

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ»**
(Н И У « Б е л Г У »)

ИНСТИТУТ ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ И ЕСТЕСТВЕННЫХ НАУК
КАФЕДРА МАТЕМАТИЧЕСКОГО И ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

**РАЗРАБОТКА WEB ПРИЛОЖЕНИЯ ДЛЯ ПОСТРОЕНИЯ ГРАФИКА
АНАЛИТИЧЕСКИ ЗАДАННОЙ ФУНКЦИИ**

Выпускная квалификационная работа
обучающегося по направлению подготовки 02.03.02. Фундаментальная
информатика и информационные технологии
заочной формы обучения, группы 07001360
Макушевой Юлии Александровны

Научный руководитель:
к.т.н., доцент
Бурданова Е.В.

БЕЛГОРОД 2017

Содержание

Введение.....	3
1 Разработка постановки задачи по созданию web-приложения	5
1.1 Существующие подходы к автоматизации построения графиков аналитически заданных функций	5
1.2 Общие сведения о web-приложениях и инструментах по их реализации	11
1.3 Выработка требований к web-приложению	15
2 Практическая реализация web-приложения	17
2.1 Выбор инструментальных средств для создания web-приложения	17
2.2 Определение архитектуры web-приложения	20
2.3 Описание порядка разработки web-приложения	24
3 Ввод web-приложения в эксплуатацию	34
3.1 Порядок размещения web-приложения в сети Интернет.....	34
3.2 Выбор способа продвижения web-приложения в поисковых системах....	41
Заключение	52
Список использованных источников	54
Приложения	57

ВВЕДЕНИЕ

В последнее время найти персональный компьютер, который не подключен к сети Интернет, весьма и весьма проблематично. Количество пользователей, регулярно пользующихся услугами сети Интернет, к середине 2016 года превысило три миллиарда человек. К Интернету подключаются как персональные компьютеры, так и иные устройства: сотовые телефоны, планшеты, телевизоры и многие другие. Используемые аппаратные и программные решения позволяют в режиме он-лайн смотреть фильмы высокого разрешения, слушать радиопередачи, передавать огромные объемы данных.

Представленные в рамках сети Интернет возможности реализуются с использованием служб (сервисов) и web-приложений. В настоящее время количество представленных в сети сервисов и приложений достаточно велико и оно непрерывно продолжает расти. Для того, чтобы полноценно использовать возможности глобальной сети, необходимо не только иметь представление о порядке использования сервисов и приложений, но и о порядке их создания.

Целью выполнения настоящей работы является создание web-приложения, предназначенного для построения графика аналитически заданной функции. Данная задача выглядит достаточно интересной. Круг пользователей, которые могут ей заинтересоваться может быть очень широким: студенты, школьники, научные работники. Учитывая то, что доступ к приложению будет реализован по средствам сети Интернет, реализованный сайт может получить большое количество посетителей.

В процессе выполнения работы будут выполнены следующие задачи:

- разработана постановка задачи для создания web-приложения – в рамках данной задачи будут изучены подходы к автоматизации построения графиков, рассмотрены инструменты для создания web-приложений и выработаны требования к проектируемому приложению;

– выполнена разработка web-приложения в соответствии с выбранным порядком реализации – в рамках данной задачи будут определены конкретные версии программных продуктов для создания приложения и выбрана методика его разработки, разработана архитектура приложения;

– определен порядок ввода готового приложения в эксплуатацию – в рамках данной задачи будет определен порядок размещения готового приложения в сети Интернет и его дальнейшего продвижения в поисковых системах (с целью увеличения количества посетителей).

Таким образом, выбранный в качестве темы дипломной работы проект выглядит в достаточной мере актуальным и интересным. Полученные в процессе написания работы результаты могут заинтересовать достаточно большое число пользователей сети Интернет.

Таким образом, объектом использования в рамках настоящей работы являются web-приложения. Предмет исследования – возможность их применения для вывода графика функции, заданной аналитически.

Методологическую основу для выполнения работы составили такие методы научного познания, как анализ, синтез и дедукция.

Сама тема использования web-приложения для решения различного рода прикладных задач исследована достаточно хорошо. Так, технические вопросы данной темы описаны в трудах Вязилова Е. (Архитектура, методы и средства Интернет-технологий), Морвиля М. и Розерфельда Л. (Информационная архитектура в Интернете), Никифорова А. (Компьютерные сети, протоколы и технологии Интернета).

Выпускная квалификационная работа содержит 56 страниц, 20 рисунков и 2 приложения на 15 страницах.

1 РАЗРАБОТКА ПОСТАНОВКИ ЗАДАЧИ ПО СОЗДАНИЮ WEB-ПРИЛОЖЕНИЯ

1.1 Существующие подходы к автоматизации построения графиков аналитически заданных функций

Перед тем как перейти к непосредственному описанию подходов к построению графиков функций, кратко рассмотрим некоторые теоретические сведения: что такое функция и какие способы ее задания существуют.

Существует несколько вариантов определения понятия «функция». Наиболее простое выглядит следующим образом: зависимость, при которой каждому значению независимой переменной ставится в соответствие единственное значение зависимой переменной, называется функцией. Таким образом, функция определяет характер зависимости одной переменной от другой (в общем случае можно рассмотреть функцию нескольких переменных).

Для того, чтобы описать характер зависимости между переменными, необходимо каким-то образом определить (задать) функцию. Существует четыре основных способа задания функции [22]:

- Аналитический – способ задания функции с помощью аналитического выражения. Например, следующее аналитическое выражение представляет собой описание функции: $y = \sin(x)$. В данном случае, каждому значению x ставится значение y . Для определения значения y необходимо использовать указанную аналитическую зависимость.

- Табличный – данный вариант задания подразумевает построение таблицы, каждая строка которой содержит значение независимой переменной и соответствующее ему значение зависимой переменной. Табличный способ задания позволяет задать описание любой зависимости. Однако, его

использование не позволяет точно определить значение зависимой переменной для тех значений независимой переменной, которые отсутствуют в таблице.

– Графический – данный вариант задания представляет собой график, который как раз и отражает зависимость между переменными. Графический способ задания является наиболее наглядным, так как позволяет оценить поведение функции на различных участках изменения независимой переменной. Однако, применение графического метода подразумевает наличие информации для его построения. В качестве данной информации может выступать аналитический или табличный способ задания.

– Словесный – данный вариант задания позволяет определить характер зависимости между переменными буквально одной фразой. Примерами подобного задания могут служить следующие высказывания: «модуль числа», «дробная часть числа» т.д. Применение словесного метода достаточно сильно ограничено. С его помощью можно описать только сравнительно несложные зависимости.

Таким образом, на практике может быть использован целый ряд различных вариантов задания функции. Однако, анализ рассмотренных вариантов показывает, что наиболее удобным и универсальным вариантом задания функции является аналитический. С его помощью можно получить все прочие варианты описания: построить таблицу значений, изобразить график, словесно определить характер зависимости.

Однако, и аналитический метод задания имеет свои ограничения. Так, в некоторых случаях выразить характер зависимости между переменными бывает попросту невозможным. В качестве примера можно привести задачу анализа результатов эксперимента. В качестве исходной информации в данном случае выступает таблица измерений определенных показателей. На основании данной таблицы требуется определить характер зависимости между показателями. Для этих целей используются методы регрессионного анализа, с использованием которых можно максимально близко описать зависимость с использованием формулы определенного вида.

Целью выполнения настоящей работы является преобразование аналитического задания функции в графический: требуется на основании имеющейся формулы построить график соответствующей функции. Рассмотрим принципы, на основании которых может быть выполнено подобное преобразование с использованием компьютерной техники.

Для построения графика функции с использованием вычислительной техники, необходимо получить еще один способ представления функции: табличный. Таблица значения функции строится для указанного интервала изменения независимой переменной. Затем, на основании каждой строки полученной таблицы выводится точка, соответствующая паре значений. Полученные точки соединяются между собой по выбранному алгоритму (подробнее о возможных подходах к соединению точек графика, см. ниже).

Перед построением таблицы значений следует определить шаг, в соответствии с которым будет изменяться значение аргумента. К выбору шага следует подходить с особой тщательностью. В случае, если шаг будет слишком большим, полученный график будет достаточно сильно отличаться от реального. В особенности это будет заметно на тех участках, на которых функция изменяется быстро.

В случае, если шаг изменения аргумента выбран очень маленьким, таблица значений будет содержать большое количество строк, многие из которых будут бесполезными (их наличие несколько не улучшит внешний вид графика). При этом, будут затрачены временные и вычислительные ресурсы для вычисления значений зависимой переменной, а также использован большой объем машинной памяти для хранения полученной таблицы.

Второй момент, который следует учесть, это порядок соединения соседних точек в графика. Существует два подхода к машинному построению графиков [14, с. 438]: точечный вывод (при котором точки выводятся настолько близко друг к другу, что дополнительное их соединение между собой не требуется) и соединение соседних точек между собой по определенному алгоритму. В первом случае требуется выполнить большое количество расчетов

(фактически – для каждой точки области вывода), во втором количество вычислений можно существенно уменьшить, однако при этом требуется выбрать способ соединения соседних точек.

Рассмотрим различные варианты соединения соседних точек. Самый простой вариант – использование прямой линии. Причем он позволяет достичь достаточно хорошие результаты, если расстояние между точками сравнительно невелико. При увеличении расстояния между соседними точками, график становится все менее плавным.

Альтернативой прямым линиям могут служить различные кривые. Сама задача расчета значения табличной функции для промежуточных значений аргумента, которые отсутствуют в таблице, носит название «интерполяции». Наибольшее распространение при решении задач интерполяции получили так называемые «сплайны». Сплайн представляет собой функцию, область определения которой разбита на конечное число отрезков, причем на каждом из данных отрезков она представляет собой полином некоторой степени. Коэффициенты данного полинома рассчитываются исходя из известных точек таблицы. В результате построенная кривая проходит через все точки, указанные в таблице.

На рисунке 1.1 приведен пример использования кубического сплайна для интерполяции функции, заданной таблицей значений. На данном рисунке красным цветом изображена кривая, полученная путем соединения соседних точек таблицы прямыми линиями.

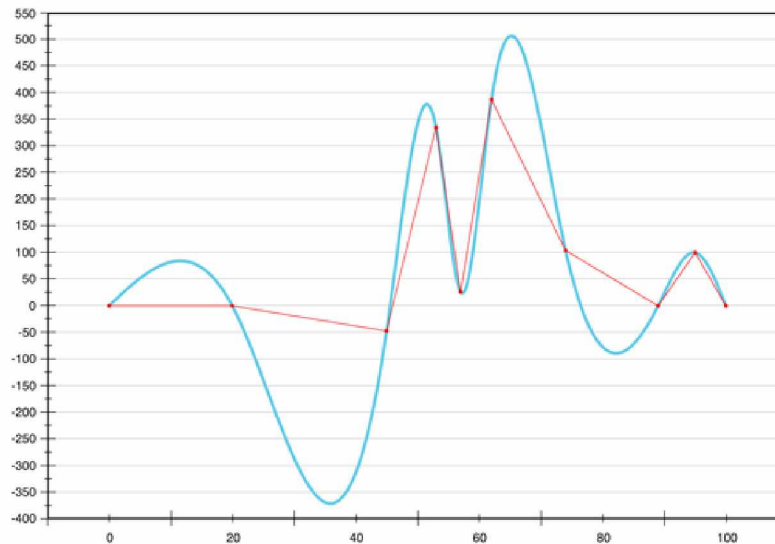


Рисунок 1.1 – Интерполяция с использованием кубического сплайна

Помимо сплайнов различного порядка на практике могут быть использованы и другие методы интерполяции значений табличной функции. Наибольшую известность среди них получили интерполяционный многочлен Лагранжа и кривые Безье [23].

Таким образом, задача соединения соседних точек графика между собой может быть решена с использованием целого ряда методов.

Еще один аспект, который должен быть учтен при машинном построении графиков – это необходимость масштабирования значений аргумента и функции. В распоряжении разработчика при построении графика имеется не бесконечная плоскость, а лишь ограниченная область экрана – окно вывода. Соответственно, для того, чтобы вывести в данную область имеющийся график, необходимо пересчитать реальные координаты точек, в экранные. Для этого используются следующие формулы [14, с. 439]:

$$x_g = X_{g \max} + \frac{X_{g \max} - X_{g \min}}{b - a} (x - a) \quad (1.1)$$

$$y_g = \frac{Y_{g \max} + Y_{g \min}}{2} - Y \frac{Y_{g \max} - Y_{g \min}}{2 * m}$$

где: $X_{g \min}$ – минимальная координата по X в области вывода;

X_{gmax} – максимальная координата по X в области вывода;

Y_{gmin} – минимальная координата по Y в области вывода;

Y_{gmax} – максимальная координата по Y в области вывода;

a – левая граница диапазона изменения аргумента функции;

b – правая граница диапазона изменения аргумента функции;

m – максимальное значение функции в диапазоне изменения аргумента;

x, y – координаты точки, для которой выполняется пересчет координат;

x_g, y_g – полученные в процессе расчета экранные координаты точки.

Ну и в заключение, следует упомянуть еще об одной задаче, которая должна быть решена при построении графика аналитически заданной функции с использованием компьютера. Речь идет о способе задания самой аналитической зависимости. Конечно, данная зависимость может быть реализована в программе. Для ее изменения потребуются откорректировать соответствующий участок кода. В случае решения ограниченной задачи: вывода графика заданной функции, данный подход вполне оправдан. Но при реализации программы, которая должна обеспечивать возможность построения графика для произвольной функции, заданной аналитически, его использование недопустимо. В данном случае необходимо выполнить разбор строкового представления аналитической зависимости и транслировать его во внутреннее описание, которое позволяет осуществлять вычисление значения функции в заданных точках.

