

**Филиал федерального государственного бюджетного образовательного учреждения  
высшего образования  
«Национальный исследовательский университет «МЭИ»  
в г. Смоленске**

**МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ ДЛЯ ОБЕСПЕЧЕНИЯ  
ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА**

---

Направление подготовки: **09.03.03 «Прикладная информатика»**

Профиль: **«Прикладная информатика в топливно-энергетическом комплексе»**

Уровень высшего образования: **бакалавриат**


Нормативный срок обучения: **4 года**

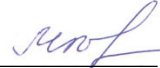
Форма обучения: **очная**

Год набора: **2023**

*Направление подготовки 09.03.03 «Прикладная информатика»  
Профиль «Прикладная информатика в топливно-энергетическом комплексе»  
Методическое обеспечение РПД Б1.О.06 «Информационные технологии»*


**Методические материалы составили:**

канд. техн. наук, доцент кафедры  
информационных технологий в экономике и управлении  А.Ю. Пучков

канд. техн. наук, доцент кафедры  
информационных технологий в экономике и управлении  М.Ю. Лебедева

«20» января 2023 г.

**Заведующий кафедрой информационных технологий в экономике и управлении:**

  
\_\_\_\_\_ д-р техн. наук, профессор М.И. Дли  
подпись \_\_\_\_\_ ФИО

«08» февраля 2023 г.

## МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ЛАБОРАТОРНЫХ РАБОТ ПО ДИСЦИПЛИНЕ

### 1 СЕМЕСТР

#### ЛАБОРАТОРНАЯ РАБОТА 1. ЛИНЕЙНЫЕ АЛГОРИТМЫ

**Цель занятия:** освоение среды разработки, изучение структуры программы, ввод и вывод данных различных типов.

#### **Теоретические сведения. Понятие линейных алгоритмов**

Любой алгоритм составляется из ряда базовых структур. Простейшей базовой структурой является следование — структура с линейными характеристиками. Из этого можно сформулировать определение.

Линейный алгоритм — это алгоритм, образуемый командами, которые выполняются однократно и именно в той последовательности, в которой записаны. Линейная структура, по сути, проста. Записать её можно как в текстовой, так и в графической форме.

Для изображения алгоритма графически используют блок-схемы. Они представляют собой геометрические фигуры (блоки), соединённые стрелками. Стрелки показывают связь между этапами и последовательность их выполнения. Каждый блок сопровождается надписью.

Как составить программу линейной структуры?

Порядок следующий:

— определите, что именно относится к исходным данными, а также каков тип/класс этих данных, выберите имена переменных;

— определите, каков тип данных будет у искомого результата, выберите название переменных (переменной);

— определите, какие математические формулы связывают результат и исходные данные;

— если требуется наличие промежуточных данных, определите класс/типы этих данных и выберите имена;

— опишите все используемые переменные;

— запишите окончательный алгоритм. Он должен включать в себя ввод данных, вычисления, вывод результатов.

Основное элементарное действие в вычислительном алгоритме – это присваивание переменной величине определенного значения. В случае, когда значение константы определяется видом ее записи, переменная величина получит конкретное значение исключительно в результате присваивания. Это может быть выполнено с помощью двух способов: при помощи команды присваивания; при помощи команды ввода.

#### **Задание к лабораторной работе:**

Выполнение заданий 1 – 2 предполагает оформление отчета в текстовом редакторе Word (Times New Roman, 12 п., одинарный), в котором отражаются этапы выполнения заданий и результаты (ответы на вопросы, скриншоты заполненных полей калькулятора, документы аттестации ИБ и т.д.)

#### **Задание 1.** Изучение линейных алгоритмов

**1.1.** Открыть «Сборник заданий по программированию» А.П. Нестерова (электронную версию Вам должен дать преподаватель), и, далее, выполнить задание Вашего варианта лабораторной работы №1.

#### **Задание 2.** Изучение линейных алгоритмов

**2.1.** Открыть «Сборник заданий по программированию» А.П. Нестерова (электронную версию Вам должен дать преподаватель), и, далее, выполнить задание Вашего варианта лабораторной работы №2.

**Контрольные вопросы:**

1. Что такое алгоритмы?
2. Перечислите виды алгоритмов.
3. Дайте определения понятию линейный алгоритм.
4. Что необходимо для графического изображения алгоритма?
5. Как составить программу линейной структуры?

## ЛАБОРАТОРНАЯ РАБОТА 2. РАЗВЕТВЛЕННЫЕ АЛГОРИТМЫ

**Цель занятия:** изучить операторы ветвления и способы их применения в программе.

### Теоретические сведения. Понятие алгоритма ветвления

Разветвляющийся алгоритм – алгоритм, в котором в зависимости от условия выполняется либо одна, либо другая последовательность действий.

Во многих случаях требуется, чтобы при одних условиях выполнялась одна последовательность действий, а при других – другая.

Вычислительный процесс называется ветвящимся, если для его реализации предусмотрено несколько направлений (ветвей). Каждое отдельное направление процесса обработки данных является отдельной ветвью вычислений. Ветвление в программе — это выбор одной из нескольких последовательностей команд при выполнении программы. Выбор направления зависит от заранее определенного признака, который может относиться к исходным данным, к промежуточным или конечным результатам. Признак характеризует свойство данных и имеет два или более значений. Ветвящийся процесс, включающий в себя две ветви, называется простым, более двух ветвей — сложным. Сложный ветвящийся процесс можно представить с помощью простых ветвящихся процессов. Направление ветвления выбирается логической проверкой, в результате которой возможны два ответа: «да» — условие выполнено и «нет» — условие не выполнено.

Следует иметь в виду, что, хотя на схеме алгоритма должны быть показаны все возможные направления вычислений в зависимости от выполнения определенного условия (или условия), при однократном прохождении программы процесс реализуется только по одной ветви, а остальные исключаются. Любая ветвь, по которой осуществляются вычисления, должна приводить к завершению вычислительного процесса.

Оператор ветвления(условия) – оператор управления, осуществляющий выполнение одного и только одного составного оператора из двух возможных в зависимости от условия.

Синтаксис оператора ветвления:

```
if (условие) { составной оператор №1 }  
else { составной оператор №2 }
```

Если составной оператор в основной или альтернативной ветви оператора управления содержит только один простой оператор, то фигурные скобки можно опустить. В таком случае синтаксис оператора ветвления примет вид:

```
if(условие) простой оператор №1;  
else простой оператор №2;
```

Ветвление — метод и форма организации действий, когда в зависимости от выполнения определённого условия совершается та либо иная последовательность шагов.

В результате совсем несложно составить алгоритм покупки мороженого с учётом наличия необходимой суммы денег.

### **Задание к лабораторной работе:**

Выполнение заданий 1 – 2 предполагает оформление отчета в текстовом редакторе Word (Times New Roman, 12 п., одинарный), в котором отражаются этапы выполнения заданий и результаты (ответы на вопросы, скриншоты заполненных полей калькулятора, документы аттестации ИБ и т.д.)

#### **Задание 1.** Изучение разветвлённых алгоритмов

**1.1.** Открыть «Сборник заданий по программированию» А.П. Нестерова (электронную версию Вам должен дать преподаватель), и, далее, выполнить задание Вашего варианта лабораторной работы №3.

#### **Задание 2.** Изучение решения геометрических задач

**2.1.** Открыть «Сборник заданий по программированию» А.П. Нестерова (электронную версию Вам должен дать преподаватель), и, далее, выполнить задание Вашего варианта лабораторной работы №4.

### **Контрольные вопросы:**

1. Что такое разветвляющийся алгоритм?
2. Каков синтаксис алгоритма ветвления?
3. Приведите пример алгоритма ветвления.
4. Что такое ветвление в программе?
5. Как составить программу разветвляющейся структуры?

## **ЛАБОРАТОРНАЯ РАБОТА 3. ЦИКЛИЧЕСКИЕ АЛГОРИТМЫ**

**Цель занятия:** научиться использовать операторы циклов, научиться использовать различные операторы циклов для вычисления конечных сумм и произведений.

### **Теоретические сведения. Понятие циклических алгоритмов**

Прежде чем приступить к основной теме статьи, следует прояснить терминологию вопроса и рассмотреть основные определения:

— цикл — вид управляющей конструкции в языках программирования. Он позволяет организовать многократное исполнение определённого набора инструкций (последовательность действий, при котором выполняется тело цикла);

— тело цикла — последовательность инструкций, обеспечивающая их многократное исполнение;

— итерация — однократное исполнение тела цикла;

— условие выхода (условие окончания) — выражение, которое определяет, станет ли в очередной раз выполняться итерация либо произойдёт завершение цикла;

— счётчик цикла — переменная, которая сохраняет номер итерации.

Счётчик не обязательно должен содержаться в цикле и счётчик совсем не обязательно должен быть один, то есть условие выхода порой зависит от нескольких изменяемых переменных. Вдобавок к этому, условие выхода иногда зависит от внешних условий (как пример — наступление определённого времени).

Работа любого цикла вне зависимости от его вида включает в себя:

— первоначальную инициализацию циклических переменных;

— проверку условия выхода из цикла;

— выполнение тела;

— обновление циклической переменной на каждой итерации.

Циклический алгоритм – описание действий, которые должны повторяться указанное число раз или пока не выполнено заданное условие. Алгоритм называется циклическим, если в нем имеются действия или наборы действий, которые необходимо выполнить более одного раза. Повторяющиеся алгоритмические действия являются телом цикла. Дополнительно каждый цикл имеет условие, по которому выполнение циклического алгоритма за-

канчивается. Перечень повторяющихся действий называют телом цикла. Циклические алгоритмы бывают двух типов:

Циклы со счетчиком, в которых какие-то действия выполняются определенное число раз;

Циклы с условием, в которых тело цикла выполняется, в зависимости от какого-либо условия.

Циклы со счетчиком используют, когда заранее известно, какое число повторений тела цикла необходимо выполнить. Например, на уроке физкультуры вы должны пробежать некоторое количество кругов вокруг стадиона.

Для счетчика от начального значения до конечного значения выполнить действие. Часто бывает так, что необходимо повторить тело цикла, но заранее не известно, какое количество раз это надо сделать. В таких случаях количество повторений зависит от некоторого условия. Такие циклы называются циклы с условием. Циклы, в которых сначала проверяется условие, а затем, возможно, выполняется тело цикла называют циклы с предусловием. Если условие проверяется после первого выполнения тела цикла, то циклы называются циклы с постусловием.

