам ЕГЭ 2021.....	19
○ Задания для тренировки.....	23
○ Ответы к заданиям.....	24
• Разбор задания номер 17 по материалам демоверсии 2022.....	25
○ Методика создания программного кода на языке PascalABCNET..	25
○ Методика создания электронной таблицы.....	27
○ Сравнение методов.....	30
○ Примеры решения других заданий по материалам ЕГЭ 2022.....	31
○ Задания для тренировки.....	36
○ Ответы к заданиям.....	37
• Список используемой литературы.....	38

**КОПИЯ
ВЕРНА**



**ДИРЕКТОР
МБОУ Лицей № 4
Мосина НВ**

Пояснительная записка

Данное методическое пособие предназначено для учителей информатики, работающих в 9-11 классах, для учащихся 10-11 классов, методистов, всех тех, кто ведет подготовку учащихся к сдаче Единого Государственного Экзамена в новой компьютерной форме.

Цель методического пособия показать основные методы решения задания номер 17. По спецификации тем ЕГЭ 2022 года это задание относится к заданиям повышенного уровня сложности. В демонстрационной версии ЕГЭ 2022 года это задание существенно изменили в сторону усложнения. Для успешного выполнения этого задания выпускникам необходимо свободно владеть базовыми навыками программирования, в том числе чтением данных из файлов и обработкой массивов, уверенно применять алгоритмы поиска наибольшего и наименьшего элементов массива, суммы и количества элементов массива с заданными параметрами, отбирать элементы массива с заданными свойствами, уверенно оперировать логическими связками, знать приоритеты логических операций, оценивать результат логического выражения. Поэтому основными *задачами* данного пособия являются:

- 1) Показать возможные методы решения задания с помощью среды программирования PascalABC.NET
- 2) Показать методику альтернативного решения с помощью электронных таблиц
- 3) Сравнить используемые методы
- 4) Предложить вниманию читателей возможные, на взгляд автора, изменения в данном задании
- 5) Предложить материал для отработки умений и навыков.

Актуальность материалов данного методического пособия объясняется тем, что умение создавать собственные программы (20 – 40 строк) для обработки целочисленной информации вызывает у многих учащихся серьезные затруднения. Об этом свидетельствует низкий процент выполнения этого задания по стране (61%), с учетом усложнения задания этот процент может существенно снизиться. Компьютерное ЕГЭ проводится только второй год, материалов для подготовки к этой форме экзамена недостаточно. Имеющиеся печатные пособия и интернет - публикации не содержат методических рекомендаций для решения, а дают зачастую только подборку вариантов или списка задач.

В пособии рассматривается возможность решения данного задания двумя способами: написанием программного кода с помощью среды программирования PascalABC.NET и при помощи организации электронной таблицы. Подробно разбираются строки программного кода с



документированием. Вниманию читателей предлагаются возможные варианты изменения этого задания, в приложениях можно использовать готовые программные коды.

Затем подробно рассматривается возможность решения данного задания альтернативным методом - при помощи организации электронной таблицы с использованием мастера функций. Скриншоты последовательности действий дают возможность пошагово детализировать разработку электронной таблицы. Акцентируется внимание учащихся на работе мастера функций, прорабатывается возможность использования функций из различных категорий (математические, статистические, логические). Проводится сравнительная характеристика методов решения посредством программирования и путем создания электронных таблиц. У учащихся есть возможность решить задачу двумя способами, сравнить полученные ответы.

Отмечу, что данные умения и навыки могут пригодиться выпускникам и для решения других заданий ЕГЭ.

Для отработки задания 17 в пособии для учеников предусмотрены материалы для тренировки различного уровня сложности с возможностью проверки правильного ответа. Задания подобраны и расположены в порядке увеличения сложности. Учитель может использовать эти задания для отработки умений и навыков учащихся, а также для проведения самостоятельных работ или домашнего задания.

В школьном курсе информатики на решение таких задач отводится недостаточное количество времени. Тема «Алгоритмизация и программирование» - одна из самых сложных тем курса информатики. Не менее сложной является тема «Расчет и обработка электронных таблиц». Изучив и проработав данное пособие, у мотивированных выпускников есть возможность выбрать понравившийся метод, овладеть им и успешно справиться на экзамене с этим непростым заданием.



Описательная часть

Что необходимо знать

В данном задании проверяемыми элементами содержания являются умения создавать собственные программы длиной в 20-40 строк для обработки целочисленной информации. **Проверяемыми умениями и способами действия** являются умения написать простую переборную программу на известном языке программирования, содержащую цикл и ветвление. Задание имеет повышенный уровень сложности, на его выполнение дается в среднем 15 минут.

При решении этого задания ученикам необходимо знать базовые алгоритмические структуры, такие как цикл и ветвление.