Таким образом, в рамках настоящего раздела были рассмотрены основные моменты, которые должны быть учтены при реализации программы для вывода графика аналитически заданной функции:

- определение шага построения таблицы значений;
- выбор способа соединения соседних точек графика;
- обеспечение пересчета координат из таблицы значений в экранные координаты;

– реализация механизма трансляции строкового представления функции во внутреннее представление.

Следующий раздел будет посвящен рассмотрению инструментальных средств, которые могут быть использованы для вывода графика функции.

1.2 Общие сведения о web-приложениях и инструментах по их реализации

Web-приложение представляет собой клиент-серверное приложение, предназначенное для работы в сети Интернет. В качестве клиента в нем выступает браузер, в качестве сервера – web-сервер, на котором расположен сайт приложения. Порядок взаимодействия клиента и сервера, как правило, выглядит следующим образом:

- браузер отправляет на сервер запрос на получение определенных данных, параметры получения данных указываются пользователем, читаются из настроек или определяются любым иным способом;
- web-сервер, получив запрос от клиента, формирует запрошенные данные, генерирует web-страницу и отправляет ее клиенту;
- клиент визуализирует полученную таблицу.

Таким образом, в процессе взаимодействия клиента и сервера используются динамические страницы. Их содержимое не определяется жестко при их создании. Содержимое динамических страниц генерируется автоматически на основании некоторого набора параметров. И если статические web-страницы создаются с использованием языка разметки HTML, то для создания динамических страниц используются совсем другие инструменты.

В настоящее время на практике используется целый ряд технологий и языков программирования, обеспечивающих возможность работы с

динамической генерацией содержимого страниц сайта. Наиболее известными инструментами по созданию динамических страниц являются: PHP, Perl, ASP.NET и Java. Каждый из указанных вариантов имеет свои плюсы и минусы и задач, которая должна быть решена в рамках данного раздела работы, как раз и является изучение различных вариантов технологии, которая наиболее полно удовлетворяет имеющимся потребностям.

Создателем языка Perl является американский программист Ларри Уолл. Работа по разработке данного языка была начата в 1987 году. Первая версия языка вышла 18.12.1987 году. Последняя версия языка (5.24.1) увидела свет 14.01.2017. Perl представляет собой высокоуровневый интерпретируемый динамический язык программирования общего назначения. Первоначально основной задачей, которую предполагалась решать с использованием нового языка программирования, была обработка текстовой информации, однако в настоящее время спектр его применения существенно расширился. Системное администрирование, сетевое программирование, разработка игр, разработка графических пользовательских интерфейсов – в этих и во многих другие сферах находит свое применение язык Perl. Так, еще одним направлением его использования стала web-приложений. Дополнительное преимущество Perl дает широкая коллекция дополнительных модулей, позволяющих его использовать для создания веб-страниц и работы с базами данных различных форматов.

PHP (Hypertext Preprocessor или Personal Home Page Tools) – скриптовый язык программирования общего назначения, основным назначением которого в настоящее время является разработка web-приложений. Создателем языка PHP стал датский программист Расмус Лердорф. Работы по реализации нового языка были начаты в 1995 году.

Первые шаги в данном направлении представляли собой набор скриптов на языке Perl, предназначенные для обработки шаблонов HTML-документов. Созданные Лердорфом инструменты как раз и получили название Personal Home Page (PHP). В определенный период времени Лердорфу стало не хватать имеющейся в Perl функциональности. Результатом стало появление нового

интерпретатора шаблонов, обрабатывающих документы, созданные PHP шаблоны.

Первая версия реализованного интерпретатора во многом повторяла функциональность Perl, но обладала несколько более простым и ограниченным синтаксисом. Следующая версия интерпретатора PHP вышла в 1997 году и пользовалась достаточно широкой популярностью (ее использовали в своей работе около 50 тысяч доменов по всему миру [18]). В настоящее время последняя версия (7.1.2) вышла 17.02.2017. Активность использования PHP со времени его создания существенно выросла, и в настоящее время (по данным на начало 2013 года) превысило 244 миллионов, что составляет около 39% всех сайтов [19].

Язык программирования Java представляет собой высокоуровневый язык программирования общего назначения. Создателем языка стала компания Sun Microsystems, в настоящее время поддержкой языка занимается корпорация Oracle. Работы по реализации нового языка были начаты в 1990 году. Основным синтаксиса нового языка стал широко известный язык программирования C.

Отличительной особенностью Java стала трансляция исходного текста в так называемый «байт-код». Для исполнения программ в виде байт-кода использовался специальный интерпретатор (Java-машина), которая могла быть разработана и функционировать на различных платформах. Подобная реализация позволяет считать Java платформонезависимым (кроссплатформенным) языком, программы на котором могут быть использованы в рамках любой операционной системы, для которой реализована Java-машина. Последняя версия языка (Java Standard Edition 8 Update 121) вышла 18.03.2014. Java представляет собой язык программирования общего назначения, который может быть использован для решения самого широкого круга задач. В частности, в настоящее время Java активно используется для реализации веб-сервисов и расширения функциональности сайтов.

ASP.NET, в отличие от описанных выше инструментов, представляет собой не язык программирования, а технологию, предназначенную для

создания веб-приложений и веб-сервисов. Создателем данной технологии стала компания Microsoft. ASP.NET стала развитием технологии ASP (Active Server Pages), автором которой также была компания Microsoft.

Необходимость в модернизации технологии ASP возникла в связи с наличием большого количества претензий, предъявляемых пользователями. Работы по реализации новой технологии стартовали в 1997 году. Во главе нового проекта встали сотрудники Microsoft Марк Андерс и Скотт Гатри. Уже в конце 1997 года была подготовлена первая версия ASP.NET. Одной из особенностей ASP.NET является возможность использования реализации функциональности сайта с любым языком программирования, входящего в состав .NET Framework (C#, Visual Basic.NET, Jscript.NET). В настоящее время технология ASP.NET получила широкое распространение имеющейся информации количество сайтов, построенных на ее основе составило около 20% по состоянию на конец 2014 года. В абсолютных цифрах это составляет более 120 миллионов сайтов [20].

Таким образом, в настоящее время существует целый ряд инструментов, с использованием которых можно создать web-приложения самой различной направленности. При этом, выбор того или иного инструментального средства следует осуществлять исходя из особенностей данного конкретного проекта. В противном случае (при выборе не подходящего средства реализации) полученное в итоге решение будет реализовывать требуемые функции не в полном объеме или не реализовывать их вовсе. Целью написания настоящего раздела дипломной работы было именно определение перечня инструментальных средств, которые могут быть использованы для реализации web-приложения. Выбор единственного инструмента для решения имеющейся задачи, будет выполнено в следующей главе работы.

1.3 Выработка требований к web-приложению

В заключительной части работы сформулируем требования, которые необходимо учесть при реализации web-приложения.

Параметры вывода графика должны задаваться посетителем. В качестве параметров вывода указываются:

- аналитическая функция – задается в виде строки, которая разбирается в процессе работы программы, при этом при вводе функции должны использоваться стандартные арифметические операции (+, −, *, /), операция возведения в степень, тригонометрические функции (sin, cos, tg, ctg), а также модуль числа;

- диапазон изменения аргумента – начальное и конечное значение аргумента функции, которое будет выведено на графике.

Без ввода указанных параметров построение графика будет попросту невозможным.

Затем необходимо обеспечить возможность определения служебных параметров, определяющих порядок формирования графика. К параметрам данной группы относятся:

- шаг построения таблицы – данный параметр определяет количество отрезков, на которые будет разбит диапазон, в котором изменяется аргумент;

- порядок соединения точек графика – в соответствии со значением данного параметра осуществляется соединение соседних точек графика, возможными значениями данного параметра являются: не соединять (соседние точки не соединяются), прямая (точки соединяются прямой линией), кубический сплайн (точки соединяются с использованием кубического сплайна) и кривая Безье (точки соединяются кривой Безье).

Параметры данной группы позволяют подобрать такой порядок построения графика, который наиболее подходит в данном, конкретном случае.

Их наличие позволяет получить действительно универсальное решение, обеспечивающее вывод действительно произвольного графика.

Еще одна группа параметров должна определять визуальное оформление области вывода:

- цвет графика – цвет линии (точек), который используется при выводе графика;
- вывод точек – параметр определяет необходимость выделения точек, на основании которых строится график, на фоне линии графика;
- цена деления осей – цена одного деления по оси OX и OY , в соответствии с данным значением выводятся линии сетки, помимо задания значения, необходимо также предусмотреть возможность скрытия горизонтальных и / или вертикальных линий сетки;

Введенные пользователем параметры могут быть сохранены в базе данных web-приложения под определенным именем, и выбраны при следующем входе на сайт из выпадающего списка.

Реализация web-приложения в полном соответствии с сформулированными требованиями, позволяет получить действительно удобный инструмент, который позволит построить график произвольной (с учетом поддерживаемых в процессе ввода операций и функций) аналитически заданной функции.

2 ПРАКТИЧЕСКАЯ РЕАЛИЗАЦИЯ WEB-ПРИЛОЖЕНИЯ

2.1 Выбор инструментальных средств для создания web-приложения

Функциональность наиболее распространенных технологий, используемых для создания сайтов, основанных на использовании динамических web-страниц, была кратко описано в первой главе дипломной работы. После определения требований к реализуемой системе, необходимо выбрать среди них то, которое наиболее подходит для создания web-приложения. В настоящее время наиболее используемым инструментом в области управления динамическими страницами, является язык программирования PHP. Рассмотрим его преимущества и недостатки в сравнении с другими, рассмотренными нами инструментами и технологиями.

Язык программирования Perl являлся «предком» PHP. Необходимость устранения недостатков функциональности Perl как раз и стало причиной создания языка PHP. Реализованные в рамках PHP возможности примерно соответствуют имеющимся в Perl. Однако, его синтаксис обладает меньшей степенью избыточности и выглядит более просто. Связано это, в первую очередь с тем, что при создании язык Perl позиционировался как язык программирования широкого назначения, тогда как основной целью разработки PHP стала именно реализация веб-приложений. Таким образом, сравнение указанных языков показывает их схожую функциональность, тогда как простота реализации программы будут существенно выше при использовании PHP.

Сравнивая Java с PHP следует отметить, что одно из основных достоинств Java (его платформонезависимость) является и его недостатком. Исполнение программ, реализованных в виде байт-кода, часто требует избыточных ресурсов и может занимать продолжительное время. Вместе с тем интерпретаторы PHP

также реализованы для различных операционных систем, что позволяет обеспечить его использование и в Windows и в Unix. При этом производительность, которые показывают скрипты на базе PHP, существенно превосходит аналогичные решения на базе Java. Функциональность, предоставляемая Java и PHP в целом схожи и, значит более предпочтительным будет использованием именно PHP.

Использование ASP.NET позволяет создавать приложения, реализованные с использованием целого ряда языков программирования, поддерживающих технологию .NET. Однако, данный плюс, как это часто бывает, является и минусом рассматриваемой технологии. Во-первых, все языка, которые могут быть использованы в рамках технологии ASP.NET подразумевают использование в качестве операционной среды платформы разработки только операционной системы Windows. Во-вторых, сами web-ресурсы, реализованные с использованием ASP.NET, часто могут быть отображены корректно только в рамках браузера Internet Explorer, реализованного, как известно, также в рамках операционной системы Windows. Таким образом, выбор в качестве технологии реализации web-приложения для вывода графика аналитически заданной функции, ASP.NET ограничит не только выбор операционной системы для разработки ресурса (что, в принципе, не является критичным), но во многом ограничит и web-браузеры, используемые пользователями web-приложения.

Таким образом, наиболее оправданным решением, которое будет использовано для реализации web-приложения для вывода графика аналитически заданной функции, будет выбор языка программирования PHP.

Следующий момент, который необходимо выбрать при реализации web-приложения – определение системы управления данными. Применительно к решаемой задаче, необходимость наличия системы управления данными, на первый взгляд, не очень важна. Однако, реализуя для пользователя возможность сохранения параметров построения графика для их последующего

использования в следующей сессии, необходимость в наличии системы управления базами данных уже выглядит не так нелепо.

На рынке имеется огромное количество решений, предназначенных для управления базами данных различного объема и степени нагрузки. Одной из достаточно известных систем данного класса является сервер баз данных MySQL. Данный сервер баз данных фактически является стандартом для реализации баз данных малого и среднего объема. Последние версии MySQL (на данный момент – 5.7.17 от 12.12.2016) представляют собой полноценный SQL-сервер, поддерживающий все присущие данному классу инструментов функциональные возможности:

- администрирование прав пользователей;
- использование структурированного языка запросов (SQL);
- применение хранимых процедур и пользовательских функций;
- создание триггеров;
- создание представлений.

При этом, указанная функциональность доступна на бесплатной основе, по открытой лицензии GNU General Public License. Еще одним доводом в пользу использования MySQL является его тесная интеграция с языком программирования PHP. Реализованные в рамках стандартной поставки PHP дополнения позволяют использовать при разработке веб-сайта все возможности сервера баз данных MySQL.

Таким образом, для целей реализации web-приложения по выводу графика аналитически заданной функции, были выбраны следующие инструменты: разработка web-приложения – PHP, создание хранилища данных – MySQL.

При этом, указанная функциональность доступна на бесплатной основе по открытой лицензии GNU General Public License. Еще одним доводом в пользу использования MySQL является его тесная интеграция с языком программирования PHP. Реализованные в рамках стандартной поставки PHP

дополнения позволяют использовать при разработке web-приложения все возможности сервера баз данных MySQL.

