

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Локтионова Оксана Геннадьевна
Должность: проректор по учебной работе
Дата подписания: 28.02.2023 13:44:26
Уникальный программный ключ:
0b817ca911e6668abb13a5d426d39e5f1c11eabbf73e943df4a4851fda56d089

МИНОБРАЗОВАНИЯ И НАУКИ РОССИИ
Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Юго-Западный государственный университет»
(ЮЗГУ)

Кафедра информационной безопасности

УТВЕРЖДАЮ
Проректор по учебной работе
О.Г. Локтионова
« 28 » 02 2021 г.



СОЗДАНИЕ САЙТОВ НА ЯЗЫКЕ JAVASCRIPT И
ОБЕСПЕЧЕНИЕ ИХ ИНФОРМАЦИОННОЙ
БЕЗОПАСНОСТИ

Методические указания по выполнению лабораторных работ
для студентов укрупненной группы специальностей и
направлений подготовки 10.00.00

Курск 2021

УДК 004.56.5(076.5)

Составитель: А.Л. Ханис

Рецензент

Кандидат технических наук, доцент кафедры
информационной безопасности А.Г. Спеваков

Создание сайтов на языке JavaScript и обеспечение их информационной безопасности : методические указания по выполнению лабораторных работ для студентов укрупненной группы специальностей и направлений подготовки 10.00.00 / Юго-Зап. гос. ун-т; сост.: А.Л. Ханис. - Курск, 2021. - 70 с.: ил. 12, табл. 1. – Библиогр.: с. 70.

Содержат краткие теоретические положения о языке программирования JavaScript, а также методах защиты информации с помощью Java-скриптов, предназначенных для защиты контента от несанкционированного копирования.

Предназначены для студентов укрупненной группы специальностей и направлений подготовки 10.00.00.

Текст печатается в авторской редакции

Подписано в печать . Формат 60x84 1/16.
Усл.печ. л. 2,38 Уч.-изд. л. 2,15. Тираж 100 экз. Заказ. Бесплатно.
Юго-Западный государственный университет.
305040, г. Курск, ул. 50 лет Октября, 94.

Введение

В сознании большинства пользователей Интернет ассоциируется с тремя основными информационными технологиями:

- электронная почта (e-mail);
- файловые архивы FTP;
- World Wide Web.

Технология World Wide Web предоставляет простой интерфейс для доступа к разнообразным сетевым ресурсам. Это привлекло внимание коммерческих структур и привело к лавинообразному росту числа пользователей WWW.

Динамические сайты, в которых веб-страницы генерируются или формируются создаются динамически в процессе исполнения запроса пользователя. Динамические сайты бывают двух типов. В первом типе сайтов, веб-страницы генерируются или формируются из данных хранящихся на сервере в базе данных. Во втором типе сайтов веб-страницы генерируются на стороне клиентского приложения (в браузере).

Для управления клиентскими сценариями просмотра страниц Web-сайта можно использовать языки программирования этих сценариев, например, JavaScript. Клиентские сценарии широко применяются для решения таких задач, как проверка информации, введенной пользователем в формы, перед ее отправкой на сервер или для программирования различных ответных реакций на действия пользователя, делающих веб-страницы интерактивными.

В данном методическом пособии рассматриваются основы языка программирования JavaScript. Рассмотрены Java-скрипты для защиты контента от несанкционированного копирования.

1. Методы ввода и вывода информации в языке JavaScript.

Краткие теоретические положения

Для ввода и вывода информации в языке JavaScript используются методы `alert()`, `prompt()`, `confirm()`. Эти методы генерируют различные окна сообщений. Метод `alert()` отображает окно предупреждения с соответствующим сообщением. Окно содержит кнопку `Ok`. Аргументом данного метода является строка.

Метод `prompt()` служит для получения данных от пользователя. При его вызове отображается окно приглашения с текстовым полем. Метод может содержать два аргумента. Оба эти аргумента должны быть строками. Первый аргумент – сообщение, которое отображается в окне. Второй аргумент – это текст по умолчанию, который должен появиться в соответствующем поле. Также этот метод возвращает значение, которое также является строкой – это текст, который ввел пользователь. Метод `confirm()` отображает окно подтверждения, которое сходно с окном предупреждения, генерируемое методом `alert()`, но содержит 2 кнопки: Ok и Cancel.

Также для вывода информации в языке JavaScript используются метод `document.write()`, который позволяет дописывать строку текста прямо в страницу.

Задание №1

Цель работы: изучить методику ввода и вывода информации в языке JavaScript.

Назначение программы: в данной программе выдается окно с запросом —“Здравствуйте, как Вас зовут?”. Пользователь вводит имя, например, Сергей. Программа выдает окно с сообщением: —Рад Вас видеть, Сергей. Спасибо Вам, что зашли на мой сайт”.

Текст программы:

```
<html>
<head>
<title>сценарий 1</title>
<script>
var nameUser;
nameUser=prompt("Здравствуйте, как вас зовут?","аноним");
alert("Рад Вас видеть"+nameUser+"Спасибо Вам, что зашли на
мой сайт");
</script>
</ head>
<body>
</ body>
</ html>
```

Задание №2

Цель работы: изучить методику применения метода `document.write ()`

Назначение программы: программа 10 раз выводит сообщение —Привет, мир!!!

Текст программы:

```
<html>
<head>
<title> Сценарий вывода информации </title>
</head>
<body>
<script type="text/javascript">
for (var i=0; i<10; i++)
{
document.write("Привет, мир!!!");
}
</script>
</body>
</html>
```

2. Переменные и массивы в языке JavaScript. Краткие теоретические положения

Рассмотрим примеры объявления и инициализации переменных в языке JavaScript:

```
var x,y,z;
var s=1.34e-5, msg="Строка", Flag=false;
```

Пока переменная не получит значение, она не определена – `undefined`. Чтобы определить тип переменной можно использовать операцию `typeof`, например, `typeof(msg)`. Она возвращает строку с типом переменной, в данном случае - `string`. Следующий оператор выводит сообщение о значении переменной и типе переменной:

```
Document.write(—Переменная msg="+msg+" – "+typeof(msg)
+"n");
```

Сообщение будет иметь вид:

Переменная msg=Строка - string

В языке JavaScript для использования нескольких данных одного типа используются массивы. Нумерация элементов массива

начинается с нуля. Язык JavaScript позволяет не перечислять элементы по одному, а сразу вывести все элементы массива. Для этого необходимо обратиться к самому массиву, в результате чего получится строка, в которой все элементы массива перечислены через запятую. Узнать длину массива можно с помощью свойства `length`. Например, следующая строка выведет размер массива `stars`:

```
alert(stars.length)
```

Пример объявления массива, задания значений элементов массива и вывода элементов массива рассмотрены в практическом занятии № 17.

Здание №3

Цель работы: изучить методику обработки массивов в языке JavaScript

Назначение программы: в программе реализован массив из 5 элементов (звезды), показан вывод первого элемента массива и всех элементов массива.

Текст программы:

```
<html>
<title>Инициализация массива</title>
<script>
//Объявляем массив и инициализируем значения элементов
var stars=new Array('Сириус','Канопус', 'Арктур', 'Капелла',
'Вега');
//Вывод первого элемента массива
alert(stars[0]);
//Вывод всех элементов массива
alert(stars);
</script>
</head>
<body>
</body>
</html>
```

Задание №4

Цель работы: разработать программу обработки массива в языке JavaScript

Назначение программы: программа должна находить сумму элементов массива, длину массива и среднее арифметическое значение элементов массива. Программа должна выводить результаты с помощью метода `document.write()` или метода `alert()`.

3. Основные операторы в языке программирования JavaScript

Краткие теоретические положения

Оператор `if..else` управляет механизмом ветвления, позволяет выбирать одно из альтернативных действий в зависимости от значения условия. Синтаксис данного оператора имеет вид:

```
if (логическое выражение)
{
операторы 1
}
else
{
операторы 2
}
```

Сначала вычисляется логическое выражение, затем если оно равно `true`, выполняются операторы 1, если оно равно `false`, то выполняются операторы 2. Альтернативная ветвь `else` может отсутствовать. В этом случае, если логическое выражение равно `false`, то управление передается следующему оператору и продолжается выполнение сценария.

Оператор `switch` позволяет сравнить значение с множеством других. Синтаксис данного оператора имеет вид:

```
switch (выражение)
{
case значение1:
операторы
case значение2:
операторы
.....
default:
операторы
}
```

В данном операторе определяется значение выражения. Значение, возвращаемое выражением, сравнивается со значениями, указанными после ключевых слов case. Если имеет место совпадение, выполняется соответствующий блок операторов. Если не найдено ни одно совпадение, то выполняются операторы после инструкции default.

Оператор for служит для создания цикла. Он имеет следующий синтаксис:

```
for (выражение инициализации; выражение условия;
выражение цикла)
{
операторы
}
```

Выражение инициализации служит для задания начального значения цикла. Выражение условия позволяет прекратить цикл, когда условие перестанет выполняться, то есть примет значение false. Выражение цикла осуществляет инкремент или декремент счетчика цикла.

