

**Министерство науки и высшего образования РФ
ФГБОУ ВО «Ульяновский государственный университет»
Факультет математики, информационных и авиационных технологий**

Санников И.А.

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ
ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ
ПО ДИСЦИПЛИНЕ**

«Информатика и программирование»

для студентов бакалавриата по направлениям 01.03.02 Прикладная математика и информатика, 15.03.04 Автоматизация технологических процессов и производств, 24.03.04 Авиастроение

Ульяновск, 2019

Методические указания для самостоятельной работы студентов по дисциплине «Информатика и программирование» для студентов бакалавриата по направлениям для студентов бакалавриата по направлениям 01.03.02 Прикладная математика и информатика, 15.03.04 Автоматизация технологических процессов и производств, 24.03.04 Авиастроение/ составитель: Санников И.А. – Ульяновск: УлГУ, 2019.

Настоящие методические указания предназначены для студентов бакалавриата по направлениям для студентов бакалавриата по направлениям 01.03.02 Прикладная математика и информатика, 15.03.04 Автоматизация технологических процессов и производств, 24.03.04 Авиастроение. В работе приведены литература по дисциплине, примеры кода для реализации лабораторных работ и методические указания для самостоятельной работы студентов.

Они будут полезны при подготовке к лабораторным занятиям и к экзамену по данной дисциплине.

*Рекомендованы к введению в образовательный процесс Ученым советом
Факультета математики, информационных и авиационных технологий УлГУ
(протокол № 2/19 от 19 марта 2019 г.).*

1. ЛИТЕРАТУРА ДЛЯ ИЗУЧЕНИЯ ДИСЦИПЛИНЫ

1. Лутошкин И.В. Теоретические основы информатики : учеб. пособие. Ульяновск: УлГУ, ИЭиБ. 2015. 92 с. URL: <http://lib.ulsu.ru/ProtectedView/Book/ViewBook/317>
2. Рацеев С.М. Программирование на языке Си : учеб. пособие. Ульяновск : УлГУ, 2015. 319 с. - URL: <http://lib.ulsu.ru/ProtectedView/Book/ViewBook/325>
3. Трофимов, В. В. Информатика в 2 т : учебник для академического бакалавриата / В. В. Трофимов ; под редакцией В. В. Трофимова. — 3-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2016. — 959 с. — (Бакалавр. Академический курс). — ISBN 978-5-9916-3894-4. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/388058>
4. Рацеев С.М. Лабораторный практикум по программированию : метод. указания. Ульяновск : УлГУ, 2014. 91 с.-URL: <http://lib.ulsu.ru/ProtectedView/Book/ViewBook/479>
5. Угаров В. В. Основы информатики: учеб.-метод. пособие. Ульяновск: УлГУ. 2015 - URL: <http://lib.ulsu.ru/ProtectedView/Book/ViewBook/337>
6. Поднебесова, Г. Б. Абстрактная и компьютерная алгебра : практикум / Г. Б. Поднебесова. — Челябинск : Южно-Уральский государственный гуманитарно-педагогический университет, 2016. — 125 с. — ISBN 978-5-906908-47-6. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <http://www.iprbookshop.ru/83852.html> (дата обращения: 22.12.2020). — Режим доступа: для авторизир. пользователей
7. ГОСТ-Эксперт – единая база ГОСТов Российской Федерации для образования и промышленности. URL: <https://gostexpert.ru/>
 - а. ГОСТ Р 52292-2004 Информационная технология. Электронный обмен информацией. Термины и определения. М.: ИПК Издательство стандартов, 2005.
 - б. ГОСТ Р ИСО/МЭК 27002-2012 Информационная технология. Методы и средства обеспечения безопасности. Свод норм и правил менеджмента информационной безопасности. М.: Стандартинформ, 2014.
 - с. ГОСТ Р 51725.17-2012 Каталогизация продукции для федеральных государственных нужд. Российский открытый технический словарь. М.: Стандартинформ, 2013.
8. Зиборов В. В. MS Visual C в среде.net. Библиотека программиста. СПб.: Питер, с.: ил. ООО Издательство «Питер», 2012/
9. Васильев А.Н., Программирование на C++ в примерах и задачах, 2017.

