

## ПРАКТИЧЕСКОЕ ЗАДАНИЕ №1

### СТРУКТУРА ПРОГРАММЫ, ВВОД/ВЫВОД

**Цель работы** – изучить структуру программы, научиться использовать переменные различных типов, освоить функции форматного ввода и вывода, арифметические операции и операции присваивания.

#### **Постановка задачи**

Набрать текст программы, представленный листингом. Проанализировать значения переменных после каждой операции присваивания. Проверить порядок выполнения операций в одном из выражений, содержащих несколько операций присваивания, разделив этот оператор-выражение на несколько операторов, выполняемых последовательно. В функциях ввода и вывода изменить спецификаторы типов, проанализировать полученные результаты.

Написать программу для вычисления следующих выражений:

$a=5, c=5$

$a=a+b-2$

$c=c+1, d=c-a+d$

$a=a*c, c=c-1$

$a=a/10, c=c/2, b=b-1, d=d*(c+b+a)$

Выражения, записанные в одной строке, записывать одним оператором-выражением. Переменные  $c$  и  $d$  объявить как целые, переменные  $a$  и  $b$  – как вещественные. Значения переменных  $b$  и  $d$  вводить с клавиатуры. После вычисления каждого выражения выводить на экран значения всех переменных.

#### **Листинг**

```
#include <stdio.h>

int main (void)
{
    int a, b = 5, c;
```

```

float x, y = -.5, z;
printf("a=");
scanf("%d", &a);
x = c = a;
printf("a=%d, c=%d, x=%f\n",a,c,x);
a += b;
printf("a=%d\n", a);
x += b+a;
printf("x=%f\n", x);
b += a--;
printf("b=%d\n", b);
x -= ++c;
printf("x=%f\n",x);
c = a/b;
printf("c=%4d\n",c);
c = a%b;
printf("c=%d\n",c);
x = 5.3;
y += ((--x-1)/x++);
printf("  x = %f\ty = %.2f\n\n  x = %.0f\ty = %.0f\n ", x - 1, y, x, y - 1);
z = a/2;
printf("  z = %f\n", z);
z = (float) a/2;
printf("  z = %f\n", z);
y = x/2;
printf("  y = %f\n", y);
y = (int)x/2;
printf("  y = %f\n", y);
z = a%2 - (x+b)/c + (x-y)/(a-1) + 1/4*a - y++ + ++b/3.;
printf("  a = %d b = %d c = %d x = %f y = %f z = %f\n", a, b, c, x, y, z);

```

```
    return 0;  
}
```

### **Контрольные вопросы**

1. Какова структура программы на языке Си?
2. Зачем нужна директива `#include`?
3. Что такое `main()`?
4. Перечислите скалярные типы данных языка Си.
5. Что определяет тип данного?
6. Что такое `void`?
7. Что такое явное и неявное приведение типов? Как и когда оно используется?
8. Что такое константа? Найдите константы в набранных вами программах.
9. Что такое переменная?
10. Как проинициализировать переменную?
11. Чем отличается оператор от операции?
12. Чем отличаются унарные операции от бинарных?
13. Какие операции относятся к арифметическим? Каков приоритет каждой из них?
14. Каков порядок выполнения операций в случае их одинакового приоритета?
15. Как выполняется операция деления в случае целочисленных операндов и в случае, когда хотя бы один из операндов вещественный?
16. Что такое выражение?
17. Какое значение вычисляет операция присваивания?
18. В каком порядке выполняются присваивания в случае, если в выражении их несколько?
19. Как и зачем используются дополнительные операции присваивания?
20. Чем отличается префиксная форма операции инкремента или

декремента от постфиксной?

21. Какие функции используются для ввода информации? Назовите их отличительные особенности.
22. Какие функции используются для вывода информации? Назовите их отличительные особенности.
23. Почему функции `scanf()` и `printf()` называются функциями форматного ввода и вывода? Как они работают?
24. Чем отличается управляющая строка функции `scanf()` от управляющей строки функции `printf()`?
25. Что такое спецификатор типа? Зачем он нужен?
26. Какие параметры указываются функции `scanf()` после управляющей строки? Сколько их должно быть?
27. Каковы последствия несоответствия типа считываемой переменной спецификатору типа?
28. Какие параметры указываются функции `printf()` после управляющей строки? Сколько их должно быть?
29. Каковы последствия несоответствия типа выводимого значения спецификатору типа?
30. Что такое управляющие символы? Зачем они нужны? Приведите примеры.