Некоммерческое акционерное общество

АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ

Кафедра электроники

АЛГОРИТМИЗАЦИЯ И ПРОГРАММИРОВАНИЕ

Методические указания по выполнению лабораторных работ для студентов специальности 5В071600 – Приборостроение

.

Алматы 2015

СОСТАВИТЕЛИ: Голубева Т.В., Алгоритмизация и программирование. Методические указания по выполнению лабораторных работ для студентов специальности 5В071600 – Приборостроение. - Алматы: АУЭС, 2015.– 32 с.

Методические указания содержат указания по подготовке к проведению лабораторных работ, в них приведены описания каждой лабораторной работы, описаны алгоритмы, приведены блок-схемы и тексты их программной реализации, дана методика проведения, перечень рекомендуемой литературы и контрольные вопросы.

Методические указания предназначены для студентов всех форм обучения специальности 5В071600 – Приборостроение

Ил. 20, библиогр. – 8 назв.

Рецензент: канд. техн. наук, доцент Матаев У.М.

Печатается по плану издания некоммерческого акционерного общества «Алматинский университет энергетики и связи» на 2015 г.

Введение

Данные методические указания предназначены для выполнения лабораторных работ на языке C++ в среде Microsoft Visual Studio. Выполнение лабораторных работ начинается с изучения интерфейса среды разработки с постепенным освоением основ программирования простейших алгоритмов и последующим усложнением задач с целью изучения линейных и нелинейных структур. Данные знания необходимы для приобретения навыков программирования микроконтроллеров.

1 Лабораторная работа №1. Начало работы в приложении Microsoft Visual Studio

Цель работы: изучить интерфейс Microsoft Visual Studio.

Задание: изучить возможности Microsoft Visual Studio в соответствии с данными рекомендациями.

Microsoft Visual Studio — это набор инструментов разработки, основанных на использовании компонентов и других технологий для создания мощных, производительных приложений.

Кроме того, среда Visual Studio оптимизирована для совместного проектирования, разработки и развертывания корпоративных решений.

Также Visual Studio позволяет создавать проекты, имеющие пользовательский интерфейс (GUI), работая с разными компонентами, такими, как формы, кнопки, списки, меню и т.д.

В рамках данного курса мы будем рассматривать лишь программы, работающие в режиме DOS.

Рассмотрим версию Visual Studio 2010.

При загрузке приложения из меню Пуск/Программы/Microsoft Visual Studio 2010 появляется главное окно с начальной страницей программы, которое представлено на рисунке 1.1.

Мы будет создавать программы, которые работают в консоли, т.е. взаимодействие с пользователем происходит посредством черного экрана.

Для создания программы необходимо нажать «Файл» / «Создать» / «Проект» или нажать на стартовой странице Visual Studio2010 быструю ссылку «Создать проект», выделенную синим цветом. После выбора создания нового проекта появится другое диалоговое окно «Создать проект», где необходимо выбрать требуемые опции, а именно: в левом столбце необходимо выбрать Visual C++/Win32, справа наверху – «Консольное приложение Win32», справа внизу вписать имя проекта (например, proga1), в графе расположение выбрать вашу папку, где будут храниться все программы, оставить галочку «Создать каталог» для решения. Диалоговое окно «Создать проект» представлено на рисунке 1.2.

После этого нужно нажать ОК. Откроется мастер настройки нашего будущего консольного приложения, который представлен на рисунке 1.3.

Требуется нажать кнопку «Готово». Тогда в главном окне Visual

Studio 2010 закроется начальная страница и откроется файл proga1.cpp (правое верхнее поле), окно вывода ошибок и предупреждений Вывод (правое нижнее поле), Командный обозреватель (левое вертикальное поле). Данный вид представлен на рисунке 1.4.

Файл proga1.cpp предназначен для текста программы (или кода программы), здесь мы будем вписывать наши операторы, переменные и функции.

Рисунок 1.1 – Начальная страница Visual Studio 2010

Окно вывода пока пусто, т.к. программа еще ни разу не была запущена. После первого запуска в этом окне будет появляться служебная информация, какой проект запускается, что проверяется, есть ли в коде программы ошибки и если есть, то какие. Благодаря данному списку ошибок можно легко найти ошибку в коде программы и исправить. После исправления ошибок следует перезапустить программу на проверку еще раз. Когда ошибки не будут обнаружены, программа запустится на выполнение задачи и появится консоль.

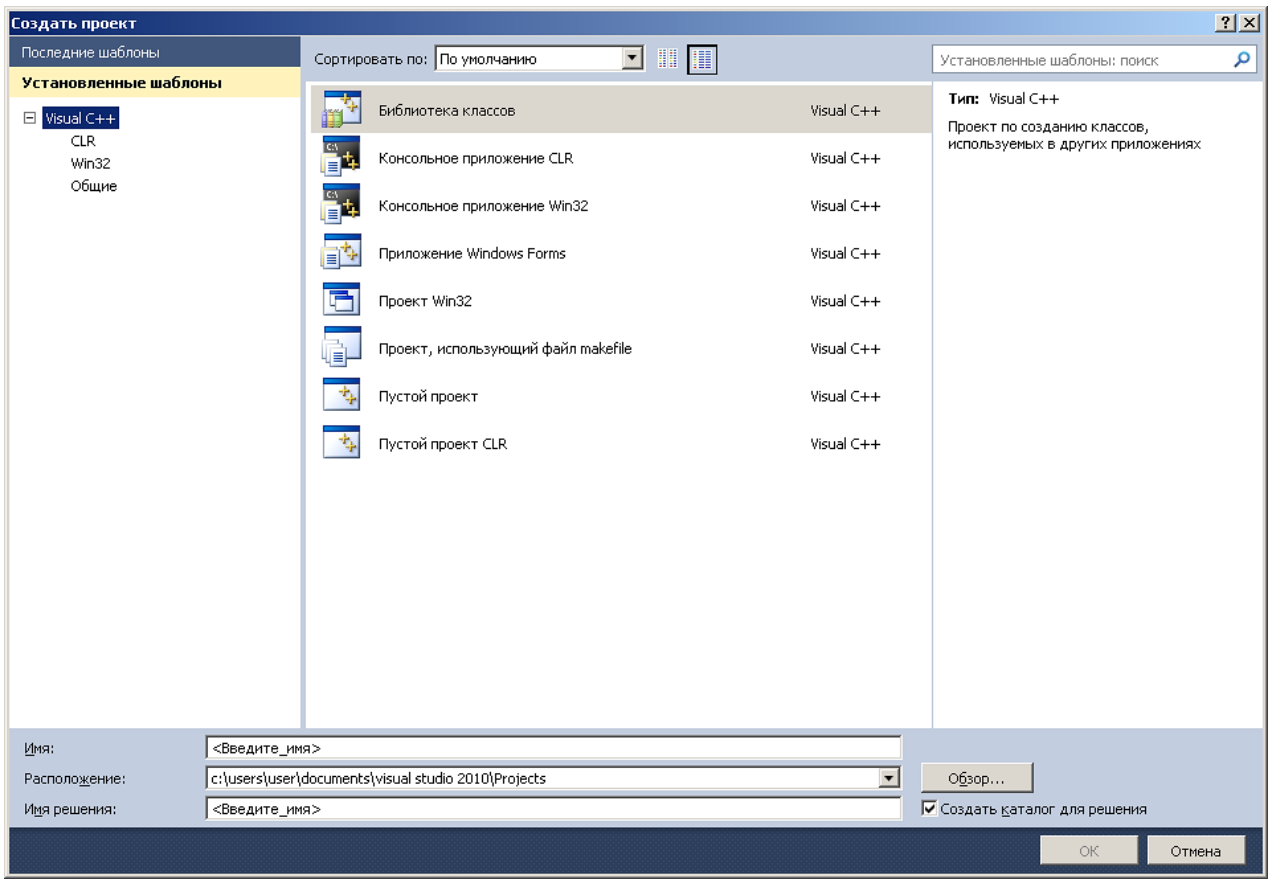

Рисунок 1.2 – Диалоговое окно Создать проект

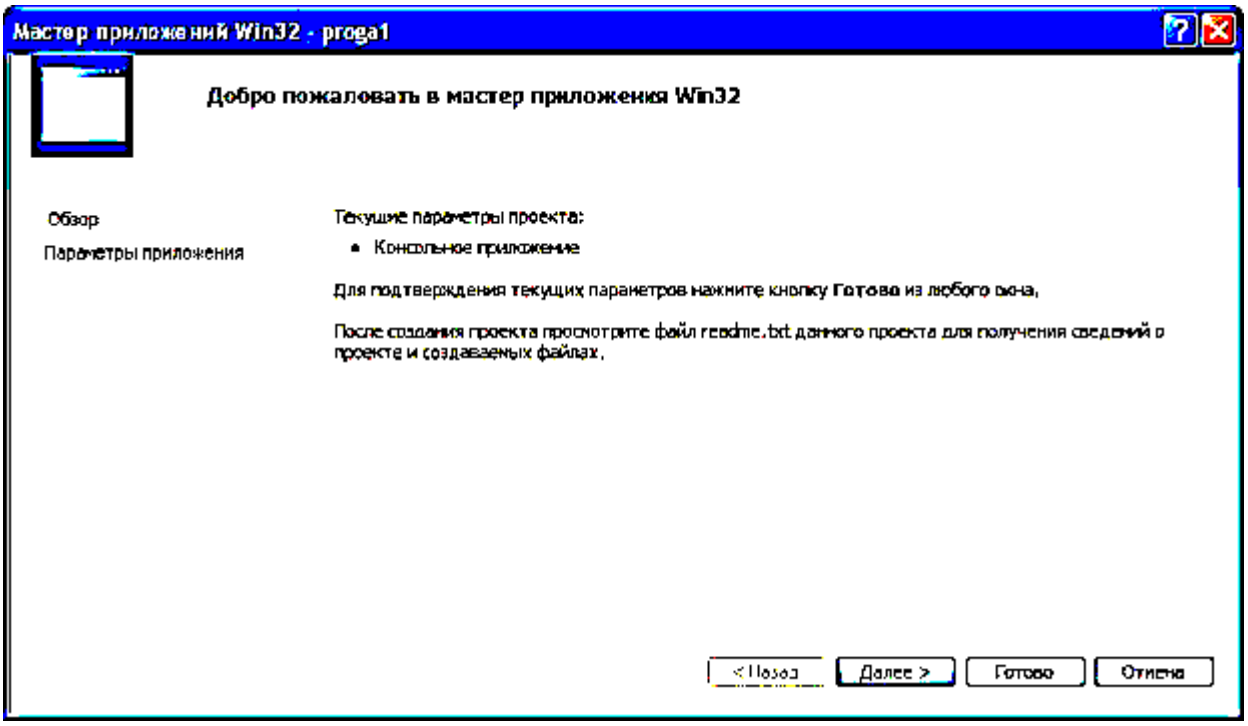

Рисунок 1.3 – Окно Мастер приложений Win32

Вместо командного обозревателя можно включить Обозреватель решений, в котором видны все файлы и папки, созданные для нашего нового проекта. Для этого необходимо под Командным обозревателем нажать кнопку Обозреватель решений. Далее в появившемся списке раскрыть папки

Файлы исходного кода и Заголовочные файлы. Окно Visual Studio с Обозревателем решений представлено на рисунке 1.5.