Структуру цикла с параметром (цикл *для*) иначе называют циклом  $i$  раз.

Эта команда выполняется таким образом: параметру  $i$  присваивается начальное значение  $a$ , сравнивается с конечным значением  $b$  и, если оно меньше или равно конечному значению  $b$ , выполняется серия команд. Параметру присваивается значение предыдущего, увеличенного на величину  $h$  – шага изменения параметра и вновь сравнивается с конечным значением  $b$ .

### **Задание к лабораторной работе:**

Выполнение заданий 1 – 3 предполагает оформление отчета в текстовом редакторе Word (Times New Roman, 12 п., одинарный), в котором отражаются этапы выполнения заданий и результаты (ответы на вопросы, скриншоты заполненных полей калькулятора, документы аттестации ИБ и т.д.)

#### **Задание 1. Изучение разветвлённых алгоритмов**

**1.1.** Открыть «Сборник заданий по программированию» А.П. Нестерова (электронную версию Вам должен дать преподаватель), и, далее, выполнить задание Вашего варианта лабораторной работы №5-7.

**1.2.** Открыть «Сборник заданий по программированию» А.П. Нестерова (электронную версию Вам должен дать преподаватель), и, далее, выполнить задание Вашего варианта лабораторной работы №9.

#### **Задание 2. Изучение итерационных циклов**

**2.1.** Открыть «Сборник заданий по программированию» А.П. Нестерова (электронную версию Вам должен дать преподаватель), и, далее, выполнить задание Вашего варианта лабораторной работы №11.

#### **Задание 3. Изучение вложенных циклов**

**3.1.** Открыть «Сборник заданий по программированию» А.П. Нестерова (электронную версию Вам должен дать преподаватель), и, далее, выполнить задание Вашего варианта лабораторной работы №12.

### **Контрольные вопросы:**

1. Что такое циклический алгоритм?
2. Что такое циклы с постусловием?
3. Перечислите типы циклических алгоритмов.
4. Что такое циклы с предусловием?
5. Что такое циклы с параметром?
6. Расскажите основные определения.

## ЛАБОРАТОРНАЯ РАБОТА 4. ОДНОМЕРНЫЕ МАССИВЫ В СИ

**Цель занятия:** научиться обрабатывать одномерные массивы, научиться производить сложную обработку массивов.

### **Теоретические сведения. Понятие одномерных массивов**

При решении задач с большим количеством данных одинакового типа использование переменных с различными именами, не упорядоченных по адресам памяти, затрудняет программирование. В подобных случаях в языке Си используют объекты, называемые массивами.

Массив — это непрерывный участок памяти, содержащий последовательность объектов одинакового типа, обозначаемый одним именем.

Массив характеризуется следующими основными понятиями:

Элемент массива (значение элемента массива) — значение, хранящееся в определенной ячейке памяти, расположенной в пределах массива, а также адрес этой ячейки памяти. Каждый элемент массива характеризуется тремя величинами:

- адресом элемента — адресом начальной ячейки памяти, в которой расположен этот элемент;
- индексом элемента (порядковым номером элемента в массиве);
- значением элемента.

Адрес массива — адрес начального элемента массива.

Имя массива — идентификатор, используемый для обращения к элементам массива.

Размер массива — количество элементов массива.

Размер элемента — количество байт, занимаемых одним элементом массива.

Графически расположение массива в памяти компьютера можно представить в виде непрерывной ленты адресов.

Для объявления массива в языке Си используется следующий синтаксис:

тип имя[размерность]={инициализация};

Инициализация представляет собой набор начальных значений элементов массива, указанных в фигурных скобках, и разделенных запятыми.

При обращении к элементам массива индекс требуемого элемента указывается в квадратных скобках []. Однако часто требуется задавать значения элементов массива в процессе выполнения программы. При этом используется объявление массива без инициализации. В таком случае указание количества элементов в квадратных скобках обязательно.

Массив — это структура данных, представленная в виде группы ячеек одного типа, объединенных под одним единым именем. Массивы используются для обработки большого количества однотипных данных. Имя массива является указателем. Отдельная ячейка данных массива называется элементом массива. Элементами массива могут быть данные любого типа. Массивы могут иметь как одно, так и более одного измерений. В зависимости от количества измерений массивы делятся на одномерные массивы, двумерные массивы, трёхмерные массивы и так далее до n-мерного массива.

### **Задание к лабораторной работе:**

Выполнение заданий 1 – 2 предполагает оформление отчета в текстовом редакторе Word (Times New Roman, 12 п., одинарный), в котором отражаются этапы выполнения заданий и результаты (ответы на вопросы, скриншоты заполненных полей калькулятора, документы аттестации ИБ и т.д.)

#### **Задание 1. Изучение одномерных массивов**

**1.1.** Открыть «Сборник заданий по программированию» А.П. Нестерова (электронную версию Вам должен дать преподаватель), и, далее, выполнить задание Вашего варианта лабораторной работы №13-14.

## **Задание 2.** Сложная обработка одномерных массивов

**2.1.** Открыть «Сборник заданий по программированию» А.П. Нестерова (электронную версию Вам должен дать преподаватель), и, далее, выполнить задание Вашего варианта лабораторной работы №16.

### **Контрольные вопросы:**

1. Какие типы массивов существуют?
2. Что такое одномерный массив?
3. Перечислите способы ввода одномерного массива.
4. Перечислите способы вывода одномерного массива на экран.
5. Чем характеризуется одномерный массив?

## **ЛАБОРАТОРНАЯ РАБОТА 5. ДВУМЕРНЫЕ МАССИВЫ В СИ**

**Цель занятия:** научиться обрабатывать двумерные массивы, научиться обрабатывать элементы массива в зависимости от их расположения относительно главной и побочной диагоналей.

### **Теоретические сведения. Понятие двумерных массивов**

Двумерные массивы в Си — это массив одномерных массивов.

Если требуется разрешить задачу с существенными объёмами информационных однотипных данных и использованием разных переменных, не прошедших сортировку по адресации памяти, то при этом возникает большое количество проблем при попытке программно реализовать процесс решения. Чтобы облегчить программное построение алгоритма разрешения стоящей проблемы, в таких случаях в программных языках и в частности Си используют объекты, называемые массивами.

Массив — это фиксированный участок памяти, содержащий комплект объектов одного типа и который имеет одно имя.

Массив характеризуется такими составляющими:

– Элемент массива. Это величина, хранящаяся в предназначенной ей ячейке памяти, которая расположена в области массива, и ещё адрес этой ячейки памяти. Любой элемент массива характеризуется следующими параметрами:

1. Его адресацией. Имеется ввиду адрес начальной ячейки памяти, в которой расположен данный элемент.

2. Индекс элемента. А именно его нумерация в составе массива. Непосредственно содержимое (значение) элемента.

3. Адресация массива. То есть адрес начального элемента. Наименование массива. То есть идентификационный признак, по которому идёт обращение к элементам массива.

– Величина массива. То есть количество элементов массива.

– Объём элемента. То есть количество байт, занимаемое одним элементом массива.

– Размер массива. То есть количество байт, отведённое в памяти для хранения всех элементов массива. Следовательно, длина массива — это размер одного элемента, умноженный на их общее число.

По своей размерности массивы в Си бывают одномерные, двумерные и многомерные. Под многомерными понимаются такие массивы, которые имеют больше чем один индекс. Многомерные массивы можно интерпретировать как комплект компонентов, которые распределены по нескольким координатам. Процедура задания двумерного массива аналогична объявлению одномерного массива, но при объявлении двумерного массива требуется задать внутри квадратных скобок размеры для всех направлений.

Чтобы получить доступ к компонентам двумерных массивов, требуется применять пару индексов, указывающих на нумерацию столбцов и строчек. Приведённые выше усло-

вия можно применить и к массивам большего размера, вплоть до N. Однако следует отметить, что массивы уже с четырьмя параметрами достаточно сложно выразить в графическом виде. Следует помнить, что нумерация положения компонентов в строчках и столбиках ведёт отсчёт от нуля. Это означает, что первый компонент обладает нулевым индексом, по этой причине в примере выше индексы указаны на единицу меньше требуемых номеров координат. Также необходимо учитывать, что запрещается делать запись информации за конечным компонентом массива. Например, когда массив включает в свой состав одиннадцать компонентов, то запрещаются попытки назначения какого-либо значения компоненту, имеющему номер одиннадцать, так как конечный компонент в данном массиве обладает индексом десять. А компонента, имеющего индекс одиннадцать, просто не существует. Если всё-таки попытаться это выполнить, то вероятны самые непредсказуемые последствия, вплоть до создания помех в работе другим программам. Но, как правило, в функции операционных систем заложена блокировка недостоверных процедур такого вида.

### **Задание к лабораторной работе:**

Выполнение заданий 1 – 2 предполагает оформление отчета в текстовом редакторе Word (Times New Roman, 12 п., одинарный), в котором отражаются этапы выполнения заданий и результаты (ответы на вопросы, скриншоты заполненных полей калькулятора, документы аттестации ИБ и т.д.)

#### **Задание 1.** Изучение обработки двумерных массивов

**1.1.** Открыть «Сборник заданий по программированию» А.П. Нестерова (электронную версию Вам должен дать преподаватель), и, далее, выполнить задание Вашего варианта лабораторной работы №17-18.

#### **Задание 2.** Изучение обработки части массива

**2.1.** Открыть «Сборник заданий по программированию» А.П. Нестерова (электронную версию Вам должен дать преподаватель), и, далее, выполнить задание Вашего варианта лабораторной работы №19.

### **Контрольные вопросы:**

1. Что такое массив?
2. Что такое двумерный массив?
3. Как инициализировать элементы двумерного массива?
4. Какими составляющими характеризуется массив?
5. Какие существуют способы обработки двумерных массивов?

## **ЛАБОРАТОРНАЯ РАБОТА 6. ПРИМЕНЕНИЕ ПРОЦЕДУР И ФУНКЦИЙ ПРИ РАЗРАБОТКЕ ПРОГРАММ**

**Цель занятия:** научиться использовать подпрограммы (процедуры и функции) для решения различных задач.

### **Теоретические сведения. Понятие процедур и функций**

Алгоритм, выполняющий некоторую относительно автономную, законченную часть основной задачи, называют вспомогательным алгоритмом, а соответствующую "вспомогательную программу" — подпрограммой (или процедурой).

Участок программы, к которому можно обращаться из различных мест программы для выполнения некоторых действий называется подпрограммой (англ. Subprogram).

