ПРАКТИЧЕСКАЯ РАБОТА № 1

Основы алгоритмизации и программирования.

Часть 1. Решение алгебраических и трансцендентных уравнений.

Задание:

Составить алгоритм и программу поиска корней кубического $x^3 + ax^2 + bx + c = 0$ и трансцендентного $g(t) = 0$ уравнений.

Варианты заданий

№ варианта	*Функция* $f(x) = x^3 + ax^2 + bx + c$			*Функция* $g(t)$
	a	b	c	
1	–1,53	–0,677	0,944	$\sqrt[3]{x} + x^3 - 5$
2	4,19	–0,553	–6,40	$x\mathrm{e}^x + x^3 - 10$
3	0,630	–13,9	12,0	$x^2\mathrm{e}^x - 10$
4	–5,94	8,97	–3,12	$x^3 + x^2 + \sqrt{x} - 7$
5	–1,32	–7,90	–5,84	$3^x + x^3 - 9$
6	7,71	14,4	0,500	$x2^x - 4$
7	–0,334	–14,2	–7,67	$x^2 \ln x - 1$
8	–4,26	–2,83	19,1	$x \ln x + x^3 - 4$
9	3,45	–10,4	–25,3	$x \sin x + x^3 - 4$
10	3,31	–19,4	–57,9	$x \cos x + x^3 - 4$
11	–2,04	–10,8	16,6	$\tanh x + x^3 - 4$
12	–1,47	–15,5	–6,47	$x \cosh x + x^3 - 7$
13	–4,23	–1,09	2,82	$x^2 \sinh x + x^3 - 8$
14	–5,70	9,74	–4,82	$x^2 \ln x + x^3 - 4$
15	8,89	19,2	–1,26	$x^2\mathrm{e}^x + x^3 - 15$
16	2,87	–8,72	–4,64	$x \cos^2 x + x^3 - 4$
17	6,03	5,68	–10,5	$x \sin^2 x + x^3 - 5$
18	–2,09	–4,17	7,08	$x^3 - \sqrt[3]{x} - 3$
19	1,73	–0,164	–0,117	$x^2 - x\mathrm{e}^x + 5$
20	0,622	–18,4	–27,4	$3^x - x^2 - 3$
21	–0,758	–5,04	–2,79	$x^3 - x \ln x - 3$
22	1,56	–0,260	–0,150	$x^3 - x \sin x - 2$
23	8,59	21,0	13,3	$x^3 - x \cos x - 3$
24	5,59	4,14	–3,56	$x \sin x \cos x + x^3 - 4$
25	3,66	–5,88	–20,7	$x^2 - \sqrt{x} - 1$
26	–2,89	–19,4	60,4	$\mathrm{e}^x \tanh x - 4$
27	–0,808	–10,9	17,0	$\sqrt{x} - x\mathrm{e}^x + 5$
28	–4,27	2,35	3,67	$\sqrt{x} + x \ln x - 2$
29	1,16	–7,46	–4,67	$x^3 - \sqrt{x} - 2$
30	–2,24	–3,22	0,133	$x^3 - x \sin x \cos x - 3$