Когда мы составляем программу в Visual Studio, получается целый проект, который автоматически создается средой Visual Studio. Задача начинающего программиста состоит в том, чтобы напечатать код своей программы в файле *.cpp и запустить программу на выполнение.

Рассмотрим заготовку кода, которую нам предлагает Visual Studio.

Первые две строчки начинаются двумя символами "//". Данные символы означают, что далее на этой строчке следует комментарий; он не воспринимается компилятором как код программы и не будет выдавать ошибку. Удалять эти две строчки не рекомендуется.

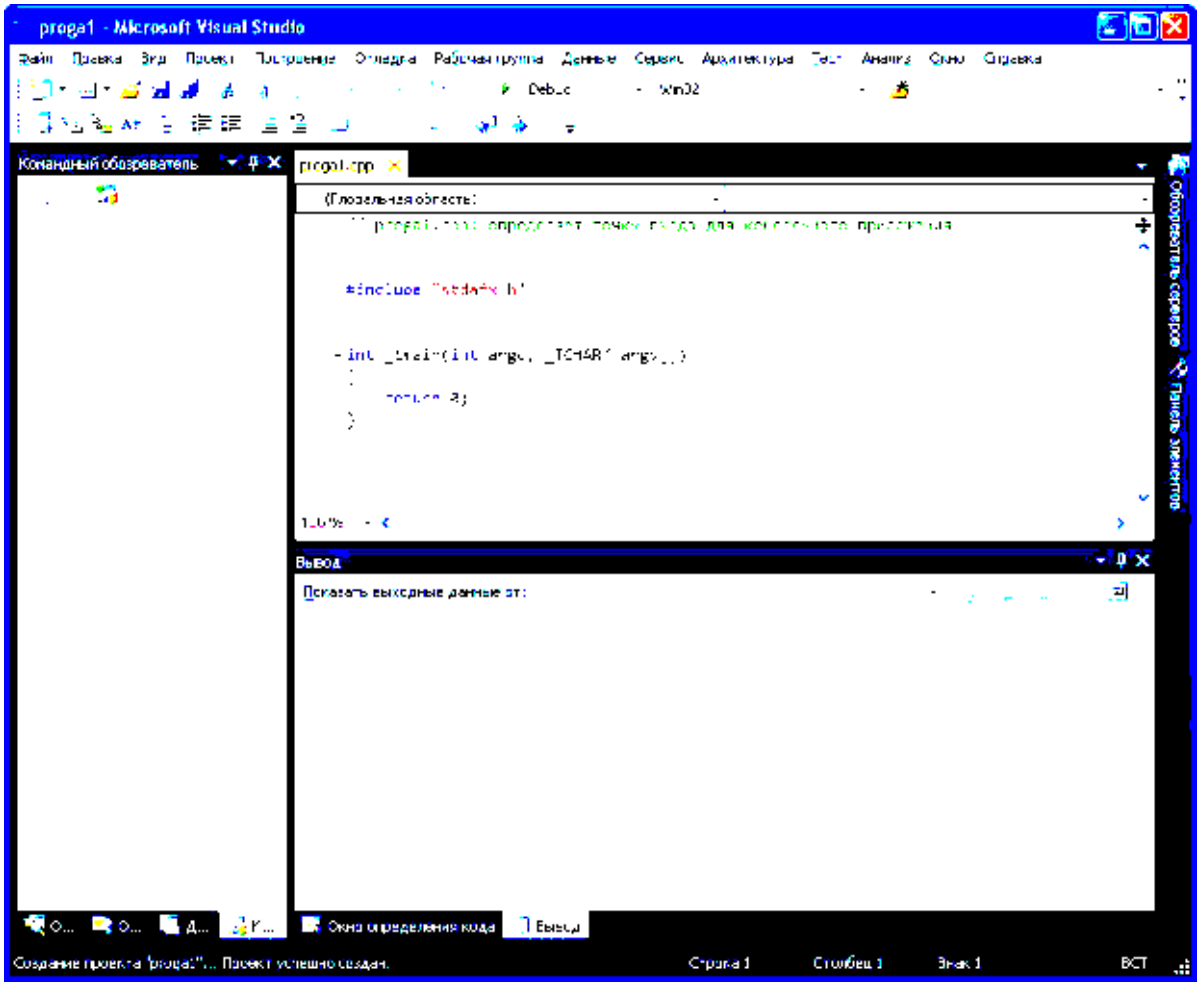

Рисунок 1.4 – Заготовка нового проекта в Visual Studio 2010

Далее следует строчка #include "stdafx.h". Данная строка необходима для сборки нашего проекта. Файл stdafx.h – это один из файлов, автоматически создаваемых для нашей программы, и его можно увидеть в левом столбце Обозревателя решений. Эту строчку также нельзя удалять.

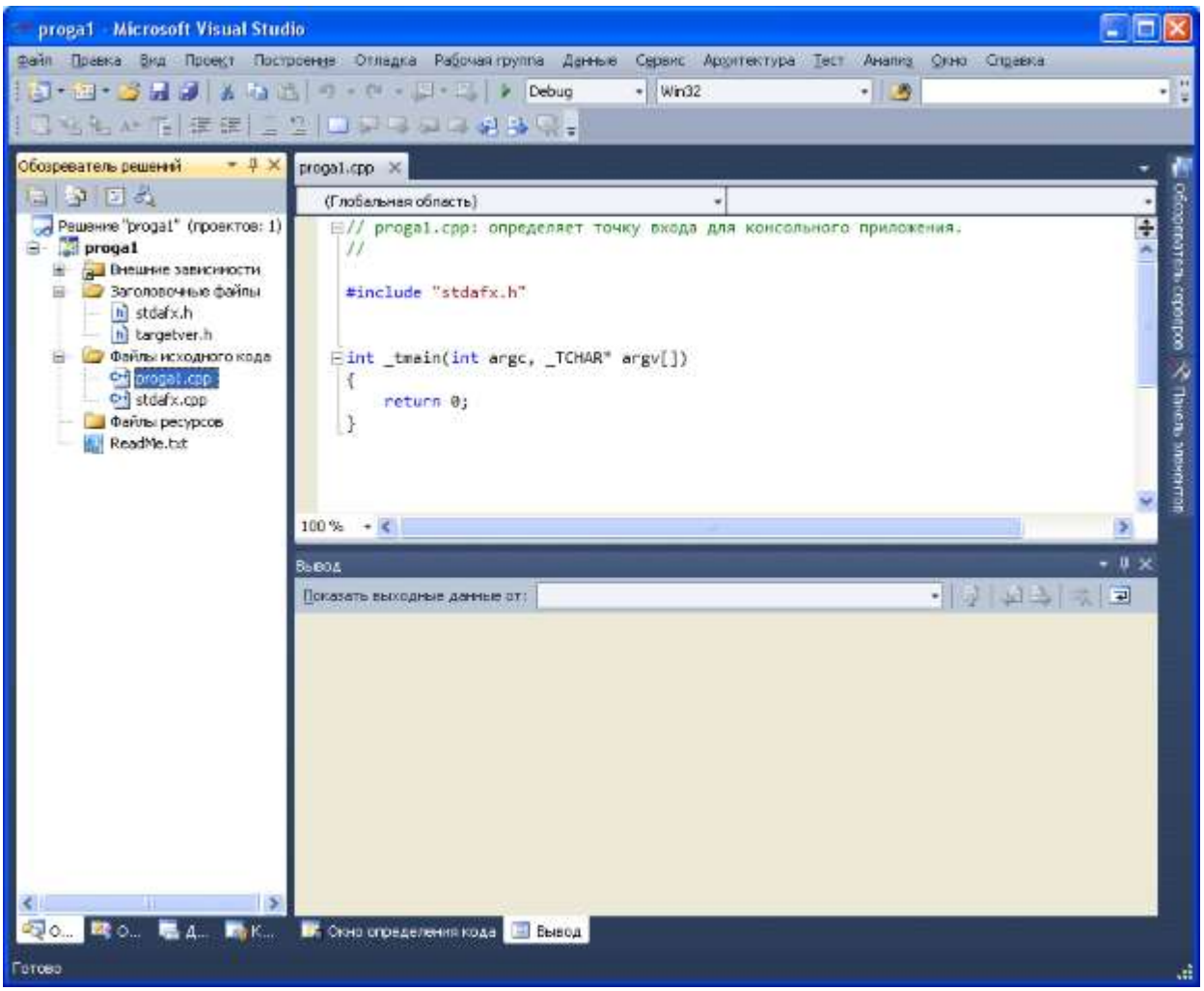

Рисунок 1.5 – Просмотр Обозревателя решений

Далее начинается функция int main(). Автоматически Visual Studio называет ее _tmain и вписывает аргументы int argc, _TCHAR* argv[]. У нас будут самые простые программы для начинающих программистов, поэтому мы изменяем эту строчку и приводим ее к виду: int main() и оставляем return 0, или void main() и стираем return 0. После проведенной нами подготовки необходимо собрать проект, нажав в меню Построение/Построить решение. Тогда в поле Вывод начинается проверка нашего кода. Сейчас проверка прошла успешно и проект построился. Окно Visual Studio после сборки проекта представлено на рисунке 1.6.

Попробуем теперь запустить нашу пустую программу. Для этого на клавиатуре необходимо нажать Ctrl+F5. Тогда появится консоль со стандартной надписью после выполнения программы "Для продолжения нажмите любую клавишу…". Консоль представлена на рисунке 1.7.