2. МЕТОДИЧЕСКИЕ УКАЗАНИЯ

Ниже приведены методические указания для самостоятельной подготовки лабораторных работ по дисциплине.

СПИСОК ЛАБОРАТОРНЫХ РАБОТ

Лабораторная работа 1. Работа с формами.	5
Лабораторная работа 2. Работа с консолью.	7
Лабораторная работа 3. Работа с массивами.	9
Лабораторная работа 4. Работа с файлами.	12

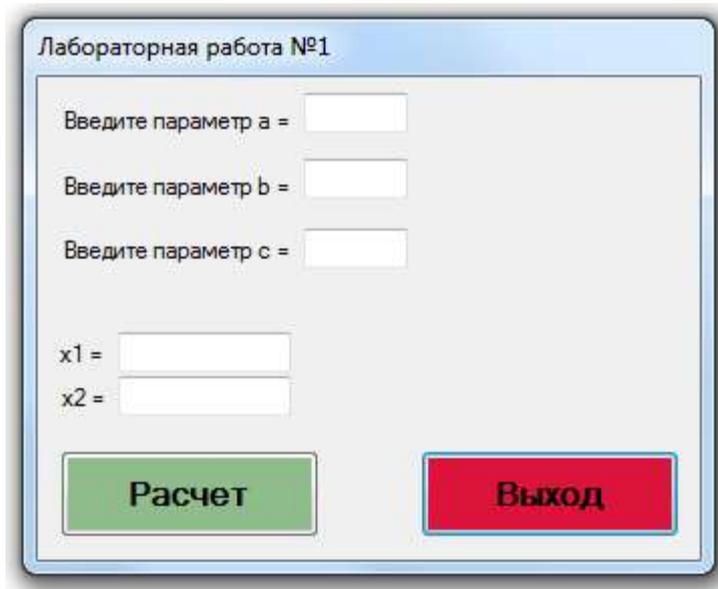
Лабораторная работа 1. Работа с формами.

Задание

Требуется написать программу-калькулятор для решения уравнений на языке программирования C++ с использованием шаблона Windows Forms Application в среде разработки Visual Studio.

Пример реализации

Пользователь вводит в соответствующие поля значения коэффициентов A, B, C квадратного уравнения: $A \cdot x^2 + B \cdot x + C = 0$. Программа выдает, в зависимости от введенных значений, результат в виде $x_1 = _$, $x_2 = _$, сообщить об отсутствие решений или о допустимости любых значений. Если пользователь ввел не корректные значения коэффициентов, программа должна сообщить об ошибке ввода данных.



Ниже приводится пример кода, реализующего алгоритм вычисления корней уравнения.

```
private: System::Void btn_start_Click(System::Object^ sender,
System::EventArgs^ e)
{
    double a, b, c;
    double x1, x2;
    double D;
    bool res, res_a, res_b, res_c;

    res_a = Double::TryParse(textBox_a->Text, a);
    res_b = Double::TryParse(textBox_b->Text, b);
    res_c = Double::TryParse(textBox_c->Text, c);

    res = res_a && res_b && res_c;

    if (res_a) {      lbl_err_a->Visible = false;  }   else {
    lbl_err_a->Visible = true;      }

    if (res_b) {      lbl_err_b->Visible = false;  }   else {
    lbl_err_b->Visible = true;      }

    if (res_c) {      lbl_err_c->Visible = false;  }   else {
    lbl_err_c->Visible = true;      }
```

```

if (res)
{
    if ( (a == 0) && (b == 0) && (c == 0) )
    {
        txtBox_x1->Text = "любое значение";
        txtBox_x2->Text = "любое значение";
    }

    if ( (a == 0) && (b == 0) && (c != 0) )
    {
        txtBox_x1->Text = "нет решений";
        txtBox_x2->Text = "нет решений";
    }

    if ( (a == 0) && (b != 0) && (c != 0) )
    {
        x1 = - c/b;
        txtBox_x1->Text = x1.ToString();
        txtBox_x2->Text = "-";
    }

    if ( (a != 0) && (b != 0) && (c != 0) )
    {
        D = b*b - 4*a*c;
        if (D >= 0)
        {
            x1 = (- b - sqrt(D)) / 2*a;
            x2 = (- b + sqrt(D)) / 2*a;
            txtBox_x1->Text = x1.ToString();
            txtBox_x2->Text = x2.ToString();;
        }
        else
        {
            txtBox_x1->Text = "-";
            txtBox_x2->Text = "-";
        }
    }
}
}
}