Начало работы с MS Visual C++

1) Создание проекта и добавление файла исходного кода

- Создайте новый проект (в меню **Файл** выберите пункт **Создать** и щелкните **Проект...)**.
- В списке типов проектов **Visual C++** выберите **Win32** и щелкните пункт **Консольное приложение Win32**.
- Введите имя проекта (по умолчанию имя решения, содержащего данный проект, совпадает с именем проекта, однако можно задать другое имя. При необходимости для проекта можно указать иное место размещения). Нажмите **ОК**, чтобы создать проект.
- В окне **Мастер приложений Win32** выберите пункт **Пустой проект** и нажмите кнопку **Готово**.
- Если обозреватель решений не отображается, в меню **Вид** выберите команду **Обозреватель решений**.
- Добавьте в проект новый файл исходного кода (в обозревателе решений щелкните правой кнопкой мыши папку **Файлы исходного кода**, выберите пункт **Добавить** и щелкните **Создать элемент...**. В узле **Код** щелкните пункт **Файл C++ (.cpp)**, введите имя файла и нажмите кнопку **Добавить**). В папке "Файлы исходного кода" в обозревателе решений отобразится файл **.cpp**, и появится окно со вкладками, в котором вводится код для файла.
- Щелкните мышью в созданной вкладке в Visual Studio и введите корректный код программы на Visual C++, в которой используется стандартная библиотека C++. обратите внимание на директиву using namespace std;. Эта директива позволяет программе использовать **cout** и **endl** без указания полных имен (**std::cout** и **std::endl**).
- В меню **Построение** выберите команду **Построить решение (F7)**. Окно **Выходные данные** отображает информацию о ходе выполнения компиляции, такую как размещение журнала построения и сообщение о статусе построения.
- В меню **Отладка** выберите команду **Начать отладку (F5)**.

2) Пример простой программы на C++

```
#include <iostream>
#include <conio.h>
using namespace std;
// Программа вычисляет произведение двух вещественных чисел
int main()
{
  // Директива, позволяющая выводить кириллический текст на экран:
  setlocale(LC_ALL,"Russian");
  double a, b, c;                                  //Объявление всех переменных
  cout << "Введите первый множитель a=";           //Вывод сообщения на экран
  cin >> a;                                        //Ввод значения первого множителя
  cout << "Введите второй множитель b=";           //Вывод сообщения на экран
  cin >> b;                                        //Ввод значения второго множителя
  c=a*b;                                           //Расчет произведения
  cout << "Произведение a*b=" << c << endl;        //Вывод результата на экран
  cout << endl << "Нажмите любую клавишу...";     //Вывод сообщения на экран
  _getch();                                        //Задержка экрана
  return 0;                                        //Возвращаемое значение main()
}
```

- Строки: `#include <iostream>` и `#include <conio.h>` называются директивами препроцессора `include`, которые сообщают компилятору где можно найти информацию о некторых используемых в программе элементах. Внутри скобок указаны названия стандартных библиотек, в которых содержатся описания процедур. В частности, библиотека **iostream** содержит описания процедур, выполняющих ввод с клавиатуры (**cin**) и вывод на экран (**cout**). Состав заголовочных файлов приведен в [1, с. 409–415].
- Следующая строка: `using namespace std;` сообщает, что имена определенные в `iostream` нужно интерпретировать стандартным образом (`std`), иначе вместо команд `cin` и `cout` пришлось бы писать `std::cin` и `std::out`, соответственно.
- Следующая строка: `// Программа вычисляет произведение двух вещественных чисел` называется комментарием к программе. Здесь может быть написано все что угодно. Компилятор, встретив символ `//`, игнорирует все символы, следующие за ним, и переходит к следующей строке.
- Следующие строки содержат собственно код программы. Любая программа на C++ должна содержать хотя бы одну функцию – это функция `int main()` , тело которой ограничивается фигурными скобками. В круглых скобках указываются аргументы функции. В данном случае функция не имеет аргументов. Идентификатор `int` указывает, что функция возвращает целочисленное значение. Для указания на возвращаемое значения используется процедура `return()`. В данном случае функция `main()` не возвращает никакого значения, поэтому в конце программы указывается `return 0;`.
- Строка: `setlocale(LC_ALL,"Russian");` позволяет выводить на экран сообщения, содержащие символы кириллицы (нижеследующие в тексте сообщения в кавычках после оператора вывода `cout <<`).
- Следующая строка в теле основной функции: `double a, b, c;` называется объявлением переменных. Все переменные используемые в программе должны быть объявлены до их использование, иначе в оперативной памяти не будет выделено место для их хранения и при компиляции программы будет выдано сообщение об ошибке. Идентификатор `double` указывает на то, что тип используемых переменных – с плавающей запятой двойной точности, для хранения которого выделятся 8 байт. Типы данных C++ приведены в таблице 1. Описание типов данных и выражений приведен в [2, с. 67–85] и [3, Chapter 2].
- Следующая процедура: `cout << "Введите первый множитель a=";` выводит на экран сообщение, заключенное в двойные кавычки.
- Процедура: `cin >> a;` присваивает, введенное с клавиатуры значение, переменной `a`.
- Процедура: `c=a*b;` вычисляет значение произведения переменных `a` и `b` и присваивает результат переменной `c`. Описание основных операций C++ приведены в таблице 2.
- Строка: `cout << "Произведение a*b=" << c << endl;` выводит результат вычислений на экран. Идентификатор `endl` (end line) после выведенной на строки переводит курсор на следующую строку.
- Процедура: `_getch();` используется здесь для задержки экрана, т.е. чтобы после выполнения программы окно ввода не закрывалось сразу же, а только после нажатия любой клавиши пользователем. Функция `_getch()` читает символ с клавиатуры без вывода на экран. Описание функции содержится в заголовочном файле `conio.h` стандартной библиотеки (поэтому в начале программы необходимо указать директиву препроцессора `#include <conio.h>`).
- Результат работы программы приведен на рисунке 1.