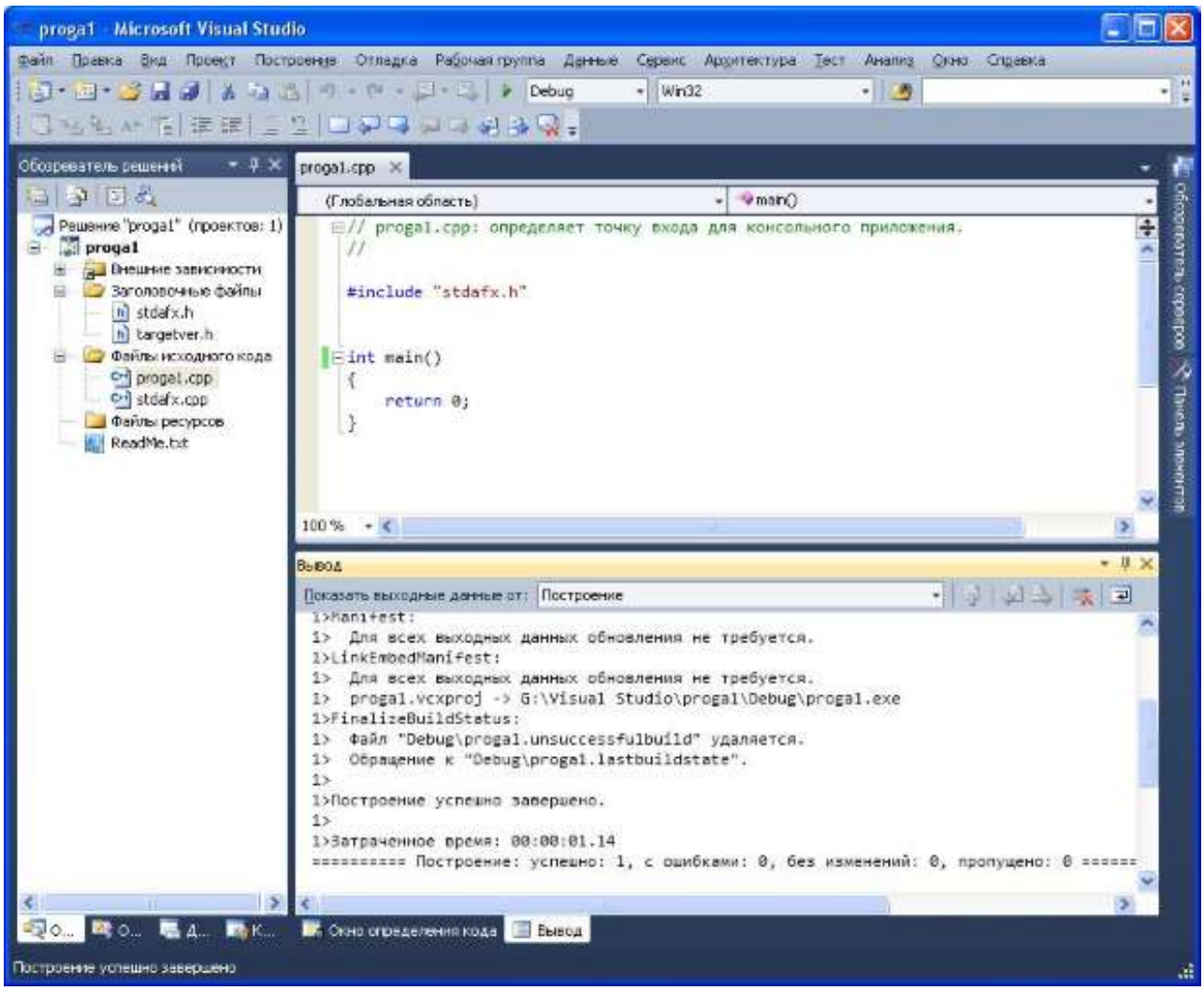

Рисунок 1.6 – Успешное построение проекта

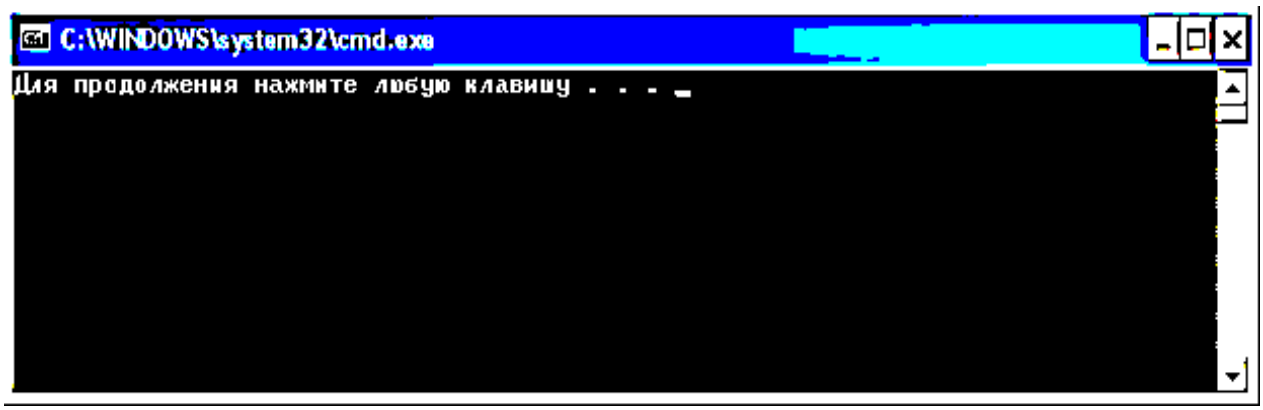

Рисунок 1.7 – Консоль выполнения программы

После просмотра консоли закройте ее, нажав на крестик.

Теперь можно составить простейшую программу, которая напишет на экране фразу "PRIVET, STUDENT!!!". В код нашей заготовки нужно добавить несколько строк:

#include <iostream> // т.к. нужно будет использовать оператор вывода на экран cout

using namespace std; // подключение пространства имен

cout<<"PRIVET, STUDENT!!!"<<endl; // вывод фразу на экран и перевод курсора на новую строку,
// чтобы стандартная фраза "Для продолжения нажмите любую клавишу…" не "налипла" на нашу фразу.

Итак, итоговый код программы представлен на рисунке 1.8.

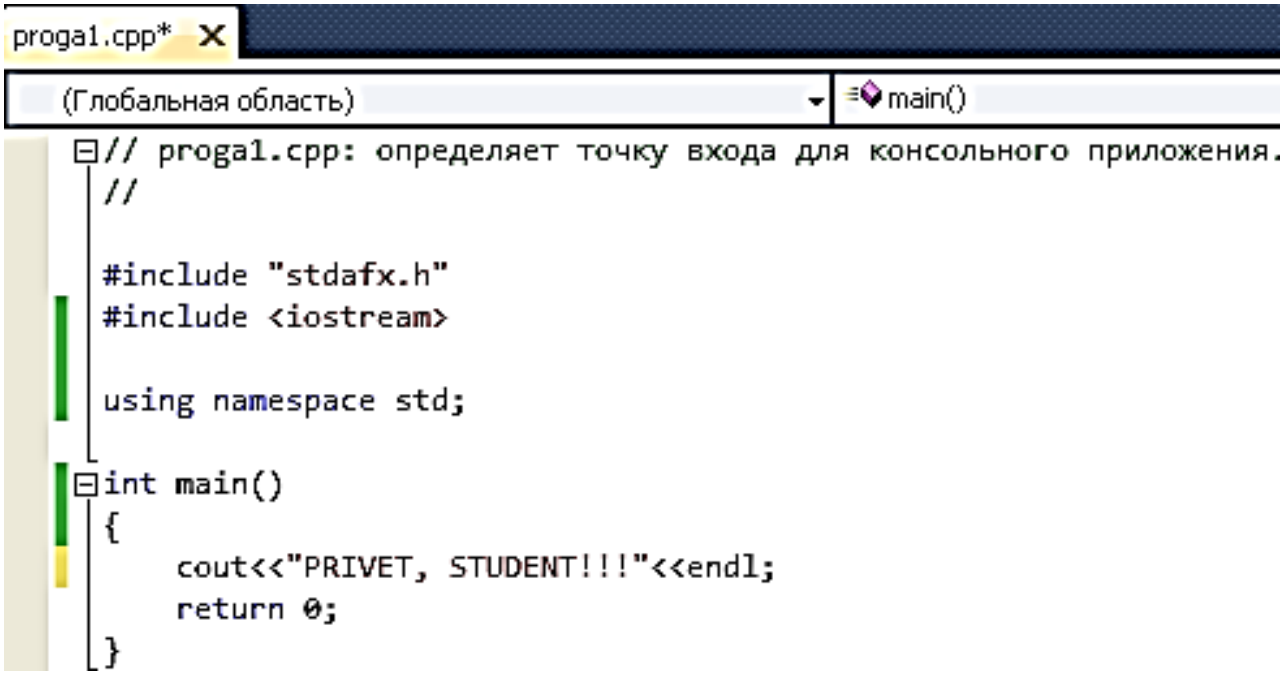

Рисунок 1.8 – Файл с кодом программы

После этого опять строим решение, т.к. код программы был изменен. Если построение выполнилось успешно, то можно запускать программу, нажав Ctrl+F5. Если же построение выдало ошибки, то нужно еще раз все проверить, исправить код, после этого построить проект еще раз. Консоль представлена на рисунке 1.9.

Рисунок 1.9. Консоль выполнения программы

В процессе работы с приложением Visual Studio может возникнуть множество вопросов. В меню программы предусмотрена справка и стандартные примеры, которые можно посмотреть. Чтобы включить справку, необходимо нажать Справка/Просмотрсправки или воспользоваться горячими клавишами Ctrl+F1. Чтобы посмотреть примеры кодов, в меню нужно выбрать Справка/Примеры.

Контрольные вопросы

1. Какие языки программирования позволяет использовать Microsoft Visual Studio?
2. Какие функции реализованы в модуле «Обозреватель решений»?
3. Каким сочетанием клавиш запускается выполнение проекта?
4. Назначение модуля «Мастер приложений.

2 Лабораторная работа №2. Программирование линейных структур. Простейшие типовые задачи.

Цель работы: изучить принципы программирования линейных структур.

Задание: изучить возможности программирования линейных структур в соответствии с данными рекомендациями.

Составить блок-схему и программу для вычисления значений функций $y = \sin x$ и $z = \ln x$ при x, который считывается с экрана (клавиатуры).

Приведем блок-схему на рисунке 2.1.

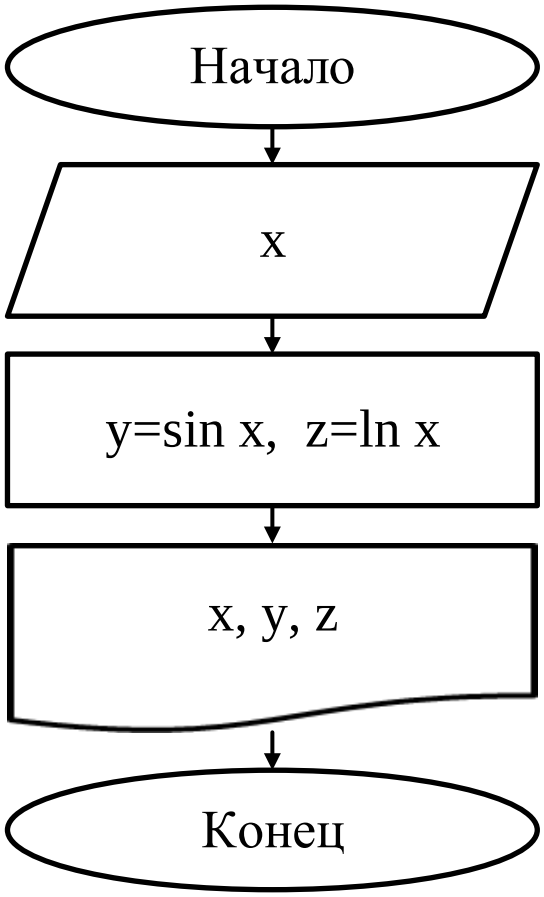

Рисунок 2.1 – Блок-схема

Код программы (Visual Studio):

```
// proga11.cpp: определяет точку входа для консольного приложения.
//
```

```
#include "stdafx.h"
#include <iostream>
#include <math.h>
using namespace std;
int main(){
        double x, y, z;
        cout<<"vvedi x=";
        cin>>x;
        y=sin(x); z=log(x);
        cout<<"pri x="<<x<<" y="<<y<<" z="<<z<<endl;
        return 0;
}
```

Результат выполнения программы (Visual Studio) приведен на рисунке 2.2:

Рисунок 2.2 – Результат выполнения программы

Задание: изменить данную программу и вычислить дополнительно функции $z_{2=}yz$ и $z_3 = \frac{y}{z}$.