Оператор while сходен с оператором for, но он не производит инициализацию и инкремент счетчика в своем объявлении. Синтаксис этого оператора следующий:

```
while (выражение условия)
{
Операторы
}
```

Оператор do..while практически идентичен оператору while, но, поскольку в нем проверка условия осуществляется в конце, он гарантирует выполнение операторов по крайней мере один раз:

```
do
{
Операторы
} while (выражение условия)
```

Задание №5

Цель работы: изучить методику применения оператора switch

Назначение программы: программа осуществляет перевод названия животных с английского языка на русский.

Текст программы:

```

<html>
<head>
<title>переводчик</title>
<script>
var trans;
var beast=prompt ("Введите название животного","dog");
switch (beast)
{
case "dog":
trans="собака";
break;
case "cat":
trans="кошка";
break;
case "cow":
trans="корова";
break;
default:
trans="неизвестное животное"
}
alert(beast+"---"+trans);
</script>
</ head>
<body>
</ body>
</ html>

```

Задание №6

Цель работы: изучить методику применения оператора while.

Назначение программы: программа осуществляет расчет факториала числа.

Текст программы:

```

<html>
<head>
<title> факториал с помощью оператора while </title>
<script>

```

```

var f=1;
var x=prompt("введите число","5");
x=+x;
var i=1;
while(i<=x)
{
f=f*i;
i++;
}
alert(f);
</script>
</head>
<body>
</body>
</html>

```

Задание №7

Цель работы: изучить методику применения оператора if..else, программу разработайте самостоятельно.

Назначение программы: программа рассчитывает значение функции:

$x+10$, при $x<0$

$F(x) = x^2 + 4$, при $0 \leq x \leq 5$

$5/x$, при $x>5$

4. Функции в языке программирования JavaScript. Краткие теоретические положения

Функция – это набор команд, объединенных под общим именем для выполнения определенной задачи. Синтаксис функции имеет следующий вид:

```

function ИмяФункции ([аргумент1] [,..аргумент N])
{
операторы
}

```

Ключевое слово function объявляет функцию с именем ИмяФункции. Функциям могут передаваться аргументы, которые перечисляются через запятую в скобках рядом с именем функции. В качестве аргументов могут выступать переменные, значения и

выражения. Функции можно объявить в любом месте внутри элемента SCRIPT. Тем не менее, рекомендуется объявить функции в блоке HEAD до начала выполнения основного кода, иначе функция может быть недоступной в момент ее вызова.

Функции могут возвращать значение. Для этого необходимо использовать оператор возврата return. Функции можно использовать в выражениях. Функция может объявляться в выражении. Обычно в этом случае функция присваивается переменной, которая затем может использоваться в выражении:

```
var cube =function(x) {return x*x*x}
alert(cube(2)+cube(3));
```

В данном случае объявляется переменная-функция, вычисляющая куб числа.

Задание №8

Цель работы: изучить методику использования функций

Назначение программы: программа рассчитывает в зависимости от выбранного режима сумму или произведение элементов массива. Для расчета суммы и произведения использованы две функции sum() и product().

Текст программы:

```
<html>
<head>
<title>Пример использования функций</title>
<script>
//Глобальный массив
var a= new Array (1,2,4,8);
function sum()
{
var s=0; //локальная переменная для хранения суммы
//Цикл по всем элементам массива a
for (var i in a)
{
s=s+a[i];
}
alert("Сумма элементов массива="+s);
}
```

```

function product()
{
var p=1; //локальная переменная для хранения произведения
for (var i in a)
{
p=p*a[i];
}
alert("Произведение элементов массива="+p);
}
</script>
</head>
<body>
<script>
var x=prompt("Найти сумму или произведение (0 или 1)?","0");
if (x=="0")
{
sum();
}
if (x=="1")
{
product();
}
</script>
</body>
</html>

```

Задание №9

Цель работы: изучить методику использования значения функции

Назначение программы: функция `sum(a,b)` возвращает сумму `a` и `b`, затем значение функции используется в выражении и результат выражения выводится на экран методом `alert`:

Текст программы:

```

<html>
<head><title>Использование значения функции</title>
<script>
function sum(a,b)

```

```

{
return (a+b);
}
</script>
</head>
<body>
<script>
var x1=5, x2=6, x3=3, x4=3;
//Отображаем (x1+x2)*(x3+x4)
alert (sum(x1,x2)*sum(x3,x4));
</script>
</body>
</html>

```

Задание №10

Цель работы: получить навыки самостоятельной разработки программ с использованием функций на языке JavaScript

Назначение программы: программа рассчитывает число сочетаний без повторений из n элементов по m элементов по формуле:

$$C_n^m = \frac{n!}{m!(n-m)!}$$

При решении задачи расчет факториала надо выполнить с помощью специальной функции.

5. Обработчики событий в языке программирования JavaScript. Краткие теоретические положения

В языке JavaScript часто используются реагирования на события, которые могут быть вызваны пользователем или браузером. К событиям относятся открытие новой страницы, перемещение указателя мыши, щелчок кнопкой мыши. Каждое событие имеет соответствующий обработчик, который автоматически реагирует на возникшее событие. Например, если посетитель веб-страницы нажимает некоторую кнопку, происходит событие Click и вызывается обработчик событий onClick. Если указатель мыши пересекает гипертекстовую ссылку, происходит событие MouseOver, которому соответствует обработчик событий onMouseOver. Рассмотрим сценарии с данными обработчиками событий.

Задание №10

Цель работы: изучить методику использования обработчиков событий onmouseover и onmouseout для динамической смены изображений-ссылок на веб-странице

Назначение программы: изначально на веб-странице отображается рисунок a.jpg. Когда посетитель наводит указатель мыши на изображение, картинка меняется на b.jpg. Это результат действия обработчика onmouseover. Когда пользователь убирает указатель с картинки, снова появляется изображение a.jpg. Здесь действует команда onmouseout.

Текст программы:

```
<html>
<head>
<title>JavaScript</title>
</head>
<body topmargin=70>
<center>
<h1> Демонстрация обработчика события наведения указателя
мыши </h1>
<br>
<a href="primer1.html" onmouseover="document.pic.src='b.jpg'"
onmouseout="document.pic.src='a.jpg'">      <img      SRC="a.jpg"
name="pic" hspace=40 border=2></a>
</center>
</body>
</html>
```

Задание №11

Цель работы: изучить методику использования обработчика событий onclick

Назначение программы: программа по ссылке переходит на поисковую систему Rambler. Но предварительно по обработчику события onclick выдается сообщение «Желаем удачного поиска информации»

Текст программы:

```
<html>
```

```

<head>
<title>JavaScript</title>
</head>
<body topmargin=70>
<center>
<h1> Демонстрация обработчика щелчка кнопки мыши </h1>
<br>
<a href="http://www.rambler.ru" onClick="alert('желаем удачного
по-иска информации');"> Поиск </a>
</center>
</body>
</html>

```

6. Окна в языке программирования JavaScript. Краткие теоретические положения

Объект `window` представляет окно браузера или фрейм. С помощью сценария можно создавать или удалять окна, загружать в них документы и записывать данные. Окно создается методом `open()` объекта `window`.

`Window.open (URL-адрес, имя, опции, замена);`

URL - адрес – строка с URL – адресом документа, который следует загрузить в открываемое окно; если вы не хотите загружать в окно никакой документ, то укажите пустую строку — “.

Имя - строка с именем окна, которое позже может стать значением атрибута `target` в тегах `<a>` ссылок, если вы не будете ссылаться на данное окно, то укажите пустую строку — “. Если вам нужно, чтобы страница загружалась в маленьком окне, то данный параметр должен иметь значение `yes`.

Опции – строка, в которой через запятую указаны дополнительные параметры оформления окна.

Замена – необязательное логическое значение, указывающие, должен ли документ по заданному URL – адресу заменить содержимое окна (применительно к уже существующим окнам).

Пример открытия окна:

```

mywindow= open(http://www.yandex.ru, —Yandex”,
—width=400, height=400”)

```

Для закрытия окна служит метод `close()`. Однако выражение `window.close()` закрывает главное окно, а не то окно, которое создано методом `open()`. В этом случае необходима ссылка на созданное окно, которую следует сохранить в глобальной переменной.

Пример закрытия окна:

```
Var objwin=window.open(—mypage.htm”, —Моя страница”)
Objwin.close()
```

При создании окна методом `open()` с помощью третьего параметра можно указать в строке множество дополнительных параметров, разделенных запятыми.

Наиболее популярными являются следующие параметры:

`width, height` – ширина и высота окна в пикселях;

`left, top` – горизонтальная и вертикальная координаты относительно левого верхнего угла экрана в пикселях;

`menubar` – разрешение/запрет отображения панели меню;

`toolbar` - разрешение/запрет отображения панели инструментов;

`scrollbars` – разрешение/запрет отображения полос прокрутки;

`resizable` – разрешение/запрет изменения размеров окна пользователем;

`location` – параметр определяет, будет ли отображаться поле ввода адреса HTML – документа;

`status` – отображение строки состояния.