Таблица 1 – Типы данных C++

Тип	Обозначение	Диапазон значений	Размер, байт
логический	bool	true и false	–
знаковый символьный	char (signed char)	–128…127	1
беззнаковый символьный	unsigned char	0…255	1
простой знаковый целый	int (short, short int, signed int, signed short, signed short int)	–32.768…32.767	2
простой беззнаковый целый	unsigned int (unsigned short, unsigned short int)	0…65.535	2
расширенный знаковый целый	long (long int, signed long int)	0…4.294.967.295	4
расширенный беззнаковый целый	unsigned long (unsigned long int)	–2.147.483.648… +2.147.483.647	4
с плавающей запятой простой точности	float	$1{,}1\times10^{-38}$… $3{,}4\times10^{+38}$	4
с плавающей запятой двойной точности	double	$2{,}2\times10^{-308}$… $1{,}7\times10^{+308}$	8
расширенный с плавающей запятой двойной точности	long double	$3{,}3\times10^{-4932}$… $1{,}1\times10^{+4932}$	10

Таблица 2 – Основные операции C++

Операции	Описание	Пример
+ –	сложение вычитание	a+b a–b
* / %	умножение деление остаток от деления	a*b a/b a%b
=	присваивание	a=b
< <= > >=	отношение	a<b a<=b a>b a>=b
== !=	равно не равно	a==b a!=b
&& ||	логическое «И» логическое «ИЛИ»	a>=b && a!=c a==b || a!=c
+= –= *= /= %=	сложение с присваиванием вычитание с присваиванием умножение с присваиванием деление с присваиванием остаток от деления с присваиванием	a+=b (эквивалентно a=a+b) a–=b (эквивалентно a=a–b) a*=b (эквивалентно a=a*b) a/=b (эквивалентно a=a/b) a%=b (эквивалентно a=a%b)
++ ––	увеличение на 1 уменьшение на 1	a++ (эквивалентно a=a+1) a–– (эквивалентно a=a–1)
?:	Операция условие a ? b : c (если a, то b, если не a, то c)	max = (x > y) x : y abs = (x > 0) x : –x

Реализация ветвлений в C++

3) Операторы ветвления (условные операторы)

- ветвление «если-то»:

```
if(условие)
   действие;
```

пример:

```
if(a<0)
   a=-a;
```

- ветвление «если-то-иначе»:

```
if(условие)
   действие_1;
else
   действие_2;
```

пример:

```
if(a>b)
   max=a;
else
   max=b;
```

- ветвление «если-то-иначе»
  с последовательностями действий:

```
if(условие)
   {
   последовательность
   действий;
   }
else
   {
   последовательность
   действий;
   }
```

пример:

```
if(b!=0)
   {
   c=a/b;
   cout << "a/b=" << c << endl;
   }
else
   {
   c=0;
   cout << "Ошибка!\a" << endl;
   }
```