При написании программ часто при реализации алгоритма работы устройства приходится повторять одни и те же операторы (например, операторы, работающие с параллельным или последовательным портом). Было бы неплохо использовать один и тот же участок кода, вместо того, чтобы повторять одни и те же операторы несколько раз. Подпрограмма может быть многократно вызвана из разных частей программы. В языках программирования для оформления и использования подпрограмм существуют специальные синтаксические средства.

Главное назначение подпрограмм — структуризация программы с целью удобства её понимания и сопровождения:

Выделение набора действий в подпрограмму и вызов её по мере необходимости позволяет логически выделить целостную подзадачу, имеющую типовое решение, и экономить память. Даже в тех случаях, когда в подпрограмму выделяется однократно производимый набор действий, это оправдано, так как позволяет сократить размеры целостных блоков кода, составляющих программу, то есть сделать программу более понятной и обзорной.

В простейшем случае подпрограмма представляет собой последовательность команд (операторов), отдельную от основной части программы и имеющую в конце специальную команду выхода из подпрограммы. Обычно подпрограмма имеет имя, по которому её можно вызвать. Вызов подпрограммы выполняется с помощью команды вызова, включающей в себя имя подпрограммы. В большинстве современных языков программирования команда вызова представляет собой просто имя вызываемой подпрограммы, за которым могут следовать её параметры.

Для обеспечения контролируемой передачи параметров в подпрограмму и возврата результатов из неё используется механизм параметров. Параметры описываются при описании подпрограммы (в её заголовке) и могут использоваться внутри процедуры аналогично переменным, описанным в ней.

В языках программирования высокого уровня используется два типа подпрограмм: процедуры и функции.

Функция — это подпрограмма специального вида, которая, кроме получения параметров, выполнения действий и передачи результатов работы через параметры имеет ещё одну возможность — она может возвращать результат. Вызов функции является, с точки зрения языка программирования, выражением, он может использоваться в других выражениях или в качестве правой части присваивания.

Процедура — это любая подпрограмма, которая не является функцией. Подпрограммы, входящие в состав классов в объектных языках программирования, обычно называются методами. Этим термином называют любые подпрограммы-члены класса, как функции, так и процедуры; когда требуется уточнение, говорят о методах-процедурах или методах-функциях.

Впрочем, деление на процедуры и функции весьма условно, потому что они очень близки. Отличаются они тем, что процедура просто выполняет группу операторов, а функция вдобавок вычисляет некоторое значение и передает его в программу.

Ячейки памяти, в которых хранятся адреса возврата из подпрограмм называются стеком. Логически эти ячейки памяти организованы так, чтобы считывание последнего записанного адреса производилось первым, а первого записанного адреса производилось последним.

Подпрограммы также делят на: подпрограммы с параметрами и подпрограммы без параметров. Подпрограмма с параметрами используется для записи многократно повторяющихся действий при разных исходных данных. Их можно разделить на два типа: подпрограммы-функции и просто подпрограммы с параметрами (их называют процедурами). Параметры, записанные в заголовке подпрограммы, называются формальными параметрами (аргументами). Для подпрограммы - это просто переменные. Параметры, подставленные в месте вызова подпрограммы, называются фактическими параметрами.

Для вызова подпрограммы без параметров нужно поместить перед ее именем символ &. Вызов подпрограммы без аргументов означает, что ей передается текущее состояние массива.

Подпрограммы также делятся по способу передачи параметров в подпрограмму: по имени, по значению, по ссылке, через стек.

### **Задание к лабораторной работе:**

Выполнение заданий 1 – 3 предполагает оформление отчета в текстовом редакторе Word (Times New Roman, 12 п., одинарный), в котором отражаются этапы выполнения заданий и результаты (ответы на вопросы, скриншоты заполненных полей калькулятора, документы аттестации ИБ и т.д.)

#### **Задание 1.** Изучение использования подпрограмм

**1.1.** Открыть «Сборник заданий по программированию» А.П. Нестерова (электронную версию Вам должен дать преподаватель), и, далее, выполнить задание Вашего варианта лабораторной работы №20-21.

#### **Задание 2.** Изучение обработки массивов

**2.1.** Открыть «Сборник заданий по программированию» А.П. Нестерова (электронную версию Вам должен дать преподаватель), и, далее, выполнить задание Вашего варианта лабораторной работы №22.

#### **Задание 3.** Изучение передачи массива как параметра

**3.1.** Открыть «Сборник заданий по программированию» А.П. Нестерова (электронную версию Вам должен дать преподаватель), и, далее, выполнить задание Вашего варианта лабораторной работы №23-24.

### **Контрольные вопросы:**

1. Что такое подпрограмма?
2. Что такое процедура?
3. Что такое функция?
4. Назовите главную функцию подпрограммы.
5. Что представляет собой подпрограмма?

## ЛАБОРАТОРНАЯ РАБОТА 7. РАЗРАБОТКА ПРОГРАММ С ИСПОЛЬЗОВАНИЕМ КОСВЕННОЙ АДРЕСАЦИИ

**Цель занятия:** освоение способа разработки программ с использованием косвенной адресации.

### Теоретические сведения. Понятие косвенной адресации

Адресация – это система задания адресов в командах микропроцессора или ЦВМ и соответствующих правил доступа к устройствам хранения информации (регистрам, ячейкам памяти, портам и др.) в соответствии с заданными адресами.

Система адресации предназначена для обеспечения доступа к к устройствам хранения информации.

Введём ряд новых понятий.

Исполнительный или абсолютный адрес (АИ) – это номер ячейки памяти (регистра), в которой хранится операнд, используемый в процессе выполнения команды, или результат операции.

Адресный код (АК) позволяет с помощью специальных преобразований определять исполнительный адрес.

Система адресации реализуется большим многообразием способов адресации.

В современных микропроцессорах нашли широкое применение практически все существующие способы адресации памяти.

Косвенная адресация – это такая адресация, при которой в адресном поле команды указан адрес ячейки памяти или регистра, в которых хранится не сам операнд, а его адрес.

Если этот адрес является абсолютным адресом операнда, то имеет место одноступенчатая косвенная адресация.

Если в ячейке памяти или регистре, адреса которых указаны в команде, содержится не абсолютный адрес операнда, а его адресный код, то имеется место многоступенчатая косвенная адресация. Обычно необходимость в ней возникает при использовании "ценных" списков, т.е. совокупности слов, связанных между собой адресами по принципу: первое слово, содержит адрес второго слова, второе слово третьего слова и т.д.

Количество косвенных адресов, обрабатываемых до получения прямого или непосредственного адреса, называется глубиной косвенной адресации.

Способы косвенной адресации:

На Рисунок 10 – Рисунок 13 приведены четыре основных способа косвенной адресации, широко используемых в микропроцессорах. На рисунках адреса, отмеченные знаком \*, могут иметь много уровневую адресацию.



Рисунок 10 Косвенно-регистровый способ адресации

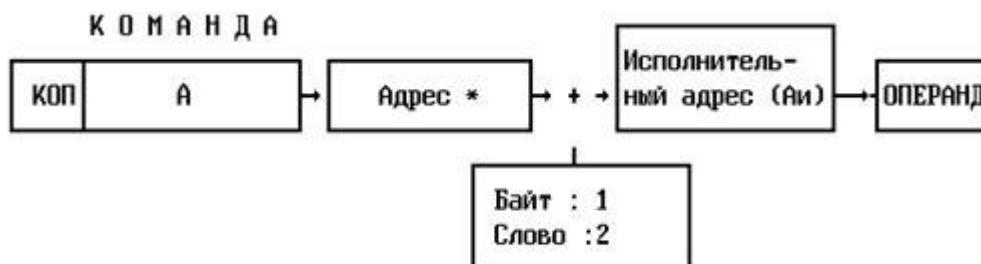


Рисунок 11 Косвенно-автоинкрементный способ адресации

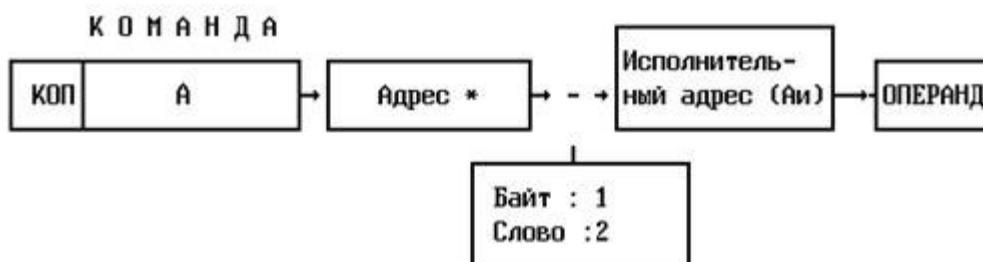


Рисунок 12 Косвенно-автодекрементный способ адресации.

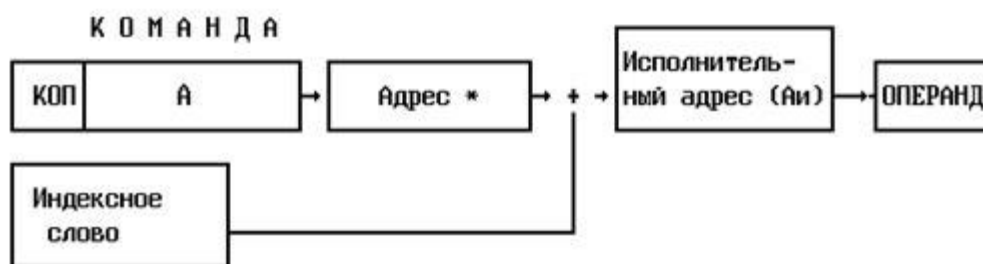


Рисунок 13 Косвенно-индексный способ адресации

Команды косвенной адресации наиболее эффективно используются при решении задач сбора, поиска и сортировки информации и при обработке списочных структур.

#### Задание к лабораторной работе:

Выполнение заданий 1 – 2 предполагает оформление отчета в текстовом редакторе Word (Times New Roman, 12 п., одинарный), в котором отражаются этапы выполнения заданий и результаты (ответы на вопросы, скриншоты заполненных полей калькулятора, документы аттестации ИБ и т.д.)

#### Задание 1. Изучение косвенной адресации

1.1. Открыть «Сборник заданий по программированию» А.П. Нестерова (электронную версию Вам должен дать преподаватель), и, далее, выполнить задание Вашего варианта лабораторной работы №27.