Контрольные вопросы

1. В каком модуле расположены описания основных математических функций?
2. Каков приоритет операторов ¦ ¦ и &&, и как вычисляются выражения, связанные этими операторами?
3. В чем смысл использования именованных констант?
4. Какие типы блоков бывают?

3 Лабораторная работа №3. Программирование нелинейных структур. Переходы по условию.

Цель работы: изучить принципы программирования нелинейных структур.

Задание: изучить возможности программирования нелинейных структур с переходами по условию в соответствии с данными рекомендациями.

Вычислить значение функции

$$Z(x,y)=\begin{cases} x+y, & \text{если } xy<0 \\ x-y, & \text{если } 0\le xy\le 5 \\ y, & \text{если } xy>5 \end{cases}$$

при различных значениях аргументов x и y (аргументы x и y считать с клавиатуры).

Решение. В данном примере функция Z определена на трех промежутках. Zзависит от двух переменных x и y.

Блок-схема представлена на рисунке 3.1.

```
// proga19.cpp: определяет точку входа для консольного приложения.
//
#include "stdafx.h"
#include <iostream>
using namespace std;
int main(){
	double x, y, Z;
	cout<<"x=";
	cin>>x;
	cout<<"y=";
	cin>>y;
	if(x*y<0){ Z=x+y;}
	else {
		if(x*y<=5){
			Z=x-y;
		}
		else {
			Z=y;
		}
	}
	cout<<"Z="<<Z<<endl;
	return 0;
}
```

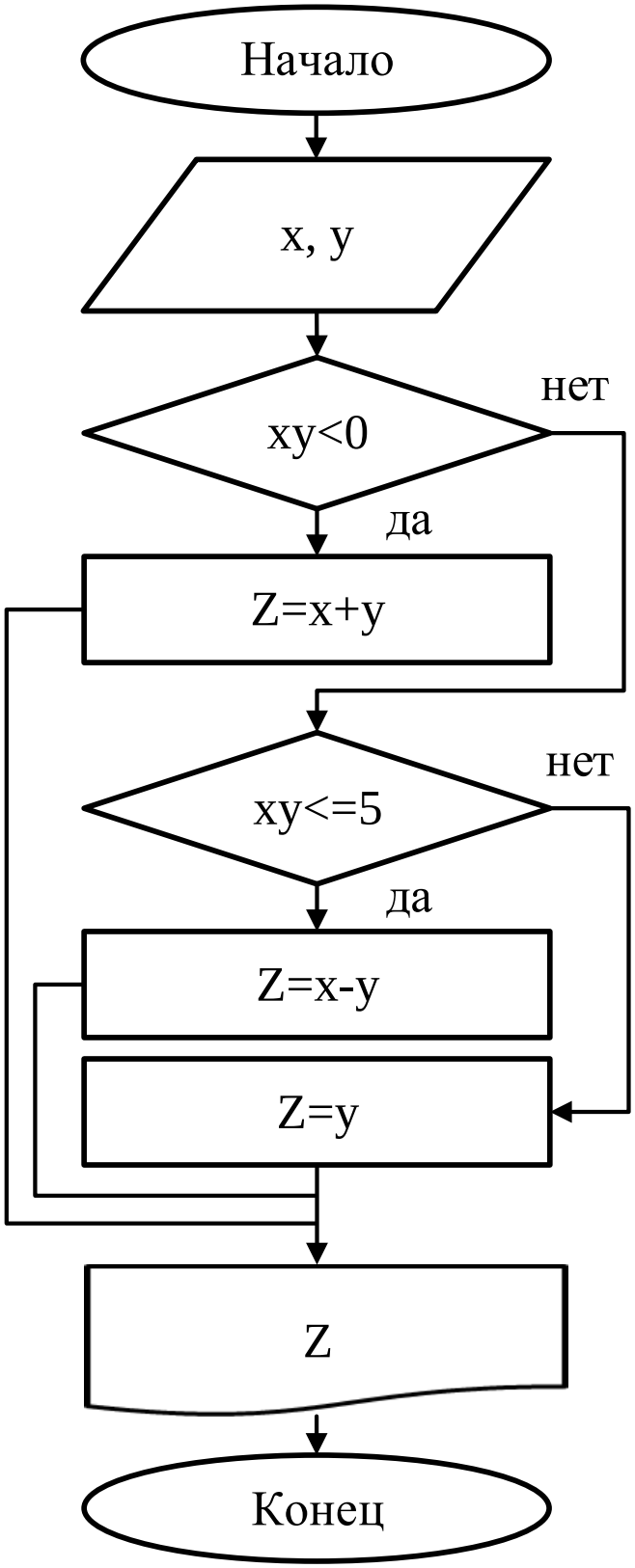

Рисунок 3.1 – Блок-схема

Код программы (Visual Studio):

Результат выполнения программы (Visual Studio) представлен на рисунке 3.2:

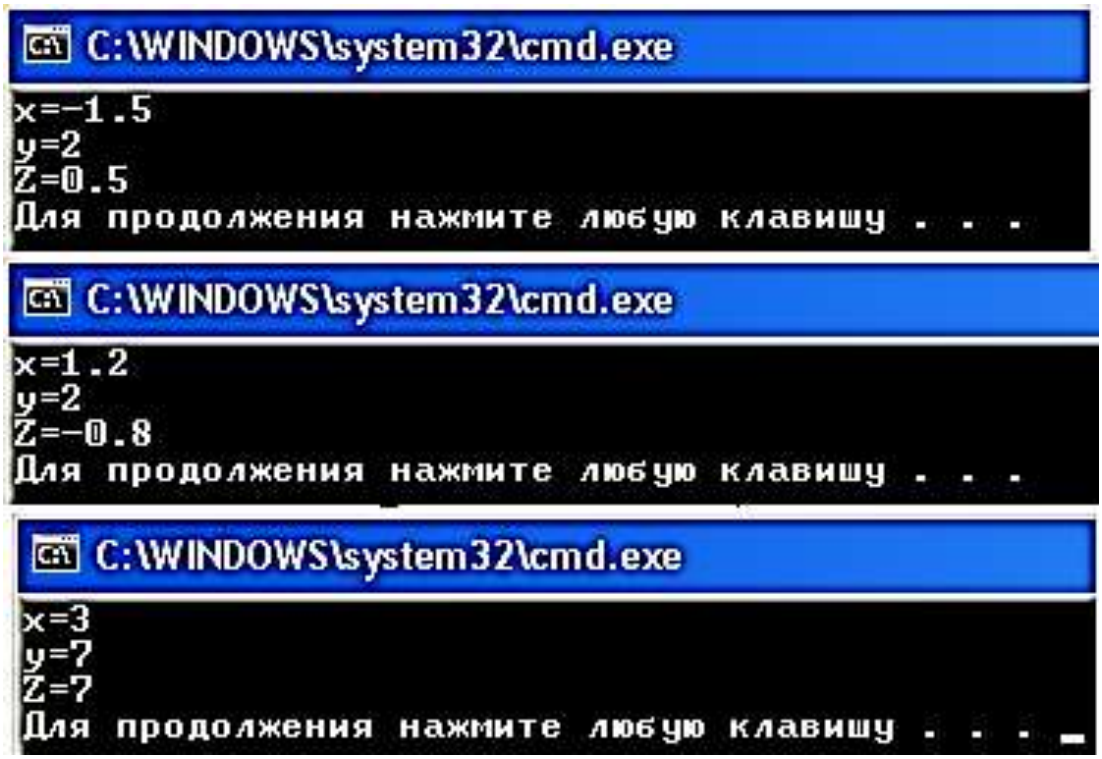

Рисунок 3.2 – Результат выполнения программы

Задание: найти наибольшее (максимальное) из трех чисел a, b, c, числа ввести с клавиатуры.

Контрольные вопросы

1. Какова может быть вложенность операторов ветвления?
2. Напишите пример оператора выбора?
3. Напишите пример условного оператора в полной форме ?
4. Напишите пример тернарной операции

4 Лабораторная работа №4. Программирование нелинейных структур. Применением операторов цикла с параметром

Цель работы: изучить принципы программирования нелинейных структур.

Задание: изучить возможности программирования нелинейных структур с операторами цикла с параметром в соответствии с данными рекомендациями.

Вычислить таблицу "x, y, Z" значений функции

$$Z(x,y) = \begin{cases} x + y, & xy < 1 \\ x - y, & xy \geq 1 \end{cases}$$

при $-1 \leq x \leq 0{,}5$ с шагом 1, $5 \leq y \leq 15$ с шагом 5 .

Решение. В данной задаче изменяются две переменные: x и y. Следовательно, нужно реализовать двумерный цикл: внешний цикл по

переменной x, внутренний цикл по переменной y. Тело цикла будет содержать вычисление функции Z и вывод на экран строчки значений x, y, Z.

В данной задаче имеем:

$x = -1$ – начальное значение переменной внешнего цикла x;

$hx = 1$ – шаг изменения переменной внешнего цикла x;

"$x \leq 0{,}5$" - условие для выполнения итерации внешнего цикла по x;

$y = 5$ – начальное значение переменной внутреннего цикла y;

$hy = 5$ – шаг изменения переменной внутреннего цикла y;

"$y \leq 15$" - условие для выполнения итерации внутреннего цикла по y.

Блок-схема с предусловием приведена на рисунке 4.1.

Код программы с оператором for:

```
// proga25for.cpp: определяет точку входа для консольного приложения.
//
#include "stdafx.h"
#include <iostream>
#include <iomanip>
using namespace std;
int main(){
        double x, y, Z;
        cout<<setw(10)<<"x"<<setw(10)<<"y"<<setw(10)<<"Z"<<endl;
        for(x=-1; x<=0.5; x=x+1){
                for(y=5; y<=15; y=y+5){
                        if(x*y<1){
                                Z=x+y;
                        }
                        else{
                                Z=x-y;
                        }

        cout<<setw(10)<<x<<setw(10)<<y<<setw(10)<<Z<<endl;
                }
        }
        return 0;}
```

Задание: изменить в данной программе шаг для переменных x до 0,5; y до 1. Произвести вычисления.

Контрольные вопросы

1. К какому виду относится оператор цикла for?
2. Когда удобнее использовать оператор цикла for?
3. Напишите пример цикла с оператором for в полной форме.
4. Как задается шаг в цикле с оператором for ?