- составное ветвление

```
if(условие_1)
   действие_1;
else if(условие_2)
   действие_2;
   ...
else if(условие_n)
   действие_n;
else
   действие_n+1;
```

пример:

```
if(a[0]==0)
   Zero=1;
else if(a[1]==0)
   Zero=2;
   ...
else if(a[n]==0)
   Zero=n;
else
   Zero=0;
```

- множественный выбор

```
switch(i)
   {
   case значение_1:
      последовательность_действий_1;
      break;
   case значение_2:
      последовательность_действий_2;
      break;
   ...
   case значение_n:
      последовательность_действий_n;
      break;
   default:
      последовательность_действий_n+1;
      }
```

пример:

```
switch(i)
   {
   case 0:
      cout << "Ноль" << endl;
      break;
   case 1:
      cout << "Один" << endl;
      break;
   ...
   case 9:
      cout << "Девять" << endl;
      break;
   default:
      cout << "Это не цифра!" << endl;
   }
```

Описание операторов ветвления приведено в [1, с. 40–44], [2, с. 77–85] и [3, Chapter 3].

4) Операторы циклов

- цикл со счётчиком:

```
for(инициализация; условие; приращение)
  действие;
```

или

```
for(инициализация; условие; приращение)
  {
  последовательность_действий
  (тело_цикла);
  }
```

пример:

```
for(i=9; i>=0; i--)
  cout << i << endl;
```

```
for(i=0; i<10; i++)
  {
  cout << "Введите a[" << i << "]=";
  cin >> a[i];
  }
```

- цикл с предусловием:

```
инициализация;
while(условие)
  {
  последовательность_действий
  (тело_цикла);
  }
```

пример:

```
int a=0;
while(a<b)
  {
  cout << a << endl;
  a++;
  }
```

- цикл с постусловием:

```
инициализация;
do
  {
  последовательность_действий
  (тело_цикла);
  }
  while(условие);
```

пример:

```
double num;
do
  }
  cout << "Введите число (0-стоп): ";
  cin >> num;
  }
  while(num!=0);
```

Описание операторов цикла приведено в [1, с. 44–49], [2, с. 86–93] и [3, Chapter 3].

5) Работа с массивами данных

Код программы заполнения массива и вывода его на экран с использованием циклов:

```
#include <iostream>
using namespace std;
// Программа заполнения массива длиной N и вывода его на экран
int main ()
     {
     setlocale(LC_ALL,"Russian");
     // Объявление пременных и массива:
     int N,i;       // N - длина массива
     double x[10]; // Объявление массива длиной 10
     ////////////////////////////////////////////////////////////////////////
     cout << "Введите длину массива (не более 10): ";
     cin >> N;      // Ввод длины массива
     ////////////////////////////////////////////////////////////////////////
     // Цикл со счетчиком, реализующий поэлементное заполнение массива:
     // Индексация массивов начинается с нуля!!!
     for(i=0;i<N;i++)
          {
          cout << "Введите x[" << i <<"] = ";
          cin >> x[i];
          }
     cout << endl;
     ////////////////////////////////////////////////////////////////////////
     // Цикл со счетчиком, реализующий вывод массива на экран:
```

```
    // Индексация массивов начинается с нуля!!!
    for(i=0;i<N;i++) cout << x[i] << "\t"; //Здесь "\t" означает табуляцию
    cout << endl;
    ////////////////////////////////////////////////////////////////////////////
    system("pause"); //Один из способов задержки экрана

    return 0;
        }
```