2.2 Определение архитектуры web-приложения

Несмотря на то, что в рамках предыдущего раздела работы были выбраны инструментальные средства для реализации web-приложения, ограничиться только ими в нашем случае не получится. Для обеспечения возможности нормального функционирования web-приложения, помимо браузера установленного на рабочей станции клиента и реализованного приложения на стороне сервера, необходима установка и настройка дополнительного программного обеспечения, которое также будет функционировать на том сервере, на котором развернут сайт.

На практике используется две схемы развертывания интернет-ресурсов: использование собственных аппаратных средств и использование услуг, предоставляемых так называемыми «хостерами». Хостер представляет собой компанию, предоставляющую своим клиентам услуги по использованию вычислительных мощностей для размещения информации на сервере, подключенном к сети Интернет на постоянной основе. Сама данная услуга и носит название «хостинг». Применительно к созданию web-приложения услуга хостинга заключается в предоставлении аппаратной платформы и программного обеспечения, необходимых для развертывания приложения и подключения доступа к сети Интернет.

На практике выделяют несколько видов хостинга [15, с. 113]:

– Виртуальный хостинг – данный вид хостинга подразумевает размещение на одном web-сервере нескольких web-сайтов. Данный вид хостинга является наиболее дешевым. В сети Интернет существуют хостеры,

предлагающие своим клиентам услуги виртуального хостинга на бесплатной основе.

– Виртуальный выделенный хостинг – данный вид хостинга во многом схож с предыдущим. В случае виртуального хостинга несколько сайтов функционируют на различных разделах одного веб-сервера. В случае использования выделенного виртуального хостинга сайты работают также на одной программно-аппаратной платформе, однако каждый из них работает в рамках собственного, выделенного web-сервера.

– Выделенный сервер – данный вид хостинга подразумевает выделение клиенту отдельной аппаратной платформы, на которой выполняется установка и настройка программного обеспечения под нужды конкретного клиента.

– Колокация – предоставление клиентам услуги размещения его оборудования на площадях хостера и его подключения к электрическим и телекоммуникационным сетям.

На период проектирования и разработки web-приложений, использование услуг сторонних хостеров не всегда оправдано. Данный вариант может быть использован только в том случае, если разработчик не имеет в своем распоряжении вычислительных ресурсов, необходимых для функционирования web-приложения. Однако, случаи, при которых на этапе проектирования web-приложения могут потребоваться значительные вычислительные мощности, достаточно редки. В качестве примера можно привести высоконагруженные системы, обслуживающие огромное количество одновременных запросов и обрабатывающих базы данных большого объема. В других случаях, когда функционирование проектируемого сайта не предполагает существенных требований к аппаратным ресурсам, в качестве сервера, на котором размещается разрабатываемый сайт, может быть использована рабочая станция самого разработчика.

На рынке программного обеспечения для этих целей предлагается специализированный комплект программ, содержащий все необходимые

инструменты. Речь идет о наборе дистрибутивов, известном под названием Денвер (Джентльменский Набор Web Разработчика). Автором данного набора является российский программист Дмитрий Котеров. В настоящее время в состав проекта Денвер входят следующие функциональные возможности [27]:

- Web-сервер Apache с поддержкой SSI, SSL, mod_rewrite, mod_php.
- Интерпретатор PHP с поддержкой GD, MySQL, SQLite.
- СУБД MySQL с поддержкой транзакций (mysqld-max).
- Система управления виртуальными хостами, основанная на шаблонах.
- Система управления запуском и завершением.
- Панель phpMyAdmin для администрирования СУБД.
- Ядро интерпретатора Perl без стандартных библиотек (поставляются отдельно).
- Эмулятор sendmail и сервера SMTP с поддержкой работы совместно с PHP, Perl, Parser.
- Установщик.

Итак, перечень программных средств, необходимых для разработки и функционирования web-приложения определен. Однако, в рамках настоящей работы не было приведено описание назначения одного из компонентов среды функционирования web-приложения, а именно – web-сервера. Данный компонент необходим для функционирования любого сайта, вне зависимости от принципов его построения. Следует отметить, что термином «web-сервер» называют как специализированное программное обеспечение, так и аппаратную платформу, на которой оно установлено.

Web-сервер представляет собой сервер, принимающий от посетителей сайта HTTP-запросы и отдающий им HTTP-ответы [15, с. 125]. На практике клиентами web-сервера являются интернет-браузеры, передающие запросы на получение ресурсов, описываемых их адресом (URL). В ответ на поступивший запрос web-сервер отдает клиенту HTML-страницу, изображение, медиа-поток,

сформированную на сервере динамическую страницу, а также любую иную информацию состав и структура которой определяется запросом клиента.

На рынке программного обеспечения представлен целый ряд web-серверов, наиболее известными из которых являются: Apache HTTP-сервер, IIS (Internet Information Server), nginx, Tomado. В настоящей работе для целей реализации web-приложения будет использоваться web-сервер Apache HTTP-сервер.

Apache HTTP-сервер представляет собой свободно распространяемый кроссплатформенный web-сервер (поддерживает операционные системы Linux, Free BSD, Mac OS, Microsoft Windows и т. д.). Первая версия web-сервера Apache вышла в 1995 году. Последняя версия web-сервера (2.4.25) была выпущена 20 декабря 2016 года. Разработкой сервера занимается сообщество разработчиков Apache Software Foundation. В настоящее время Apache получил широкое распространение и по данным на начало 2016 года использовался для организации работы более чем 300 миллионов сайтов, что составляет 33,56% от их общего числа [26].

Отличительной особенностью Apache является возможность применения в процессе его работы модулей, разрабатываемых как сообществом Apache Software Foundation, так и сторонними разработчиками. С использованием модулей в рамках Apache реализуется возможность поддержки языков программирования (PHP, Perl, Python и т. д.). В настоящее время известно более пятидесяти модулей, реализующих в рамках Apache различные расширения стандартной функциональности.

В рамках Apache имеется возможность создания виртуальных хостов (как уже упоминалось ранее – размещение на одном веб-сервере нескольких сайтов), что позволяет использовать Apache для поддержки виртуального хостинга.

Таким образом, в рамках были определены все компоненты среды исполнения web-приложения, принципы их развертывания и взаимодействия.

2.3 Описание порядка разработка web-приложения

В проектируемом web-приложении можно выделить три основных компонента:

- Управление пользовательскими настройками – к данному блоку web-приложения относится форма ввода настроек, база данных, в которой они хранятся и методы взаимодействия с этой самой базой данных.

- Разбор текстового представления аналитической функции – пользователь web-приложения в качестве одного из параметров работы приложения будет вводить информацию о функции, график которой следует построить. Ввод описания функции будет осуществляться в виде текстовой строки, для обеспечения возможность выполнения расчета значений функции необходимо преобразовать введенную строку в вид, пригодный для выполнения вычислений.

- Непосредственный вывод графика – в соответствии с указанными пользователем параметрами будет осуществляться вывод графика функции.

Рассмотрим некоторые детали разработки приложения именно в таком порядке.

Для обеспечения удобства работы пользователя с web-приложением указанным им параметры могут быть сохранены в виде именного набора. При этом для каждого из пользователей подобных наборов настроек может быть несколько. Сохраненный набор доступен для выбора после возобновления сеанса работы с приложением.

Для идентификации пользователя при его работе с приложением, необходимо обеспечить этап авторизации. В процессе авторизации пользователь указывает имя для входа в систему и личный пароль. При успешной авторизации именно на основании имени пользователя и реализуется чтение сохраненных пользователем настроек.

Для сохранения настроек в рамках web-приложения будет использована база данных MySQL. Определим структуру данных, необходимую для хранения настроек. Логическая модель данных приведена на Рисунке 2.1 (скрипт по созданию базы данных приведен в Приложении А).

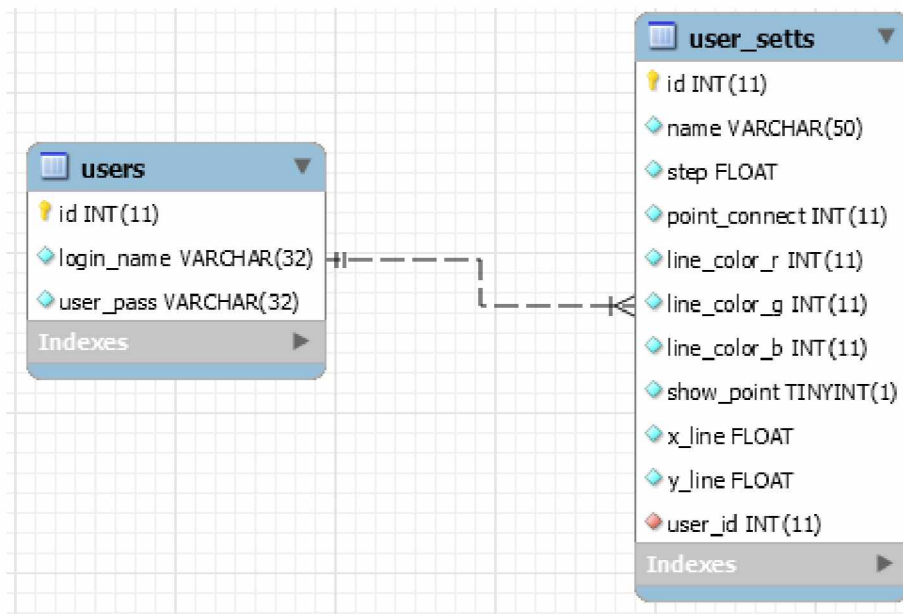


Рисунок 2.1 – Схема базы данных

Приведем более подробное описание структуры данных (см. Таблица 2.1).

Таблица 2.1

Описание структуры данных

Название поля	Тип поля	Описание
users – список пользователей		
Id	INT	Уникальный идентификатор. Значение генерируется автоматически
login_name	VARCHAR(32)	Имя пользователя. Значение является уникальным в рамках таблицы
user_pass	VARCHAR(32)	Пароль пользователя
user_setts – пользовательские настройки		
id	INT	Уникальный идентификатор. Значение генерируется автоматически
user_id	INT	Идентификатор пользователя, которым была создана настройка. Ссылка на поле id в таблице users

name	VARCHAR(50)	Имя набора настроек, присвоенное пользователем
step	FLOAT	Шаг изменения независимой переменной (аргумента) при построении таблицы значений
point_connect	INT	Порядок соединения точек графика. Значение представляет собой перечисление, которое содержит следующие значения: <ul style="list-style-type: none"> – 0 – точки не соединяются; – 1 – точки соединяются прямой линией; – 2 – точки соединяются кубическим сплайном; – 3 – точки соединяются кривой Безье.
line_color_r	INT	Компонент красного (RED) цвета графика в представлении RGB
line_color_g	INT	Компонент зеленого (GREEN) цвета графика в представлении RGB
line_color_b	INT	Компонент синего (BLUE) цвета графика в представлении RGB

Таблица 2.1. Продолжение

Название поля	Тип поля	Описание
show_point	BOOLEAN	Признак необходимости вывода точек на графике
x_line	FLOAT	Шаг вывода сетки по оси OX
y_line	FLOAT	Шаг вывода сетки по оси OY

Авторизация пользователя web-приложения выполняется на главной странице сайта. Для успешной авторизации пользователь должен корректно указать имя для входа в систему и пароль. Проверка корректности указания имени пользователя и пароля осуществляется на основании содержащейся в таблице users базы данных MySQL.

В данной таблице хранится имя пользователя (поле login_name), пароль (поле user_pass). Для успешной авторизации в личном кабинете указанное пользователем имя для входа должно содержаться в таблице users, пароль

должен соответствовать значению в базе данных. Ниже приведен фрагмент программы на языке PHP, осуществляющий проверку корректности указанных пользователем учетных данных и возможность его входа в личный кабинет.

Листинг 2.1. Проверка учетных данных пользователя

```

/* Формируем текст запроса для проверки учетных данных */
$check_user_query = "Select u.id, u.user_pass From users
u Where u.login_name='$login_name'";
/* Выполняем запрос. Если произойдет ошибка - вывести ее.
*/
$check_user_res = mysql_query($check_user_query) or
die(mysql_error());
/* Определяем количество записей */
$check_user_row = mysql_num_rows($check_user_res);
/* Если ни одна запись не выбрана */
if ($check_user_row == 0)
{
    /* Выводим страницу с ошибкой */
    Header('Location:login_baduser.php');
}
Else
{
    /* Проверяем пароль */
    if ($user_pass != $check_user_row['user_pass'])
    {
        /* Выводим страницу с ошибкой */
        Header('Location:login_badpass.php');
    }
    else
    {
        /* Переходим к началу работы */
        Header('Location:graph.php?user_id='.$check_user_
row['id']);
    }
}
}

```

После успешной авторизации выполняется чтение списка сохраненных пользователем настроек и формирование на его основе выпадающего списка. Ниже приведен фрагмент программы, выполняющий чтение данных о пользовательских настройках.

Листинг 2.2. Получение пользовательских настроек

```

/* Формируем текст запроса на получение пользовательских
настроек */
$setts_query = "Select setts.* From user_setts setts
Where setts.user_id = " + $user_id;
/* Выполняем запрос. Если произойдет ошибка - вывести ее.
*/
$setts_res      =      mysql_query($news_query)      or
die(mysql_error());

/* Цикл обхода списка новостей и их вывода */
while ($setts_row = mysql_fetch_array($setts_res)) {
    /* Выводим очередную запись в список */
    <option
value="<?=$setts_row['id']?>"><?=$setts_row['name']?></op
tion> }

```

После построения списка пользователь может выбрать необходимый его элемент и, как следствие, установить все настройки, которые будут использованы при построении графика. Управление списком пользовательских настроек осуществляется на отдельной странице сайта.

Описав порядок работы с пользовательскими настройками, перейдем к рассмотрению следующего элемента web-приложения – разбору текстового представления аналитической зависимости (функции). В процессе разбора текстовое представление аналитической зависимости будет преобразовано в дерево, соответствующее обратной польской нотации. Данный метод описания позволяет описать любое арифметическое выражение с учетом приоритета операций. На Рисунке 2.2 приведен пример подобной записи, соответствующий выражению $(2 + x) * x + \sin(x)$.