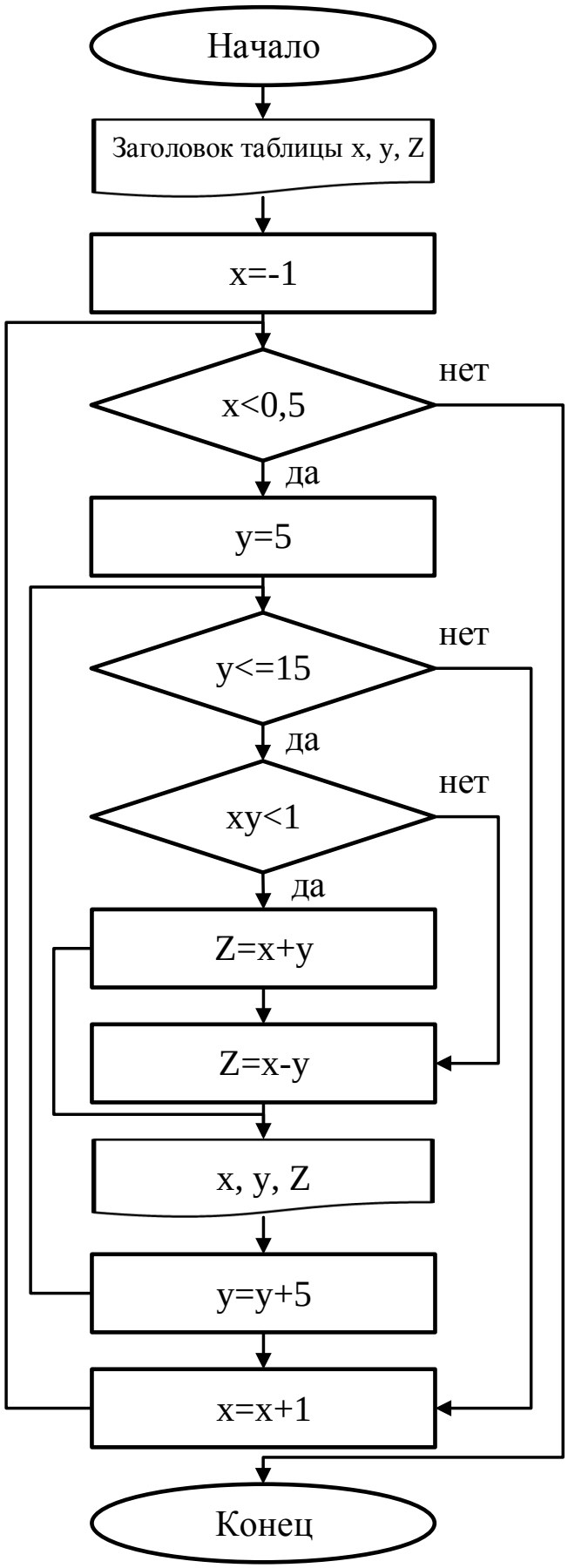

Рисунок 4.1 – Блок-схема с предусловием

5 Лабораторная работа №5. Программирование нелинейных структур. Применением операторов цикла с предусловием и постусловием

Цель работы: изучить принципы программирования линейных структур.

Задание: изучить возможности программирования линейных структур в соответствии с данными рекомендациями.

Вычислить таблицу "x, y, Z" значений функции

$$Z(x,y) = \begin{cases} x + y, & xy < 1 \\ x - y, & xy \geq 1 \end{cases}$$

при $-1 \leq x \leq 0{,}5$ с шагом 1, $5 \leq y \leq 15$ с шагом 5 .

Решение. В данной задаче изменяются две переменные: x и y. Следовательно, нужно реализовать двумерный цикл: внешний цикл по переменной x, внутренний цикл по переменной y. Тело цикла будет содержать вычисление функции Z и вывод на экран строчки значений x, y, Z.

В данной задаче имеем:

$x = -1$ – начальное значение переменной внешнего цикла x;
$hx = 1$ – шаг изменения переменной внешнего цикла x;
"$x \leq 0{,}5$" - условие для выполнения итерации внешнего цикла по x;
$y = 5$ – начальное значение переменной внутреннего цикла y;
$hy = 5$ – шаг изменения переменной внутреннего цикла y;
"$y \leq 15$" - условие для выполнения итерации внутреннего цикла по y.
Блок-схема с постусловием приведена на рисунке 5.1.

Код программы с оператором while:

```
// proga25while.cpp: определяет точку входа для консольного приложения.
//
#include "stdafx.h"
#include <iostream>
#include <iomanip>
using namespace std;
int main(){
    double x, y, Z;
    cout<<setw(10)<<"x"<<setw(10)<<"y"<<setw(10)<<"Z"<<endl;
    x=-1;
    while(x<=0.5){
        y=5;
        while(y<=15){
            if(x*y<1){
                Z=x+y;
            }
```

```
                    else{
                            Z=x-y;
                    }

        cout<<setw(10)<<x<<setw(10)<<y<<setw(10)<<Z<<endl;
                    y=y+5;
                }
                x=x+1;
        }
        return 0;
    }
```

Задание: изменить в данной программе шаг для переменных x до 0,5; y до 1. Произвести вычисления.

Контрольные вопросы

1. Основное отличие оператора цикла с постусловием от оператора с предусловием.
2. Когда рациональнее использовать оператор цикла с постусловием?
3. Возможно ли написание по одному заданию решений с постусловием и предусловием по выбору программиста?
4. В каких случаях может произойти зацикливание при использовании оператора цикла с постусловием?

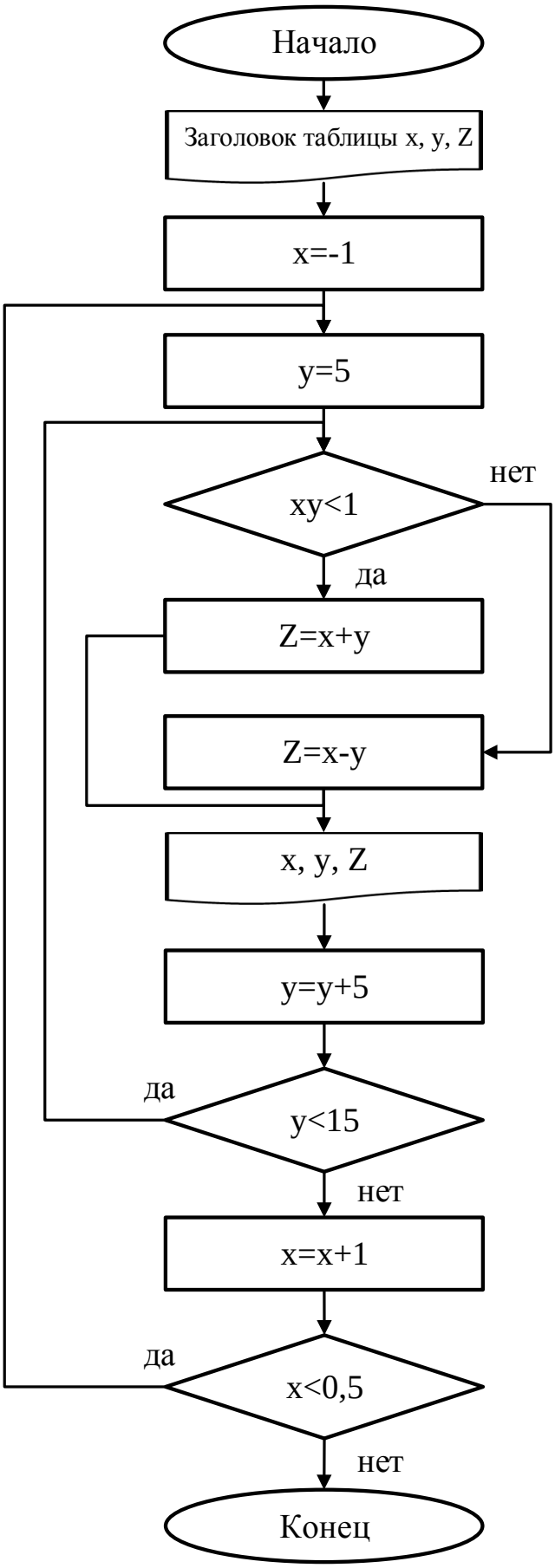

Рисунок 5.1 – Блок-схема с постусловием

6 Лабораторная работа №6. Работа с одномерными массивами

Цель работы: изучить принципы работы с одномерными массивами.

Задание: изучить возможности программной обработки одномерных массивов.

Массив $a(10)$ задан формулой $a_i = 3i - 5$. Вычислить сумму положительных элементов массива и поменять местами первый и последний элементы.

Решение.

Обозначим за S сумму положительных элементов массива. При расчете S требуется дополнительное условие: "$a_i > 0$". Первый элемент массива имеет индекс 0, последний элемент имеет индекс 9, поэтому будем менять местами a_0 и a_9. Из-за перемены мест элементов массив изменится, поэтому выведем его еще раз.

Блок-схема представлена на рисунках 6.1 и 6.2.

Код программы (Visual Studio) с оператором for:

```
// proga28.cpp: определяет точку входа для консольного приложения.
//
#include "stdafx.h"
#include <iostream>
#include <iomanip>
using namespace std;
int main(){
	double S, tmp, a[10];
	int i;
	S=0;
	for(i=0; i<10; i=i+1){
		a[i]=3.0*i-5.0;
		cout<<setw(3)<<a[i];
		if(a[i]>0){
			S=S+a[i];
		}
	}
	cout<<endl;
	cout<<"S="<<S<<endl;
	tmp=a[0];
	a[0]=a[9];
	a[9]=tmp;
	for(i=0; i<10; i=i+1){ cout<<setw(3)<<a[i]; }
	cout<<endl;
	return 0;}
```