#### Задание 2. Изучение косвенной адресации

1. Открыть «Сборник заданий по программированию» А.П. Нестерова (электронную версию Вам должен дать преподаватель), и, далее, выполнить задание Вашего варианта лабораторной работы №29.

#### Контрольные вопросы:

1. Что такое адресация?
2. Что такое косвенная адресация?
3. Перечислите способы косвенной адресации.
4. Что такое адресный код?
5. Для чего предназначена система адресации?

## ЛАБОРАТОРНАЯ РАБОТА 8. РАЗРАБОТКА ТЕСТА ДЛЯ ПРОГРАММЫ И ЕЁ ТЕСТИРОВАНИЕ

**Цель занятия:** освоение способов и методов тестирования программы, создание теста для программы.

### **Теоретические сведения. Тестирование программы**

Есть множество разных типов тестов, которые вы можете применить, чтобы убедиться, что изменения в вашем коде работают по сценарию. Не все типы тестирования идентичны, хотя здесь мы рассмотрим, насколько основные практики тестирования отличаются друг от друга.

Тестирование: ручное или автоматизированное?

Сначала надо понять различия между ручными и автоматизированными тестами. Ручное тестирование проводится непосредственно человеком, который нажимает на кнопки в приложении или взаимодействует с программным обеспечением или API с необходимым инструментарием. Это достаточно затратно, так как это требует от тестировщика установки среды разработки и выполнения тестов вручную. Имеет место вероятность ошибки за счет человеческого фактора, например, опечатки или пропуска шагов в тестовом сценарии.

Автоматизированные тесты, с другой стороны, производятся машиной, которая запускает тестовый сценарий, который был написан заранее. Такие тесты могут сильно варьироваться в зависимости от сложности, начиная от проверки одного единственного метода в классе до отработки последовательности сложных действий в UI, чтобы убедиться в правильности работы. Такой способ считается более надежным, однако его работоспособность все еще зависит от того насколько скрипт для тестирования был хорошо написан.

Автоматизированные тесты – это ключевой компонент непрерывной интеграции и непрерывной доставки, а также хороший способ масштабировать ваш QA процесс во время добавления нового функционала для вашего приложения. Однако в ручном тестировании все равно есть своя ценность.

С одной стороны, очень важно удостовериться, что пользователи смогут использовать ваше приложение («Я не могу войти в систему», «Я не могу сохранить данные» и т.п.), но с другой стороны не менее важно проверить, что ваша система не ломается при вводе неверных данных или неожиданных действиях. Вам нужно предвидеть, что произойдет, когда пользователь сделает опечатку, попытается сохранить неполную форму или использует неправильный API. Вам нужно проверить, сможет ли кто-то из пользователей легко скомпрометировать данные, получить доступ к тому или иному ресурсу, к которому у него не должно быть доступа. Хороший набор тестов должен попытаться сломать ваше приложение и помочь понять предел его возможностей.

### **Задание к лабораторной работе:**

Выполнение заданий 1 – 3 предполагает оформление отчета в текстовом редакторе Word (Times New Roman, 12 п., одинарный), в котором отражаются этапы выполнения заданий и результаты (ответы на вопросы, скриншоты заполненных полей калькулятора, документы аттестации ИБ и т.д.)

#### **Задание 1. Создание теста для программы**

**1.1.** Открыть «Сборник заданий по программированию» А.П. Нестерова (электронную версию Вам должен дать преподаватель), и, далее, выполнить задание Вашего варианта лабораторной работы №30.

#### **Задание 2. Создание теста для программы**

**2.1.** Открыть «Сборник заданий по программированию» А.П. Нестерова (электронную версию Вам должен дать преподаватель), и, далее, выполнить задание Вашего варианта лабораторной работы №32.

#### **Задание 3. Тестирование программы**

**3.1.** Открыть «Сборник заданий по программированию» А.П. Нестерова (электронную версию Вам должен дать преподаватель), и, далее, выполнить задание Вашего варианта лабораторной работы №33.

**Контрольные вопросы:**

1. Для чего проводится тестирование?
2. Какие виды тестирования существуют?
3. Охарактеризуйте виды тестирования.
4. Перечислите виды теста.
5. Перечислите минусы ручного тестирования.

**РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА**

1. Ишкова Э. А. Самоучитель С#. Начала программирования. Серия: Самоучитель. Издательство: Наука и Техника, 2013 г. 496 с.
2. Васильев А. Программирование на С# для начинающих. Основные сведения. Серия: Российский компьютерный бестселлер. – М.: Эксмо, 2018. – 592 с.
3. Справочник по С#. URL: <https://docs.microsoft.com/ru-ru/dotnet/csharp/language-reference/>
4. Пахомов Б. И. С# для начинающих. — СПб.: БХВ-Петербург, 2014. — 432 с. URL: [https://lesmatveev.narod.ru/knigi\\_proect/ci\\_sharp.pdf](https://lesmatveev.narod.ru/knigi_proect/ci_sharp.pdf)

## 2 СЕМЕСТР

### ЛАБОРАТОРНАЯ РАБОТА 1 ОСНОВЫ РАБОТЫ В СИСТЕМЕ MATHCAD

#### *Цель работы:*

получение основных навыков работы с системой Mathcad (построение графиков функций, решение задач вычислительной математики).

#### *Задачи лабораторной работы:*

- изучение основных элементов интерфейса системы Mathcad;
- изучение работы с формульным редактором;
- получение навыков работы со встроенными функциями и элементами графической визуализации;
- изучение основных объектов входного языка системы Mathcad;
- освоение символьных преобразований в системе Mathcad.

#### **1. Выполнение лабораторной работы**

На основе изучения теоретического материала учебного пособия, представленного в файле «Книга\_Mathcad» (глава 1, пункты 1.1-1.6; глава 2, пункты 2.1-2.4), и практических примеров, представленных в файле «Пример\_1», выполнить задания 1-10 в соответствии с вариантом.

#### **ЗАДАНИЯ**

- 1) Построение графиков функций.
  - 1.1. Построить графики для любых 2-х функций из предложенного списка.
  - 1.2. Построение графиков по табличным данным.
- 2) Найти символьные производные от функций и вычислить их значения в соответствующей точке.
- 3) Вычислить пределы.
- 4) Найти интегралы.
- 5) Вычислить значение определенного интеграла. Проверить с использованием символьных вычислений и формулы Ньютона-Лейбница.
- 6) Вычислить значение интеграла с помощью встроенной функции.
- 7) Найти частные производные (первого и второго порядка) от функций.
- 8) Определить корень уравнения на заданном интервале, с использованием встроенной функции «*root*».
- 9) Определить корень уравнения по формуле Ньютона на заданном интервале, используя элементы программирования. Сравнить полученные результаты.
- 10) Решить задачу.

#### **2. Подготовка отчета по лабораторной работе**

1. Отчет должен отражать выполнение основных этапов работы в виде иллюстраций с листингами программ системы Mathcad и полученными результатами.
2. Отчет должен содержать ответы на контрольные вопросы.

#### **3. Контрольные вопросы**

1. С помощью какого оператора можно вычислить выражение?
2. Как вставить текстовую область в документ Mathcad?
3. Чем отличается глобальное и локальное определение переменных? С помощью каких операторов определяются?
4. Как изменить формат чисел для всего документа?

5. Присваивание переменным значений. Выполнение операций с переменными. Задавание ранжированных переменных.
6. Как вставить встроенную функцию в документ Mathcad?
7. С помощью каких операторов можно вычислить интегралы, производные, суммы и произведения? Как определить дискретные переменные с произвольным шагом? Какой шаг по умолчанию?
8. Как определить индексированную переменную?
9. Какая системная переменная определяет нижнюю границу индексации элементов массива?
10. Как построить графики: поверхности; полярный; декартовый?
11. Как построить несколько графиков в одной системе координат?
12. Как изменить масштаб графика?
13. Как определить координату точки на графике?
14. Какие функции используются для построения трехмерных графиков?
15. Какие функции для решения одного уравнения в Mathcad вы знаете? В чем их отличие?
16. Какие аргументы функции «root» не обязательны?
17. В каких случаях Mathcad не может найти корень уравнения?
18. Как изменить точность, с которой функция «root» ищет корень?

## ЛАБОРАТОРНАЯ РАБОТА 2 РЕШЕНИЕ ЗАДАЧ ВЫЧИСЛИТЕЛЬНОЙ МАТЕМАТИКИ В СИСТЕМЕ MATHCAD

*Цель работы:*

получение основных навыков работы с системой Mathcad (решение задач вычислительной математики).

*Задачи лабораторной работы:*

- освоение методов решения систем нелинейных алгебраических уравнений в системе Mathcad;
- освоение методов решения систем обыкновенных дифференциальных уравнений в системе Mathcad

### 1. Выполнение лабораторной работы

#### 1.1. Решение систем нелинейных алгебраических уравнений в системе Mathcad.

На основе изучения теоретического материала учебного пособия, представленного в файле «Книга\_Mathcad» (глава 2, пункт 2.9.), и практических примеров, представленных в файле «Пример\_2», решить 2 системы нелинейных уравнений в соответствии с вариантом.

