

Единый
государственный
экзамен по
информатике

ЕГЭ по информатике, задание №8

Часть 1
Задание №8

ОСНОВНЫЕ ТЕРМИНЫ

Язык программирования

Переменная

Линейный алгоритм

Разветвляющийся алгоритм

Циклический алгоритм

Операторы

Что необходимо знать

Для успешного выполнения задания №8 учащийся должен знать (в соответствии со спецификатором и кодификатором КИМ ЕГЭ—2017):

1. Основные конструкции языка программирования.
2. Система программирования.
3. Читать и отлаживать программы на языке программирования.

Язык программирования — формальная знаковая система, предназначенная для записи компьютерных программ. Язык программирования определяет набор лексических, синтаксических и семантических правил, задающих внешний вид программы и действия, которые выполнит исполнитель (компьютер) под ее управлением.

Структурное программирование — методология разработки программного обеспече-

ния, в основе которой лежит представление программы в виде иерархической структуры блоков. Предложена в 70-х годах XX века Э. Дейкстрой, разработана и дополнена Н. Виртом.

В соответствии с данной методологией

1. Любая программа представляет собой структуру, построенную из трёх типов базовых конструкций:

- последовательное исполнение (линейный алгоритм) — однократное выполнение операций в том порядке, в котором они записаны в тексте программы;
- ветвление (условный оператор) — однократное выполнение одной из двух или более операций, в зависимости от выполнения некоторого заданного условия;
- цикл (повторение) — многократное исполнение одной и той же

операции до тех пор, пока выполняется некоторое заданное условие (условие продолжения цикла).

В программе базовые конструкции могут быть вложены друг в друга произвольным



образом, но никаких других средств управления последовательностью выполнения операций не предусматривается. Разработка программы ведётся пошагово, "сверху—вниз"

Обратите внимание:

- Уровень сложности задания — базовый
- Максимальный балл за выполнение задания — 1 балл (первичный)
- Примерное время выполнения задания — 3 минуты

Язык программирования Pascal

Алфавит языка - это конечный набор символов, которые используются при написании любой конструкции на языке.

Алфавит языка Паскаль можно разбить на 3 группы :

1. буквы строчные и прописные буквы латинского и русского алфавита;
2. цифры - арабские 0..9
3. специальные символы - (+, -, *, /, =, <, >, <=, >=, ::, {} , := ()), служебные слова : and, begin, const, div, do, else, for, и т.д. и указывается тип переменных списка.

Переменная - это область памяти, названная собственным именем, которая может менять

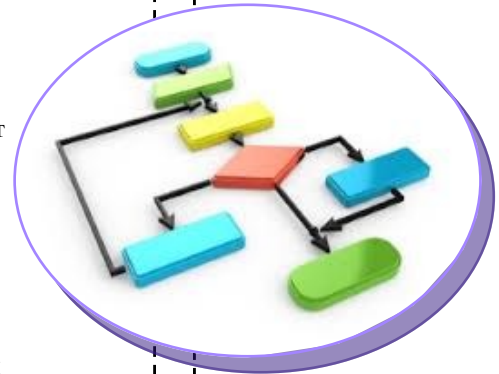
свое значение в процессе выполнения программы. Переменная характеризуется именем, типом и значением.

Константа - не изменяет своего значения в процессе выполнения программы, она может быть задана явно своим значением или обозначена именем.

Имя (идентификатор) переменной или константы задается латинскими буквами и арабскими цифрами. В качестве идентификатора нельзя использовать служебные

слова.

Идентификатор должен быть уникальным, т.е. в данном блоке про-



граммы один идентификатор не может быть использован для обозначения более чем одного объекта.

Типы переменных

Следует выделять следующие типы данных:

Целые

Integer (диапазон значений: -32768 ... 32767)

Longint (диапазон значений: -2147483648 ... 2147483647)

Byte (диапазон значений: 0 255)

Word (диапазон значений: 0 ... 65535)

Вещественные

Real (диапазон значений: 2,9 10⁻³⁹..... 1,7 10³⁸)

Логический

Boolean (значения: TRUE, FALSE)

Символьный

Char (значения: 1 символ)

Строковый

String (значения: массив символов, по умолчанию длина 256 символов)

Массивы

array (далее идет описание типа переменных, входящих в данный одномерный/двумерный массив)

Здесь приведены только те типы переменных, которые наиболее часто используются в заданиях данного уровня сложности.

Стандартные математические функции

abs(x) - абсолютное значение x

sqr(x) - квадрат x.

sqrt(x) - квадратный корень из x.

random(x) - случайное число из диапазона 0 <= ... < x

int(x) - целая часть числа x.

frac(x) - дробная часть числа x.

trunc(x) - целая часть в форме longint.

round(x) - значение x округленное до следующего целого числа.

Данные функции редко используются в заданиях такого уровня, однако понимание их необходимо для дальнейшего успешного изучения программирования

Правила записи арифметических выражений

1. Все данные, входящие в арифметическое выражение, должны быть одного типа. Допускается использовать вместе данные целого и действительного типа.
2. Записывать все составные части в одну строку без подстрочных и надстрочных индексов.
3. Использовать скобки одного типа - круглые. ([{ и другие скобки применять запрещается)
4. Нельзя записывать подряд 2 знака арифметических операций.
5. Вычисления выполняются слева направо в соответствии со старшинством операций:
 - 1) вычисление функций;
 - 2) * / DIV (деление нацело)
MOD (получение остатка от деления)
 - 3) + -