Код программы заполнения матрицы и вывода ее на экран с использованием циклов:

```
#include <iostream>
using namespace std;
// Программа заполнения матрицы размером MxN и вывода ее на экран
int main ()
    {
    setlocale(LC_ALL,"Russian");
    // Объявление пременных и матрицы:
    int M, N, i, j;    // M - кол-во строк, N - кол-во столбцов матрицы
    double x[10][10];  // Объявление массива размером 10x10
    ////////////////////////////////////////////////////////////////////////////
    cout << "Введите количество строк матрицы (не более 10):    ";
    cin >> M; // Ввод количества строк матрицы
    cout << "Введите количество столбцов матрицы (не более 10): ";
    cin >> N; // Ввод количества столбцов матрицы
    ////////////////////////////////////////////////////////////////////////////
    // Два цикла со счетчиком, реализующих поэлементное заполнение матрицы:
    // Индексация массивов начинается с нуля!!!
    for(i=0; i<M; i++)
        {
        for(j=0; j<N; j++)
            {
            cout << "Введите x[" << i+1 << "][" << j+1 <<"] = ";
            cin >> x[i][j];    // Ввод значения элемента матрицы
            }
        }
    cout << endl;
    ////////////////////////////////////////////////////////////////////////////
    // Два цикла со счетчиком, реализующих вывод матрицы на экран:
    // Индексация массивов начинается с нуля!!!
    for(i=0; i<M; i++)
        {
        for(j=0; j<N; j++) cout << x[i][j] << "\t"; //Здесь "\t" означает табуляцию
        cout << endl;
        }
    ////////////////////////////////////////////////////////////////////////////
    system("pause"); //Один из способов задержки экрана
    return 0;
        }
```

Как видно из приведенных кодов программ размеры массива и матрицы должны быть заданы при их объявлении, поэтому при вводе значений размеров массива и матрицы возникает ограничение на вводимое значение. В случае, когда заранее длина массива или размеры матрицы не известны можно воспользоваться динамическим массивом, чтобы избежать неопределенности при объявлении массивов и матриц.

Код программы заполнения массива и вывода его на экран на основе использования динамического массива приведен в приложении.

Код программы заполнения матрицы и вывода ее на экран на основе использования динамического массива приведен в приложении.