Логические значения для данных параметров указывают как `yes` и `no`.

Также возможен следующий вариант синтаксиса команды открытия окна:

```
Window.open(—адрес URL’, ‘имя окна’, config=‘параметр 1,
параметр 2, параметр n’)
```

Параметр `config` показывает, что следующие атрибуты относятся к конфигурации нового окна и определяют его внешний вид. Все атрибуты параметра `config` перечисляются через запятую без пробелов и заключаются в одинарные кавычки.

У объекта `window` имеется синоним `self`, используемый при обращении к окну, содержащему текущий элемент.

Задание №12

Цель работы: изучить методику создания и использования окон в языке JavaScript

Назначение программы: в маленькое окно загружается фотография (файл foto.jpg). По нажатию кнопки «Увеличить» создается новое окно большого размера и в него загружается данная фотография.

Текст программы: программа реализована в виде двух файлов. Имя второго файла – foto2.html.

```

<html>
<head>
<title>JavaScript</title>
<script language="JavaScript">
function openwin()
{
    window.open('foto2.html','joe', config='height=300, width=400,
toolbar=no, menubar=no, scrollbars=no, resizable=no, loca-tion=no,
status=no')
    self.name="main window"
}
</script>
</head>
<body>
<h3>Фотография</h3>
 <br> <br>
<form>
<button type="SUBMIT" onClick="openwin()">Увеличить
</button>
</form>
</body>
</html>

```

Файл foto2.html

```

<html>
<head>
<title> Фотография </title>
</head>
<body background="foto.jpg">

```

```
</body>
</html>
```

Задание №13

Цель работы: получить навыки самостоятельной работы создания фотогалереи в языке JavaScript.

Назначение программы: на веб-странице имеются несколько фотографий маленького размера. При щелчке мышью на фотографии увеличенная фотография должна открываться или в отдельном фрейме или в отдельном окне. Добавьте кнопки перехода к следующей и предыдущей фотографии.

7. Анимация изображения в языке программирования JavaScript. Краткие теоретические положения

Для анимации изображения необходимо через равные промежутки времени изменять атрибут `src` в изображении. Для создания анимации используют одну из функций объекта `Window`: `setTimeout ()` или `setInterval ()`. Рассмотрим пример анимации изображения.

Задание №14

Цель работы: изучить методику использования анимации изображений в языке JavaScript.

Назначение программы: В данном сценарии последовательно происходит анимация изображений `img5.jpg`, `img4.jpg`, `img3.jpg`, `img2.jpg`, `img1.jpg`, `img0.jpg`. Это происходит с помощью функции `animation ()`. Данная функция получает управление после щелчка кнопкой мыши на рисунке. После чего функция изменяет рисунок, уменьшает счетчик и порождает другой свой экземпляр, выполнение которого откладывается на одну секунду. Когда счетчик становится отрицательным, процесс анимации прекращается.

Текст программы:

```
<html>
<head>
<meta charset="utf-8">
<title>Анимация изображения</title>
<script>
```

```

//Создаём массив для хранения изображений
numerals=new Array(6);
//В цикле создаём объекты изображений и загружаем рисунки
for(var i=1;i<6;i++){
//Создаём объекты изображений
numerals[i]=new Image();
// Загружаем все рисунки
numerals[i].src="img"+i+".jpg";
}
//Функция, осуществляющая анимацию
function animation(count){
//Загружаем текущий рисунок
document.images[0].src="img"+count+".jpg";
count--; //Уменьшаем счётчик
//Пока есть изображения, рекурсивно вызываем функцию
animation()
if (count>=0){
//Открываем вызов функции на секунду
setTimeout("animation("+count+");",1000);
}
}
</script>
</head>
<body>
<h2 style="text-align : center;">Анимация изображения</h2>
<p style="text-align : center;" >
 </p>
</body>
</html>

```

8. Создание меню в языке программирования JavaScript.

Краткие теоретические положения

Одна из главных задач, которую необходимо решать программисту при разработке сайта, это создание удобной системы навигации (меню). Для этого можно использовать обычные гиперссылки, а можно воспользоваться средствами языка JavaScript.

Рассмотрим пример создания простого горизонтального меню. Для создания такого меню использовался обычный маркированный список со ссылками внутри, а также каскадные таблицы стилей: стиль `ul li` позволяет выстроить список в линию и убрать маркер; стиль `li a` описывает внутренние отступы и внешний вид меню. Переход по ссылкам в меню осуществляется при помощи обработчика событий `onClick`.

Задание №18

Цель работы: изучить методику создания меню в языке JavaScript.

Назначение программы: программа создает горизонтальное меню следующего вида:

О нас Новости Услуги Контакты

Рис 7. Горизонтальное меню.

Текст программы:

```
<html>
<head>
<title>Простое горизонтальное меню</title>
<!--Определяем стиль -->
<style type="text/css">
ul li{list-style-type:none;display:inline}
li a{font-family:Tahoma,Helvetica,sans-serif;text-decoration:none;color:#fff}
li a{display:block;float:left;padding:4px 10px;background:193D6A}
li a:hover{background:#408BE8;padding:7px 14px;position:relative;top:-3px}
</style>
</head>
<body>
<ul>
<li><a href="#" onClick="location.href='about.html';">
О нас</a></li>
<li><a href="#" onClick="location.href='news.html';">
Новости</a></li>
```

```
<li><a href="#" onClick="location.href='serv.html';">
Услуги</a></li>
<li><a href="#" onClick="location.href='cont.html';">
Контакты</a></li>
</ul>
</body>
</html>
```

Дополнительные задания:

1. Создайте веб-страницы, на которые указывают пункты меню и убедитесь в правильной работе сайта.

2. Измените цветовую раскраску меню следующим образом: фон меню – зеленый цвет, маркер, выбирающий пункт меню – красный цвет, текст – черный цвет.

Задание №19

Цель работы: получить навыки самостоятельной разработки меню в языке JavaScript.

Назначение программы: реализация вертикального раскрывающегося меню. В исходном состоянии меню выглядит подобно кнопке (прямоугольник с надписью). При наведении указателя мыши на данный прямоугольник меню раскрывается, показывая опции (пункты) своего подменю. При уходе указателя мыши с прямоугольника раскрывшегося меню последнее сворачивается и принимает исходное состояние. Внешний вид меню представлен на рис. 8. При щелчке мышью на пункте меню загружается соответствующая поисковая система.



Рис 8. Вертикальное раскрывающиеся меню.

9. Защита информации с помощью аутентификации в языке программирования JavaScript. Краткие теоретические положения

Объект Form служит контейнером для пересылки данных от пользователя на сервер. Объект Document является контейнером всех форм на странице. Чтобы обратиться к отдельному элементу в конкретной форме, необходимо воспользоваться конструкцией document.Имя Формы.ИмяЭлемента.

Document. Myform.myelement.value = “Мое текстовое поле”;

Рассмотрим основные свойства, методы и события формы

Характеристика	Название
Свойства	action, elements[], length, method, name, target
Методы	reset(), submit()
События	onReset, onSubmit

Таблица 1. Свойства, методы и события формы

Свойство **action** задает URL-адрес сценария на сервере, которому отправляется форма:

```
<form name="myform"
```

```
action=http://www.myserver.com/cgi-bin/registration.asp
```

```
method="post">
```

```
<!--Объекты формы- - >
```

```
</form>
```

Свойство **elements[]** представляет собой массив, в который включены все элементы формы. Ссылаться на элементы можно по их индексу, который начинается с нуля.

```
formName.element[2].value
```

Свойство **length** определяет количество элементов на форме.

Свойство **method** определяет способ отправки формы серверу. На практике чаще всего используются методы GET и POST. В методе GET данные передаются на адрес, указанный в свойстве action, через строку поиска (часть адреса, идущая после знака ?) в URL. В методе POST данные передаются в теле HTML – запроса.

Этот метод следует использовать при передаче серверу большого объема данных.

Свойство **name** ссылается на внутренние имена форм.

Свойство **target** указывает окно назначения браузера, в котором должен отображаться результат отправки формы.

Метод **reset** сбрасывает значения всех элементов формы в исходное состояние.

Метод **submit ()** используется для передачи данных формы.

Событие **onReset** восстанавливает значения по умолчанию в полях формы и возникает при нажатии кнопки Reset или при выполнении метода reset ().

Событие **onSubmit** запускается в момент отправки пользователем данных формы на сервер.

Рассмотрим элементы управления в формах.

