B6 (базовый уровень, время – 2 мин)

Тема: Оператор присваивания в языке программирования[1].

Что нужно знать:

- переменная – это величина, которая имеет имя, тип и значение; переменная может изменяться во время выполнения программы
- оператор присваивания служит для записи значения в переменную
- если в переменную записывают новое значение, старое стирается
- знаки **+**, **-**, *****, **/** используются для обозначения операций сложения, вычитания, умножения и деления
- запись вида **a div b** означает результат целочисленного деления **a** на **b** (остаток отбрасывается)
- запись вида **a mod b** означает остаток от деления **a** на **b**
- запись вида **a := b + 2*c + 3;** означает «вычислить значения выражения справа от знака присваивания := и записать результат в переменную **a**»; при этом значения других переменных (кроме **a**) не изменяются
- для многократного выполнения одинаковых операций используют циклы;
- цикл с переменной выполняется N раз, в этом примере переменная **i** принимает последовательно все значения от 1 до N с шагом 1

```
for i:=1 to N do begin
  { что-то делаем }
end;
```

- цикл с условием выполняется до тех пор, пока условие в заголовке цикла не нарушится;

```
while { условие } do begin
  { что-то делаем }
end;
```

- главная опасность при использовании цикла с условием – **зацикливание**; эта такая ситуация, когда цикл работает бесконечно долго из-за того, что условие все время остается истинным

Пример задания:

Определите значение переменной c после выполнения следующего фрагмента программы.

```
a := 5;
a := a + 6;
b := -a;
c := a - 2*b;
```

Решение:

1) для решения нужно использовать «ручную прокрутку» программы, то есть, выполнить вручную все действия
2) наиболее удобно и наглядно это получается при использовании таблицы, где в первом столбце записаны операторы программы, а в остальных показаны изменения переменных при выполнении этих операторов
3) здесь используются три переменные: a, b, c; до выполнения программы их значения анм неизвестны, поэтому ставим в таблице знаки вопроса:

	a	b	c
	?	?	?

[1] Здесь рассматривается только язык Паскаль, который является наиболее распространенным в школах России.

4) после выполнения оператора **a := 5;** изменяется значение переменной a:

	a	b	c
	?	?	?
a := 5;	5		

5) оператор **a := a + 6;** означает «вычислить значение выражения **a + 6** используя текущее значение **a** (равное 5), и записать результат обратно в переменную **a**»; таким образом, новое значение равно 5 + 6 = 11:

	a	b	c
	?	?	?
a := 5;	5		
a := a + 6;	11		

6) следующий оператор, **b := –a;** изменяет значение переменной **b**, записывая в нее **–a**; учитывая, что в **a** записано число 11, находим, что **b** будет равно –11:

	a	b	c
	?	?	?
a := 5;	5		
a := a + 6;	11		
b := –a;		–11	

7) последняя команда, **c := a – 2*b**, изменяет значение переменной **c**; при текущих значениях **a = 11** и **b = –11** результат выражения равен 11 – 2*(–11) = 33, это число и будет записано в переменную **c**:

	a	b	c
	?	?	?
a := 5;	5		
a := a + 6;	11		
b := –a;		–11	
c := a – 2*b;			33

8) таким образом, правильный ответ – 33.

Возможные ловушки и проблемы:

- нельзя забывать про знак переменных и про то, что «минус на минус дает плюс»

Ещё пример задания:

Определите значение переменной c после выполнения следующего фрагмента программы.

```
a := 40;
b := 10;
b := - a / 2 * b;
if a < b then
    c := b - a
else
    c := a - 2 * b;
```

Решение:

1) для решения нужно использовать «ручную прокрутку» программы
2) выполним начальные присваивания:

	a	b	c
	?	?	?
a := 40;	40	10	

`b := 10;`			

3) самый сложный оператор, содержащий «подводный камень»:
   `b := - a / 2 * b;`
   не забываем, что умножение и деление имеют равный приоритет, и в такой цепочке сначала выполнится деление, а потом умножение
4) результат:
   `b := - (40 / 2) * 10 = - 20 * 10 = - 200`

	a	b	c
	?	?	?
`a := 40;` `b := 10;`	40	10	
`b := -a/2*b;`		-200	

5) очевидно, что теперь условие «`a < b`» ложно, поэтому выполняется оператор, стоящий после слова `else`: `c := a-2*b = 40 – 2*(-200) = 440.`
6) Ответ: 440.