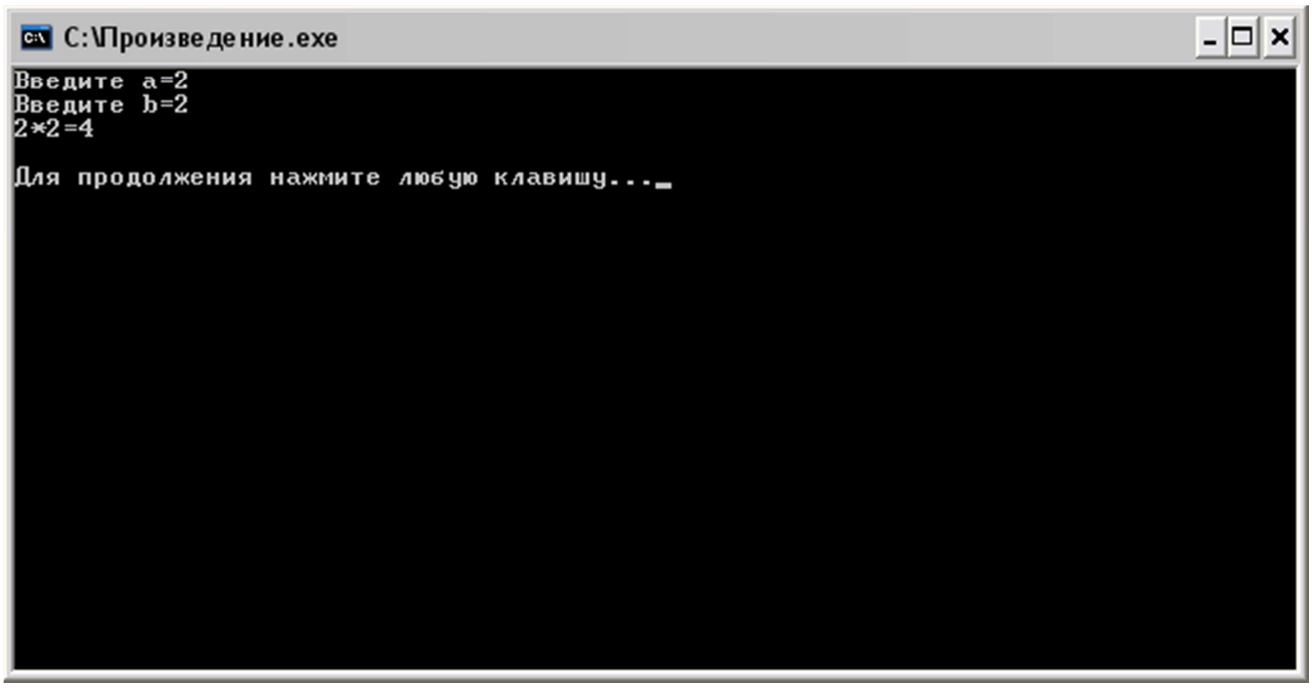

Рисунок 1 – Результат работы программы

6) Создание титульного листа и аннотации к программе

- При разработке программы необходимо вывести на экран титульный лист, затем – краткую аннотацию (назначение программы и описание входных и выходных данных), после чего результат выполнения программы.
- Для вывода текста на экран необходимо пользоваться процедурой:
  `cout << "Текст для вывода" << endl;`
- Для задержки экрана можно использовать процедуру:
  `_getch();`
- Для очистки всего экрана от предыдущего вывода можно использовать команду:
  `system("cls");`

Примеры титульного листа и аннотации к программе приведены на рисунках 2 и 3, соотвественно.

7) Библиографический список

1. Павловская, Т.А. C/C++. Программирование на языке высокого уровня / Т.А. Павловская. – СПб.: Питер, 2003. – 461 с.
2. Савич, У. Программирование на C++ / У. Савич. – СПб.: Питер, 2004. – 781 с.
3. Шилдт, Герб (Schildt, Herb) Руководство по C++ для начинающих (C++ Beginner's Guide). – http://go.microsoft.com/fwlink/?LinkId=115303

8) Содержание отчета:

1. Титульный лист.
2. Текст задания по лабораторной работе № 1 для своего варианта.
3. Текст программы на C++ с комментариями.
4. Контрольный пример (со скриншотом результата работы программы).