Поля ввода данных предназначены для ввода однострочной информации. Пример данного элемента:

```
<input type="text" name="CustName" size=40 maxlength=60>
```

Поля для паролей предназначены для ввода паролей. Обычно браузеры вместо вводимых символов на экран выводят звездочки (*). Пример этого элемента:

```
<input type="password" name="pass" size=20>
```

Многострочные поля ввода предназначены для ввода информации, которая не помещается в обычных полях ввода. Чтобы создать текстовое поле из десяти строк по 80 символов в каждом, наберите следующий код:

```
<textarea name="Comments" rows = 10 cols=80></textarea>
```

Флажки предназначены для обозначения логических данных, принимающих значение true или false. Пример этого элемента:

```
<input type="checkbox" name="hobby" value="Литература">
```

Использование флажка на форме:

Литература

Переключатели предназначены, чтобы пользователь выбрал только один вариант из нескольких возможных. Пример этого элемента:

```
<input type="radio" name="trade">
```

value =”student”> Студент

Использование переключателя на форме:

Студент

Списки также предназначены для выбора элемента. Однако в отличие от переключателей из списка можно выбрать несколько элементов. Пример списка:

```
<select name =”writer” size=2 multiple>
<option> Иванов </option>
<option> Петров </option>
<option> Сидоров </option>
</select>
```

Если параметр size=1, то список раскрывающийся. При size=2 список будет прокручиваться внутри окна. Параметр multiple служит для выбора нескольких элементов из списка.

Кнопки

В стандарте HTML определены четыре типа кнопок, которые можно включать в формы: Button (обычная кнопка), Submit (кнопка отправки данных на сервер), Reset (кнопка сброса полей формы), Image (кнопка с изображением).

Задание №20

Цель работы: изучить методику создания формы регистрации на сайте

Назначение программы: программа создает типовую форму регистрации на сайте следующего вида

Регистрация на сайте

Имя: *

Фамилия: *

Адрес Email: *

Логин: *

Пароль: *

Повтор Пароля: *

Страна:

Ваш пол: Мужской
 Женский

С условием регистрации согласен.

Поля, помеченные звездочкой (*), являются обязательными для заполнения.

Рис. 9 Форма регистрации на сайте.

С помощью функции `validateForm ()` при отправке формы на сервер (см. текст программы) происходит проверка заполнения всех полей формы.

Текст программы:

```

<html>
<head>
<title>Регистрация на сайте</title>
<script>
//Функция проверки формы на наличие пустых полей
function validateForm()
{
var validateFlag=true;//возвращаемая функция логическая
переменная
for (var i=0; i<6;i++)//Проверяем первые шесть элементов
{
// В действительности переменная countElem не нужна. Она
введена для
// улучшения читабельности кода
var countElem=document.regform.elements[i];
if (countElem.value=="") //Текущий элемент пустой?
{//Если да, то
validateFlag=false; //Устанавливаем флаг в false

```

```

//Выдаем предупреждение для текущего элемента
alert ("Введите значение в поле"+countElem.name);
break; // Цикл уже можно не продолжать
}
}
alert('Поздравляем, Вы успешно прошли регистрацию');
return validateFlag;
}
</script>
</head>
<body>
<h2>Регистрация на сайте</h2>
<!--Создаем форму-->
<form name="regform" method="post" onsubmit="return validate-
Form();">
  <!--Для удобства форматирования элементы формы
располагаем в таблице-->
  <table width="100%">
    <tr>
      <!--Создаем поля ввода данных-->
      <td> Имя: <sup>*</sup></td>
      <!--Поле Имя-->
      <td><input type="text"size="35" name="firstname"></td>
    </tr>
    <tr>
      <td>Фамилия: <sup>*</sup></td>
      <!--Поле Фамилия-->
      <td><input type="text"size="35" name="lastname"></td>
    </tr>
    <tr>
      <td>Адрес Email: <sup>*</sup></td>
      <!--Поле Email-->
      <td><input type="text"size="35" name="email"></td>
    </tr>
    <tr>
      <td>Логин: <sup>*</sup></td>
      <!--Поле Логин-->

```

```

<td><input type="text" size="35" name="login"></td>
</tr>
<tr>
<td>Пароль: <sup>*</sup></td>
<!--Поле Пароль-->
<td><input type="password" size="20" name="pass1"></td>
</tr>
<tr>
<td>Повтор Пароля: <sup>*</sup></td>
<!--Поле Повтор Пароля-->
<td><input type="password" size="20" name="pass2"></td>
</tr>
<tr>
<td>Страна:</td>
<!--Далее список-->
<td><select size="1" name="country">
<!--Вариант списка-->
<option value="Russia">Россия</option>
<option value="Ukraine">Украина</option>
<option value="Belorussia">Белоруссия</option>
<option value="USA">США</option>
<option value="France">Франция</option>
<option value="Great Britain">Великобритания</option>
<option value="other">Другая</option>
</td>
</tr>
<tr>
<td>Ваш пол:</td>
<td>
<!--Формируем переключатели -->
<input type="radio" name="sex" value="male">Мужской<br/>
<input type="radio" name="sex" value="female">Женский<br/>
</td>
</tr>
<tr>
<td colspan=2>

```

```

-->
<!--Флажок со своим описанием занимает всю строку в таблице
-->
<input type="checkbox" name="consent">
С условием регистрации согласен.</td>
</tr>
<tr>
<td colspan=2> Поля, помеченные звездочкой (<sup>*</sup>),
являются обязательными для заполнения.</td>
</tr>
<tr>
<!--Заканчивают форму две стандартные кнопки-->
<td><input type="submit" value="Зарегистрировать"></td>
<td><input type="reset" value="Очистить"></td>
</tr>
</table>

</form>
</body>
</html>

```

Дополнительные задания:

1. Усовершенствуйте форму регистрации на сайте таким образом, чтобы происходила проверка на совпадение полей: «пароль» и «повтор пароля».
2. Усовершенствуйте форму регистрации на сайте таким образом, чтобы происходила проверка поля адрес E-mail на наличие символа @.

Задание №21

Цель работы: изучить методику защиты паролем конкретной страницы сайта.

Назначение программы: если пароль неверный, то пользователь переадресовывается, если верный, то выдается сообщение и загружается текущая страница. Данный Java-скрипт следует вставить между тегами <head> и </head> той Web-страницы, которую следует защитить паролем. В данном демонстрационном примере пароль password, в случае неверного

ввода пользователь переадресовывается на Web-страницу <http://studia.scriptic.ru/>

Текст программы:

```
<head>
<title>Защита паролем. Если пароль неверный то пользователь
пе-реадресовывается, если верный то выдается сообщение и
загружается страница.</title>
<script language=JavaScript>
<!--//
р = prompt('Пожалуйста, введите пароль');
// Запрос пароля
if (р=='password') {
// Не забудьте сменить пароль
alert('Вход разрешен')
// текст сообщения о том что пароль принят
} else {
top.location.href="http://studia.scriptic.ru/"
// Страница на которую будет направлен пользователь если
пароль не верен
}
//-->
</script>
</head>
```

Задание №22

Цель работы: изучить методику защиты паролем сайта, при которой на каждой странице сайта используется отдельный логин и пароль.

Назначение программы: В данном демонстрационном примере рассматривается защита сайта из трех страниц. Например, для страницы news.htm используется логин «пример1» и пароль «при-мер1». Данный Java-скрипт может быть реализован на главной странице сайта.

Текст программы:

```
< head>
< script language="JavaScript">
<!-- Begin
```

```

function Login(){
var done=0;
var username=document.login.username.value;
username=username.toLowerCase();
var password=document.login.password.value;
password=password.toLowerCase();
if (username=="пример" && password=="пример") { win-
dow.location="kontakt.htm"; done=1; }
if (username=="пример1" && password=="пример1") { win-
dow.location="news.htm"; done=1; }
if (username=="пример2" && password=="пример2"){ win-
dow.location="serv.htm"; done=1;}
// и так далее...
if (done==0) { alert("Неверный пароль или имя пользователя!"); }
}
// End -->
</script>
</head>
<body>
<center>
<form name=login>
<table width=225 border=1 cellpadding=3 bgcolor="#808080">
<tr><td colspan=2>
<p align="center"><b><font color="#FFFFFF"
size="3"></font></b></p>
</td></tr>
<tr><td>
<font size="3" color="#FFFFFF">
Имя:
</font>
</td><td>
<font size="3" color="#FFFFFF">
<input type=text name=username>
</font>
</td></tr>
<tr><td>
<font size="3" color="#FFFFFF">

```

Пароль:

```

</font>
</td><td>
<font size="3" color="#FFFFFF">
<input type=text name=password>
</font>
</td></tr>
<tr><td colspan=2 align=center>
<font size="3" color="#FFFFFF">
<input type=button value="Перейти на сайт" onClick="Login()">

</font>
</td></tr>
</table>
</form>
</center>
</body>

```

10. Использование форм в языке программирования JavaScript

В данном параграфе рассмотрим примеры использования форм в задачах выбора вариантов и передачи информации на сервер.

Задание №23

Цель работы: изучить методику решения задач выбора вариантов с использованием форм и флажков "checkbox".

Назначение программы: программа создает форму, в которой пользователю предлагается выбрать хобби. Программа анализирует ответ пользователя и выдает сообщение, каковы хобби пользователя.