```

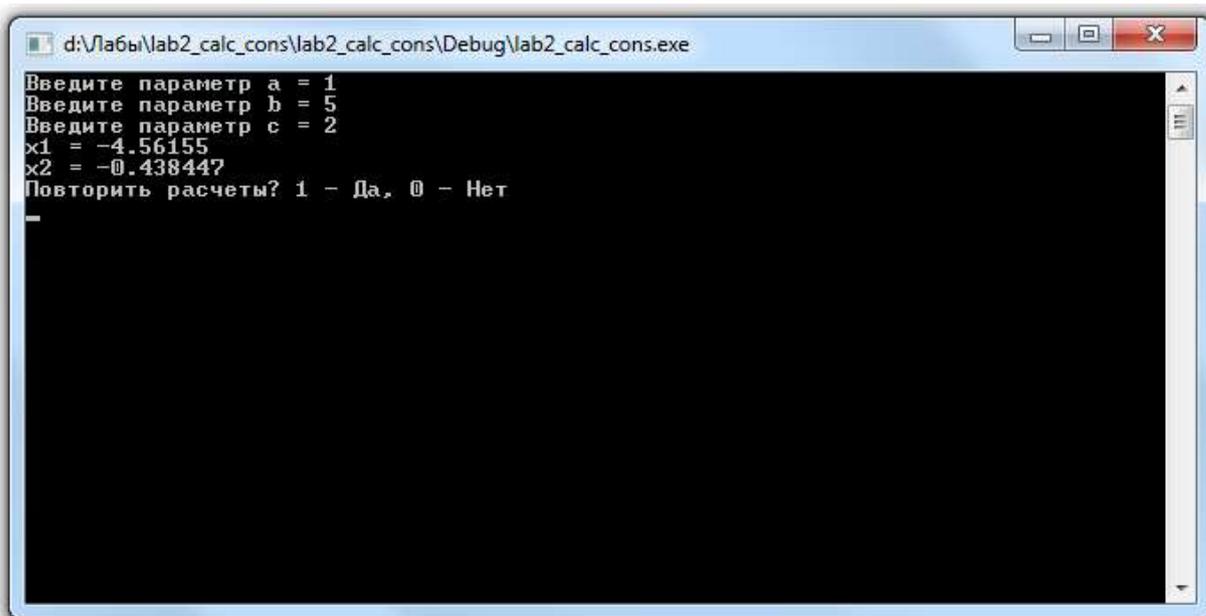
Лабораторная работа 2. Работа с консолью.

Задание

Требуется написать консольную программу-калькулятор для решения квадратных уравнений на языке программирования C++ с использованием шаблона Win32 Console Application в среде разработки Visual Studio.

Пример реализации

Аналогично заданию лабораторной работы №1.



```
d:\Лаб\lab2_calc_cons\lab2_calc_cons\Debug\lab2_calc_cons.exe
Введите параметр a = 1
Введите параметр b = 5
Введите параметр c = 2
x1 = -4.56155
x2 = -0.438447
Повторить расчеты? 1 - Да, 0 - Нет
-
```

Ниже приводится пример кода, реализующего алгоритм вычисления корней уравнения с вводом/выводом информации в консоль.

```
#include <iostream>
#include <math.h>

using namespace std;

void main()
{
    setlocale(LC_ALL, "RUS");

    bool flag = true;
    int rep;

    double a, b, c;
    double x1, x2;
    double D;

    while (flag)
    {
        cout << "Введите параметр a = ";
        cin >> a;

        cout << "Введите параметр b = ";
        cin >> b;

        cout << "Введите параметр c = ";
```

```

cin >> c;

if ( (a == 0) && (b == 0) && (c == 0) )
{
    cout << "Бесконечно много решений\n";
}

if ( (a == 0) && (b == 0) && (c != 0) )
{
    cout << "Нет решений\n";
}

if ( (a == 0) && (b != 0) && (c != 0) )
{
    x1 = - c/b;
    cout << "x = " << x1 << endl;
}

if ( (a != 0) && (b != 0) && (c != 0) )
{
    D = b*b - 4*a*c;

    if (D >= 0)
    {
        x1 = (- b - sqrt(D)) / 2*a;
        x2 = (- b + sqrt(D)) / 2*a;

        cout << "x1 = " << x1 << endl;
        cout << "x2 = " << x2 << endl;
    }
    else
    {
        cout << "D < 0\n";
    }
}

cout << "Повторить расчеты? 1 - Да, 0 - Нет \n";
cin >> rep;

if (rep == 0)
{
    flag = false;
}

system("cls");
}

system("pause");
}