Таблица – Варианты заданий для лабораторной работы

1	$\begin{cases} x + y + z = 4 \\ x + 2y + 3z = 5 \\ x^2 + y^2 + z^2 = 14 \end{cases}$	2	$\begin{cases} \sqrt{4x + y - 3z + 7} = 2 \\ \sqrt[3]{2y + 5x + z + 25.5} = 3 \\ \sqrt{y + z} - \sqrt{6x} = 0 \end{cases}$
3	$\begin{cases} x + y + z = 13 \\ x^2 + y^2 + z^2 = 61 \\ xy + xz = 2yz \end{cases}$	4	$\begin{cases} x^2 + y^2 = z^2 \\ xy + yz + zx = 47 \\ (z - x)(z - y) = 2 \end{cases}$
5	$\begin{cases} x^2 + y^2 = 2(xy + 2) \\ x + y = 6 \end{cases}$	6	$\begin{cases} x + xy + y = 11 \\ x^2y + xy^2 = 30 \end{cases}$
7	$\begin{cases} x + y^2 = 7 \\ xy^2 = 12 \end{cases}$	8	$\begin{cases} x^2 - y = 23 \\ x^2y = 50 \end{cases}$
9	$\begin{cases} (x^2 - y^2)xy = 180 \\ x^2 - xy - y^2 = -11 \end{cases}$	10	$\begin{cases} 3x^2 - 2xy + 5y^2 - 35 = 0 \\ 5x^2 - 10y^2 - 5 = 0 \end{cases}$
11	$\begin{cases} \frac{12}{\sqrt{x-1}} + \frac{5}{\sqrt{y+\frac{1}{4}}} = 5 \\ \frac{8}{\sqrt{x-1}} + \frac{10}{\sqrt{y+\frac{1}{4}}} = 6 \end{cases}$	12	$\begin{cases} x + y - 2\sqrt{xy} = 4 \\ x + y = 10 \end{cases}$
13	$\begin{cases} \sqrt{\frac{3x}{x+y}} - 2 + \sqrt{\frac{x+y}{3x}} = 0 \\ xy - 54 = x + y \end{cases}$	14	$\begin{cases} \frac{1}{4}\sqrt[3]{x^2 + y^2} - \frac{1}{2}\sqrt[3]{17} = 0 \\ \sqrt{x+y} + \sqrt{x-y} = 6 \end{cases}$
15	$\begin{cases} \sqrt{x^2 + y^2} - \sqrt{x^2 - y^2} = y \\ x^4 - y^4 = 144a^4 \end{cases}$	16	$\begin{cases} x^2 + xy + y^2 = 84 \\ x + \sqrt{xy} + y = 14 \end{cases}$
17	$\begin{cases} x^2 + y^2 = \frac{5}{2}xy \\ x - y = \frac{1}{4}xy \end{cases}$	18	$\begin{cases} x^2 + xy + y^2 = 13 \\ x + y = 4 \end{cases}$
19	$\begin{cases} x^2 - xy + y^2 = 7 \\ x - y = 1 \end{cases}$	20	$\begin{cases} \frac{x}{y} + \frac{y}{x} = \frac{25}{12} \\ x^2 - y^2 = 7 \end{cases}$

### 1.2. Решение обыкновенных дифференциальных уравнений в системе Mathcad

На основе изучения теоретического материала учебного пособия, представленного в файле «Книга\_Mathcad» (глава 2, пункт 2.12.), и практических примеров, представленных в файле «Пример\_4», решить систему обыкновенных дифференциальных уравнений в соответствии с вариантом.

При отсутствии начальных условий в задании принять их равными нулю.

## 2. Подготовка отчета по лабораторной работе

1. Отчет должен отражать выполнение основных этапов работы в виде иллюстраций с листингами программ системы Mathcad и полученными результатами.
2. Отчет должен содержать ответы на контрольные вопросы.

## 3. Контрольные вопросы

1. Назовите функции для решения систем уравнений в MathCAD и особенности их применения.
2. Какой знак равенства используется в блоке решения? Какой комбинацией клавиш вставляется в документ?
3. Какие выражения не допустимы внутри блока решения уравнения?
4. Опишите способы использования функции «Find».
5. В каких случаях MathCAD не может найти решение системы уравнений?
6. Дайте сравнительную характеристику функциям «Find» и «Minerr».
7. Назовите функции для решения систем дифференциальных уравнений в Mathcad и особенности их применения.
8. Какой знак равенства используется в блоке решения? Какой комбинацией клавиш вставляется в документ?
9. Дайте сравнительную характеристику различных функций для решения систем дифференциальных уравнений.

## ЛАБОРАТОРНАЯ РАБОТА 3 ОБРАБОТКА ЭКСПЕРИМЕНТАЛЬНЫХ ДАННЫХ С ПОМОЩЬЮ МЕТОДОВ РЕГРЕССИОННОГО АНАЛИЗА

*Цель работы:*

получение основных навыков обработки экспериментальных данных в системе Mathcad.

*Задача лабораторной работы:*

освоение метода обработки данных с помощью линейной регрессии в системе Mathcad.

### 1. Выполнение лабораторной работы

Широко распространенной задачей обработки данных является представление их совокупности некоторой функцией  $y(x)$ . Задача регрессии заключается в получении параметров этой функции такими, чтобы функция приближала «облако» исходных точек (заданных векторами  $VX$  и  $VY$ ) с наименьшей среднеквадратичной погрешностью.

Чаще всего используется линейная регрессия вида  $y(x)=a+b \cdot x$ . К линейной регрессии можно свести нелинейную регрессию вида  $y=f(x)$ , (где  $f(x)$  – некоторая нелинейная функция). Для осуществления линейной регрессии в систему Mathcad встроен ряд приведенных ниже функций:

- $\llcorner corr \llcorner$  ( $VX, VY$ ) – возвращает скаляр – коэффициент корреляции Пирсона;
- $\llcorner intercept \llcorner$  ( $VX, VY$ ) – возвращает значение параметра  $a$  (смещение линии регрессии по вертикали);
- $\llcorner slope \llcorner$  ( $VX, VY$ ) – возвращает значение параметра  $b$  (угловой коэффициент линии регрессии).

Пример решения представлен на рисунке 1.

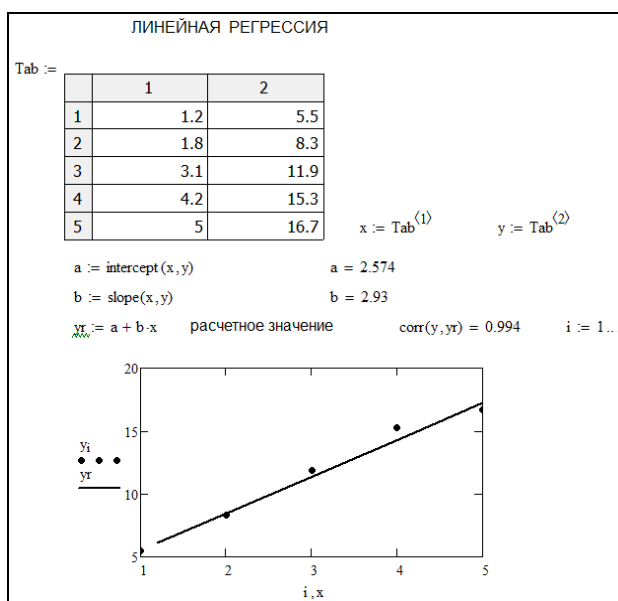


Рисунок 1 – Построение линейной регрессии

Как видно из рисунка 1, прямая регрессии проходит в «облаке» исходных точек с максимальным среднеквадратичным приближением к ним (коэффициент корреляции = 0,994).

### 1.1. Пример выполнения задания – построение простой линейной регрессии.

Пусть заданы значения  $(x_k, y_k)$ ,  $k = 1, 2, \dots, 12$ .

Пример решения представлен на рисунке 2.

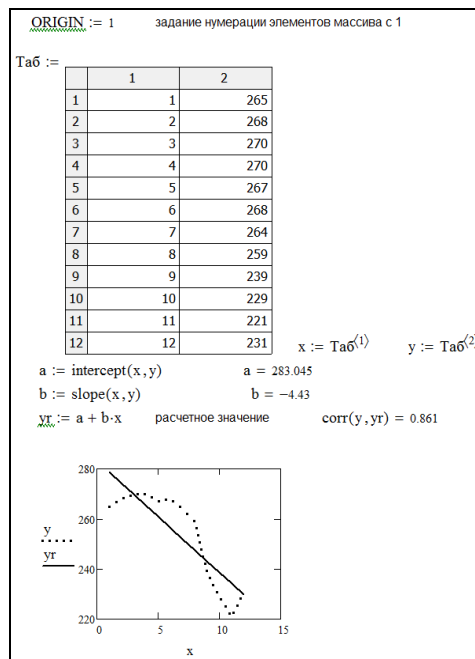


Рисунок 2 – Построение линейной регрессии

### 1.2. Пример выполнения задания – построение простой линейной регрессии с использованием предварительной линеаризации.

Достаточно часто встречаются задачи, в которых требуется выполнить предварительную линеаризацию уравнений.

Требуется найти параметры зависимости  $Y = X/(BX - A)$ .

Приведем данную зависимость к виду  $y(x) = a + bx$ .

Решение представлено на рисунке 3:

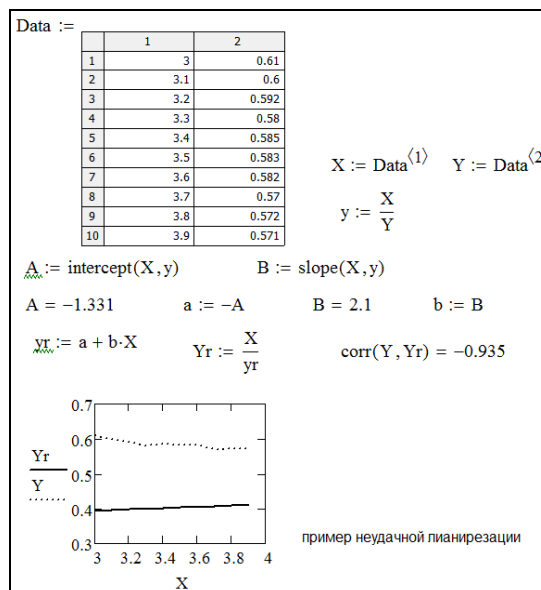


Рисунок 3 – Построение простой линейной регрессии

### 1.3. Пример выполнения задания – построение линейной регрессии общего вида.

В системе Mathcad реализована возможность выполнения линейной регрессии общего вида. При данной регрессии заданная совокупность точек приближается функцией вида:

$$F(x, K_1, K_2, \dots, K_n) = K_1 \cdot F_1(x) + K_2 \cdot F_2(x) + \dots + K_n \cdot F_n(x).$$

Таким образом, функция регрессии является линейной комбинацией функций  $F_1(x)$ ,  $F_2(x)$ , ...,  $F_n(x)$ , причем сами эти функции могут быть нелинейными.

Для реализации линейной регрессии общего вида используется функция *linfit* ( $VX$ ,  $VY$ ,  $F$ ). Она возвращает вектор коэффициентов линейной регрессии общего вида  $K$ , при котором среднеквадратичная погрешность приближения «облака» исходных точек, координаты которых хранятся в векторах  $VX$  и  $VY$ , оказывается минимальной. Вектор  $F$  должен содержать функции  $F_1(x)$ ,  $F_2(x)$ , ...,  $F_n(x)$ , записанные в символьном виде (рисунок 4). Вектор  $VX$  должен содержать абсциссы, упорядоченные в порядке их возрастания, а вектор  $VY$  – содержать ординаты, соответствующие абсциссам в векторе  $VX$ .

Пример решения представлен на рисунке 4.