Ваше хобби

- литература

- спорт

- туризм

Рис. 10 Форма выбора варианта с помощью флажков "checkbox"

Текст программы:

```

<script>
function raport() {
var num=document.myform.length;
var hob="Ваше хобби";
for (var i=0; i<num-1; i++) {
if (document.myform.hobby[i].checked==true) {
hob=hob+" " + document.myform.hobby[i].value;}
}
window.alert(hob);
}
</script>
<form name="myform">
<input type="checkbox" name="hobby" value="литература">
литература</br>
<input type="checkbox" name="hobby" value="спорт">
спорт</br>
<input type="checkbox" name="hobby" value="туризм">
туризм</br>
<input type="button" value="Отправить" onClick="raport()">
</form>

```

Задание №24

Цель работы: изучить методику передачи информации на сервер с помощью формы.

Назначение программы: программа передает на сервер следующую информацию: e-mail пользователя и сообщение. Но поскольку программы —anurprogramm.php”, которая обрабатывает информацию на сервере не существует, при нажатии кнопки «Отправить» будет выдано сообщение —Невозможно найти удалённый сервер”.

Ваш e-mail

Сообщение

Сброс

Отправить

Рис. 11 Форма отправки сообщения на сервер

Текст программы:

```

<html>
<head>
<title>Форма</title>
<style type="text/css">
form
{
width:300px;
padding:20px;
border-with:1px;
border-style:solid;
background:#e0e0e0
}
</style>
</head>
<body>
<form          action="http://anyserver.ru/anyprogramm.php"
method="post">
  <h2>Сообщение автору </h2>
  Ваш E-mail:

```

```

<input type="text" name="from" value="" size=30/>
<br/>
Сообщение: <br>

<textarea name='text'cols=30 rows=8></textarea>
<br/>
<input type="reset"/>
<input type="submit" name="Отправить" value="Отправить"/>
</form>
</body>
</html>

```

Задание №25

Цель работы: получить навыки самостоятельной разработки программ для решения задач выбора варианта с использованием форм и переключателей “radio”.

Назначение программы: программа реализует тестовый вопрос, в котором из 4 возможных вариантов ответов надо выбрать один правильный. Программа должна проанализировать ответ пользователя и выдать сообщение:

1. Ответ правильный.
2. Ответ неправильный. Вам надо прочитать теоретический материал по сетевым атакам.

Какая сетевая атака связана с превышением допустимых пределов функционирования сети?

- Подслушивание (Sniffing).
- Отказ в обслуживании (DoS –атака).
- Атака Man in – the – Middle (человек в середине).
- Угадывание ключа.

Рис. 12 Форма выбора варианта с помощью переключателей “radio”.

10. Слои в языке программирования JavaScript. Краткие теоретические положения

Для создания слоев используется элемент DIV, являющийся блочным элементом разметки. Рассмотрим свойства, указывающие способ размещения и точное позиционирование слоев. Свойство position указывает способ размещения элементов. Оно может принимать четыре значения:

- absolute – задает абсолютное свободное позиционирование;
- fixed – аналогично absolute, но при прокрутке элемент не перемещается;
- relative – определяет относительное свободное позиционирование;
- static – задает статическое позиционирование, при котором элемент страницы отображается внутри общего «потока» текста.

Свойства left, right, top и bottom определяют точное позиционирование элементов:

- left – задает горизонтальную позицию левой границы свободно позиционированного элемента;
- right – определяет горизонтальную позицию правой границы свободно позиционированного элемента;
- top – задает вертикальную позицию верхней границы свободно позиционированного элемента;
- bottom – определяет вертикальную позицию нижней границы свободно позиционированного элемента.

Например, следующий фрагмент программы выводит слой в правом нижнем углу:

```
<div id=third style="position:absolute; bottom:50px;right:50px;
Background-color:red;">
Слой в правом нижнем углу
</div>
```

Свойства height и width определяют высоту и ширину блока, определяющего слой.

Свойство visibility используется для того, чтобы делать слои видимыми и скрытыми. Оно может принимать следующие значения:

- hidden – скрывает элемент;
- collapse – если не используется с таблицами, то аналогично значению hidden;

`inherit` – значение по умолчанию, которое указывает, что видимость наследуется от родителя;

`visible` – делает элемент видимым.

Задание №26

Цель работы: изучить методику использования и позиционирования слоев.

Назначение программы: программа выводит на экран три слоя, причем третий слой является прозрачным, поэтому сквозь него виден второй слой.

Текст программы:

```
<html>
<head>
<title>Перекрывающиеся слои</title>
</head>
<body>
<h2>Перекрывающиеся слои </h2>
<div id=first style="position:absolute; top:50px;left:50px;
height:150px; width:150px; background-color:orange;">
1 слой
</div>
<div id=second style="position:absolute; top:70px; left:70px;
height:150px; width:150px; background-color:red;">
2 слой
</div>
<div id=third style="position:absolute; top:90px;left:90px;
height:150px; width:150px;">
3 слой
</div>
</body>
</html>
```

Задание №27

Цель работы: изучить методику управления видимостью слоев.

Назначение программы: программа создает невидимый слой, который можно отобразить или вновь скрыть щелчком кнопки мыши по ссылке.

Текст программы:

```
<html>
<head>
<title> Управление видимостью </title>
<script>
//Функция, осуществляющая изменение видимости слоя
function vis(){
//Получаем доступ к слою
var Layer=document.getElementById('lay');
//выясняем состояние слоя
if (Layer.style.visibility=='hidden') {
//если слой невидим, то делаем видимым
Layer.style.visibility='visible';
}
else{
//если слой видим, то делаем невидимым
Layer.style.visibility='hidden';
}
}
</script>
</head>
<body>
<h2> Упраление видимостью </h2>
<div id=control style="position:absolute; top:50px;left:50px;">
<a href="javascript:vis();">Скрыть/Показать слой </a>
</div>
<div id=lay style="position:absolute; top:70px;left:50px;
height:100px;width:250px; background-color:orange;
visibility:hidden">
```

Этот слой по умолчанию невидим. Чтобы его скрыть или увидеть,

необходимо щелкнуть кнопкой мыши на ссылке

```
</div>
```

```
</body>
</html>
```

12. Защита контента от несанкционированного копирования информации в языке программирования JavaScript. Краткие теоретические положения

Рассмотрим 3 варианта Java-скриптов, предназначенных для защиты контента сайта от несанкционированного копирования. Java-скрипты необходимо вставлять между тегами <head> и </head>, например после тега </title>.

1. Рассмотрим Java-скрипт, позволяющий скопировать текст, но после вставки текста в конце появляется ссылка на источник, например: Источник: © www.swsu.ru

```
<script type="text/javascript">
function addLink() {
var body_element = document.getElementsByTagName('body')[0];
var selection;
selection = window.getSelection();
var pagelink = "<p>Источник:
<a href='"+document.location.href+"'>"+"</a> © www.swsu.ru";
var copytext = selection + pagelink;
var newdiv = document.createElement('div');
newdiv.style.position='absolute';
newdiv.style.left='-99999px';
body_element.appendChild(newdiv);
newdiv.innerHTML = copytext;
selection.selectAllChildren(newdiv);
window.setTimeout(function() {
body_element.removeChild(newdiv);
},0);
}
document.oncopy = addLink;
</script>
```

2. Рассмотрим второй Java-скрипт, который не только запрещает выделять текст на странице, но и пользоваться правой кнопкой мышки.

```
<script language="JavaScript">
```

```
document.onselectstart=function(){return false}
document.oncontextmenu=function(){return false}
document.onmousedown=function(){return false}
</script>
```

3. Рассмотрим третий Java-скрипт, который защищает контент от несанкционированного копирования прозрачным слоем. При попытке копирования пользователь копирует прозрачный слой.

```
<html>
<head>
<title>Издательство НТ Пресс</title>
<div style="position: relative;">
//далее защищаемый прозрачной картинкой контент сайта

</div>
</head>
<body>
</body>
</html>
```

Задание №28

Цель работы: изучить методику защиты контента сайта от несанкционированного копирования

Назначение программы: программа реализует сайт, состоящий из трех страниц. Сайт на произвольную тему разработайте самостоятельно или используйте сайт из практического задания №8. Каждую страницу сайта защитите от несанкционированного копирования разными видами Java-скриптов, рассмотренными в кратких теоретических положениях.

Список контрольных вопросов

1. Какие методы ввода и вывода информации существуют в языке JavaScript? Поясните на примерах.