Как видно из приведенного рисунка, вершинами дерева являются операции и операнды (числа и переменные). В случае, если в вершине действия располагается арифметическая операция, слева и справа от нее располагаются соответствующие операнды. Функция в такой записи представляется в виде унарной операции, операнд которой располагается справа.

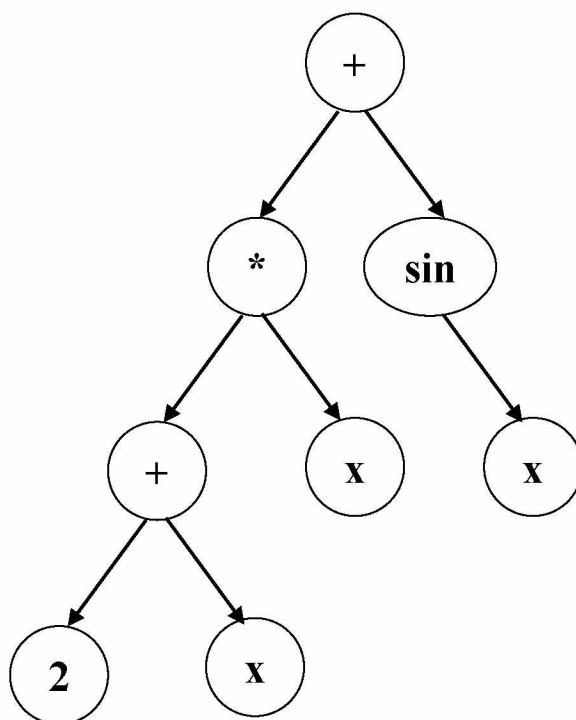


Рисунок 2.2 – Представление выражения в виде дерева

Подобное представление арифметического выражения позволяет достаточно просто вычислить его значение. Для этого удобно использовать рекурсивную функцию, которой в качестве параметра передается узел дерева. Функция может возвращать следующие значения:

- в случае, если в узле дерева расположена константа, возвращается ее значение;
- в случае, если в узле дерева расположена переменная, возвращается ее текущее значение;
- в случае, если в узле дерева располагается арифметическая операция или функция, выполняется расчет значений операндов (с использованием рекурсивного вызова функции для левого и правого поддеревьев), после чего к полученным значениям применяется соответствующая арифметическая операция или функция.

На РНР дерево, представляющее собой арифметическое выражение, описывается с использованием следующей структуры данных.

Листинг 2.3. Описание класса для хранения арифметического выражения

```

class PolishTree{
// Тип узла (переменная, константа, арифметическая
операция, функция)
public $nodeType;
// Значение узла – строковое представление значения
public $nodeValue;
// Указатель на левое поддерево
public $left = NULL;
// Указатель на правое поддерево
public $right = NULL;
}

```

Таким образом, порядок преобразования текстового представления аналитической зависимости следующий:

- вызывается анализатор выражения, который получает на вход строку и возвращает построенное дерево или статус ошибки, если исходное выражение записано некорректно;
- при необходимости вычисления выражения вызывается рекурсивная функция, получающая на вход узел дерева и возвращающая рассчитанное значение.

На практике, при использовании языка программирования РНР предпочтительнее использовать не дерево, а строковое представление выражения, представленного в виде польской инверсной записи. В процессе разбора данного выражения встреченные числа помещаются в стек, при появлении в строке оператора, из стека извлекается необходимое количество операндов, вычисляется значение текущего действия, после чего полученный результат помещается в стек. В реализованной программе для этого написана функция

```
function calc_rpn($str, $x)
```

Данная функция обеспечивает вычисление значения выражения, представленного в виде строки \$str для аргумента, равного \$x.

В заключительной части раздела рассмотрим порядок визуализации графика. Вывод графика осуществляется непосредственно в палитровое изображение, которое создается с использованием функции `php imagecreate`:

```
resource = imagecreate(int $width, int $height);
```

В качестве параметров данной функции передается размер создаваемого изображения. Для построения графика используются следующие функции:

- создание цвета, в качестве параметров передается созданное ранее изображение и компоненты цвета

```
int imagecolorlocate(resource $image, int $red, int $green, int $blue)
```

- вывод закрашенного прямоугольника, в качестве параметров передается изображение, координаты левого верхнего и правого нижнего угла и требуемый цвет;

```
bool imagefilledrectangle ( resource $image , int $x1 , int $y1 , int $x2 , int $y2 , int $color )
```

- вывод прямоугольника, в качестве параметров передается изображение, координаты левого верхнего и правого нижнего угла и требуемый цвет;

```
bool imagerectangle ( resource $image , int $x1 , int $y1 , int $x2 , int $y2 , int $color )
```

- вывод линии, в качестве параметров передается изображение, координаты концов линии и требуемый цвет;

```
bool imageline ( resource $image , int $x1 , int $y1 , int $x2 , int $y2 , int $color )
```

– вывод эллипса, в качестве параметров передается изображение, ширина и высота эллипса и требуемый цвет;

```
bool imageellipse ( resource $image , int $cx , int $cy ,
int $width , int $height , int $color )
```

– вывод текста, в качестве параметров передается изображение, координаты левого верхнего угла, выводимый текст и требуемый цвет;

```
bool imagestring ( resource $image , int $font , int $x ,
int $y , string $string , int $color )
```

– генерация изображения в формате PNG, в качестве параметра передается изображение, прочие параметры не используются.

```
bool imagepng ( resource $image [, mixed $to [, int
$quality [, int $filters ]]] )
```

Алгоритм формирования графика выглядит следующим образом:

- создание изображения с использованием функции `imagecreate`;
- построение таблицы значений функции;
- формирование области построения (вывод осей, линий разметки);
- вывод в область построения точек из построенной таблицы;
- соединение точек с использованием указанного пользователем метода;
- генерация изображения в формате PNG;
- генерация HTML страницы, содержащей построенное изображение и передача ее клиенту.

Таким образом, в рамках второй главы дипломной работы был подробно описан процесс создания web-приложения, а именно:

- выбраны инструментальные средства для реализации;
- описана архитектура приложения;

– описан порядок разработки программы.

Реализованное web-приложение может быть впоследствии использовано для исследования поведения функций. Возможность задания вида функции пользователем (то есть функция не выбирается из списка заранее определенных), существенно расширяет возможности использования приложения.

В рамках заключительной главы будет рассмотрен процесс размещения реализованного web-приложения в сети Интернет и методы его продвижения в поисковых системах.

3 ВВОД WEB-ПРИЛОЖЕНИЯ В ЭКСПЛУАТАЦИЮ

3.1 Порядок размещения web-приложения в сети Интернет

В процессе разработки web-приложения, оно было развернуто непосредственно на рабочей станции, с использованием набора инструментов Денвер. Данный подход достаточно удобен в процессе разработки и тестирования, так как позволяет выполнять все действия по настройке и модернизации сайта непосредственно на своей рабочей станции. При этом, необязательно даже наличие подключения к сети Интернет. Однако, после того как разработка web-приложения закончена, для размещения созданного продукта необходимо использовать инструменты, позволяющие организовать доступ пользователей посредством сети Интернет.

Как было отмечено ранее, размещение сайта в сети Интернет осуществляется с использованием услуг компаний-хостеров. На рынке представлены целый ряд различных предложений, которые отличаются предоставляемыми услугами (поддержка PHP, MySQL, наличие сервисов FTP, SMTP и т.д.) и ресурсами (возможность работы под нагрузкой, предоставляемое дисковое пространство). Естественно, что в зависимости от предоставляемых клиенту возможностей, отличается и стоимость услуг.

На рынке услуг хостинга можно найти и бесплатные предложения. Учитывая характер реализованного проекта (выполнение учебного задания), оптимальным вариантом как раз и будет воспользоваться одним из бесплатных предложений компаний-хостеров. Продемонстрируем размещение сайта в сети Интернет с использованием хостера 000 webhost (сайт <https://ru.000webhost.com/>).

Процесс размещения web-приложения начинается с регистрации на сайте компании-хостера. Нажмем на главной странице сайта кнопку

«Зарегистрироваться бесплатно» и укажем свои учетные данные (см. Рисунок 3.1).

Начните
БЕСПЛАТНУЮ
Регистрацию!

————— 100% БЕСПЛАТНО —————

da-makusheva@mail.ru

.....|

www.fun-graph.ru

□
□
https://www.fun-graph.ru.000webh...

Рисунок 3.1 – Указание параметров регистрации

Далее необходимо подтвердить указанный при регистрации адрес электронной почты. Для этого на указанный адрес отправляется письмо, в котором содержится ссылка для подтверждения. После перехода по ней адрес электронной почты считается подтвержденным, а сам процесс регистрации – завершенным.

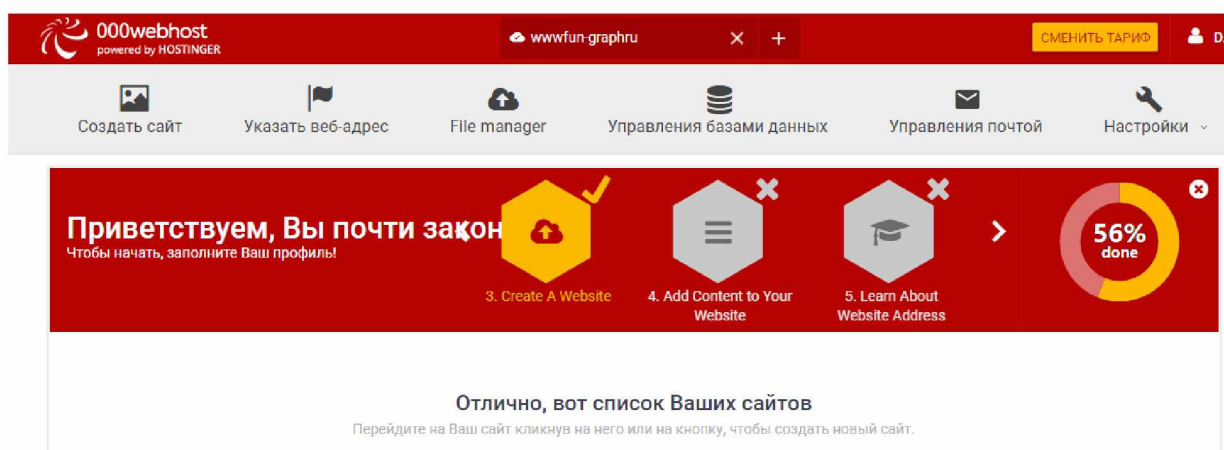


Рисунок 3.2 – Панель управления сайтами

После входа в личный кабинет на сайте компании-хостера, становится доступным панель управления сайтами (см. Рисунок 3.2).

Начнем работу по размещению сайта с создания базы данных. Для этого нажмем кнопку «Управления базами данных». На странице управления данными нажмем кнопку «Новая база данных». В появившемся окне укажем имя новой базы данных, пользователя для работы с ней и пароль пользователя. В результате созданная база данных появится в списке (см. Рисунок 3.3).

Имя Базы	Пользователь	Хост	
id1644654_graph	id1644654_graph_us...	localhost	Управлять ▼

Новая База Данных

Рисунок 3.3 – Окно управления базами данных

Как видно из приведенного рисунка, при создании базы данных выполняется добавление служебных префиксов к именам базы данных и пользователя. Нажмем на кнопку «Управлять» и выберем пункт меню «PhpMyAdmin» (PhpMyAdmin утилита управления сервером MySQL, реализованная с использованием языка PHP).

Log in

Username:

Password:

Рисунок 3.4 – Указание учетных данных для подключения к базе данных

В окне авторизации введем параметры, указанные при добавлении базы данных (см. Рисунок 3.4). В результате будет запущен инструмент администрирования базы данных MySQL. В нашем распоряжении уже имеется скрипт, который создает все необходимые объекты базы данных (см.

Приложение А). Для его выполнения нажмем кнопку «Import», выберем сохраненный ранее файл и укажем его кодировку (см. Рисунок 3.5).



Рисунок 3.5 – Параметры импорта данных

Нажмем кнопку «GO», в результате выбранный скрипт будет выполнен в рамках базы данных и, соответственно, будут созданы все необходимые таблицы (см. Рисунок 3.6).

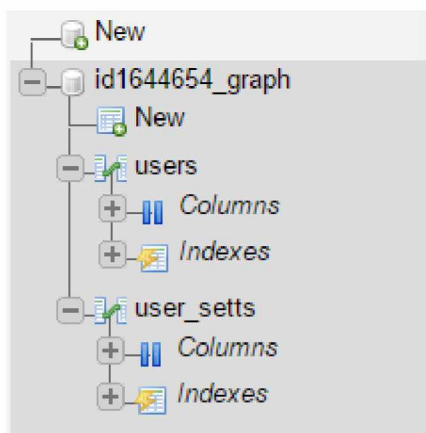


Рисунок 3.6 – Созданные таблицы базы данных

Для продолжения работ по размещению сайта, необходимо выложить все входящие в его состав файлы (скрипты PHP). Для этого нажмем на панели управления сайтами кнопку «File manager» и в появившемся окне нажмем кнопку «Загрузить файлы сейчас». В результате будет открыта панель встроенного файлового менеджера (см. Рисунок 3.7).