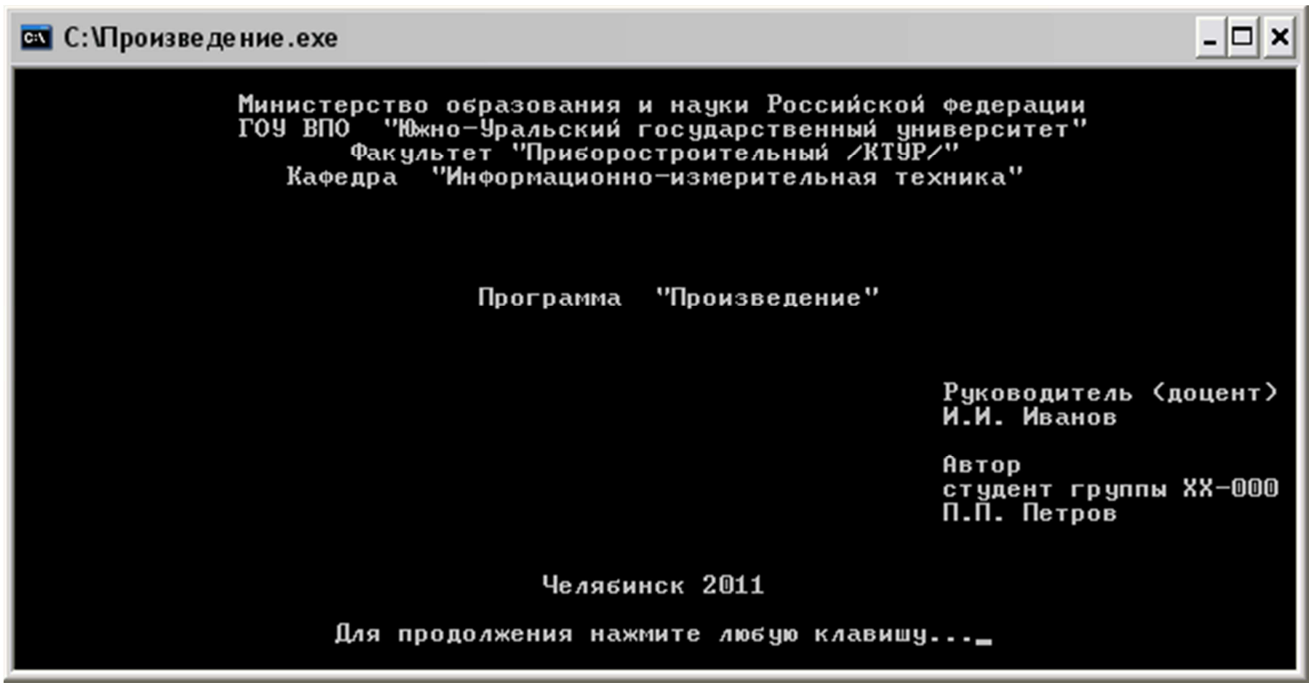

Рисунок 2 – Пример титульного листа программы

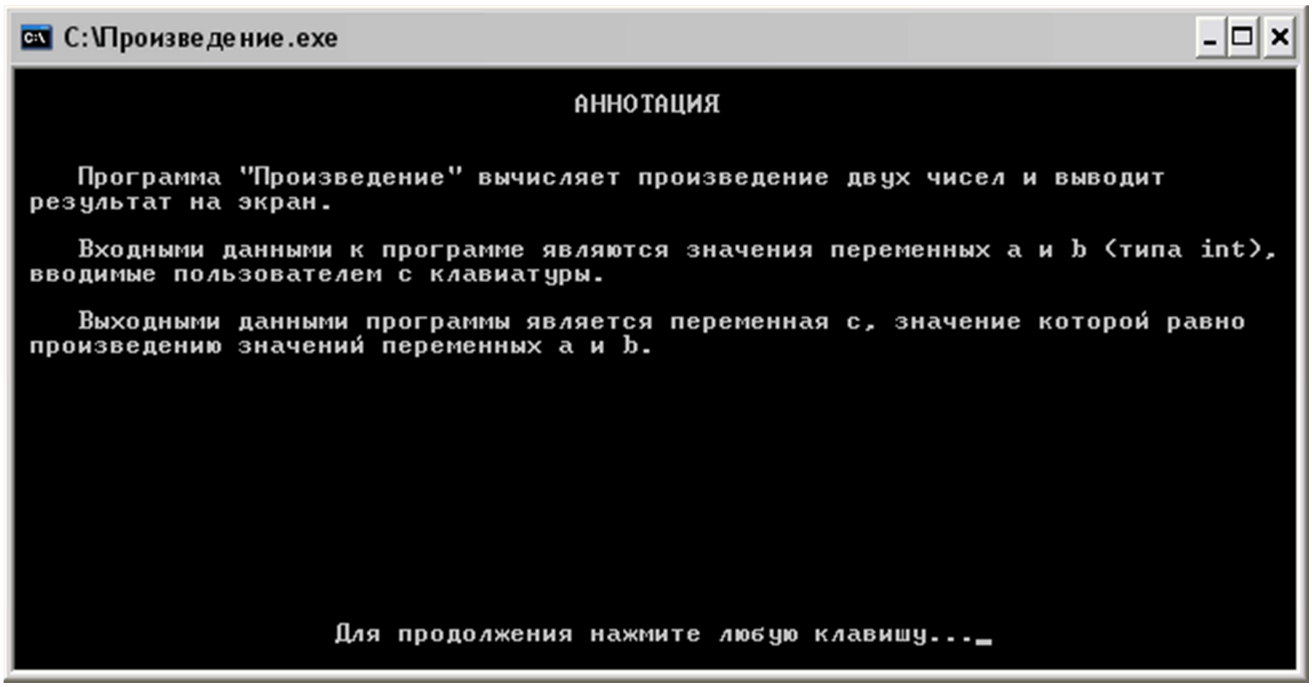

Рисунок 3 – Пример аннотации к программе