2. Нужно ли при объявлении переменной в языке JavaScript указывать ее тип?
3. Каким образом можно вывести значение всех элементов массива в языке JavaScript?
4. Какие операторы языка JavaScript служат для реализации механизмов ветвления. Запишите общий вид этих операторов.
5. Какие операторы языка JavaScript служат для реализации механизмов цикла. Запишите общий вид этих операторов.
6. Дайте определение функции и запишите ее общий вид в языке JavaScript.
7. Для какой цели в языке JavaScript используется оператор return?
8. Приведите примеры двух-трех событий и соответствующих им обработчиков этих событий.
9. Каким образом можно программным путем можно обратиться к полю формы и присвоить ему определенное значение? Приведите пример.
10. Каким образом программа определяет адрес URL-сценария на сервере по которому передаются данные формы?
11. Какие методы передачи данных формы на сервер существуют и в чем их различие?
12. Дайте краткую характеристику основных элементов управления, используемых в формах (поля ввода данных, поля ввода паролей, флажки, переключатели, списки)
13. С какой целью при работе с формами используются метод Reset и событие onReset?
14. Какие существуют методы защиты контента сайта от не-санкционированного копирования информации. В чем их различие?

Список литературы

1. Дунаев В. В. Сценарии для Web-сайта: PHP и JavaScript. – 2-е изд., перераб. и доп. – СПб.: БХВ – Петербург, 2008. -576 с.
2. Днепров А. Г. JavaScript на 100%. – СПб.: Питер, 2008. -304 с.

3. Колисниченко Д.Н. Современный сайт на PHP и JavaScript. - СПб.: Питер, 2009. – 176 с.
4. Дронов В.А. JavaScript и AJAX в Web-дизайне. - СПб.: БХВ-Петербург, 2008. – 728 с.
5. Прохоренок Н.А., Дронов В. А.: HTML, JavaScript, PHP и MySQL. - СПб.: БХВ-Петербург, 2015. – 766 с.
6. Гарнаев А.Ю. Web-программирование на Java и JavaScript. - СПб.: БХВ-Петербург, 2005. – 1040 с.
7. Дуванов А.А. Web-конструирование. DHTML. - СПб.: БХВ-Петербург, 2003. – 512 с.
8. Кисленко Н. П. HTML. Самое необходимое. БХВ - Петербург, 2008. – 353с.
9. Гаевский А. Ю., Романовский В. А. 100% самоучитель по созданию Web – страниц и Web-сайтов. HTML и JavaScript. - Триумф, 2008. -454 с.

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Юго-Западный государственный университет»
(ЮЗГУ)

Кафедра информационной безопасности

УТВЕРЖДАЮ
Проректор по учебной работе
О.Г. Локтионова
« 09 » 2021 г.



РАЗРАБОТКА И ЗАЩИТА WEB-ПРИЛОЖЕНИЙ С СЕРВЕРНЫМИ СЦЕНАРИЯМИ НА ЯЗЫКЕ PHP

Методические указания по выполнению лабораторных работ
для студентов укрупненной группы специальностей и
направлений подготовки 10.00.00

Курск 2021

УДК 004.56.5(076.5)

Составитель: А.Л. Ханис

Рецензент

Кандидат технических наук, доцент кафедры
информационной безопасности А.Л. Марухленко

Разработка и защита Web-приложений с серверными сценариями на языке PHP : методические указания по выполнению лабораторных работ для студентов укрупненной группы специальностей и направлений подготовки 10.00.00 / Юго-Зап. гос. ун-т; сост.: А.Л. Ханис. - Курск, 2021. - 33 с.: ил. 2, табл. 1. – Библиогр.: с. 33.

Содержат краткие теоретические положения об основных операторах языка программирования PHP, а также о реализации авторизации и регистрации пользователей с помощью PHP-скриптов.

Предназначены для студентов укрупненной группы специальностей и направлений подготовки 10.00.00.

Текст печатается в авторской редакции

Подписано в печать . Формат 60x84 1/16.
Усл.печ. л. 1,86 Уч.-изд. л. 1,68. Тираж 100 экз. Заказ. Бесплатно.
Юго-Западный государственный университет.
305040, г. Курск, ул. 50 лет Октября, 94.

ВВЕДЕНИЕ

Официальное название языка PHP – Hypertext Processor (гипертекстовый препроцессор). Он является языком сценариев, выполняющихся на сервере. Когда браузер пытается получить доступ к указанному URL, по которому расположена PHP-страница, он выполняет сначала запрос к веб-серверу, который активизирует синтаксический анализатор PHP. Затем этот анализатор выполняет PHP-код, расположенный в запрашиваемом файле, и возвращает полученный результат веб-серверу как часть HTML-кода. Данный HTML-код, в свою очередь, передается в браузер пользователя для отображения. Именно этот результат и видит пользователь, когда запрашивает PHP-страницу. Следует отметить, что при этом пользователь не имеет возможности просмотреть сам PHP-код, в отличие от JavaScript, а также не требуется наличия специальных интерпретаторов в браузере.

Согласно руководству по PHP «Цель языка – помочь веб-разработчику быстро создавать динамически генерируемые страницы».

Обычно PHP используется для следующих целей:

- выполнение системных функций – создание, открытие, чтение, запись и закрытие файлов, выполнение системных команд, создание каталогов, изменение прав доступа;
- сбор данных с помощью форм – сохранение данных в файле, отправка данных по e-mail, возврат обработанных данных пользователю;
- доступ к базам данных и генерация содержимого «на лету» (технология отображения данных в зависимости от запроса пользователя) или создание веб-интерфейса для добавления, удаления и изменения элементов в базе данных;
- запись на компьютер пользователя файлов «cookie» и доступа к переменным «cookie»;
- запуск сессий и использование переменных объектов сессии;
- проверка имени и пароля (аутентификация) пользователя для ограничения его доступа к разделам веб-сайта;
- шифрование данных;

- организация электронных платежей и др.

Следует отметить межплатформенность PHP (он способен работать на различных платформах и с различными веб-серверами). При своей многофункциональности, стабильности и надежности в работе PHP достаточно нетребователен к системным ресурсам веб-сервера. Также он не требует приобретения лицензии на использование, что делает его особенно привлекательным как для коммерческих, так и для личных целей.

Основные преимущества PHP проявляются при совместном использовании с базой данных MySQL. В этом случае вы можете раскрыть полный потенциал динамических Web-страниц и перейти к созданию системы Web-публикации, в основе которого лежит именно широкое использование баз данных. На сегодняшний день MySQL является лидером среди систем управления базами данных для разработки интернет-приложений. Область применения MySQL самая разнообразная – от создания форумов, гостевых книг, досок объявлений до интернет-магазинов и корпоративных веб-сайтов. Если веб-проект содержит много информации, то в нем не обойтись без базы данных. Именно база данных позволяет структурировать информацию и обеспечивает оперативный и бесперебойный доступ к данным. Учитывая возможности, стабильность, надежность, безопасность и распространенность MySQL, выбор в пользу этой базы данных становится очевидным.

SQL (Structured Query Language) – структурированный язык запросов. SQL создан для работы с реляционными базами данных. Он позволяет пользователям взаимодействовать с базами данных (просматривать, искать, добавлять и удалять данные). Таблицы в реляционной базе данных связаны между собой при помощи отношений. Поэтому при выполнении запроса обеспечивается возможность объединить данные из нескольких таблиц. SQL как часть системы MySQL можно охарактеризовать как язык структурированных запросов плюс наиболее распространенный язык, используемый для доступа к базам данных.

Лабораторная работа

Разработка и защита Web-приложений с серверными сценариями на языке PHP

1. Установка и настройка веб-сервера с PHP

Цель: научиться устанавливать на персональный компьютер программное обеспечение, необходимое для программирования и отладки веб-сценариев.

Задание: установить на flash-накопитель веб-сервер Apache, интерпретатор языка PHP, текстовый редактор Notepad++, веб-браузер Firefox с расширением Firebug.

Ход работы.

Для изучения дисциплины и полноценной работы необходимо установить веб-сервер Apache (<http://apache.org/>) и интерпретатор языка программирования PHP (<http://php.net/>). Установка этого ПО достаточно сложное дело для начинающих пользователей (как это делается и для чего это необходимо можно прочесть здесь <http://web.diwaxx.ru/web-server-doma.php>), поэтому мы будем использовать готовую сборку Denwer (<http://www.denwer.ru/>). Инструкция по установке находится по адресу <http://www.denwer.ru/base.html>. Устанавливать Denwer необходимо на flash-накопитель, чтобы была возможность программировать как на занятиях, так и в домашних условиях. Устанавливайте Denwer в каталог первого уровня flash-накопителя, например f:\WebServers. Под конец установки будет задан вопрос, как именно Вы собираетесь запускать и останавливать комплекс. Есть два альтернативных варианта:

1. Создавать виртуальный диск при загрузке машины (при этом, инсталлятор обеспечит, чтобы это происходило автоматически), а при остановке серверов виртуальный диск не отключать.

2. Создавать виртуальный диск только по явной команде старта комплекса. И, соответственно, отключать диск от системы — при остановке серверов.

Первый режим наиболее удобен, если комплекс

устанавливается на жесткий диск

компьютера, а не на flash-накопитель, поэтому необходимо выбрать второй вариант. На вопрос создавать ярлыки для запуска и остановки Denwer на рабочем столе отвечайте отрицательно. Для запуска комплекса будем использовать файлы Run.exe, Stop.exe и Restart.exe в каталоге X:\WebServers\denwer\, где X:\WebServers\ - диск и папка, в которую установлен комплекс.