Возможные ловушки и проблемы:
- нужно не забыть, что умножение и деление имеют одинаковый приоритет, то есть выполняются по порядку слева направо (если нет скобок)

Еще пример задания:

В результате выполнения фрагмента программы

```
while n <> 0 do begin
  write ( 2*(n mod 10)+1);
  n := n div 10;
end;
```

на экран выведено число 13717. Укажите все числа, которые могли находиться в переменной **n** *до выполнения этого цикла.*

Решение:

1) прежде всего, заметим, что для вывода используется оператор **write**, который не переходит на следующую строку; поэтому числа в цикле будут выводиться в одной строке «вплотную» друг к другу, без промежутков
2) вспомним, что **n mod 10** – остаток от деления числа на 10 – это последняя цифра числа в десятичной системе счисления;
3) операция **n div 10** (деление нацело на 10) равносильна отбрасыванию последней цифры в десятичной системе счисления
4) эти две операции выполняются пока значение переменной **n** не станет равно нулю
5) анализируя алгоритм, можно придти выводу, что этот фрагмент программы выводит на экран числа $2d_1+1, 2d_2+1, \ldots$, где d_i – это i-ая цифра с конца числа
6) подумаем, в каком интервале находится значение $2d_i+1$, если d_i – это цифра от 0 до 9: получаем интервал от $2\cdot 0+1=2$ до $2\cdot 9+1=19$
7) поэтому разбить цепочку 13717 на части можно следующими способами:
   1 – 3 – 7 – 17
   1 – 3 – 7 – 1 – 7
   13 – 7 – 17
   13 – 7 – 1 – 7

8) в любом варианте разбиения каждое число получено по формуле $n_i = 2d_i + 1$, поэтому можно сразу определить цифры соответствующих чисел по формуле $d_i = \frac{n_i - 1}{2}$:

1 – 3 – 7 – 17	0 – 1 – 3 – 8
1 – 3 – 7 – 1 – 7	0 – 1 – 3 – 0 – 3
13 – 7 – 17	6 – 3 – 8
13 – 7 – 1 – 7	6 – 3 – 0 – 3

9) вспоминаем, что цифры числа в цикле обрабатываются, начиная с последней, поэтому в ответе нужно перечислить числа 836, 3036, 8310 и 30310.
10) таким образом, правильный ответ – 836, 3036, 8310, 30310.

Возможные ловушки и проблемы:
- нужно уметь анализировать работу алгоритма, «прокручивать» его в уме
- можно забыть, что цифры числа обрабатываются в обратном порядке, начиная с последней

Задачи для тренировки[2]:

1) Определите значение целочисленных переменных a и b после выполнения фрагмента программы:
```
a := 3 + 8*4;
b := (a div 10) + 14;
a := (b mod 10) + 2;
```
2) Определите значение целочисленных переменных a и b после выполнения фрагмента программы:
```
a := 1819;
b := (a div 100)*10+9;
a := (10*b-a) mod 100;
```
3) Определите значение целочисленных переменных a и b после выполнения фрагмента программы:
```
a := 42;
b := 14;
a := a div b;
b := a*b;
a := b div a;
```
4) Определите значение целочисленных переменных x, y и t после выполнения фрагмента программы:
```
x := 5;
y := 7;
t := x;
x := y mod x;
y := t;
```
5) Определите значение целочисленных переменных a и b после выполнения фрагмента программы:
```
a :=6*12 + 3;
b :=(a div 10)+ 5;
a :=(b mod 10)+ 1;
```
6) Определите значение целочисленных переменных x и y после выполнения фрагмента программы:
```
x := 336
y := 8;
x := x div y;
y := x mod y;
```

[2] Источники заданий:
1. Демонстрационные варианты ЕГЭ 2004-2011 гг.
2. Гусева И.Ю. ЕГЭ. Информатика: раздаточный материал тренировочных тестов. — СПб: Тригон, 2009.
3. Крылов С.С., Лещинер В.Р., Якушкин П.А. ЕГЭ-2010. Информатика. Универсальные материалы для подготовки учащихся / под ред. В.Р. Лещинера / ФИПИ. — М.: Интеллект-центр, 2010.
4. Якушкин П.А., Ушаков Д.М. Самое полное издание типовых вариантов реальных заданий ЕГЭ 2010. Информатика. — М.: Астрель, 2009.
5. М.Э. Абрамян, С.С. Михалкович, Я.М. Русанова, М.И. Чердынцева. Информатика. ЕГЭ шаг за шагом. – М.: НИИ школьных технологий, 2010.
6. Самылкина Н.Н., Островская Е.М. ЕГЭ 2011. Информатика. Тематические тренировочные задания. — М.: Эксмо, 2010.

7) Определите значение целочисленных переменных a и b после выполнения фрагмента программы:

```
a :=1686;
b :=(a div 10) mod 5;
a := a - 200*b;
```

8) Определите значение целочисленных переменных x и y после выполнения фрагмента программы:

```
x := 11;
y := 5;
t := y;
y := x mod y;
x := t;
y := y + 2*t;
```

9) Определите значение целочисленных переменных x и y после выполнения фрагмента программы:

```
x := 19;
y := 3;
z := y*2;
y := x mod y;
x := x - z;
y := y + z;
```

10) Определите значение целочисленных переменных x, y и z после выполнения фрагмента программы:

```
x := 13;
y := 3;
z := x;
x := z div y;
y := x;
```

11) В результате выполнения фрагмента программы

```
while n <> 0 do begin
  write ( 2*(n mod 5 + 3) );
  n := n div 10;
end;
```

на экран выведено число 10614. Какое число хранилось до этого в переменной **n**, если известно, что все цифры в нем нечетные?