Цикл – это управляющая структура, предназначенная для повторения группы алгоритмических действий. Последовательность повторяющихся действий называется телом цикла. Единичное повторение тела цикла называется итерацией. Параметр или счетчик цикла – это величина, которая изменяет свое значение при каждом следующем витке (итерации) цикла. Очень важно следить за изменением параметра цикла, так как от этого зависит выход из цикла. При решении этого задания лучше употреблять цикл с параметром, который в синтаксисе языка программирования PascalABC.NET имеет следующий вид:

For параметр:=начальное значение **to** конечное значение **do**

Begin

Операторы тела цикла;

End;

Например, для перебора чисел в диапазоне значений от 2094 до 5760 и дальнейших действий с перебираемым числом в программе нужно написать:

for n := 2094 **to** 5760 **do**

Begin

Операторы тела цикла по обработке n;

End;

Если нужно перебирать числа в обратном порядке от 5760 до 2094 нужно организовать цикл с шагом -1:

for n := 5760 **downto** 2094 **do**

Begin

Операторы тела цикла по обработке n;

КОПИЯ
ВЕРНА



МБОУ лицей № 4
Мосина НВ

End;

В операторах тела цикла необходимо сформировать условие отбора, поэтому нужно применить алгоритмическую структуру ветвление.

Ветвление – алгоритмическая конструкция, предназначенная для выбора одного из двух возможных путей алгоритма. Путь выбирается в зависимости от того проверяемое логическое выражение дает истину или ложь. Ветвление бывает: полное, неполное, составное, вложенное. В зависимости от сложности задачи нужно уметь применять разные виды ветвления.

В синтаксисе языка программирования PascalABCNET ветвление имеет следующий вид:

If логическое выражение **Then**

Begin

Группа действий, если логическое выражение истинно;

End

Else

Begin

Группа действий, если логическое выражение ложно;

End

При организации логического выражения следует применить логические связки:

-конъюнкция (логическое умножение), обозначается служебным словом **and**;

-дизъюнкция (логическое сложение), обозначается служебным словом **or**;

-инверсия (логическое отрицание), обозначается служебным словом **not**;

Необходимо понимать, что приоритет логических операций в логическом выражении выстроен следующим образом: действие в скобках, инверсия, конъюнкция, дизъюнкция. Внимание! В языке PascalABCNET проверяемые условия нужно взять в скобки, чтобы поднять приоритет операции сравнения.

При организации условия отбора нужно уметь применять арифметические операции: **div** – целочисленное деление, **mod** – остаток от целочисленного деления.

Так, например, если следует отобрать числа кратные (делящиеся) на 2 и при этом не кратные 3 нужно записать следующее условие отбора:



If $(n \bmod 2 = 0)$ and $(n \bmod 3 \neq 0)$ then действия.

Если следует отобрать числа, оканчивающиеся (последняя цифра) на 2 или оканчивающиеся на 3, при этом не кратные 4 нужно записать следующее условие отбора:

If $((n \bmod 10 = 2) \text{ or } (n \bmod 10 = 3))$ and $(n \bmod 4 \neq 0)$ then действия.

Обратите внимание на то, что дополнительными скобками поднят приоритет связки or.

Если следует отобрать числа, оканчивающиеся (последняя цифра) на А в шестнадцатеричной системе счисления или оканчивающиеся на 36 в восьмеричной системе счисления, нужно записать следующее условие отбора:

If $(n \bmod 16 = 10) \text{ or } (n \bmod 8 = 3)$ and $(n \div 8 \bmod 8 \neq 3)$ then действия.

Обратите внимание, что операции div и mod имеют одинаковый приоритет и в выражении выполняются слева направо. В этом примере сначала выполнится связка and, затем связка or, поэтому дополнительные скобки ставить не нужно.

Если в задании просят обратиться к файлу, расположенному на диске, для чтения входных данных, чаще целых чисел, нужно уметь организовать процесс чтения, для этого используются следующие процедуры:

Объявляем файловую переменную fl, которая будет связана с реальным файлом на диске '17-demo2022.txt'

```
var fl: text;
```

Связываем файловую переменную с рабочим файлом, расположенном на диске

```
assign(fl, '17-demo2022.txt');
```

Открываем файл для чтения данных

```
reset(fl);
```

Процесс чтения данных выполняем в цикле. Для чтения компонент файла нужно воспользоваться процедурой

```
Readln(fl, имя переменной для размещения считанного числа).
```

После чтения данных из файла его нужно обязательно закрыть процедурой

```
Close(имя файловой переменной)
```

Внимание! При первом запуске программы она должна быть записана на диск в ту же папку, что и рабочий файл, иначе они не найдут друг друга.

Считанные данные лучше всего сохранить в массиве. Массив – совокупность данных, хранящаяся в оперативной памяти рядом, каждому

элементу которой соответствует порядковый номер или индекс элемента. Для описания массива нужно определить его имя, размер, тип хранящихся в нем элементов.