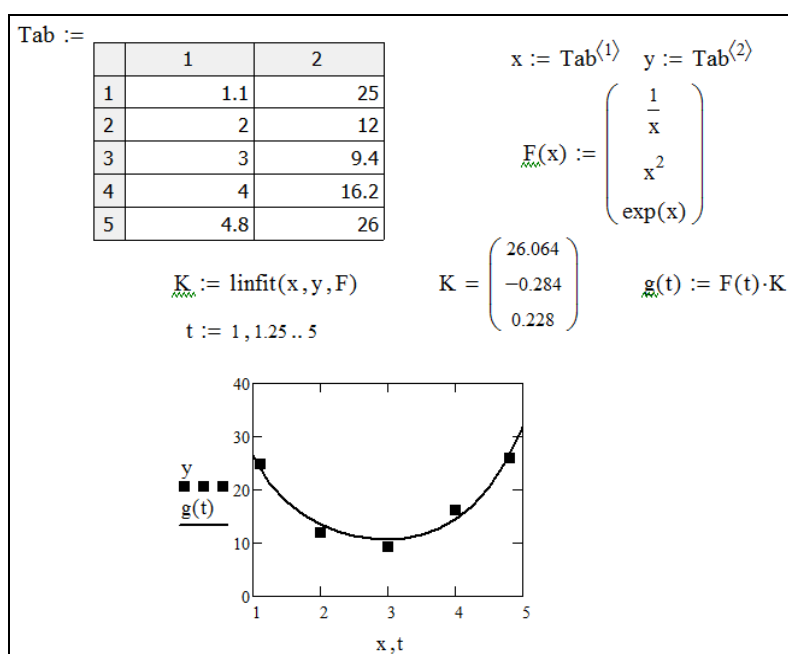


Рисунок 4 – Построение линейной регрессии общего вида

#### 1.4. Пример выполнения задания – построение полиномиальной регрессии.

В системе Mathcad введена функция для обеспечения полиномиальной регрессии при произвольной степени полинома регрессии: *regress* ( $VX$ ,  $VY$ ,  $n$ ). Она возвращает вектор  $VS$ , запрашиваемый функцией *interp* ( $VS$ ,  $VX$ ,  $VY$ ,  $x$ ), содержащий коэффициенты многочлена  $n$ -й степени, который наилучшим образом приближает «облако» точек с координатами, хранящимися в векторах  $VX$  и  $VY$  (рисунок 5). Для вычисления коэффициентов полинома регрессии используется функция «*submatrix*».

Пример решения представлен на рисунке 5.

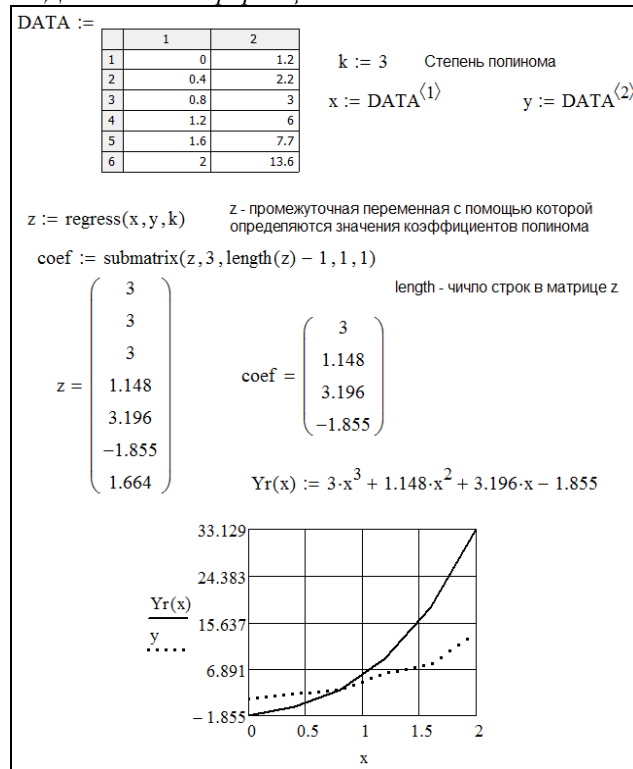


Рисунок 5 – Построение полиномиальной регрессии

Функция «*regress*» создает единственный приближающий полином, коэффициенты которого вычисляются по всей совокупности заданных точек. Иногда полезна другая функция полиномиальной регрессии, дающая локальные приближения отрезками полиномов второй степени: *loess* (*VX*, *VY*, *span*) – возвращает вектор *VS*, используемый *interp* (*VS*, *VX*, *VY*, *x*) для наилучшего приближения данных *VX* и *VY* отрезками полиномов второй степени. Аргумент *span* > 0 указывает размер локальной области приближаемых данных (рекомендуемое начальное значение – 0,75). Чем больше «*span*», тем сильнее сказывается сглаживание данных. При больших значениях «*span*» эта функция приближается к *regress*(*VX*, *VY*, 2)..

### 1.5. Пример выполнения задания – построение нелинейной регрессии общего вида.

Под нелинейной регрессией общего вида подразумевается нахождение вектора *K* параметров произвольной функции  $F(x, K1, K2, \dots, Kn)$ , при котором обеспечивается минимальная среднеквадратичная погрешность приближения «облака» исходных точек. Для проведения нелинейной регрессии общего вида используется функция *genfit* (*VX*, *VY*, *VS*, *F*). Она возвращает вектор *K* параметров функции *F*, дающий минимальную среднеквадратичную погрешность приближения функцией  $F(x, K1, K2, \dots, Kn)$  исходных данных. Функция *F* должна быть представлена вектором с символьными элементами, причем они должны содержать аналитические выражения для исходной функции и ее производных по всем параметрам. Вектор *VS* должен содержать начальные значения элементов вектора *K*, необходимые для решения системы нелинейных уравнений регрессии итерационным методом. На рисунке 6 показан пример выполнения нелинейной регрессии общего вида.

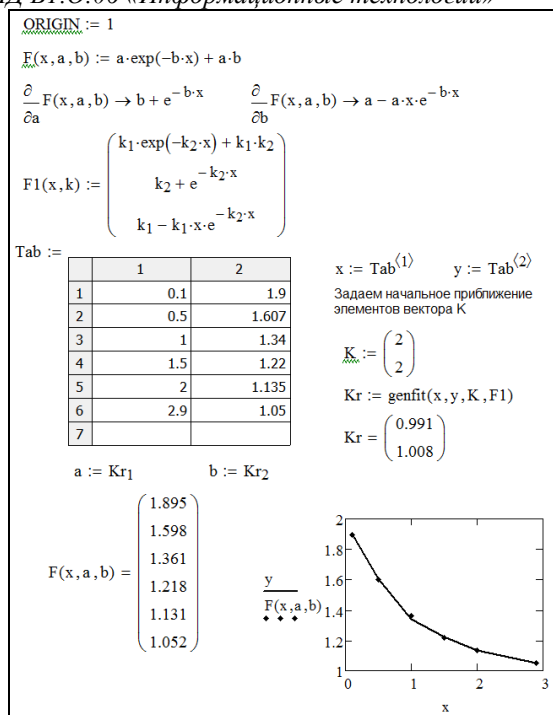


Рисунок 6 – Построение нелинейной регрессии общего вида

При решении этой задачи возникают две проблемы. Во-первых, надо вычислить значения производных по переменным  $a$  и  $b$ . В документе на рисунке 6 это сделано с помощью символьных операций (третья строка документа после заголовка), что наглядно показывает пользу от таких операций. Вторая проблема связана с необходимостью применения функции «genfit» в ее стандартном виде. Поэтому пришлось заменить параметр  $a$  на  $k_1$ , а параметр  $b$  на  $k_2$ .

### ЗАДАНИЕ

По заданным экспериментальным данным с помощью предложенных выше методов определить коэффициенты регрессии, построить графики экспериментальной и расчетной зависимостей, рассчитать коэффициент корреляции, определить суммарную ошибку расчетов:  $((Y_p - Y_{\text{э}}) / Y_{\text{э}}) \cdot 100, \%$ .

### 2. Подготовка отчета по лабораторной работе

Отчет должен отражать выполнение основных этапов работы в виде иллюстраций с листингами программ системы Mathcad и полученными результатами.

Отчет должен содержать ответы на контрольные вопросы.

### 3. Контрольные вопросы

1. Для чего используются функции «corr», «intercept», «slope» в системе Mathcad?
2. Какая функция в системе Mathcad предназначена для построения регрессии общего вида?
3. Какие функции в системе Mathcad предназначены для построения полиномиальной регрессии?
4. Какая функция в системе Mathcad предназначена для построения нелинейной регрессии?
5. Для чего предназначена функция «submatrix» в системе Mathcad?
6. Укажите каким образом в системе Mathcad вычисляются частные производные?

## ЛАБОРАТОРНАЯ РАБОТА 4 ВВЕДЕНИЕ В ВЕБ-ТЕХНОЛОГИИ

### ВВЕДЕНИЕ

CMS, от английского ContentManagementSystem (система управления контентом), - это программное обеспечение, позволяющее пользователям размещать или изменять уже размещенную на сайте информацию без привлечения разработчиков сайта. Это значит, что пользователю не обязательно обладать навыками программирования или знаниями языка HTML, чтобы, например, опубликовать на своем сайте новость, статью или добавить изображение. Часто наряду с термином CMS можно услышать также термин "движок сайта", которым обычно пользуются веб-мастера в своем профессиональном сленге.

Состоят CMS обычно из двух частей:

- back-office - это часть системы, отвечающая за функциональность и хранение информации;

- front-office - это часть системы, обеспечивающая интерфейс с пользователем.

Принцип работы всех CMS основан на разделении контента (содержания) и дизайна (оформления) сайта. Обычно дизайн сайта меняется редко, тогда как изменения контента могут происходить не только каждый день, но и даже каждый час. Поэтому в своей работе CMS используют так называемые шаблоны - специальные "пустые" заготовки страниц, в которых дизайн сайта уже прописан и осталось лишь наполнить их информацией. Пользователю достаточно воспользоваться специальным WYSIWYG-редактором. Этот редактор по внешнему виду очень похож на привычные текстовые редакторы офисных приложений, поэтому пользователю не составляет особого труда освоить его. А наличие в системе большого количества готовых шаблонов дает возможность выбрать подходящий дизайн буквально в считанные минуты.

Информация хранится в базе данных, например, в MySQL и вызывается из нее при загрузке страниц сайта.