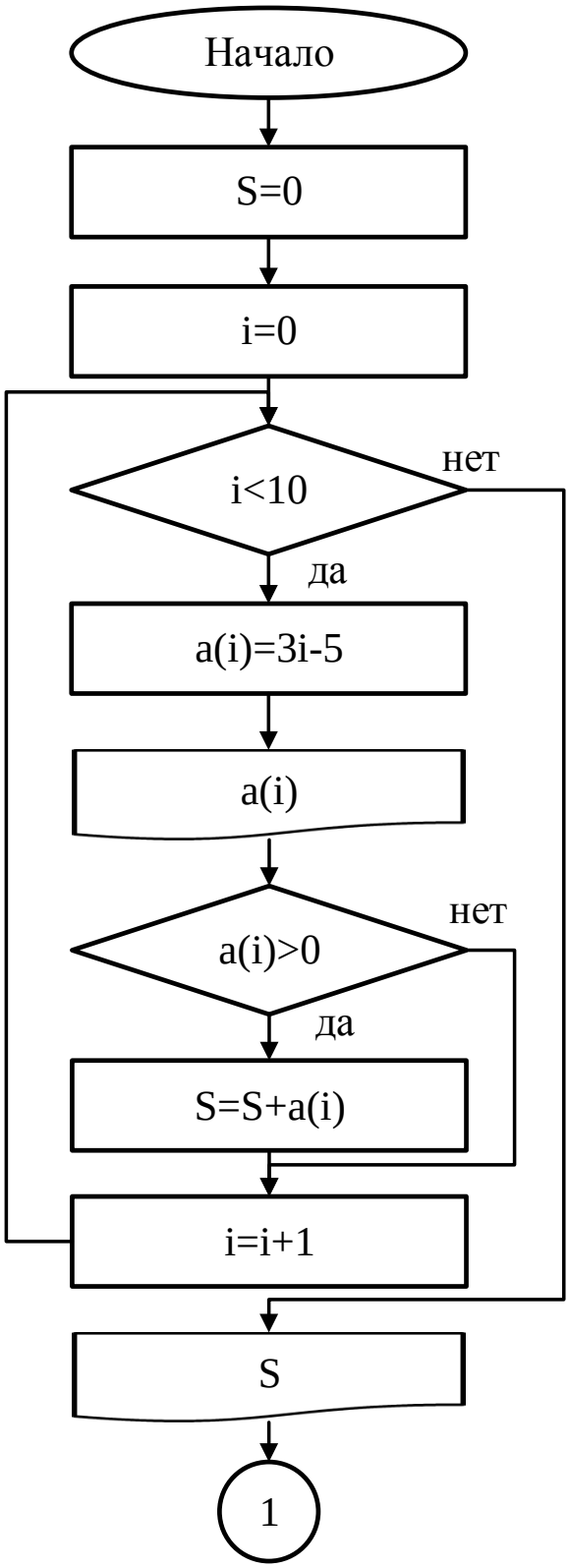

Рисунок 6.1 – Блок-схема

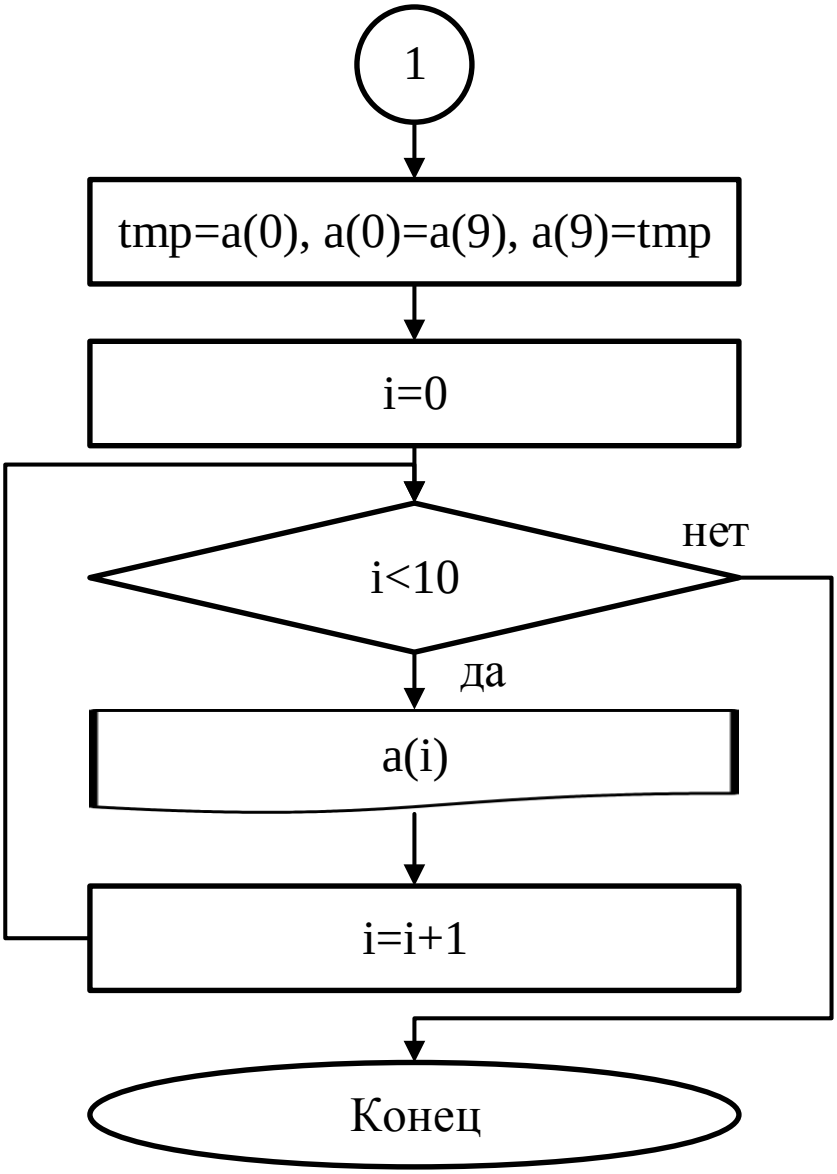

Рисунок 6.2 – Продолжение блок-схемы

Результат выполнения программы представлен на рисунке 6.3.

Рисунок 6.3 – Результат выполнения программы

Задание: вычислить сумму положительных элементов массива и поменять местами первый и пятый элементы.

Контрольные вопросы

1. Какие типы данных может содержать массив?
2. Приведите примеры инициализации массива?
3. Что такое массив?
4. Приведите пример прохождения указателя по одномерному массиву?

7 Лабораторная работа №7. Работа с двумерными массивами

Цель работы: изучить принципы работы с двумерными массивами.

Задание: изучить возможности программной обработки двумерных массивов.

Массив $a(6,8)$ задан формулой $a_{i,j} = 3i - 5j$. Найти сумму элементов во второй строке, поменять местами первую и третью строки, найти произведение по формуле $P = \Pi_{i=0}^{2}(a_{i,1} - a_{i,0})$.

Решение. Данную задачу можно разбить на несколько этапов:

1) задать массив по формуле и вывести его на экран;
2) найти сумму элементов во второй строке и вывести ее на экран;
3) поменять местами первую и третью строки;
4) вывести на экран измененный массив;
5) найти произведение по формуле и вывести его на экран.

Каждый указанный этап решается с помощью циклов. Все циклы целесообразно использовать с предусловием (в программе – оператор for). На первом этапе следует организовать двумерный цикл по i, по j, т.к. необходимо задать значениями и вывести на экран весь массив. На втором этапе перед циклом необходимо задать $S = 0$, а в теле цикла по j насчитывать значение S.Индекс i задаем как 1, что соответствует второй строке. Следует отметить, что нельзя считать сумму по какой-то определенной строке (столбце, диагонали) внутри двумерного цикла, т.к. в этом случае S будет больше в разы (зависит от количества строк/столбцов).На третьем этапе индексы строк берем $i = 0$ – для первой строки, $i = 2$ – для третьей строки. Замена строк организуется в одномерном цикле по j. На четвертом этапе матрица a должна быть отображена на экране. Нам требуется показать, что замена строк выполнена верно. На пятом этапе перед циклом следует задать $P = 1$, а в теле цикла насчитывать произведение. Цикл организуем одномерный по i от 0 до 2 включительно, т.к. согласно формуле изменяется только i.

Блок-схема для примера приведена на рисунках 7.1 – 7.4 .