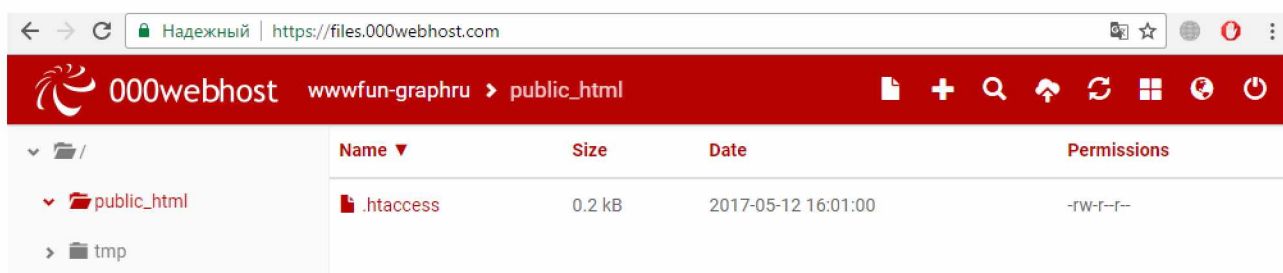


Рисунок 3.7 – Панель управления файлами

В результате, после размещения на сайте всех необходимых скриптов будет возможным зайти через браузер по указанном доменному имени (см. Рисунок 3.8).

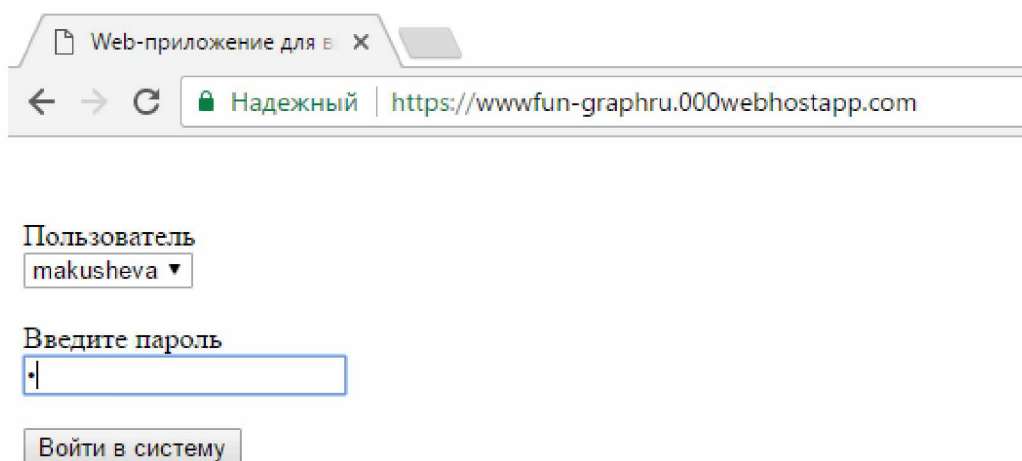


Рисунок 3.8 – Авторизация на размещенном сайте

Продемонстрируем работу размещенного web-приложения. Авторизуемся в системе и укажем параметры формирования графика (см. Рисунок 3.9).



Рисунок 3.9 – Задание параметров формирования графика

Нажав на кнопку «Вывести график» получим график функции $y = \sin x$ на промежутке от -3 до 3 (см. Рисунок 3.10).

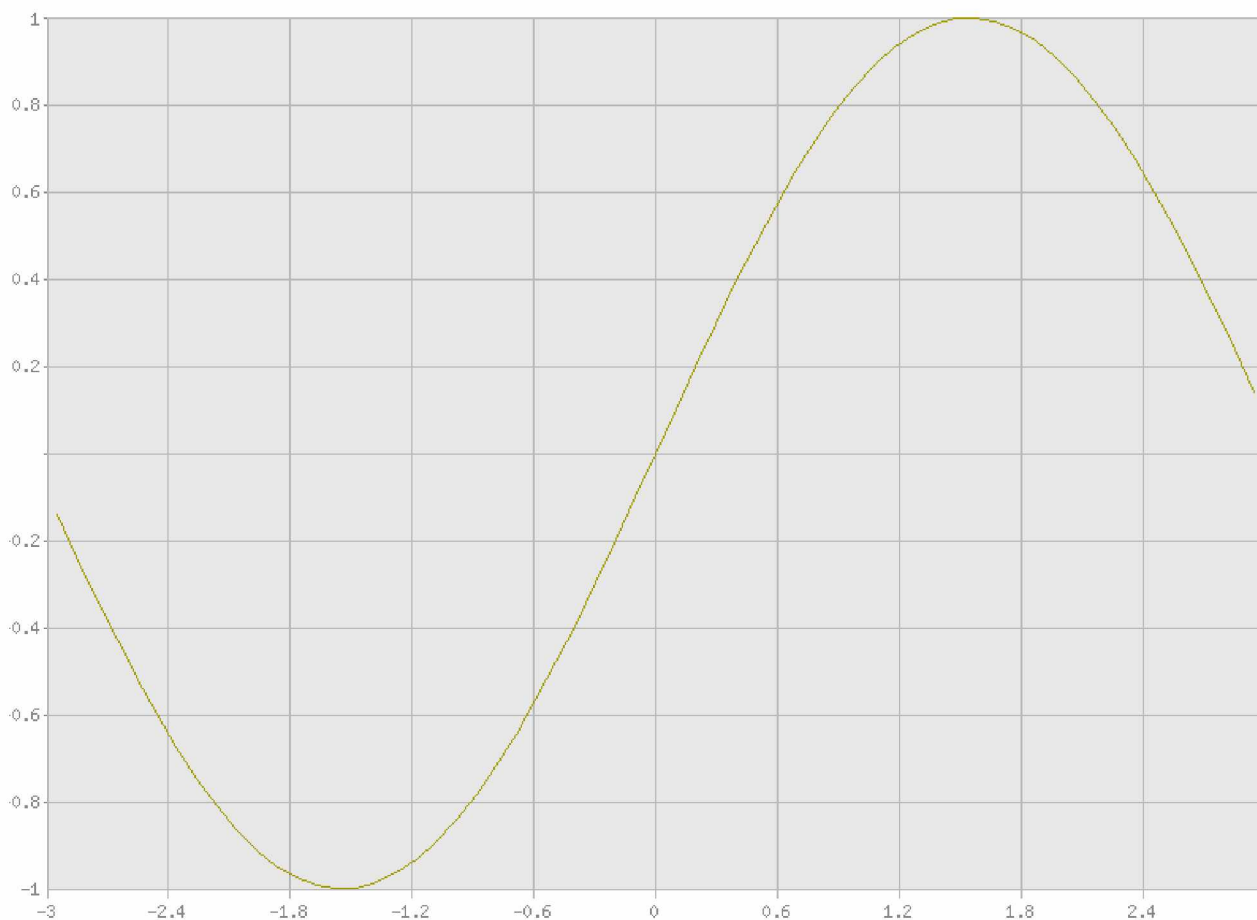


Рисунок 3.10 – Вывод графика функции $y = \sin x$

Соединение точек графика на данном рисунке было выполнено с использованием прямых линий. Как видно из данного рисунка выбранный метод в некоторых случаях позволяет построить гладкий график. Однако, в том случае, если функция меняется слишком быстро, полученные результаты могут быть неприемлемыми.

Web-приложение для в ×

← → ↻ Надежный | <https://wwwfun-graphru.000webhostapp.com/graph.php>

Функция

Диапазон изменения аргумента от до

Рисунок 3.11 – Параметры формирования графика

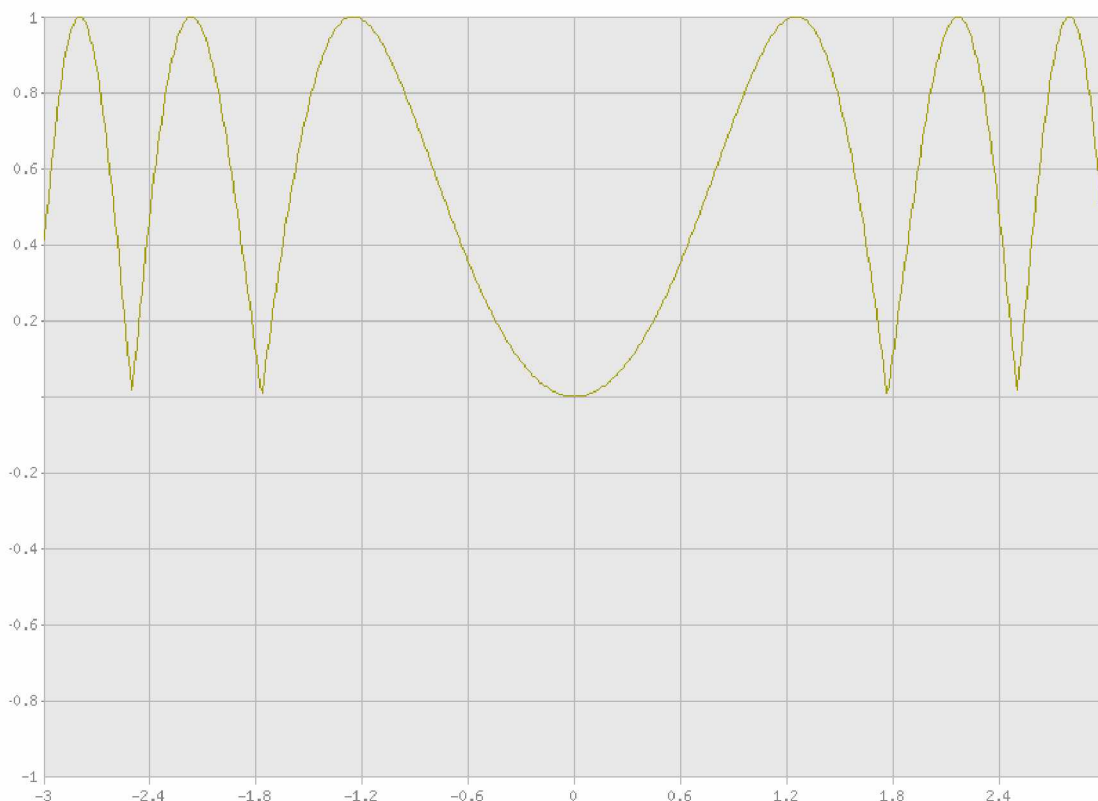


Рисунок 3.12 – График функции $y = |\sin x^2|$

Программа позволяет формировать и более сложные графики. Например, сформируем график функции $y = |\sin x^2|$ на промежутке от -3 до 3 . Для этого укажем соответствующие параметры (см. Рисунок 3.11).

Нажав на кнопку «Вывести график» получим следующий график (см. Рисунок 3.12). Продемонстрированные возможности показывают, что реализованное web-приложение вполне пригодно для организации исследования поведения функций для различных диапазонов изменения аргумента.

Таким образом, в процессе написания настоящего раздела был выполнен весь комплекс мероприятий по размещению разработанного web-приложения на виртуальном хостинге, предоставленном 000webhost. В процессе размещения был использован бесплатный тариф. В результате размещенное на данном хостинге web-приложение доступно не по доменному имени, которое было указано при регистрации. К нему добавлен постфикс 000webhostapp.com. Но для целей выполнения настоящей работы подобный

вариант размещения вполне допустим. Тем более, что выбранный хостер предоставляет поддержку PHP и MySQL.

3.2 Выбор способа продвижения web-приложения в поисковых системах

Процесс привлечения новых посетителей на сайт называется «продвижением» [2, с. 34]. Под продвижением сайта понимается комплекс мер, направленный на увеличение посещаемости сайта пользователями, относящимися к целевой группе.

Одним из факторов, влияющих на посещаемость сайта, является его позиция в результатах поиска по ключевым словам, относящимся к тематике сайта. Поисковые системы упорядочивают результаты поиска по степени близости (релевантности) сайта введенному запросу. Повысить позицию сайта в результатах поиска можно при помощи внутренней организации сайта. В частности для этого может быть предпринят целый ряд шагов.

Выбор названия домена. Поисковые системы учитывают, помимо содержимого сайта, наличие в названии домена ключевых слов. Анализ выполняется как для ключевых слов указанных транслитом, так и для слов набранных кириллицей (для русскоязычных доменов). Помимо всего прочего учитывается и географическая принадлежность домена.

Выбор заголовка страницы. В качестве наименования страниц желательно выбирать ключевые слова, которые могут быть указаны пользователем при поиске. При этом желательно, чтобы форма, в которой они употреблены, соответствовала той, в которой они будут указаны при вводе поискового запроса.

Подбор текста на странице. При наборе содержательной части страницы желательно указать ключевые слова, при этом необходимо стремиться, чтобы соотношение ключевых слов к их общему числу было как можно выше.

Увеличение количества ссылок. Для повышения количества ссылок на страницы (один из параметров, по которому сортируются результаты поиска), необходимо обеспечить наличие как можно большего количества ссылок между страницами сайта.

Использование внутренних атрибутов HTML-разметки. С использованием стандартных тегов HTML можно также повысить позицию сайта в результатах поиска. Для этих целей могут быть использованы следующие теги: <description>, <keywords>, <alt> (для указания альтернативной надписи при выводе изображений).

Помимо повышения позиции сайта при помощи организации его внутренней структуры, данный показатель может быть повышен и с использованием внешних факторов. В частности могут быть предприняты следующие шаги [18, с. 138].

Регистрация сайта в поисковых системах. Поисковые системы используют для индексации страниц специализированные программы – роботы. Программа-робот перемещается по страницам и, используя специализированную информацию, размещенную администратором сайта, помещает страницу во внутренний каталог поисковой системы (индексирует страницу). Для того, чтобы страница попала в каталог поисковой системы как можно раньше, имеется возможность отправить соответствующую заявку до того момента, как робот выполнит регистрацию в автоматическом режиме.

Регистрация сайта в тематических каталогах. Положительное влияние на позицию сайта в результатах поиска оказывает количество ссылок на него в тематических каталогах ссылок.

Участие сайта в тематических материалах (обзорах, новостях и т. д.). Упоминание ресурса в различных тематических обзорах, новостных лентах и иных медиа-событиях также положительно влияет на рейтинг сайта.