Основные операторы

Оператор ввода	Оператор вывода	Оператор присваивания
<p>Read(<список переменных>); Readln(<список переменных>);</p> <p>После выполнения данного оператора программа останавливается и ждет ввода данных с клавиатуры. Если вводятся числовые данные, то их можно ввести через пробел друг за другом, а можно каждый отдельно нажимая в конце ввода Enter. Переменный в <списке переменных> разделяются запятой. Ln за словом Read означает, что после ввода следует перевести курсор на следующую строку.</p>	<p>Write('комментарий', <переменные>); Writeln('ком - рии', <переменные>);</p> <p>Переменный в <списке переменных> разделяются запятой. Для вывода целых и действительных чисел можно указывать форматы в операторе Write. Формат указывается через двоеточие после переменной.</p>	<p>: =</p> <p>Оператор применяется для присваивания переменной программы некоторого значения, которое может быть выражено числовым значением, другой переменной или некоторым уравнением</p>

Структура программы

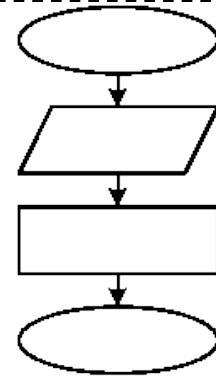
PROGRAM <имя программы>;
VAR - описание переменных;
PROCEDURE - описание процедур;
FUNCTION - описание функций;
 {начало раздела операторов}
BEGIN
 операторы ввода, вывода и
 обработки данных
END. {конец раздела операторов,
 конец программы}

Следует учитывать, что представлены не все элементы структуры программы, в частности:
USES - модули;
 {раздел описаний }
LABEL - описание меток;
CONST - описание констант;
TYPE - описание типов.
 Однако, они не встречаются в программах к заданиям такого уровня.

Линейный алгоритм

Линейные алгоритмы описывают решение задач с последовательным выполнением действий. Обычно такие действия идут в следующем порядке :

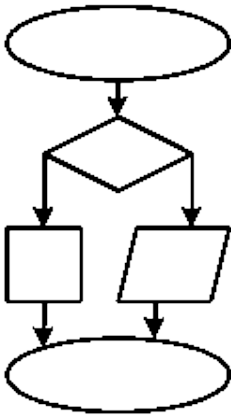
- **ввод исходных данных** (может отсутствовать, тогда данные задаются внутри программы),
- **последовательные команды** - обычно вычислительного характера;
- **вывод результатов** (должен присутствовать обязательно).



Алгоритм «Ветвление»

Разветвляющийся алгоритм - это алгоритм, в котором в зависимости от выполнения или невыполнения некоторого условия выполняется либо одна, либо другая последовательность действий. Оператор if или

условный оператор
Условный оператор может записываться в полной и неполной форме.



Полная форма условного оператора :

```
if<условие> then <оператор1>
  else <оператор2>
```

Каждое слово ELSE пишется под своим словом IF.

У

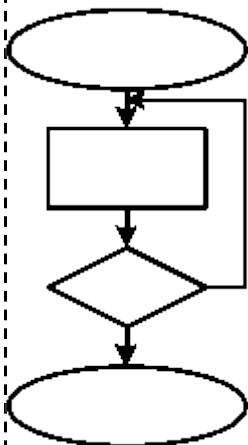
словие в операторе IF..THEN...ELSE может быть простым или сложным. Простое условие представляет собой логическое выражение, состоящее из двух выражений одинакового типа, соединенных знаком операции отношения (>, <, =, <=, >=). Сложное условие состоит из простых, соединенных знаками логических операций : OR(или) NOT(не) AND(и). При этом каждое условие ограничено с обеих сторон круглыми скобками.

Внимание !!!!
Перед словом ELSE
нельзя ставить точку с
запятой.

Алгоритм «Цикл»

Цикл - процесс многократного повторения каких-либо действий. Действия, повторяющиеся в цикле, называются **телом цикла**. Различают 3 вида циклов:

- 1) Цикл с параметром «Для»
- 2) Цикл с предусловием «Пока»
- 3) Цикл с постусловием «До»



1) Оператор цикла с параметром применяют тогда, когда заранее известно число повторений одной и той же последовательности операторов. Начальные и конечные значения параметра цикла могут быть представлены константами, переменными или арифметическими выражениями.

```
for <счетчик> :=<начальное значение> to (downto)<конечное значение> do
  <оператор>
```

2) Цикл с предусловием WHILE используется тогда, когда число повторений оператора цикла заранее не известно, а задается некоторое условие продолжение цикла.

```
While <лог выр-ние / условие> do оператор
```

Сначала выполняется значения условия. Пока оно истинно выполняются операторы циклической части.

3) Этот оператор отличается от цикла с предусловием WHILE тем, что проверка условия производится после очередного выполнения тела цикла. Это обеспечивает выполнение цикла хотя бы один раз. Repeat операторы until <логическое выражение>



Полезные ресурсы

<http://kpolyakov.spb.ru>

<https://ege.sdamgia.ru>

<http://foxford.ru/>

ЕДИННЫЙ
ГОСУДАРСТВЕННЫЙ
ЭКЗАМЕН ПО

учитель информатики
МАОУ лицей № 64
(г. Краснодар)

www.aliyaismailovna.ru
aiya-ismailovna@yandex.ru

Данный теоретический блок является частью учебного материала, направленного на усвоение навыков решения задания №8 части первой единого государственного экзамена по информатике. Практический блок содержит примеры заданий с решением.

Список источников:

1. Демо-вариант ЕГЭ 2017 г.
2. Информатика. Тематические тестовые задания. 2017 г. Авторы: Крылов С.С., Ушаков Д.М. ЕГЭ 2016.
3. Курс лекций для начинающих изучение языка Паскаль. Автор :Комарова И.Н.
4. Тренировочные и диагностические работы Статград.
5. Учебник «Информатика» 10-11 классы (ФГОС, углублённый уровень). Авторы: К.Ю. Поляков и Е.А. Еремин.