При запуске комплекса брандмауэр операционной системы может заблокировать запуск веб-сервера Apache. В этом случае брандмауэр потребует подтверждения Ваших намерений. Щелкните на Don't Block Anymore (не блокировать в дальнейшем).

Возможно, что на том компьютере, где Вы захотите запустить Denwer с Вашего flash-накопителя, Denwer не сможет создать виртуальный диск с той буквой, которую Вы задали при установке комплекса, т.к. в системе уже будет такой диск. В этом случае необходимо поменять букву виртуального диска, изменив в файле конфигурации X:\WebServers\denwer\CONFIGURATION.txt строку:

```
subst_drive = Z:
```

где необходимо поменять букву Z на любую другую, для которой в системе нет диска. Ни в коем случае не меняйте ничего больше в файле конфигурации, это может привести к нарушению работы комплекса.

Кроме Denwer Вам также потребуется текстовый редактор (не путать с текстовым процессором, подобным MS Word), желательно с подсветкой синтаксиса языков программирования и разметки, и современный веб-браузер с инструментами отладки. В качестве текстового редактора можно использовать Notepad++ (<http://notepad-plus-plus.org/>), а в качестве веб-браузера — Firefox с расширением Firebug. Их также необходимо установить на flash-накопитель.

Для проверки работы Denwer запустите файл X:\WebServers\denwer\Run.exe и наберите в браузере <http://localhost>. Если по этому адресу откроется служебная страница

Denwer, значит все работает исправно, иначе идем на

страницу <http://www.denwer.ru/base.html> и пытаемся разобраться с настройками сети и прокси-сервером.

Для остановки Denwer выполните файл X:\WebServers\denwer\Stop.exe.

1. Вывод сообщения в языке PHP

Задания:

1. Откройте текстовый редактор (или Dreamweaver) и наберите следующий код:

Листинг № 1.

```
<html>
<head><title>Вставка кода PHP</title></head>
<body>
<h1>Пример страницы с PHP кодом</h1>
<?
print("<h2> PHP-фрагмент </h2>");
?>
</body>
</html>
```

Или

Листинг 2

```
<?
print("<html><head><title>Вставка кода
PHP</title></head><body>");
print("<h1>Пример страницы с PHP кодом</h1>");
print("<h2> PHP-фрагмент </h2>");
print("</body></html>");
?>
```

После этого выполните следующие действия:

1. Сохраните данный текстовый файл в каталоге **C:\WebServers\home\localhost\www\xp** (каталог **xp** необходимо создать) под именем **php_start1.php** (обратите внимание, расширение у файла .php).

2. Запустите web-сервер **Apache** и в строке адреса браузера наберите **http://localhost/xp**. Вы должны увидеть внутри виртуального каталога **xp** свой файл **php_start.php**.

3. Щелкните на нем и, если вы правильно набрали приведенный код, у вас должна загрузиться страница (рис 1.).

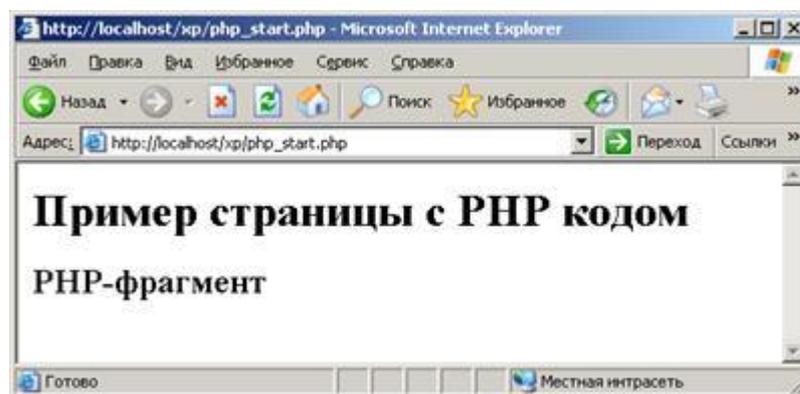


Рис. 1 – Web-страница программы «вывод сообщения»

4. Посмотрите html-код данной страницы (меню Вид → Просмотр HTML кода).

1. Передача переменных

1. Создайте новый html-файл (forma.html), содержащий следующий код (листинг 12):

Листинг №3.

```
<html>
<head>
<title>ВВОД значений в форму</title>
</head>
<body>
<form action="age.php" method="get">
```

```

    <p>ваше имя: <input name="user_name" size="20"
type="text"></p>
    <p>год рождения: <input name="user_yare" size="20"
type="text"></p>
    <input name="b1" type="submit" value="отправить"><input
name="b2" type="reset" value="очистить">
    </form>
    </body>
    </html>

```

2. Создайте файл **age.php** и наберите в нем код, представленный в листинге №4.

Листинг №4.

```

<html>
<head>
<title>вычисление возраста</title>
</head>
<body>
<p>Добро пожаловать <? echo ($user_name) ?></p>
<?
$yare = date("Y");
$user_age = $yare - $user_yare;
print ("<p>вам $user_age лет</p>");
?>
</body>
</html>

```

echo(); - функция отвечающая за вывод значений переменных, во многом, аналогична функции **print()**.

date(); - функция возвращающая текущую дату в виде строки. Функция имеет большое количество параметров.

Параметры функции date():

G - час, 24-часовой формат без ведущих нулей; т.е. от «0» до «23»

i - минуты; т.е. от «00" до «59"
j - день (число) месяца без ведущих нулей; т.е. от «1" до «31"
m - месяц; т.е. от «01" до «12"
H - час, 24-часовой формат; т.е. от «00" до «23"
n - месяц без ведущих нулей; т.е. от «1" до «12"
s - секунды; т.е. от «00" до «59"
Y - год, 4 цифры; например, «1999"
y - год, 2 цифры; например, «99"
z - день года; т.е. от «0" до «365"

пример использования функции: date()

\$today = date("j, n, Y"); переменная \$today примет значение: 10, 3, 2001 (число, месяц, год)

\$today = date("H:i:s"); переменная \$today примет значение: 17:16:54 (часы, минуты, секунды)

3. Введем имя пользователя, например, **Dik**, а год его рождения **1973**, нажав кнопку «отправить», данные будут переданы файлу **age.php**, и строка адреса примет вид:

http://localhost/xp/age.php?user_name=Dik&user_yare=1973&b1=%EE%F2%EF%F0%E0%E2%E8%F2%FC

– все передаваемые данные располагаются за символом «?»

– все данные собраны в виде: имя переменной = значение переменной;

– переменная b1 имеет значение «отправить», текст, содержащий кириллицу. Для таких переменных браузер автоматически выполняет URL-кодирование.

1. Самостоятельно создайте форму, в которой вводится имя студента, и его год его рождения. Данные из формы должны передаваться в php-файл, который определяет, на каком курсе учится человек (предполагая, что студент поступил в колледж в 16 лет).

2. Алгоритмические конструкции на PHP

Применение **условного оператора** аналогично использованию в других языках программирования. Синтаксис полностью идентичен языку Си.

```
if ($a > $b) {
    print «а больше чем b»;
} else {
    print «а не больше чем b»;
}
```

Возможно использование сокращенного условного оператора без оператора else, также возможно любое вложение условных операторов:

```
if ($a > $b) {
    print «а больше чем b»;
} elseif ($a == $b) {
    print «а равно b»;
} else {
    print «а меньше чем b»;
}
```

В качестве условия могут выступать стандартные логические операции:

- > (больше)
- < (меньше)
- == (проверка равенства)
- != (не равно)

Также возможны логические связки:

логическое сложение - **\$a || \$b** (возможно **\$a or \$b**).
Выражение будет истинным если хотя бы одно логическое выражение истинно.

логическое умножение - **\$a && \$b** (возможно **\$a and \$b**).
Выражение будет истинным только если оба логических выражения истинны.

логическое умножение - ! \$a. Истинность выражения принимает противоположное значение.

В языке PHP существует функция **isset()**, которая очень активно применяется вместе с условным оператором. Ее назначение проверять наличие переменной. Например, мы создаем страницу, которая проверяет текущую дату с днем рождения зарегистрированного пользователя (которое хранится в переменной \$data_r) и в случае совпадения, поздравляет его с праздником. Но выполнять операцию сравнения можно лишь в случае, если в данную страницу передана переменная \$data_r.

1. Создайте файл data_r.php и наберите в нем следующий программный код (листинг5):

Листинг №.5

```
<html>
<head><title>Проверка условия в языке PHP</title></head>
<body>
<H1>проверка даты рождения</H1>
<form action="data_r.php" method="get">
<P>месяц рождения: <input name="m" size="5"
type="text"></P>
<P>день рождения: <input name="d" size="5" type="text"></P>
<P><input name="b" type="submit" value="проверить"></P>
</form>
<?
if(isset($b)){
$day = date("j");
$month = date("n");
if ($m == $month && $d == $day){
print("<h2>С днем рождения</h2>");
}
}
?>
</body>
```

</html>

3. Циклические конструкции

Смысл циклических конструкций в языке PHP такой же, как и в других языках программирования. Синтаксис полностью идентичен языку Си.