Например, для описания массива из 5000 целых чисел, нужно написать такой фрагмент программы:

```
Const {константа n получает значение равное 5000}  
  n = 5000;
```

```
var
```

```
{создаем массив из 5000 элементов}
```

```
A: array[1..n] of integer;
```

Для обращения к элементам массива последовательно лучше организовать цикл с параметром, так, например для чтения целых чисел из файла и записи их в массив нужно написать следующий фрагмент программы:

```
for i := 1 to n do  
  readln(f1, A[i]);
```

Если в задаче нужно найти максимальный или минимальный элементы числовой последовательности с заданными свойствами, необходимо написать следующий фрагмент кода.

Для поиска максимального элемента, например кратного 3

{Максимальный элемент инициализируем самым малым значением последовательности чисел в файле}

```
Max_el := -10000;
```

{Выполняем цикл по последовательному перебору элементов, задавая условие отбора максимума с заданными ограничениями}

```
for i := 1 to n do  
  if (A[i] mod 3 = 0) and (A[i] > Max_el) then Max_el := A[i];
```

А вот для поиска минимального элемента с этими же свойствами нужно изменить программу так:

```
Min_el := 10000;
```

```
for i := 1 to n do  
  if (A[i] mod 3 = 0) and (A[i] < Min_el) then Min_el := A[i];
```

Если в задаче нужно найти сумму элементов числовой последовательности с заданными свойствами, необходимо написать следующий фрагмент кода.

Для поиска суммы элементов, например кратных 3 и при этом не оканчивающихся на 3.

КОПИЯ
ВЕРНА

ДИРЕКТОР
МБОУ ЛИЦЕЙ № 4
МОСКИНА НВ



{Инициализируем ячейку для накопления суммы}

Sum:=0;

{Выполняем цикл по последовательному перебору элементов, задавая условие отбора для элементов и накапливая их в ячейку Sum.}

for i := 1 to n do

if (A[i] mod 3 = 0) and (A[i] mod 10 <> 3) then Sum:= Sum+A[i];

Если в задаче нужно перебирать пары рядом стоящих элементов массива нужно написать следующий код

for i := 1 to n-1 do

if условие обработки A[i] и A[i+1] then обработка A[i] и A[i+1]

А вот если в задаче потребуется перебрать тройки рядом стоящих чисел, то программный код следует изменить вот так:

for i := 1 to n-2 do

if условие обработки A[i] и A[i+1] A[i+2] then обработка A[i], A[i+1], A[i+2]

А вот программный код для перебора всех четверок рядом стоящих элементов числовой последовательности:

for i := 1 to n-3 do

if условие обработки A[i] и A[i+1] A[i+2] A[i+3]

then обработка A[i], A[i+1], A[i+2], A[i+3]

Например, если нужно найти количество пар элементов последовательности в которых хотя одно число делится на 3, а сумма элементов пары не более максимального элемента последовательности, кратного 3.

K:=0; {инициализируем счетчик количества пар и перебираем пары}

for i := 1 to n-1 do

if ((A[i] mod 3 = 0) or (A[i+1] mod 3 = 0)) and A[i] + A[i+1] <= Max_el

then k:=k+1;

Обратим внимание, что Max_el был найден выше.

Автор постарался подробно изложить теоретические сведения по данному вопросу. Теперь вооружившись знаниями, попробуем на практике реализовать и отработать программные коды и получить результаты.



Разберем задание 17 по материалам демоверсии 2021 года.

Задача №1.

Рассмотрим множество целых чисел, принадлежащих числовому отрезку $[1016; 7937]$, которые делятся на 3 и не делятся на 7, 17, 19, 27. Найдите количество таких чисел и максимальное из них. В ответе запишите два целых числа: сначала количество, затем максимальное число.

Составим программу на языке программирования PascalABC.NET.

```
var
  n, k, max: integer;
begin
  {инициализируем счетчик нужных элементов}
  k := 0;
  {перебираем диапазон чисел по возрастанию}
  for n := 1016 to 7937 do
    {задаем условие отбора}
    if (n mod 3 = 0) and (n mod 7 <> 0) and
      (n mod 17 <> 0) and (n mod 19 <> 0)
      and (n mod 27 <> 0) then
      begin
        {счётчик k наращивает количество таких}
        k := k + 1;
        {переменная max фиксирует очередное найденное}
        {в конце цикла в max сохранит наибольшее из}
        {отобранных}
        max := n
      end;
  {Вывод результата}
  writeln(k, ' ', max)
end.
```

Таким образом, за один проход по числовой последовательности удалось ответить на два вопроса задачи.

Предположим, что задача будет сформулирована так:

Задача №2.

Рассмотрим множество целых чисел, принадлежащих числовому отрезку $[1016; 7937]$, которые делятся на 3 и не делятся на 7, 17, 19, 27. Найдите сумму таких чисел и минимальное из них. В ответе запишите два целых числа: сначала сумму, затем минимальное число.

Составим программу на языке программирования PascalABC.NET.

```
var
  n, k, min: integer;
```