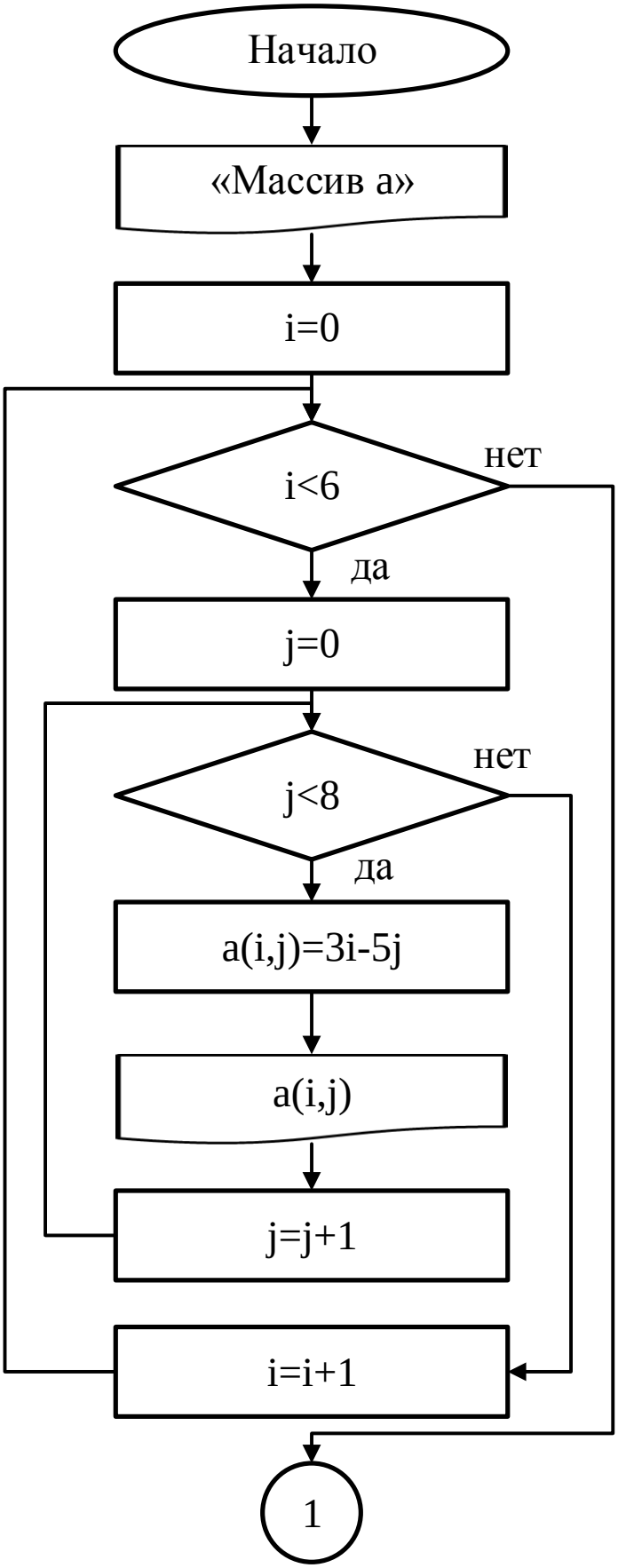

Рисунок 7.1 – Блок-схема

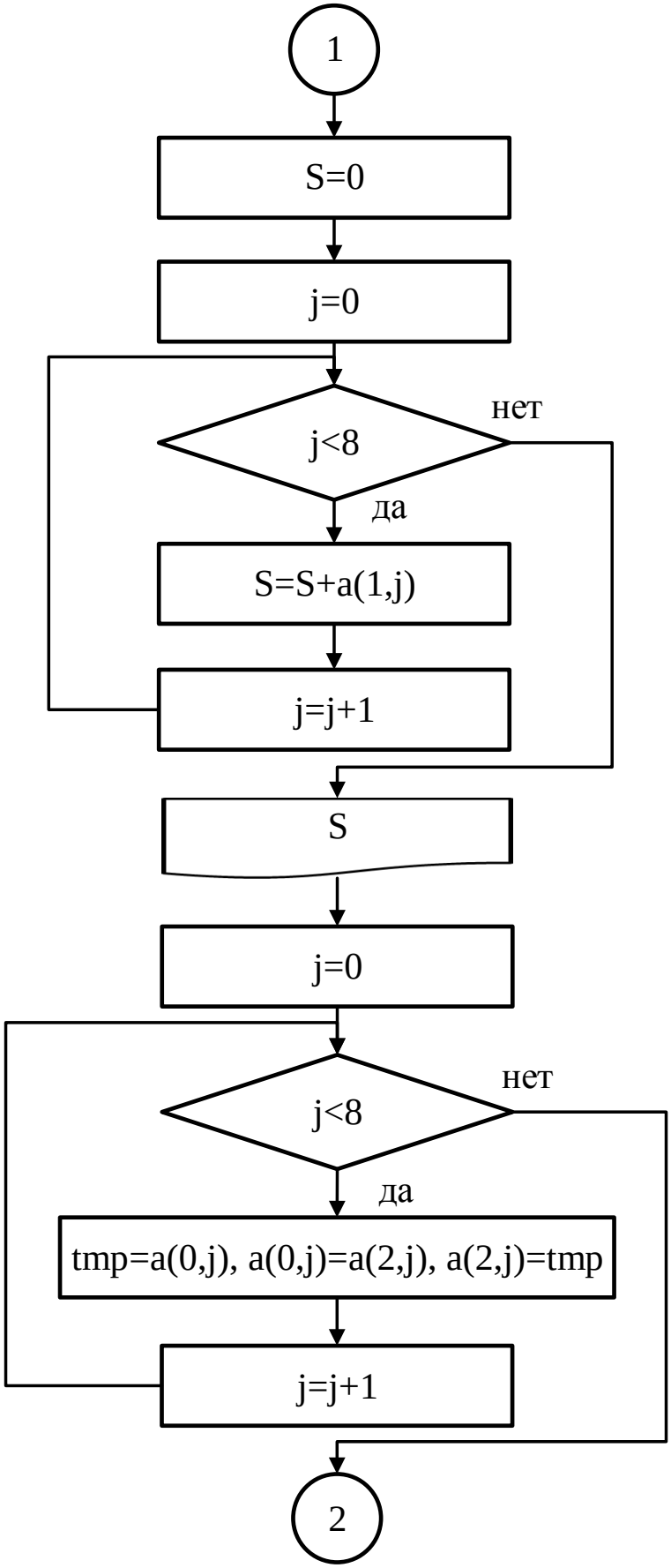

Рисунок 7.2 – Продолжение блок-схемы

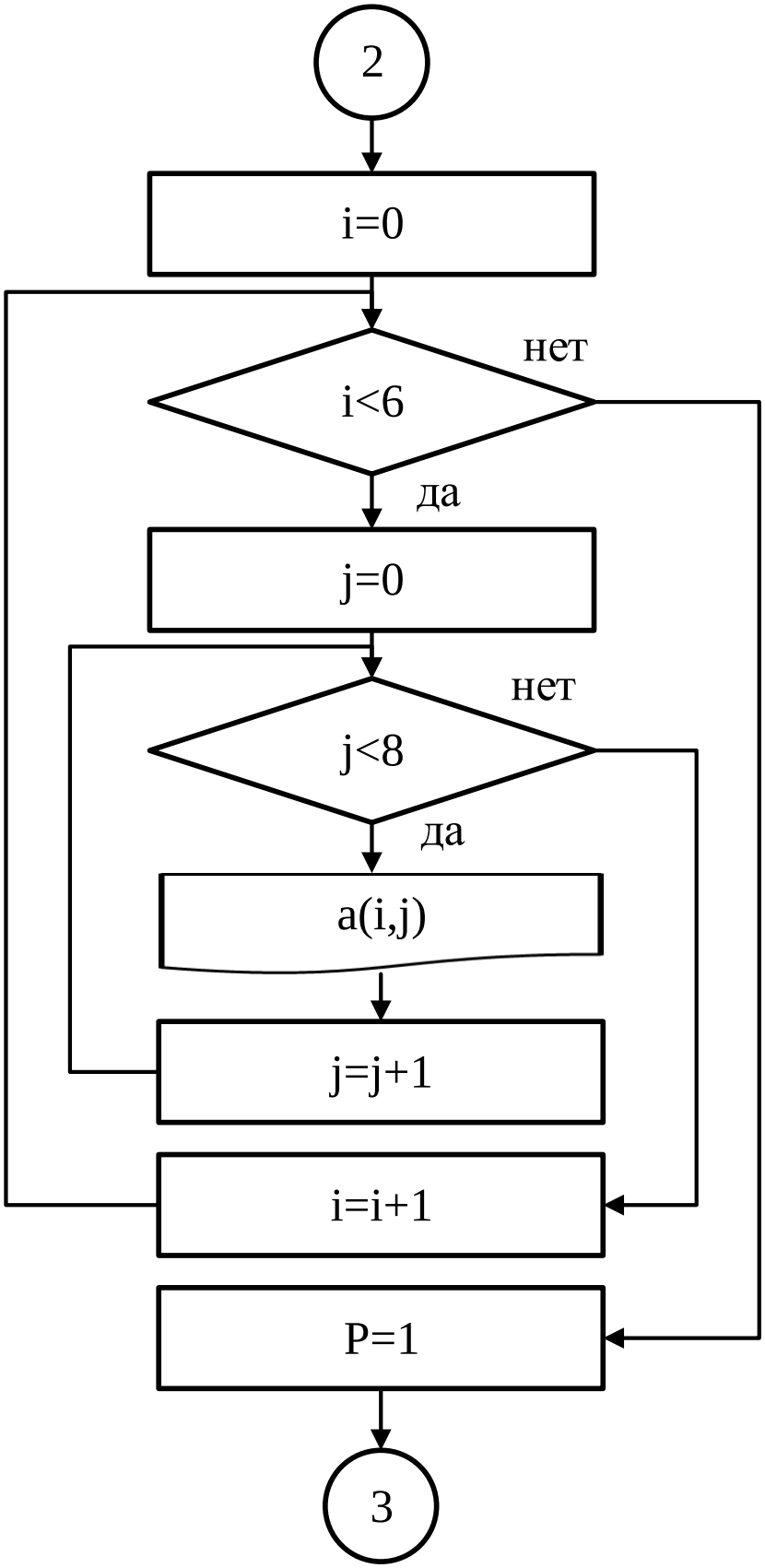

Рисунок 7.3 – Продолжение блок-схемы

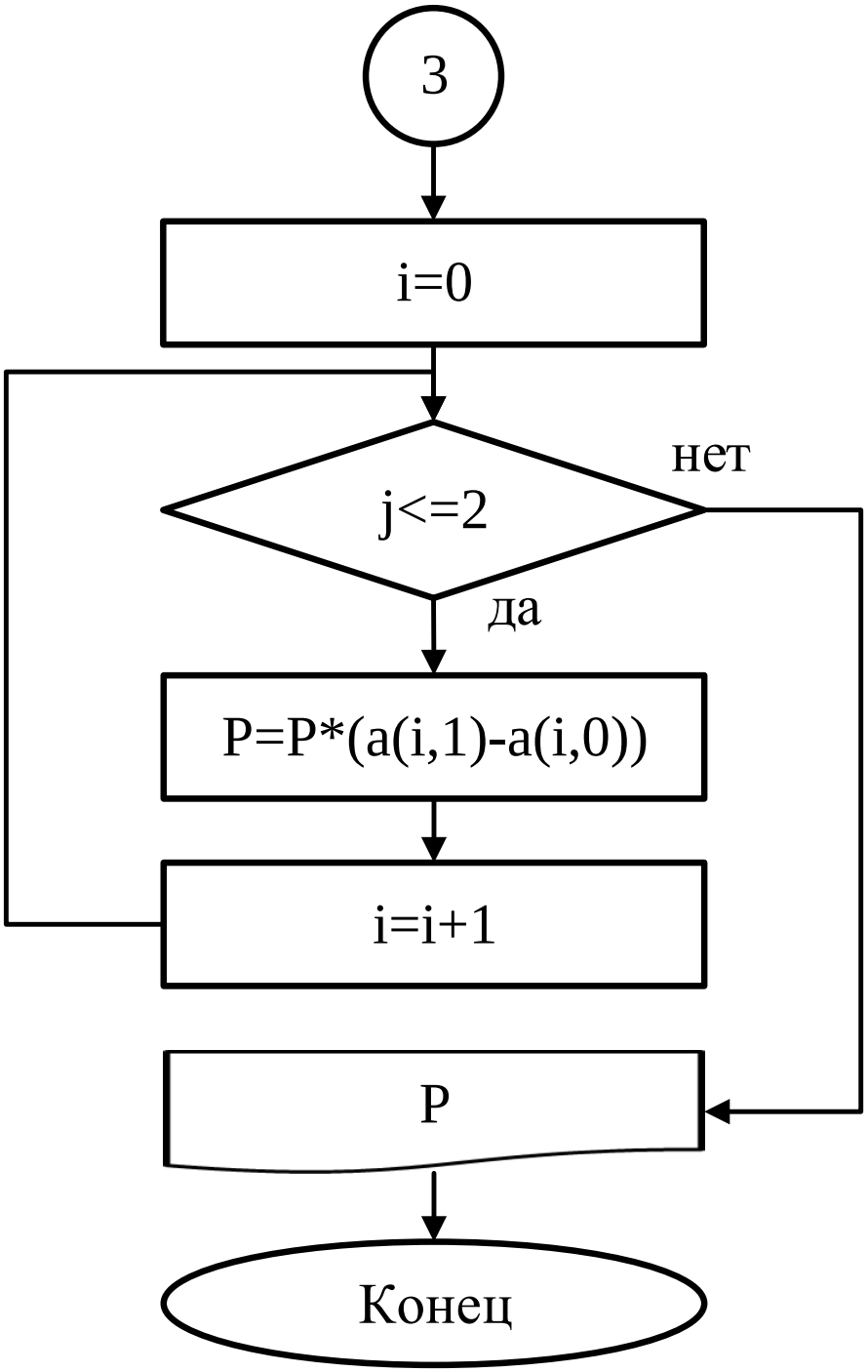

Рисунок 7.4 – Продолжение блок-схемы

Код программы (Visual Studio) с оператором for:

```
// proga29.cpp: определяет точку входа для консольного приложения.
//
#include "stdafx.h"
#include <iostream>
#include <iomanip>
using namespace std;
int main()
```

```
{
	double a[6][8];
	double S, tmp, P;
	int i, j;
	cout<<"Massiv a:"<<endl;
	for(i=0; i<6; i=i+1){
		for(j=0; j<8; j=j+1){
			a[i][j]=3.0*i-5.0*j;
			cout<<setw(5)<<a[i][j];
		}
		cout<<endl;
	}
	S=0;
	for(j=0; j<8; j=j+1){
		S=S+a[1][j];
	}
	cout<<"S="<<S<<endl;
	for(j=0; j<8; j=j+1){
		tmp=a[0][j];
		a[0][j]=a[2][j];
		a[2][j]=tmp;
	}
	for(i=0; i<6; i=i+1){
		for(j=0; j<8; j=j+1){
			cout<<setw(5)<<a[i][j];
		}
		cout<<endl;
	}
	P=1;
	for(i=0; i<2; i=i+1){
		P=P*(a[i][1]-a[i][0]);
	}
	cout<<"P="<<P<<endl;
	return 0;
}
```

Результат выполнения программы представлен на рисунке 7.5.