```

Лабораторная работа 3. Работа с массивами.

Задание

Требуется написать консольную программу для обработки статических и динамических массивов на языке программирования C++ с использованием шаблона Win32 Console Application в среде разработки Visual Studio.

Размер массива определяется в программе. Пользователю в консоли выдаются запросы на ввод элементов массива определенного размера. После окончания ввода пользователю выводятся в строке через табуляцию введенные элементы массива. На следующих строках выводятся значения минимального, максимального и рассчитанное значение среднеарифметического элементов массива. В следующей строке пользователю выводится запрос на выбор метода сортировки элементов массива – по возрастанию или по убыванию. В зависимости от выбора пользователя программа осуществляет сортировку массива и вывод его элементов в строку через табуляцию. После окончания работы пользователю предлагается начать с начала или выйти из программы.

Необходимо реализовать данную программу с использованием статических и динамических массивов. В случае реализации с динамическими массивами в начале у пользователя запрашивается размер массива.

Пример реализации для случая статического массива

```
Введите 3 элемента массива:
введите элемент массива a[1] = 1
введите элемент массива a[2] = 2
введите элемент массива a[3] = 3
1      2      3
Минимальное значение = 1
Максимальное значение = 3
Среднеарифметическое = 2
Сортировать по возрастанию – 1, по убыванию – 0
0
3      2      1
Продолжить работу – 1, выйти – 0
0
```

Пример реализации для случая динамического массива

```
Введите размер массива n =
3
введите элемент массива a[1] = 1
введите элемент массива a[2] = 2
введите элемент массива a[3] = 3
1      2      3
Минимальное значение = 1
Максимальное значение = 3
Среднеарифметическое = 2
Сортировать по возрастанию – 1, по убыванию – 0
0
3      2      1
Продолжить работу – 1, выйти – 0
0
```

Ниже приводится пример кода, частично реализующего алгоритм работы с массивами и вводом/выводом информации в консоль.

```

#include <iostream>
using namespace std;

const int n = 3;
int mas1[n];
int m;

void set_mas_static()
{
    for (int i=0; i < n; i++)
    {
        cout << "mas1[" << i+1 << "] : ";
        cin >> mas1[i];
    }
}

void set_mas_dynamic()
{
    for (int i=0; i < m; i++)
    {
        cout << "mas2[" << i+1 << "] : ";
        cin >> mas2[i];
    }
}

void get_mas_static()
{
    for (int i=0; i < n; i++)
    {
        cout << mas1[i] << "\t";
    }
}

void get_mas_dynamic()
{
    for (int i=0; i < m; i++)
    {
        cout << mas2[i] << "\t";
    }
}

int get_max_static()
{
    int max = 0;
    for (int i=0; i < n; i++)
    {
        if (max <= mas1[i])
        {
            max = mas1[i];
        }
    }
    return max;
}

int get_max_dynamic()
{
    int max = 0;
    for (int i=0; i < m; i++)
    {
        if (max <= mas2[i])
        {
            max = mas2[i];
        }
    }
    return max;
}

```

```

void main()
{
    setlocale(LC_ALL, "russian");

    set_mas_static();
    get_mas_static();
    cout << "\nМаксимальный элемент массива = " << get_max_static() << endl;

    cout << "Введите размер массива = ";
    cin >> m;

    int *mas2 = new int[m];

    set_mas_dynamic();
    get_mas_dynamic();
    cout << "\nМаксимальный элемент массива = " << get_max_dynamic();
    cout << endl;

    delete[] mas2;

    cout << "\n";
    system("pause");
}

```

Ниже приводится пример кода, реализующего алгоритм сортировки массива «пузырьком».

```

for (int i=0; i < n - 1; i++)
{
    for (int j=0; j < n - i - 1; j++)
    {
        if (mas[j] >= mas[j+1])
        {
            int tmp = mas[j];
            mas[j] = mas[j+1];
            mas[j+1] = tmp;
        }
    }
}

```

Лабораторная работа 4. Работа с файлами.