Упоминание сайта в форумах, социальных сетях. В последнее время социальные сети, форумы и подобные «клубы по интересам» получили огромное распространение. В последнее время даже используется специальный

термин «социальные медиа». Для продвижения сайта в социальных медиа может быть организована целенаправленная компания, в процессе которой при обсуждении различных тем пишутся посты, в которых упоминается «раскручиваемый» сайт. Данная акция, во-первых, повышает узнаваемость сайта в социальной сети, во-вторых, увеличивает количество ссылок на сайт.

Участие сайта в различных рейтингах. Размещение на страницах сайта счетчиков посещений позволят не только проанализировать структуру посетителей сайта, но и участвовать в рейтингах, которые формируют компании, организующие подсчет посетителей. Рейтинг строится в отдельных тематических категориях. Высокая позиция в рейтинге позволит привлечь на сайт дополнительных посетителей. Наиболее известны следующие компании, предоставляющие счетчики посещений: Liveinternet (<http://www.liveinternet.ru>), Яндекс.Метрика (<https://metrika.yandex.ru/list/>), счетчик Mail.ru (<http://top.mail.ru>), Rambler Top 100 (<http://top100.rambler.ru>).

Участие сайта в баннерных программах. В рамках сети Интернет существует еще один механизм рекламы сайта. Речь идет о баннерной рекламе. Специализированные ресурсы позволяют клиентам зарегистрировать сайт в программах баннерного обмена. Существуют специализированные баннерные сети, объединяющие большое количество интернет-ресурсов, в которых и будет осуществляться показ баннеров. Особым преимуществом баннерных сетей является возможность самостоятельного определения целевой аудитории, которой будет демонстрироваться баннер. Данная процедура носит название «таргетинга». В процессе настройки имеется возможность указания категории сайтов, на которых будет демонстрироваться баннер, время суток, дни недели, когда он будет показываться. Примеры баннерных сетей: TBN.ru (<http://tbn.ru/>), сеть AdRiver (<http://www.adriver.ru/>), RLE Banner network (<http://www.rle.ru/>).

Участие в программах контекстной рекламы. В последнее время появился новый механизм рекламы: так называемая «контекстная» реклама. Суть ее заключается в следующем: в процессе путешествия пользователя по сети Интернет, посещения им сайтов определенной тематики, указания поисковых

запросов, собирается статистика о его предпочтениях. После сбора статистики, пользователю демонстрируются ссылки на тех участников каталога контекстных ссылок, которые подходят под предпочтения пользователя. Примеры систем контекстной рекламы: Яндекс.Директ (<http://direct.yandex.ru>), Google AdWords (<https://adwords.google.com/>), Begun (<http://www.begun.ru/>).

Комплексное использование указанных мероприятий позволяет существенно увеличить количество посетителей сайта.

Итак, основные методы, используемые для повышения посещаемости сайта, определены. Далее необходимо спланировать комплекс мероприятий, который необходимо реализовать для раскрутки вполне конкретного ресурса: web-приложения для построения графика функции, заданной аналитически.

Комплекс мероприятий, направленный на улучшение позиции сайта при выдаче результатов запросов поисковыми системами носит название SEO (Search Engine Optimization – поисковая оптимизация). Все SEO-мероприятия можно разделить на две группы. Первая группа предполагает соблюдение определенных требований при создании сайта, вторая – увеличение количества ссылок на сайт в сети Интернет. Порядок продвижения разработанного web-приложения будет описан с разбивкой мероприятий на указанные группы.

В процессе создания сайта необходимо учесть следующие положения:

- Выбор названия домена
- Выбор заголовков страниц
- Подбор текстового наполнения страниц
- Увеличение количества ссылок
- Использование атрибутов HTML-разметки.

Whois-сервис

Для получения информации введите имя домена или IP-адрес:

Например, test.ru или 194.85.61.42

Информация о домене FUN-GRAPH.RU

Доменное имя свободно! Вы можете его зарегистрировать.

Рисунок 3.13 – Результат проверки доступности доменного имени

В качестве названия домена для сайта, представляющего собой сервис по построению графика аналитически заданной функции, необходимо выбрать фразу, содержащую информацию о его функциональном назначении. Для названия сайта web-приложения можно предложить следующий вариант: fun-graph.ru. Указанное доменное имя свободно (результаты проверки см. Рисунок 3.13) и может быть использовано для размещения web-приложения.

Заголовок страницы задается с использованием тэга title. Формат данного тэга имеет следующий вид: <title>Заголовок</title>.

В процессе создания текстового наполнения страниц сайта необходимо использовать ключевые слова, соответствующие его функциональному предназначению (функция, график и им подобные). Выполнение данного требования не должно приводить к снижению качества смысловой нагрузки текста. Перегрузка страницы ключевыми словами приводит к тому, что текст становится трудночитаемым и пользователь может покинуть страницу до окончательного ее изучения.

При создании структуры сайта необходимо максимально увеличить количество ссылок между его страницами. Ссылки «В начало», «Корзина» и т. д. позволяют увеличить индекс, определяющий количество ссылок для страниц

сайта. Данный параметр, среди прочих, определяет положение сайта в результатах поиска.

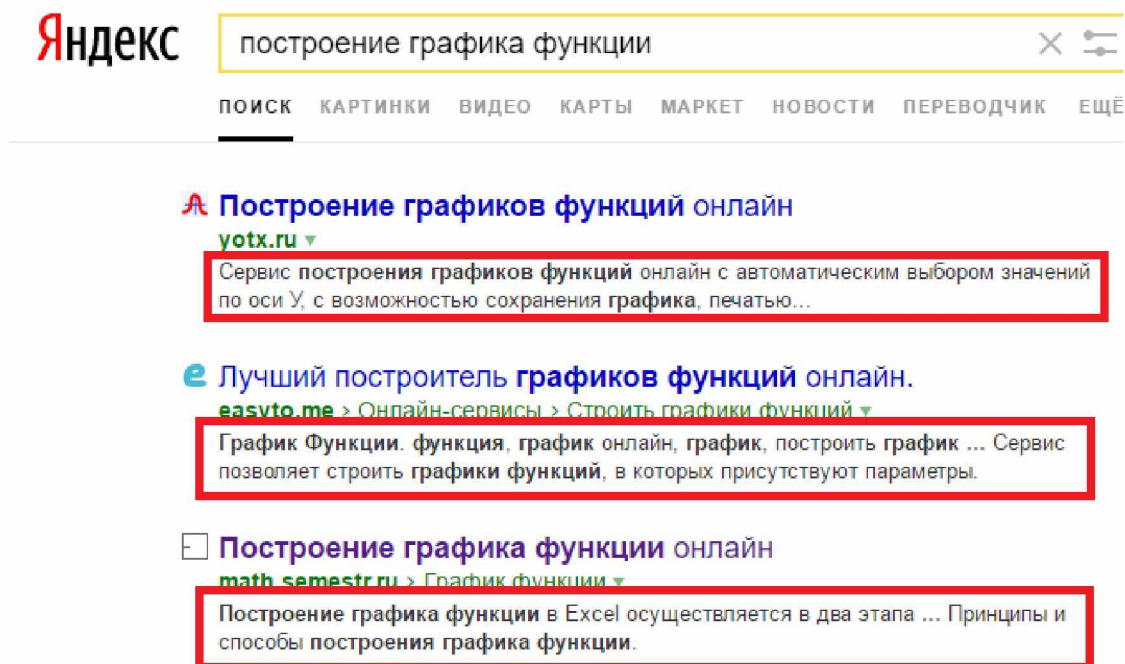


Рисунок 3.14 – Сниметы при выдаче результатов поиска

На повышение позиции сайта в поисковых системах могут повлиять следующие HTML тэги: description, keywords, alt.

Тэг description используется для создания описания страницы. Формат тэга следующий: `< meta name="description" content="Содержимое мета-тега «description»">`. Ключевой особенностью данного тэга является то, что на его основе поисковые системы формируют так называемые «сниметы» – краткое описание страницы, выводимое при выдаче результатов поиска. На Рисунке 3.14 приведен пример поиска в поисковой системе Яндекс по запросу «построение графика функции».

Сниметы отмечены на рисунке красной рамкой. Данная информация не повышает позицию сайта в рейтинге, однако ее содержание позволяет пользователю выбрать именно продвигаемый сайт среди всех предложенных поисковой системой вариантов. Пользователь старается минимизировать количество шагов до выбора интересующего сайта. Именно поэтому он старается оценить подходит ему сайт или нет на этапе просмотра результатов поиска.

Грамотно подобранное содержимое тэга description как раз и позволяет оценить все достоинства сайта уже на этапе вывода результатов поиска. В результате вероятность перехода пользователя именно на данный сайт повышается. В случае, если тэг description не заполнен, поисковая система может формировать сниппет из случайных порций текстовой информации, содержащейся на странице. Таким образом, для повышения вероятности перехода на сайт непосредственно со страницы вывода результатов поиска, необходимо обязательно заполнять тэг description для каждой из страниц сайта, при этом содержимое данного тэга должно подбираться сообразно его последующему применению (содержать лаконичное и информативное описание соответствующей страницы).

Тэг keywords используется для задания списка ключевых слов, относящихся к содержимому страницы. Формат данного тэга следующий: `< meta name="keywords" content="Перечень ключевых слов">`. Ключевые слова указываются через запятую в единственном числе. Содержимое тэга keywords является еще одним параметром, который оценивает поисковая машина при определении позиции сайта в результатах поиска. Однако не следует чрезмерно усердствовать при заполнении данного тэга. Чрезмерное количество ключевых слов, указание одних и тех же слов во всех возможных формах может привести к обратному эффекту – поисковая машина проигнорирует указанное значение. В процессе создания сайта для web-приложения вывода графика функции необходимо использовать данный тэг и заполнять его несколькими ключевыми словами, относящимися к содержимому страницы.

Тэг alt используется для определения текстового сообщения, используемого для альтернативного показа изображений. В том случае, если браузер по какой-либо причине не может вывести изображение, на экране отображается содержимое тэга alt. Также содержимое указанного тэга выводится на экране в процессе загрузки браузером изображения с сервера. Формат тэга alt имеет следующий вид: ``.

Выполнение комплекса мероприятий доработке информационного содержимого и структуры сайта является лишь одним из методов поисковой оптимизации. Более существенных результатов в продвижении сайта в результатах поиска и повышении его узнаваемости можно достичь в размещении ссылок на него на сторонних ресурсах. Рассмотрим некоторые из возможных вариантов.

Регистрация сайта в поисковых системах. Как уже упоминалось выше, поисковые системы имеют возможность автоматического сбора информации о содержимом сайта и его регистрации. Однако, владелец сайта имеет возможность и самостоятельной регистрации сайта. Рассмотрим порядок регистрации в некоторых популярных поисковых системах.

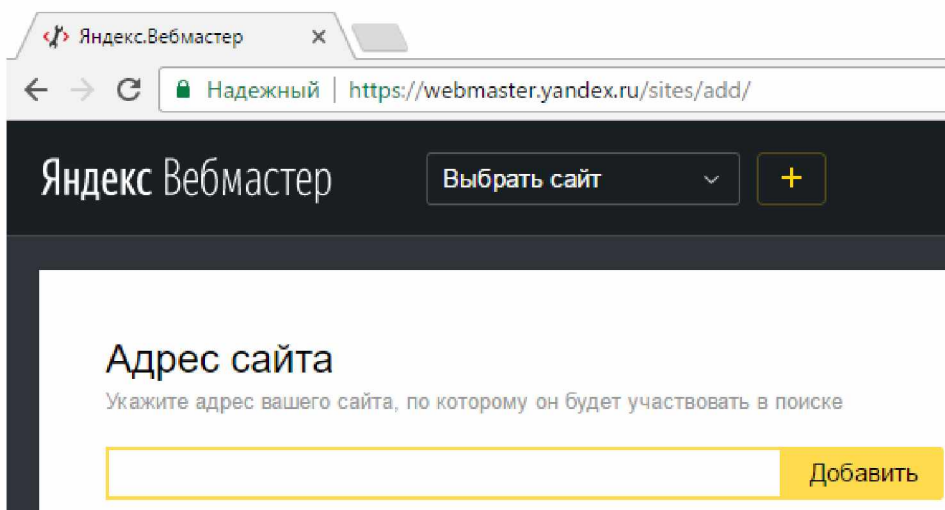


Рисунок 3.15 – Ссылка для регистрации сайта в поисковой системе Яндекс

Для регистрации в поисковой машине Яндекс используется специализированный сервис Яндекс.Вебмастер (<http://webmaster.yandex.ru/>). В рамках данного сервиса владелец ресурса может самостоятельно сообщить поисковой машине по появлении нового сайта, а также выполнить иные действия, связанные с регистрацией сайта и сбором статистики. Для регистрации нового сайта необходимо выбрать соответствующий пункт в меню сервиса (см. Рисунок 3.15).

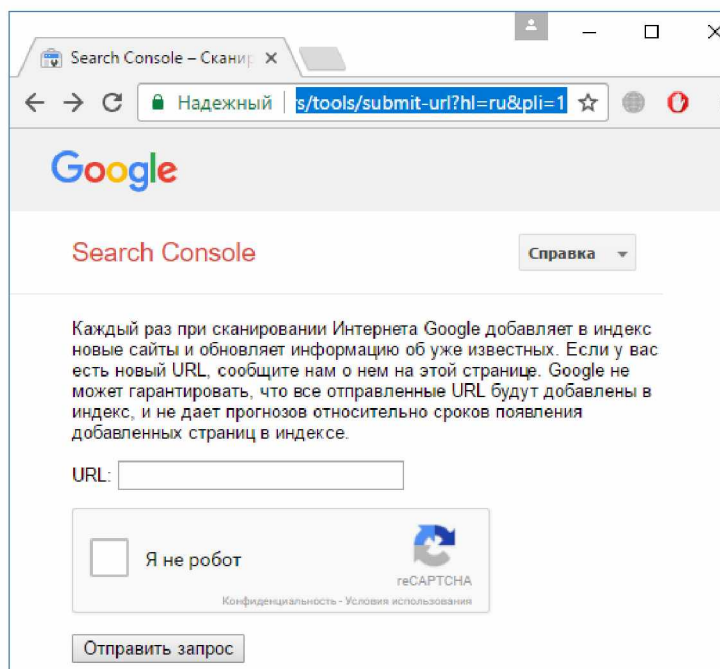


Рисунок 3.16 – Регистрация сайта в поисковой системе Google.