12) Определите значение переменной **b** после выполнения следующего фрагмента программы, где **a** и **b** – вещественные (действительные) переменные:

```
a := -5;
b := 5 + 7 * a;
b := b / 2 * a;
```

13) Определите значение переменной **b** после выполнения следующего фрагмента программы, где **a** и **b** – вещественные (действительные) переменные:

```
a := 5;
b := 5 - 3 * a;
b := b / 2 * a;
```

14) Определите значение переменной **b** после выполнения следующего фрагмента программы, где **a** и **b** – вещественные (действительные) переменные:

```
a := 5;
b := 5 + 5 * a;
```

```
b := b / 2 * a;
```

15) Определите значение переменной **b** после выполнения следующего фрагмента программы, где **a** и **b** – вещественные (действительные) переменные:

```
a := 7;
b := 7 + 3 * a;
b := b / 2 * a;
```

16) Определите значение переменной **c** после выполнения следующего фрагмента программы:

```
a := 100;
b := 30;
a := a - b*3;
if a > b then
     c := a - b
else c := b - a;
```

17) Определите значение переменных **a** и **b** после выполнения следующего фрагмента программы:

```
a := 2468;
b := (a mod 1000)*10;
a := a div 1000 + b;
```

18) Определите значение переменной **c** после выполнения следующего фрагмента программы:

```
a := 6;
b := 15;
a := b - a*2;
if a > b then
     c := a + b
else c := b - a;
```

19) Определите значение переменной **c** после выполнения следующего фрагмента программы:

```
a := -5;
b := 14;
b := b + a*2;
if a > b then
     c := a + b
else c := b - a;
```

20) Определите значение переменной **c** после выполнения следующего фрагмента программы:

```
a := -5;
b := 3;
a := a - b*2;
if a > b then
     c := b - a
else c := a - b;
```

21) Определите значение переменной **c** после выполнения следующего фрагмента программы:

```
a := -5;
b := -3;
a := a - b*3;
if a > b then
     c := b + a
else c := a - b;
```

22) Определите значение переменной **c** после выполнения следующего фрагмента программы:

```
a := -2;
```

```
b := -3;
a := b + a*3;
if a < b then
     c := a - b
else c := b – a;
```

23) Определите значение переменной **c** после выполнения следующего фрагмента программы:

```
a := 40;
b := 10;
b := a - 2*b;
if a < 2*b then
     c := a
else c := b;
```

24) Определите значение переменной **c** после выполнения следующего фрагмента программы, в котором **a**, **b** и **c** – переменные вещественного (действительного) типа:

```
a := 120;
b := 100;
a := a + b / 2;
if b < a / 2 then
     c := b + a
else c := b + a / 2;
```

25) Определите значение переменной **S** после выполнения следующего фрагмента программы:

```
S:=1; i:=1;
while i < 5 do begin
  S := S + i*(i+1);
  i := i + 1;
end;
```

26) Определите значение переменной **S** после выполнения следующего фрагмента программы:

```
S:=0; i:=7;
while i > 1 do begin
  S := S + i div 2;
  i := i - 1;
end;
```

27) Определите значение переменной **P** после выполнения следующего фрагмента программы:

```
P:=1; i:=3;
while i <= 9 do begin
  P := P * (i div 3);
  i := i + 1;
end;
```

28) Определите значение переменной **c** после выполнения следующего фрагмента программы:

```
a:= 7;
a:= a - 4;
b:= -a;
c:= -a + 2*b;
```

29) Определите значение переменной **c** после выполнения следующего фрагмента программы:

```
a:= 5;
a:= 12 – a*a;
b:= -a;
```

```
c:= 10*a - b;
```

30) Определите значение переменной **c** после выполнения следующего фрагмента программы:

```
x:= 2.5E+02;          { 2.5E+02 = 2.5·10^+02 = 250}
x:= x + 0.5E+02;      { 0.5E+02 = 0.5·10^+02 = 50}
y:= -x;
c:= -2*y - x;
```

31) Определите значение переменной **c** после выполнения следующего фрагмента программы:

```
m:= 67;
m:= m + 13;
n:= m/4 – m/2;
c:= m - n;
```

32) Определите значение переменной **c** после выполнения следующего фрагмента программы:

```
x:= 8 + 2*5;
y:= (x mod 10) + 14;
x:= (y div 10) + 3;
c:= x - y;
```

33) Определите значение переменной **c** после выполнения следующего фрагмента программы:

```
a := 30;
b := 6;
a := a / 2 * b;
if a > b then
     c := a - 3 * b
else c := a + 3 * b;
```

34) (http://ege.yandex.ru) Определите значение переменной **c** после выполнения следующего фрагмента программы:

```
a := 30;
b := 6;
a := a / 5 * b;
if a > b then
     c := a - 4 * b
else c := a + 4 * b;
```