Работа CMS не требует установки дополнительного ПО, поскольку сама система находится на сервере, а доступ к ней осуществляется через обычный интернет-браузер.

### ЗАДАНИЕ

1. Установить, настроить и запустить Open Server (портативный локальный WAMP/WNMP сервер, имеющий многофункциональную управляющую программу и большой выбор подключаемых компонентов) и систему управления содержимым сайта с открытым исходным кодом WordPress (см. файл «Установка\_ПО»).

2. Осуществить персонализацию и наполнение веб-страницы материалом.

Перейти в консоль администратора по адресу localhost/admin. Ввести логин и пароль, которые указали ранее. В левом столбце выбрать «Внешний вид» и перейти на вкладку «Темы». Производим выбор темы «TwentySixteen».

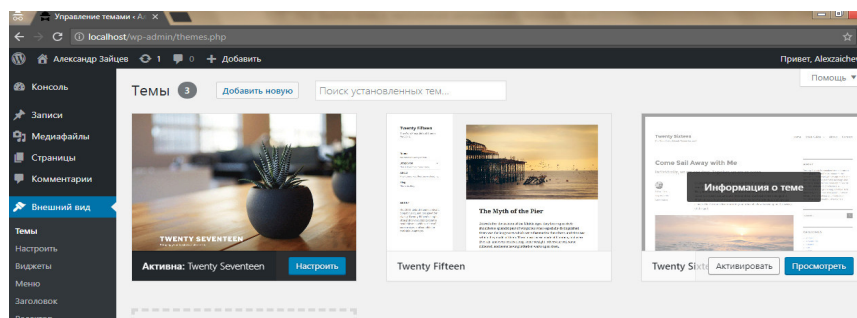


Рисунок – Выбор темы оформления веб-страницы

Свойства сайта - вписать «Лабораторная работа».

Настроить цвета. Изображение заголовка. Добавить изображение.

Фоновое изображение. Возможно добавить изображение на фон сайта Меню. Это область в которой отображаются рубрики и страницы.

Виджеты. Это правая область сайта, в которой расположены поиск и т.д. (настроить на свое усмотрение).

Дополнительные стили. Данный раздел использовать не будем.

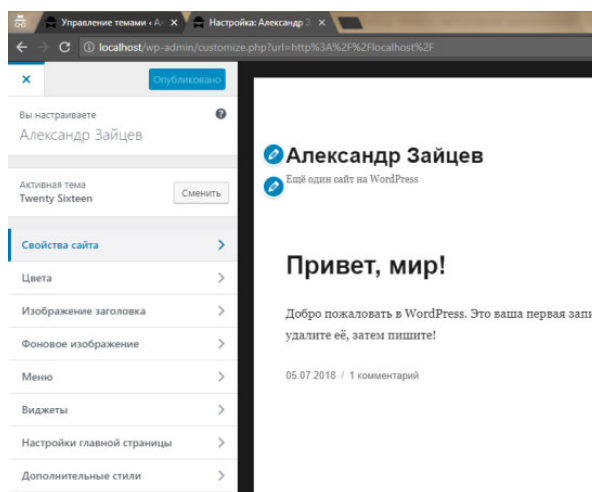


Рисунок – Внесение изменения

Вернуться в консоль администратора и выбрать раздел рубрики. Нажимаем справа раздел записи, далее рубрики и добавляем новую рубрику «Мои увлечения» Нажимаем «Добавить новую рубрику», представлено на рисунке.

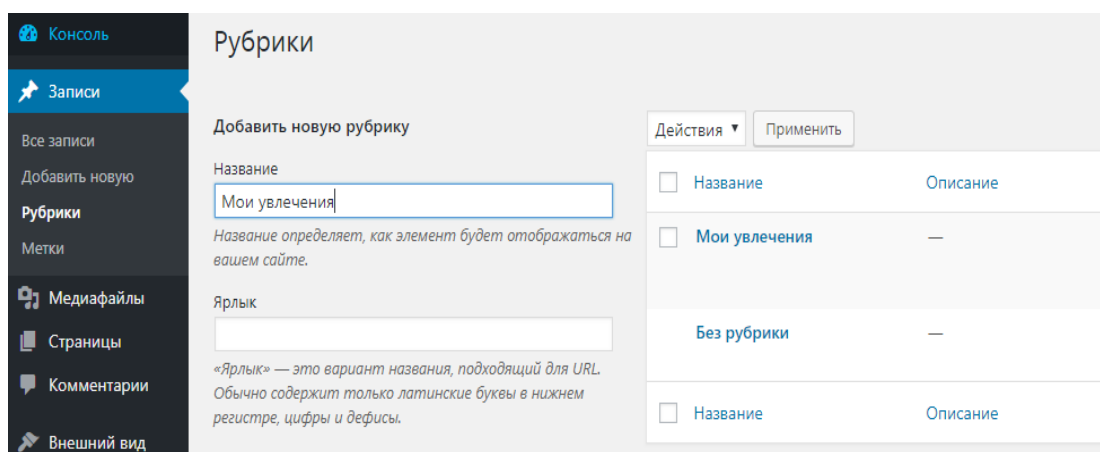


Рисунок – Создание рубрики

После создания рубрики нажать «Записи» и «Добавить новую». Создать 3 записи с вашими увлечениями. Поместить их в рубрику «Мои увлечения», как представлено на рисунке. Далее - «Опубликовать».

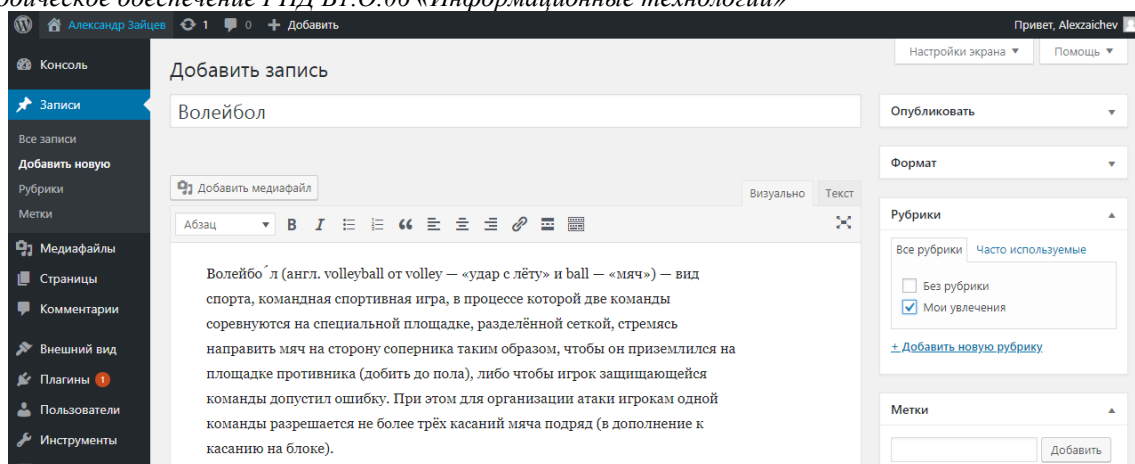


Рисунок – Добавление записи

Перейти в раздел «Страницы» и «Добавить новую». Добавить 2 страницы: «О себе»; «Мой институт» (см. рисунок).

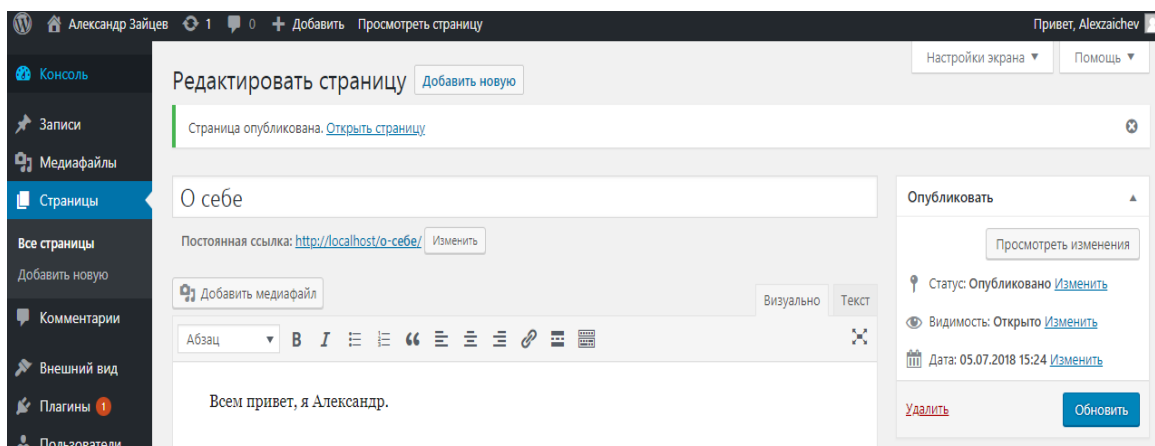


Рисунок – Добавление страницы

После публикации записей в рубриках и страниц перейти в раздел «Внешний вид», далее «Меню». Создать меню, как представлено на рисунке.

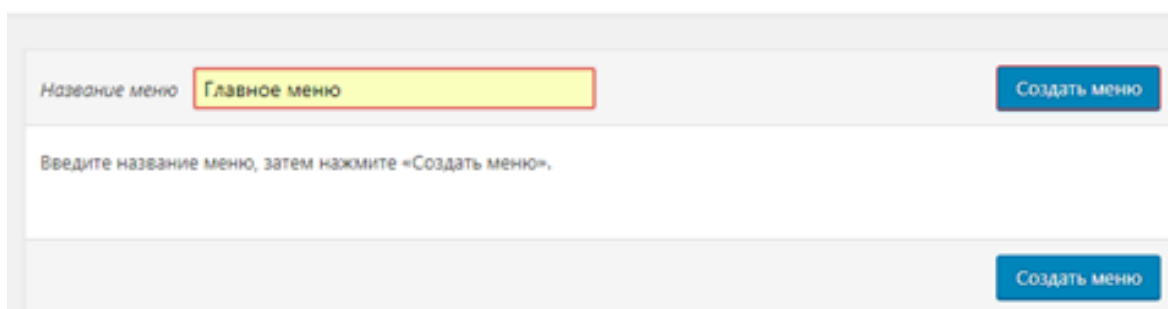


Рисунок 2.6 – Создание меню

Выбрать рубрику. Выставить галочку в «Область отображения Основное меню», как представлено на рисунке. Нажать «Сохранить меню».