Задание

Требуется написать консольную программу на языке программирования C++ с использованием шаблона Win32 Console Application в среде разработки Visual Studio, которая по заданной пользователем во входном текстовом файле матрице MxN определяла бы чётные и нечётные элементы и на их позиции в выходной файл записывала 0 и 1 соответственно.

Пример реализации

Входной файл: in.txt

```
1   2   3   4
5   6   7   8
9   10  11  12
13  14  15  16
```

Выходной файл: out.txt

```
1   0   1   0
1   0   1   0
1   0   1   0
1   0   1   0
```

Ниже приводится пример кода, реализующего алгоритм работы с файлами и вводом/выводом информации в консоль.

```
#include <iostream>
#include <fstream>
#include <string>

using namespace std;

int getRow(ifstream &fin)
{
    string str;
    int row = 0;

    fin.clear();
    fin.seekg(0, ios_base::beg);

    while (!fin.eof())
    {
        getline(fin, str);
        if (str != "") { row++; }
    }
    return row;
}

int getCount(ifstream &fin)
{
    int count = 0;
    int tmp;

    fin.clear();
    fin.seekg(0, ios_base::beg);
```

```

        while (!fin.eof())
        {
            fin >> tmp;
            count++;
        }
    return count;
}

void setMatrix(ifstream &fin, int **matrix, int row, int column)
{
    fin.clear();
    fin.seekg(0,ios_base::beg);

    while (!fin.eof())
    {
        for (int i=0; i < row; i++)
            for (int j=0; j < column; j++)
            {
                fin >> matrix[i][j];
            }
    }
}

void setMatrixToConsole(int **matrix, int row, int column)
{
    for (int i=0; i < row; i++)
    {
        for (int j=0; j < column; j++)
        {
            cout << matrix[i][j] << "\t";
        }
        cout << endl;
    }
}

void setTransformMatrix(int **matrix, int row, int column)
{
    for (int i=0; i < row; i++)
        for (int j=0; j < column; j++)
        {
            if ( matrix[i][j] % 2 == 0 ) matrix[i][j] = 0;
            else matrix[i][j] = 1;
        }
}

void delMatrix(int **matrix, int row, int column)
{
    for (int i=0; i < row; i++)
    {
        delete[] matrix[i];
    }
}

void setMatrixToFile(int **matrix, int row, int column)
{
    ofstream fout("d:/out.txt");

    if (fout.is_open())
    {
        for (int i=0; i < row; i++)
        {
            for (int j=0; j < column; j++)

```

```

        {
            fout << matrix[i][j] << " ";
        }
        fout << endl;
    }
    cout << "Файл успешно записан" << endl;
}
else cout << "Файл не записан" << endl;

fout.close();
}

int main()
{
    setlocale(LC_ALL, "rus");

    ifstream fin("d:\\in.txt");

    int column = 0;
    int row = 0;

    if (fin.is_open())
    {
        cout << "Файл успешно открыт" << endl;
        row = getRow(fin);
        column = getCount(fin) / row;
        cout << "Количество строк = " << row << ", Количество столбцов = " <<
column << endl;
    }
    else
    {
        cout << "Файл не открылся" << endl;
        return 1;
    }

    int** matrix;
    matrix = new int*[row];

    for (int i = 0; i < row; i++)
    {
        matrix[i] = new int[column];
    }

    setMatrix(fin, matrix, row, column);
    setMatrixToConsole(matrix, row, column);

    setTransformMatrix(matrix, row, column);
    cout << endl;

    setMatrixToConsole(matrix, row, column);

    setMatrixToFile(matrix, row, column);

    delMatrix(matrix, row, column);
    fin.close();

    system("pause");
    return 0;
}

```