Аналогичные возможности имеются и у другой популярной поисковой системы Google. Доступ к указанному сервису осуществляется по адресу <http://www.google.ru/intl/ru/addurl.html>. Внешний вид страницы приведен на Рисунке 3.16.

Применительно к web-сервису формирования графика функции необходимо зарегистрировать сайт во всех указанных системах.

Еще одна возможность повышения упоминаемости сайта – регистрация его в тематических каталогах. Крупнейшие поисковые системы, работающие в русскоязычном сегменте Интернет, имеют собственные каталоги ссылок: Яндекс.Каталог (http://yasa.yandex.ru/add_free.xml), Google каталог (регистрация производится через сторонний ресурс Dmoz).

Яндекс Каталог

Бесплатная регистрация в Каталоге Яндекса

Адрес сайта:

Название:

Описание:

Категория:

Жанр сайта:

Регион:

Мобильная версия:


Защита от автоматических регистраций: 

Рисунок 3.17 – Форма регистрации сайта в каталоге Yandex

При регистрации сайта в каталоге указываются все его контактные данные, а также определяется категория, к которой относится данный сайт. В рамках каталога Яндекс имеется возможность бесплатной регистрации (форма регистрации см. Рисунок 3.17), а также платная регистрация, предоставляющая клиенту расширенные возможности по управлению продвижением сайта.

Аналогичная возможность предоставляется и в рейтингах Mail.ru (<http://top.mail.ru/add>) и Rambler (<https://top100.rambler.ru/>).

На начальном этапе развития сайта вывод графика функции предполагается его регистрация во всех перечисленных выше каталогах на бесплатной основе.

Повышения уровня цитируемости сайта в различных тематических материалах и социальных сетях является достаточно специфической задачей,

выполнение которой, как правило, поручается компаниям, занимающимся раскруткой ресурсов на профессиональной основе. В настоящей работе данный метод раскрутки сайта рассматриваться не будет.

Регистрация сайта в системах баннерного обмена и контекстной рекламы также требуют привлечение профессиональных специалистов в области рекламы для подготовки баннеров и графического оформления объявлений. Данные средства раскрутки сайта, в силу специфики сферы его использования и отсутствия в ней коммерческой составляющей, использовать не планируется.

Итак, определены следующие методы раскрутки сайта:

- Оптимизация содержимого сайта.
- Регистрация сайта в поисковых системах Яндекс и Google.
- Регистрация сайта в тематических каталогах Яндекс.Каталог, Рейтинг Mail.ru и Rambler Top 100.

Таким образом, в процессе выполнения настоящей работы была не просто выполнена разработка web-приложения, реализующего затребованный функционал, но также выполнен весь комплекс мероприятий по вводу сайта в эксплуатацию.

ЗАКЛЮЧЕНИЕ

Итак, выполнение работы подошло к концу и настало время подвести некоторые итоги. Целью выполнения работы явилась создание web-приложения, предназначенного для построения графика аналитически заданной функции.

Пояснительная записка к дипломной работе содержит три главы. В рамках первой главы были подробно описаны теоретические основы построения графиков аналитически заданных функций. Кроме этого, внимание в ней также было уделено принципам построения web-приложений и используемым при этом инструментальным средствам. В заключение первой главы на основании выполненного теоретического обзора и имеющихся пожеланий к функциональности web-приложения, были сформулированы требования к нему.

Вторая глава работы носила практический характер и содержала подробное описание основных моментов реализации web-приложения: выбор инструментальных средств, определение архитектуры и, собственно, сама практическая реализация приложения.

В заключительной главе были рассмотрены вопросы, связанные с размещением реализованного web-приложения в сети Интернет и его продвижения в поисковых системах.

Таким образом, в процессе выполнения работы были последовательно выполнены все этапы, необходимые для успешной реализации проекта: выполнен обзор теоретических принципов решения задачи (изучена предметная область), сформулированы требования, выбраны инструментальные средства и осуществлена практическая реализация приложения. В заключении выбран порядок ввода реализованного продукта в эксплуатацию.

В конечном итоге было получено полностью функциональное решение, реализующее все сформулированные требования. При этом для реализации сайта было использовано следующее программное обеспечение:

- Язык программирования: PHP 5.3.13.
- Сервер управления базами данных: MySQL 5.1.

Реализованное web-приложение может быть впоследствии использовано для исследования поведения функций. Возможность задания вида функции пользователем (то есть функция не выбирается из списка заранее определенных), существенно расширяет возможности использования приложения.

Для размещения сайта в процессе разработки и тестирования использовался набор инструментов Денвер 3. Размещение рабочей версии web-приложения, было выполнено на площадке компании-хостера 000webhost.

Выполнение работы показало разнообразие инструментов, которые доступны для использования современным веб-разработчикам и позволяет оценить возможности ими предоставляемыми. Наличие данной информации позволяет оценить порядок создания тех или иных web-проектов и по-новому взглянуть на реально действующие web-проекты (с точки зрения разработчика).

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Алиев, В. Интернет – проще простого! [Текст] / В. Алиев В., Д. Винницкий. – СПб.: Питер, 2012. – 192 с.
2. Баловсяк, Н. Эффективный Интернет. Трюки и эффекты [Текст] / Н. Баловсяк, С. Болушевский. – СПб.: Питер, 2009. – 304 с.
3. Берд, Дж. Веб-дизайн. Руководство разработчика [Текст] Дж. Берд. – СПб.: Питер, 2012. – 224 с.
4. Вин, Ч. Как спроектировать современный сайт [Текст] / Ч. Вин. – СПб.: Питер, 2011. – 192 с.
5. Вязилов, Е. Архитектура, методы и средства Интернет-технологий [Текст] / Е. Вязилов. – М.: Красанд, 2009. – 512 с.
6. Годин, В.В. Управление информационными ресурсами [Текст] / В.В. Годин, И.К. Корнеев. – 2005. – 345 с.
7. Гроховский, Л. Продвижение порталов и интернет-магазинов [Текст] / Л. Гроховский. – СПб.: Питер, 2013. – 224 с.
8. Круг, С. Как сделать сайт удобным. Юзабилити по методу Стива Круга [Текст] / С. Круг. – СПб.: Питер, 2010. – 208 с.
9. Морвиль, М. Информационная архитектура в Интернете [Текст] / М. Морвиль, Л. Розенфельд. – М.: Символ-Плюс, 2010. – 608 с.
10. Никифоров, А. Компьютерные сети, протоколы и технологии Интернета [Текст] А. Никифоров. – СПб.: БХВ-Петербург, 2005 с. – 832 с.
11. Рагойша, А. Интернет для начинающих и не только... [Текст] / А. Рагойша. – М.: Красико-Принт, 2004. – 96 с.
12. Райтман, М. Как найти и скачать в Интернете любые файлы [Текст] / М. Райтман. – СПб.: БХВ-Петербург, 2012. – 368 с.
13. Райтман, М. Поисковая оптимизация сайтов. Исчерпывающее руководство [Текст] / М. Райтман. – М.: Эксмо, 2012. – 528 с.

14. Тюкачев, Н. Программирование графики в Delphi [Текст] / Н. Тюкачев, И. Илларионов, В. Хлебостроев. – Спб.: БХВ-Петербург, 2008. – 784 с.
15. Фейт, С. TCP/IP. Архитектура, протоколы, реализация (включая IPv6 и IP Security) [Текст] / С. Фейт. – М.: Лори, 2009. – 424 с.
16. Шкляр, Л., Розен Р. Архитектура веб-приложений [Текст] / Л. Шкляр, Р. Розен. – М.: Эксмо, 2011. – 640 с.
17. Пользователи Интернета в мире // Интернет в России и в мире [Электронный ресурс]. – Режим доступа: http://www.bizhit.ru/index/polzovateli_interneta_v_mire/0-404 (Дата обращения: 25.03.2017)
18. Сайт поддержки языка PHP. История развития языка PHP [Электронный ресурс]. – Режим доступа: <http://php.net/manual/en/history.php.php> (Дата обращения: 25.03.2017)
19. Сайт CyberSecurity.ru. Данные о статистике использования PHP для создания сайтов [Электронный ресурс]. – Режим доступа: <http://www.cybersecurity.ru/net/169098.html>
20. Портал Lab.by. Статья «Тенденции развития мировых веб-технологий в 2013-2014 году» [Электронный ресурс]. – Режим доступа: <http://la.by/blog/tendencii-mirovyh-veb-tehnologiy-v-2013-2014-godu> (Дата обращения: 25.03.2017)
21. Функции и способы задания функций // Автор24 [Электронный ресурс]. – Режим доступа: https://author24.ru/spravochniki/matematika/funkcii_i_sposoby_zadaniya_funkciy/ (Дата обращения: 25.03.2017)
22. Способы задания функции. Примеры // Helpmatan [Электронный ресурс]. – Режим доступа: <http://helpmatan.ru/page101.php> (Дата обращения: 25.03.2017)

23. Интерполяция данных: соединяем точки так, чтобы было красиво // Хабрахабр [Электронный ресурс]. – Режим доступа: <https://habrahabr.ru/post/264191/> (Дата обращения: 25.03.2017)
24. Сайт CyberSecurity.ru. Данные о статистике использования PHP для создания сайтов [Электронный ресурс]. – Режим доступа: <http://www.cybersecurity.ru/net/169098.html> (Дата обращения: 25.03.2017)
25. Портал Lab.by. Статья «Тенденции развития мировых веб-технологий в 2013-2014 году» [Электронный ресурс]. – Режим доступа: <http://la.by/blog/tendencii-mirovyh-veb-tehnologiy-v-2013-2014-godu> (Дата обращения: 25.03.2017)
26. Netcraft. January 2016 Web Server Survey [Электронный ресурс]. – Режим доступа: <https://news.netcraft.com/archives/2016/01/26/january-2016-web-server-survey.html> (Дата обращения: 21.04.2017)
27. Сайт проекта Denwer. Описание функциональности пакета [Электронный ресурс]. – Режим доступа: <http://www.denwer.ru> (Дата обращения: 21.04.2017)

ПРИЛОЖЕНИЯ

Скрипт по созданию базы данных

```

-- phpMyAdmin SQL Dump
-- version 3.5.1
-- http://www.phpmyadmin.net
--
-- Хост: 127.0.0.1
-- Время создания: Апр 22 2017 г., 16:42
-- Версия сервера: 5.5.25
-- Версия PHP: 5.3.13

SET SQL_MODE="NO_AUTO_VALUE_ON_ZERO";
SET time_zone = "+00:00";

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES cp1251 */;

--
-- База данных: `graph`
--
-----

--
-- Структура таблицы `users`
--

CREATE TABLE IF NOT EXISTS `users` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `login_name` varchar(32) NOT NULL,
  `user_pass` varchar(32) NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `login_name` (`login_name`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=1 ;

-----

```

```

--
-- Структура таблицы `user_setts`
--

CREATE TABLE IF NOT EXISTS `user_setts` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `user_id` int(11) NOT NULL,
  `name` varchar(50) NOT NULL,
  `step` float NOT NULL,
  `point_connect` int(11) NOT NULL,
  `line_color_r` int(11) NOT NULL,
  `line_color_g` int(11) NOT NULL,
  `line_color_b` int(11) NOT NULL,
  `show_point` tinyint(1) NOT NULL,
  `x_line` float NOT NULL,
  `y_line` float NOT NULL,
  PRIMARY KEY (`id`),
  KEY `user_id` (`user_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=1 ;

/*!40101 SET
CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET
CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET
COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;

```

Набор PHP-скриптов

```
// Файл #db_login.php
<?php
$db_host='localhost';
$db_database='graph';
$db_username='root';
$db_password='';
$link='';

function dbcon()
{
    global $db_host, $db_database, $db_username, $db_password;
    $dbc = mysql_connect($db_host, $db_username, $db_password)
        or die ('Не удалось соединиться с БД: '. mysql_error());
    mysql_select_db($db_database)
        or die ('Не удалось выбрать БД');
    mysql_query("set names 'cp1251'");
    return $dbc;
}
?>

// Файл index.php
<?php
    echo          '<meta          http-equiv="Content-Type"
content="text/html; charset=windows-1251">';
    include("#db_login.php");

    $link = dbcon();

    echo "<title>Web-приложение для вывода графика аналитически
заданной функции</title>";
    echo "<br><br>Пользователь<br>";
    $query = "SELECT * FROM users order by login_name";

    $result = mysql_query($query) or die(mysql_error());

    echo "<form action='login.php' method='post'>";

    echo "<select name='login_name'> ";

    while ($row = mysql_fetch_array($result))
    {
        echo "<option value = $row[0]>$row[1]</option>";
    }

    echo "</select>";
    echo "<br><br>Введите пароль<br>";
    echo "<INPUT TYPE=password VALUE='' name=user_pass>";
    echo "<br><br><INPUT TYPE=submit VALUE='Войти в систему'>";
    echo "</form>";
    mysql_free_result($result);
```



```

mysql_close($link);
?>

// Файл login.php
<?php
    include("#db_login.php");

    $link = dbcon();

    session_start();
    $user_pass = $_POST['user_pass'];
    $login_name = $_POST['login_name'];

    /* Формируем текст запроса для проверки учетных данных */
    $check_user_query = "Select u.id, u.user_pass From users u
Where u.id='$login_name'";

    /* Выполняем запрос. Если произойдет ошибка - вывести ее.
*/
    $check_user_res = mysql_query($check_user_query) or
die(mysql_error());
    /* Определяем количество записей */
    $check_user_rows = mysql_num_rows($check_user_res);
    /* Если ни одна запись не выбрана */
    if ($check_user_rows == 0)
    {
        /* Выводим страницу с ошибкой */
        Header('Location:login_baduser.php');
    }
    else
    {
        $check_user_row = mysql_fetch_array($check_user_res);

        /* Проверяем пароль */
        if ($user_pass != $check_user_row['user_pass'])
        {
            /* Выводим страницу с ошибкой */
            Header('Location:login_badpass.php');
        }
        else
        {
            session_start();
            $_SESSION['user_id'] = $check_user_row['id'];
            /* Переходим к началу работы */
            Header('Location:graph.php');
        }
    }
}
?>

// Файл login_badpass.php
<?php