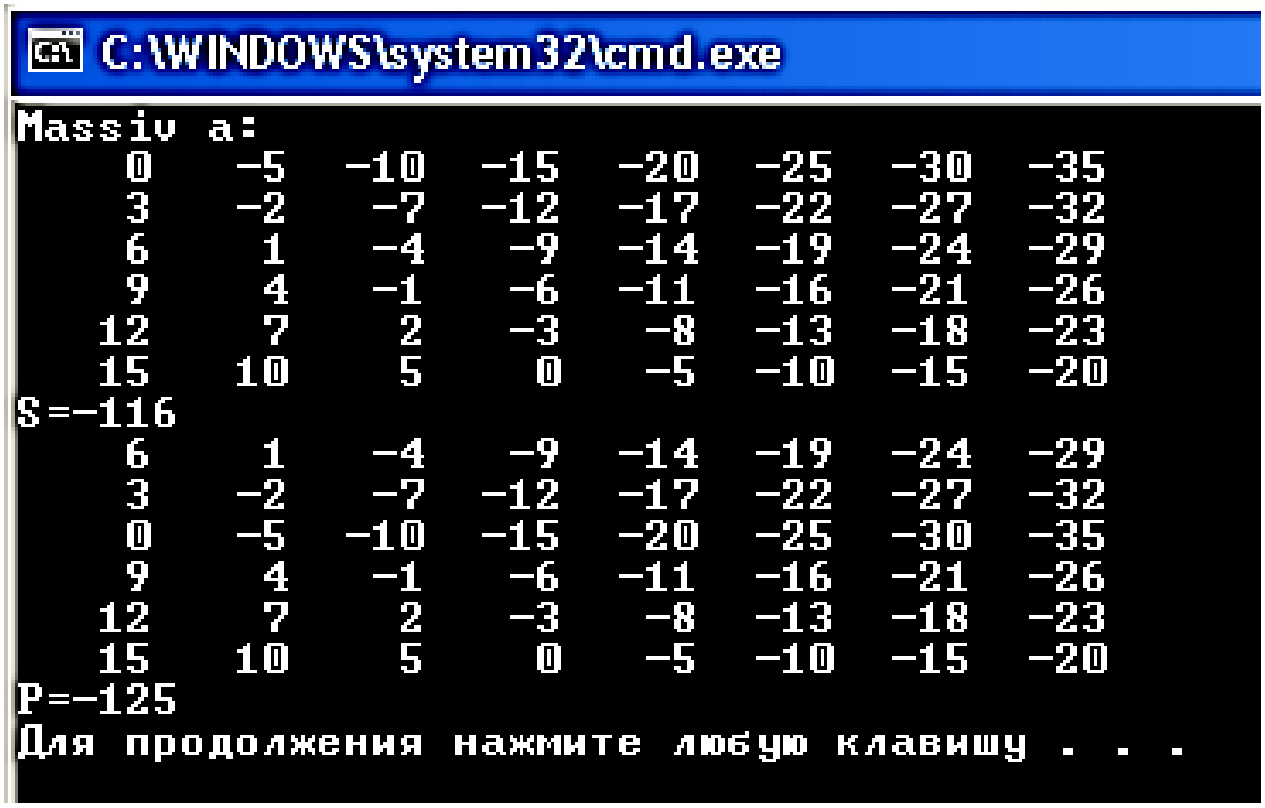

Рисунок 7.5 – Результат выполнения программы

Задание: вычислить сумму положительных элементов массива и поменять местами первый и пятый столбец.

Контрольные вопросы

1. Почему в программе на С++ необходимо, чтобы был известен размер массива?
2. Можно ли выполнить прямое присваивание массивов объявленных так: int x[10], y[10]?
3. Когда, с какой целью и почему возможно объявление безразмерных массивов?
4. Может ли значение элемента массива использоваться в качестве индекса другого элемента массива?

8 Лабораторная работа №8. Сортировка и поиск данных с применением функций

Цель работы: изучить принципы работы с одномерными массивами.

Задание: изучить возможности программной реализации основных методов сортировки.

Сортировка методом простого включения (сдвиг-вставка, вставками, вставка и сдвиг).

Хотя этот метод сортировки намного менее эффективен, чем сложные алгоритмы (такие как быстрая сортировка), у него есть ряд преимуществ:

1) прост в реализации;

2) эффективен на небольших наборах данных, на наборах данных до десятков элементов может оказаться лучшим;

3) эффективен на наборах данных, которые уже частично отсортированы;

4) это устойчивый алгоритм сортировки (не меняет порядок элементов, которые уже отсортированы);

5) может сортировать массив по мере его получения;

6) не требует временной памяти, даже под стек.

На каждом шаге алгоритма выбираем один из элементов входных данных и вставляем его на нужную позицию в уже отсортированной последовательности до тех пор, пока набор входных данных не будет исчерпан. Метод выбора очередного элемента из исходного массива произволен; может использоваться практически любой алгоритм выбора (рисунок 8.1).

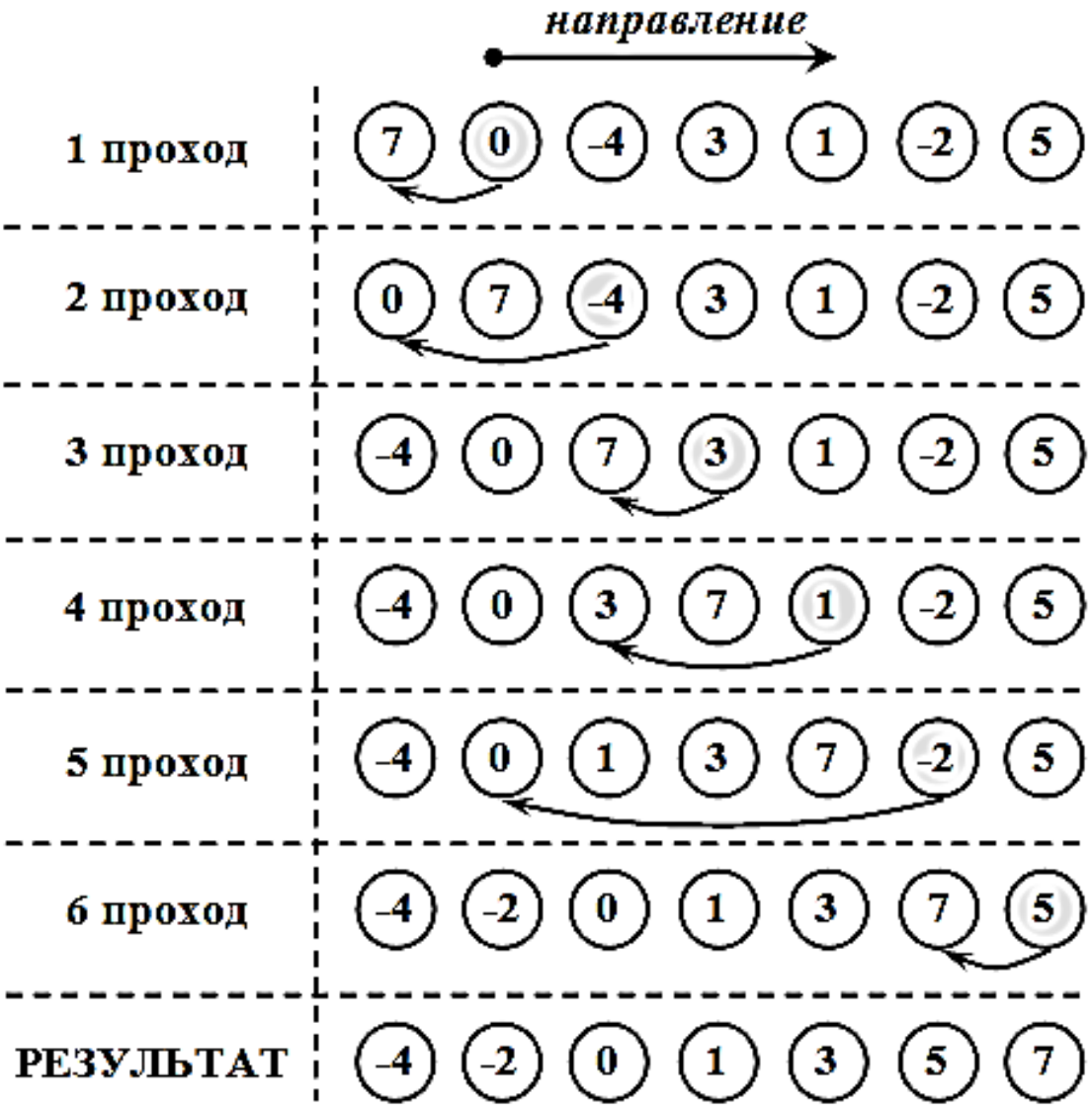

Рисунок 8.1 – Демонстрация сортировки по неубыванию методом простого включения

Ниже приводится пример функции сортировки.

```
//Описание функции сортировки методом простого включения
void InsertSort (int k,int x[max]) {
  int i,j, temp;
  for (i=0;i<k;i++) {
  //цикл проходов, i - номер прохода
    temp=x[i];
    //поиск места элемента
    for (j=i-1; j>=0 && x[j]>temp; j--)
      x[j+1]=x[j];//сдвигаем элемент вправо, пока не дошли
      //место найдено, вставить элемент
      x[j+1]=temp;
  }
}
```

Задание: написать программу сортировки 10-значного одномерного массива. Ввод первоначальных данных произвести с клавиатуры.

Контрольные вопросы

1. Какая программная структура обязательно присутствует в программе сортировки массива?
2. Приведите пример программы для сортировки одномерного массива методом, отличным от приведенного в описании.
3. Назовите основные методы сортировки одномерного массива.
4. Напишите алгоритм сортировки одномерного массива из десяти элементов методом пузырька.

Список литературы

Основная

1 Бимагамбетов Т.С. Программирование на языке С++. Учебное пособие.-А.,2007.

2 Лафоре Р. ООП в С++.-СПб.: «Питер», 2011.

3 Культин Н. С/С++ в задачах и примервх.-СПб.: «БХВ-Петербург», 2004, 2008, 2011.

4 Федоренко Ю.П. Алгоримы и программы на С++ BUILDER.-М.: «ДМК Пресс», 2010.

5 Шлее М. Qt4.5 профессиональное программирование на С++.-СПб.: «БХВ-Петербург», 2010.

Дополнительная

6 Алексеев Е.Р. Программирование на Microsoft Visual C++ и Turbo C++ Explorer.-М., 2007.

7 Архангельский А.Я. Приемы программирования в С++Builder 6 и 2006.-М.: «Бином», 2010.

8 Архангельский А.Я. С++ BUILDER работа с документами Excel.-М.: «Бином», 2009.

Содержание

Сводный план 2015 г., поз. 160

Татьяна Викторовна Голубева

АЛГОРИТМИЗАЦИЯ И ПРОГРАММИРОВАНИЕ

Методические указания по выполнению лабораторных работ для студентов специальности 5В071600 – Приборостроение

Редактор Л.Т. Сластихина
Специалист по стандартизации Н.К. Молдабекова

Подписано в печать _____ Формат 60х84 1/16
Тираж 100 экз. Бумага типографская №1
Объем 2 уч.-из.л. Заказ ___ Цена 1000 тн.

Копировально-множительное бюро
некоммерческого акционерного общества
«Алматинский университет энергетики и связи»
050013, Алматы, Байтурсынова, 126