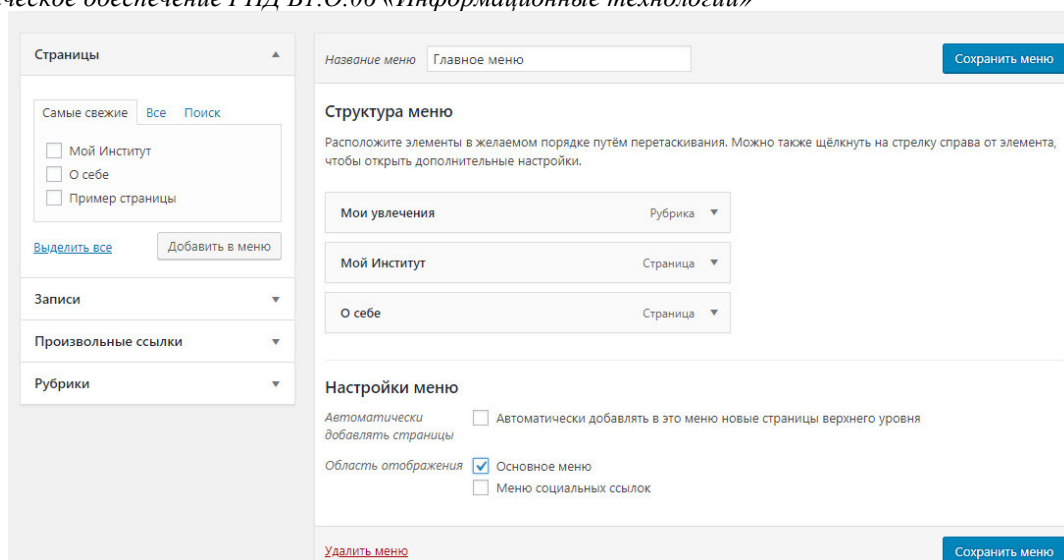


Рисунок – Настройка меню

Переходим на главную страницу сайта по адресу localhost. Справа сверху должны появиться элементы, которые вы выбрали, как показано на рисунке.

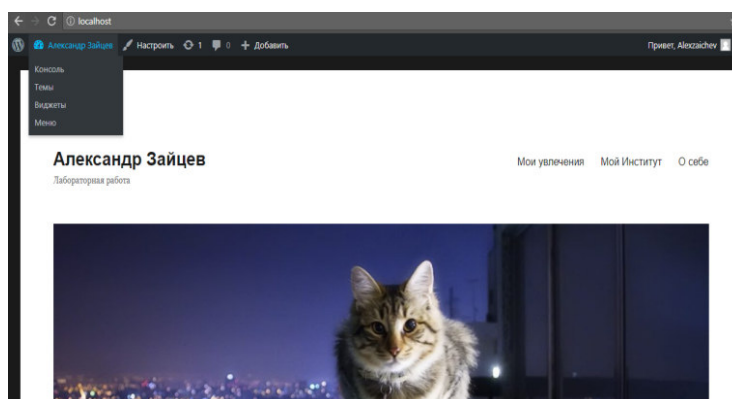


Рисунок – Проверка элементов

### 3. Создание и настройка контактной формы ContactForm 7.

Контактная форма помогает обрабатывать заявки посетителей сайта и оперативно реагировать на полученную информацию.

Перейти в раздел «Плагины», затем «Добавить новый». Ввести в поиске ContactForm 7 и установить, как представлено на рисунке.

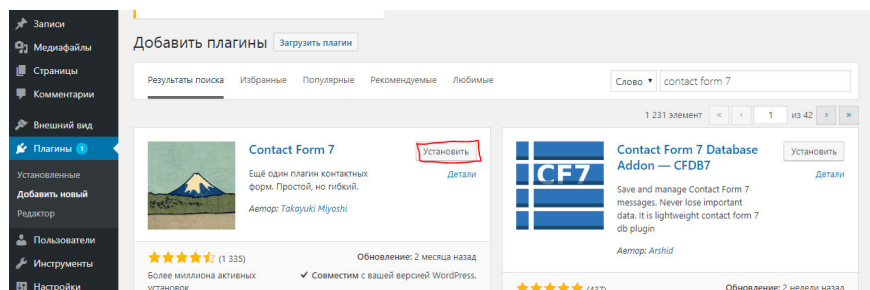
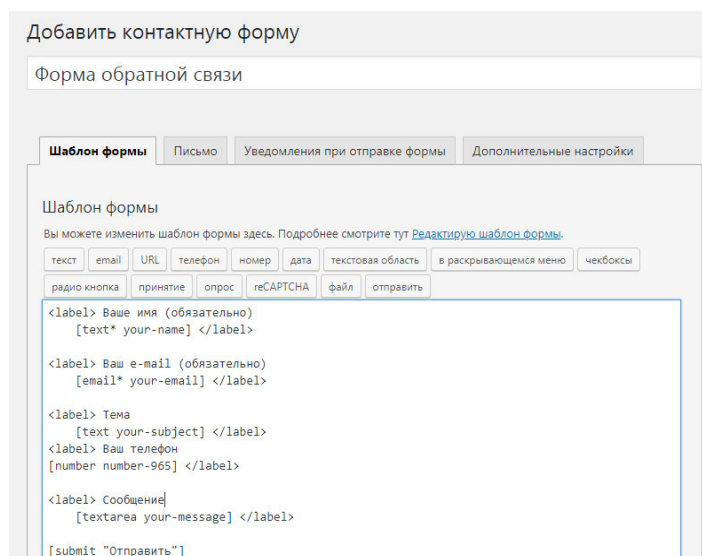


Рисунок – Установка плагина

Активировать плагин. При успешной активации плагина, в боковом меню появится раздел ContactForm 7. Открыть и перейти «Добавить новую». Заполнить название: Форма обратной связи. Добавляем форму для ввода сотового телефона путем использования, встроенного элемента как представлено на рисунке.



```
<label> Ваше имя (обязательно)
[text* your-name] </label>

<label> Ваш e-mail (обязательно)
[email* your-email] </label>

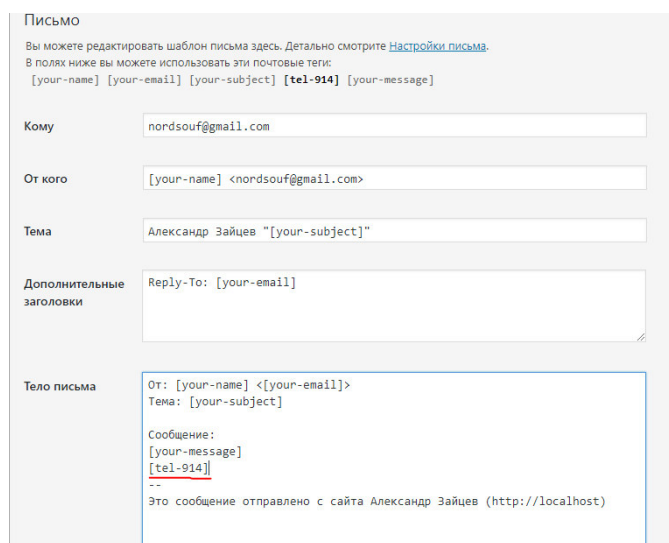
<label> Тема
[text your-subject] </label>
<label> Ваш телефон
[number number-965] </label>

<label> Сообщение
[textarea your-message] </label>

[submit "Отправить"]
```

Рисунок – Форма обратной связи

Перейти на вкладку «Письмо» и произвести настройку письма, которое будет вам приходиться с формы обратной связи. Сохранить настройки.



```
От: [your-name] <[your-email]>
Тема: [your-subject]

Сообщение:
[your-message]
[tel-914]
--
Это сообщение отправлено с сайта Александр Зайцев (http://localhost)
```

Рисунок – Настройка письма

Копируем шорткод, расположенный на синем фоне (пример: [contact-form-7 id=«23» title=«Форма обратной связи»]). Переходим на вкладку страницы и создаем новую страницу. Название страницы «Напишите нам» и копируем на эту страницу ваш шорткод, как представлено на рисунке. Публикуем страницу и добавляем её в меню.

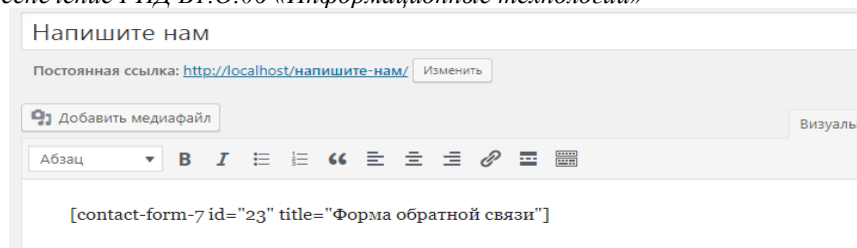


Рисунок – Шорткод

Проверить корректность работы формы обратной связи. Отправить тестовое сообщение. Успешная отправка показана на рисунке.

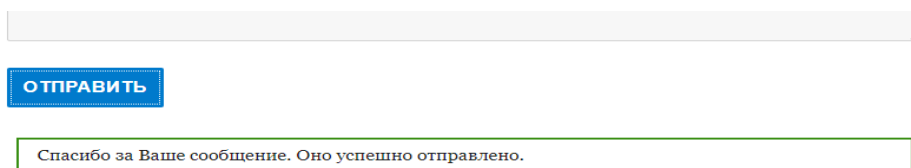


Рисунок – Успешная отправка сообщения

4. Подготовить отчет о проделанной работе.

#### **Контрольные вопросы.**

1. Опишите процесс инсталляции системы создания и управление сайтом «WordPress» на локальном веб-сервере.
2. Опишите стандартную операцию «О подключении к базе данных».
3. Для чего нужен `phpmyadmin` в данном случае?
4. Опишите выбор темы оформления в среде создания и управления сайтом Wordpress.
5. Опишите создание рубрики в среде создания и управления сайтом Wordpress.
6. Как добавить записи в среде создания и управления сайтом Wordpress?
7. Как добавить новые страницы в среде создания и управления сайтом Wordpress?
8. Опишите процесс создания меню в среде создания и управления сайтом Wordpress.
9. Опишите инструкцию настройки меню в среде создания и управления сайтом Wordpress.
10. Как установить плагины в среде создания и управления сайтом Wordpress?
11. Опишите процесс создания формы обратной связи в среде создания и управления сайтом Wordpress.
12. Как осуществить настройку обратных писем в среде создания и управления сайтом Wordpress?