```

```

        echo "<title>Web-приложение для вывода графика аналитически
заданной функции</title>";
        echo "Неверно указан пароль! <br><br>";
        echo
            "<tr><td><a
align=center>Авторизация</a></td></tr>";
?>

```

```

// Файл login_baduser.php
<?php
    echo "<title>Web-приложение для вывода графика аналитически
заданной функции</title>";
    echo "Пользователь незарегистрирован в системе4! <br><br>";
    echo
        "<tr><td><a
align=center>Авторизация</a></td></tr>";
?>

```

```

// Файл graph.php
<?php
    session_start();
    echo "<header content='text/html; charset=Windows-1251'>";
    echo "<title>Web-приложение для вывода графика аналитически
заданной функции</title>";

    echo "<form action='show_graph.php' method='post'>";

    echo "<br><br>Функция ";
    if (empty($_SESSION['func_show'])) {
        echo "<INPUT VALUE='sin(x)' name=func_show>";
    } else {
        echo " <INPUT VALUE=' " . $_SESSION['func_show'] . "'
name=func_show>";
    }
    echo "<br><br>Диапазон изменения аргумента от ";
    echo "<INPUT VALUE='-3' name=x_start>";
    echo " до ";
    echo "<INPUT VALUE='3' name=x_end>";
    echo "<br><br><INPUT TYPE=submit VALUE='Вывести график'>";
    echo "</form>";

?>

```

```

// Файл show_graph.php
<?php
    include("polish.php");

    error_reporting(0);
    session_start();
    $user_id = $_SESSION['user_id'];
    $func_show = $_POST['func_show'];
    $x_start = $_POST['x_start'];

```

```

$x_end = $_POST['x_end'];
$_SESSION['func_show'] = $func_show;
$_SESSION['x_start'] = $x_start;
$_SESSION['x_end'] = $x_end;
$x_step = 0.1;

$x_values = array();
$y_values = array();

$x = $x_start;
$step_no = 0;
$has_error = false;
$error = "";

$count_step = ($x_end - $x_start) / $x_step;

while (!$has_error && $step_no <= $count_step) {
    $x_values[] = $x;
    try {
        $y_values[] = calc_rpn($func_show, $x);
    } catch (Exception $e) {
        $has_error = true;
        $error = "Ошибка: " . $e->getMessage();
    }
    //echo $step_no . ": x = " . $x_values[$step_no] . ", y = " .
    $y_values[$step_no] . "<br>";
    $x = round($x + $x_step, 2);
    $step_no++;
}

if ($has_error) {
    echo $error . "<br><br>";
    echo "                "<tr><td><a href='graph.php'
align=center>Назад</a></td></tr>";
}
else {
    $W = 800;
    $H = 600;

    $MB = 20;
    $ML = 8;
    $M = 5;

    $LW = imagefontwidth(2);

    $count=count($x_values);

    $max=0;

    for ($i=0; $i < $count; $i++) {
        $max
        =
        $max
        <
        abs($y_values[$i])?abs($y_values[$i]):$max;

```

```

}

$max = intval($max + ($max/10));

$county = 10;

$im = imagecreate($W, $H);

$bg[0] = imagecolorallocate($im,255,255,255);
$bg[1] = imagecolorallocate($im,231,231,231);
$bg[2] = imagecolorallocate($im,212,212,212);
$c = imagecolorallocate($im,184,184,184);
$text = imagecolorallocate($im,136,136,136);
$bar = imagecolorallocate($im,161,155,0);

$text_width=0;

for ($i=1; $i <= $county; $i++) {
    $strl = strlen(($max / $county) *$i) * $LW;
    if ($strl>$text_width) $text_width=$strl;
}

$ML += $text_width;

$RW = $W - $ML - $M;
$RH = $H / 2 - $MB - $M;

$X0 = $ML;
$Y0 = $H / 2 - $MB;

$step = 2*$RH / $county;

imagefilledrectangle($im, $X0, $Y0-$RH, $X0+$RW, $Y0+$RH,
$bg[1]);
imagerectangle($im, $X0, $Y0+$RH, $X0+$RW, $Y0-$RH, $c);

for ($i=1;$i<=$county;$i++) {
    if ($i < $county / 2) {
        $y=$Y0+$step*$i;
    } else {
        $y=$Y0-$step*($i - $county / 2);
    }
    imageline($im,$X0,$y,$X0+$RW,$y,$c);
    imageline($im,$X0,$y,$X0-($ML-
$text_width)/4,$y,$text);
}

for ($i=0;$i<$county;$i++) {

```

```

        imageline($im,$X0+$i*($RW/$county),$Y0+$RH,$X0+$i*($RW/$cou
nty),$Y0-$RH,$c);
    }

    $dx=($RW/$count)/2;

    $pi=$Y0-($RH/$max*$y_values[0]);
    $px=intval($X0+$dx);

    for ($i=1;$i<$count;$i++) {
        $x=intval($X0+$i*($RW/$count)+$dx);

        $y=$Y0-($RH/$max*$y_values[$i]);
        imageline($im,$px,$pi,$x,$y,$bar);
        $pi=$y;
        $px = $x;
    }

    $ML==$text_width;

    for ($i=1;$i<=$count;$i++) {
        if ($i <= $count / 2) {
            $y=$Y0+$step*$i;
            $str=-($max/($count / 2))*$i;
        } else {
            $y=$Y0-$step*($i - $count / 2);
            $str=($max/($count / 2))*($i - $count / 2);
        }

        imagestring($im,2, $X0-strlen($str)*$LW-$ML/4-2,$y-
imagefontheight(2)/2,$str,$text);
    }

    for ($i=0;$i<=$count;$i++) {

        imageline($im,$X0+$i*($RW/$county),$Y0+$RH,$X0+$i*($RW/$cou
nty),$Y0+$RH+5,$c);
        $str=$x_values[$i * $count / $county];
        imagestring($im, 2, $X0+$i*($RW/$county)-
(strlen($str)*$LW)/2, $Y0+$RH+7,$str,$text);
    }

    header("Content-Type: image/png");

    ImagePNG($im);

    imagedestroy($im);
}
?>

```

```

// Файл polish.php
<?php

error_reporting(0);

class PolishTree{
// Тип узла (переменная, константа, арифметическая операция,
функция)
public $nodeType;
// Значение узла - строковое представление значения
public $nodeValue;
// Указатель на левое поддерево
public $left = NULL;
// Указатель на правое поддерево
public $right = NULL;
}

function calc_rpn($str, $x)
{
    // Стек для разбора выражения
    $stack = array();

    // Массив для сохранения результата
    $out = array();

    // Описание приоритета операций
    $prior = array (
        "sin"=> array("prior" => "5", "assoc" => "right"),
        "cos"=> array("prior" => "5", "assoc" => "right"),
        "tg"=> array("prior" => "5", "assoc" => "right"),
        "ctg"=> array("prior" => "5", "assoc" => "right"),
        "abs"=> array("prior" => "5", "assoc" => "right"),
        "^"=> array("prior" => "4", "assoc" => "right"),
        "*"=> array("prior" => "3", "assoc" => "left"),
        "/"=> array("prior" => "3", "assoc" => "left"),
        "+"=> array("prior" => "2", "assoc" => "left"),
        "-"=> array("prior" => "2", "assoc" => "left"),
    );

    // Заменяем x на его значение
    if ($x >= 0)
        $token = str_replace("x", strval($x), $str);
    else
        $token = str_replace("x", "(0-" . strval(-$x) . ")",
$str);

    // Удаляем пробелы
    $token = preg_replace("/\s/", "", $token);
    // Заменяем запятые на точки
    $token = str_replace(",", ".", $token);
    // Разбиваем строку на символы
    $token = str_split($token);

```

```

// Корректируем при необходимости строку
if (preg_match("/[\+\-\*\\/\^\^]/", $token['0']))
{
    array_unshift($token, "0");
}

// Начали с цифры
$lastnum = TRUE;
// Имя функции пока пустое
$func_name = "";

// Цикл обхода элементов строки
foreach ($token as $key => $value)
{
    // Встретили оператор?
    if (preg_match("/[\+\-\*\\/\^\^]/", $value))
    {
        $endop = FALSE;

        while ($endop != TRUE)
        {
            $lastop = array_pop($stack);
            if ($lastop == "")
            {
                $stack[] = $value;
                $endop = TRUE;
            }
            else
            {
                /* Получаем параметры оператора */
                $curr_prior = $prior[$value]['prior'];
//приоритет текущего оператора
                $curr_assoc = $prior[$value]['assoc'];
//ассоциативность текущего оператора

                $prev_prior = $prior[$lastop]['prior'];
//приоритет предыдущего оператора

                switch ($curr_assoc)
                {
                    case "left":

                        switch ($curr_prior)
                        {
                            case ($curr_prior >
$prev_prior):
                                $stack[] = $lastop;
                                $stack[] = $value;
                                $endop = TRUE;
                                break;

                            case ($curr_prior <=
$prev_prior):

```

```

                                $out[] = $lastop;
                                break;
                            }
                            break;
                        case "right":
                            switch ($curr_prior)
                            {
                                case ($curr_prior >=
$prev_prior):
                                    $stack[] = $lastop;
                                    $stack[] = $value;
                                    $endop = TRUE;
                                    break;
                                case ($curr_prior <
$prev_prior):
                                    $out[] = $lastop;
                                    break;
                            }
                            break;
                    }
                }
            }

            $lastnum = false;
        }
        // Встретили цифру или точку
        elseif (preg_match("/[0-9\\.]/", $value))
        {
            if ($lastnum == TRUE)
            {
                $num = array_pop($out);
                $out[] = $num.$value;
            }
            else
            {
                $out[] = $value;
                $lastnum = TRUE;
            }
        }
        // Открывающаяся скобка
        elseif ($value == "(")
        {
            // Имя функции не пустое?
            if ($func_name != "")
            {
                // помещаем ее в стек
                $stack[] = $func_name;
            }
        }
    }
}

```



```

        $func_name = "";
    }

    $stack[] = $value;
    $lastnum = FALSE;
}
// Буква
elseif (preg_match("/[a-z\\.]/", $value))
{
    // Добавляем ее к имени функции
    $func_name = $func_name . $value;
    $lastnum = FALSE;
}
// Закрывающаяся скобка
elseif ($value == ")")
{
    $found_bracket = FALSE;
    while ($found_bracket != TRUE)
    {
        $op = array_pop($stack);

        if ($op == "(")
        {
            $found_bracket = TRUE;
        }
        else
        {
            $out[] = $op;
        }
    }

    $lastnum = FALSE;
}
}

$stack1 = $stack;
$rpnp = $out;

while ($stack_el = array_pop($stack1))
{
    $rpnp[] = $stack_el;
}

$token = strtok(implode(" ", $rpnp), ' ');
$stack = array();

while ($token !== false)
{
    if (in_array($token, array('*', '/', '+', '-', '^')))
    {
        if (count($stack) < 2)

```

```

        throw new Exception("Недостаточно данных в
стеке для операции '$token'");
        $b = array_pop($stack);
        $a = array_pop($stack);
        switch ($token)
        {
            case '*': $res = $a*$b; break;
            case '/': if ($b <> 0) {
                $res = $a/$b;
            } else {
                throw new Exception("Неверный
аргумент для оператора деления!");
            }
            break;
            case '+': $res = $a+$b; break;
            case '-': $res = $a-$b; break;
            case '^': $res = pow($a,$b); break;
        }
        array_push($stack, $res);
    }
    elseif (is_numeric($token))
    {
        array_push($stack, $token);
    }
    elseif (in_array($token, array('sin', 'cos', 'tg', 'ctg',
'abs')))
    {
        if (count($stack) == 0)
            throw new Exception("Недостаточно данных в
стеке для функции '$token'");
        $a = array_pop($stack);
        switch ($token)
        {
            case 'sin': $res = sin($a); break;
            case 'cos': $res = cos($a); break;
            case 'tg': if (cos($a) <> 0) {
                $res = tan($a);
            } else {
                throw new Exception("Неверный
аргумент для функции tg!");
            }
            break;
            case 'ctg': if ($a <> 0) {
                $res = 1/tan($a);
            } else {
                throw new Exception("Неверный
аргумент для функции ctg!");
            }
            break;
            case 'abs': $res = abs($a); break;
        }
        array_push($stack, $res);
    }

```

```
    }
    else
    {
        throw new Exception("Недопустимый символ в
выражении: $token");
    }

    $token = strtok(' ');
}
if (count($stack) > 1)
    throw new Exception("Количество операторов не
соответствует количеству операндов");

return array_pop($stack);
}

?>
```