Цикл «Пока»

```
$i = 1;
while ($i <= 10) {
    print $i++;
}
```

Цикл «С параметром»

```
for ($i = 1; $i <= 10; $i++) {
    print $i;
}
```

4. Практические задания:

1. Для изучения условного оператора создайте тест, состоящий из пяти вопросов (файл **test.htm**). После выбора правильных ответов, данные передаются в новый файл **analyse_test.php**, где вычисляется количество правильных ответов и выводится соответствующее сообщение. Обратите внимание, при ответе пользователь мог специально или случайно пропустить вопрос, поэтому перед проверкой каждого ответа на правильность нужно проверить, а передана ли соответствующая переменная в php-файл.

2. Создайте PHP-страницу, в которой пользователь вводит в форму количество строк и количество столбцов. В результате по введенным значениям строится таблица. Примерный вид экрана представлен на рисунке 2.

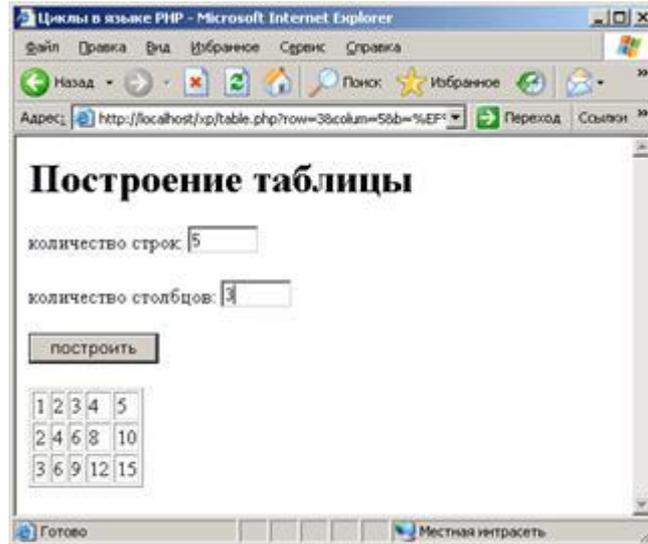


Рис. 2 – Web-страница программы «построение таблицы»

Работа с сессиями. Реальная авторизация и регистрация.

Цель занятия: научиться работать с сессиями.

Задание:

Создание авторизации и регистрации.

Ход работы.

Веб-сервер не поддерживает постоянного соединения с клиентом, и каждый запрос обрабатывается, как новый, безо всякой связи с предыдущими. То есть, нельзя ни отследить запросы от одного и того же посетителя, ни сохранить для него переменные между просмотрами отдельных страниц. Для решения этих двух задач и были изобретены сессии.

Собственно, сессии, если в двух словах - это механизм, позволяющий однозначно идентифицировать браузер и создающий для этого браузера файл на сервере, в котором хранятся переменные сеанса.

Теперь создайте файл `registration.php`, который будет содержать форму регистрации пользователя.

```

<form method="POST" action="registration.php">
  <p>Имя пользователя: <input type="text"
name="username"></p>
  <p>Email: <input type="text" name="email"></p>
  <p>Пароль: <input type="password" name="password"></p>
  <p>Пароль(повторить): <input type="password"
name="password2"></p>
  <p><input type="submit"></p>
</form>

```

В самом начале файла необходимо вызвать функцию `session_start()`, она создает сессию или продолжает текущую на основе текущего идентификатора сессии, который передается через запросы, такие как GET, POST или cookie. В большинстве случаев используют сессии на cookie, поэтому перед функцией `start_session()` не должно быть функций, возвращающих сообщение в браузер. Затем делаем проверку, аутентифицирован ли пользователь.

```

if ($_SESSION['username']) {

    echo 'Вы уже зарегистрированы';
    return 0;
}

```

Далее проверяем, существует ли POST запрос, если да, то проверяем совпадают ли пароли и заносим в переменную сессии данные из POST запроса.

```

if ($_POST) {

    if ($_POST['password'] == $_POST['password2']) {
        $_SESSION['username'] = $_POST['username'];
        echo 'Пользователь ' . $_SESSION['username'] . ' зарегистрирован';
        return 0;
    } else {
        echo 'Введенные пароли не совпадают';
    }
}

```

```
return 0;
}
}
```

В index.php заменяем все POST на SESSION.

Теперь необходимо создать файл logout.php. Этот файл будет содержать функцию, которая разрушит все данные, зарегистрированные в сессии – session_destroy().

```
<?php
session_start();

session_destroy();
header('Location: index.php');
?>
```

Перед тем как отправить форму, ее нужно проверить. Проверка формы будет осуществляться посредством javascript. Для этого у формы определим событие onsubmit="return checkForm(this)". Функция checkForm будет вызываться перед отправкой данных. Проверять будем имя пользователя, заполнено он или нет и email.

Шаблон email`а будет [английские_буквы]@[английские_буквы].[английские_буквы]. Такую проверку можно сделать, используя регулярные выражения - это формальный язык поиска и осуществления манипуляций с подстроками в тексте, основанный на использовании. По сути это строка-образец, состоящая из символов и метасимволов и задающая правило поиска. Более подробно

о регулярных выражениях можно почитать на Википедии (http://ru.wikipedia.org/wiki/Регулярные_выражения).

Вообще регулярное выражение для проверки электронной почты довольно громоздкое и сложное для понимания, поэтому воспользуемся простой формой [a-zA-Z]*@[a-zA-Z]*\.[a-zA-Z].

В начале функции определим все переменные, которые нам понадобятся, и массивы с ошибками.

```
var el, // Сам элемент
    elName, // Имя элемента формы
    value, // Значение
    type; // Атрибут type для input-ов
reg = /[a-zA-Z]*@[a-zA-Z]*\.[a-zA-Z]/;
var errorList = [];

var errorText = {
    1 : "Не заполнено поле 'Имя'",
    2 : "Не заполнено поле 'E-mail'",
}
```

Далее проходим по всем элементам формы и проверяем все теги input. Если поля для имени пустое или электронная почта не соответствует шаблону, записываем номера ошибок в массив.

```
for (var i = 0; i < form.elements.length; i++) { el =
form.elements[i];
    elName = el.nodeName.toLowerCase();

    value = el.value;
    if (elName == "input") {
        type = el.type.toLowerCase();
        switch (type) {
            case "text" :

                if (el.name == "name" && value == "")
                    errorList.push(1);
                if (el.name == "email" && !value.match(reg))
                    errorList.push(2);
                break;

            default :
```

```
break;
}
}
}
```

Затем проверяем массив с ошибками. Если он пуст, то возвращаем true, иначе формируем текст для ошибки, выводим это на экран и возвращаем false.

```
if (!errorList.length) return true;

var errorMsg = "При заполнении формы допущены следующие
ошибки:\n\n"; for (i = 0; i < errorList.length; i++) {
  errorMsg += errorText[errorList[i]] + "\n";
}
alert(errorMsg);
return false;
```

Чтение и запись в файл. Регистрация с записью в файл. Авторизация из файла.

Цель занятия: научиться работать файлами.

Задание:

Создание авторизации и регистрации на файлах.

Ход работы.

Для работы с файлами в php существует несколько функций.

fopen – функция для открытия файла.

```
$fp = fopen('filename', 'param');
```

filename и param это обязательные параметры. Первый отвечает за имя файла, который необходимо открыть, а второй определяет режим файла:

1. r – открытие файла только для чтения.

2. r+ - открытие файла одновременно на чтение и запись.

3. w – создание нового пустого файла. Если на момент вызова уже существует такой файл, то он уничтожается.

4. w+ - аналогичен r+, только если на момент вызова файл такой существует, его содержимое удаляется.

5. a – открывает существующий файл в режиме записи, при этом указатель сдвигается на последний байт файла (на конец файла).

6. a+ - открывает файл в режиме чтения и записи при этом указатель сдвигается на последний байт файла (на конец файла). Содержимое файла не удаляется.

Записывать данные в файл при помощи PHP можно при помощи функции `fwrite()`. Это функция принимает 2 обязательных параметра и 1 необязательный. В качестве обязательных параметров выступает дескриптор файла и режим файла:

```
$test = fwrite($fp, $mytext);
```

По завершению работы с файлом, его нужно закрыть, используя функцию `fclose($fp)`.

Теперь в файле `registration.php` после строки `if ($_POST) {` поставим проверку файла, если он существует, то открываем его и перемещаем указатель в конец строки, если нет, то создаем его.

```
$fp = fopen('users.txt', 'a+');
if (!$fp) {
    $fp = fopen('users.txt', 'w+');
}
```

В условие проверки паролей сформируем строку с данными, которые будут разделены знаком `&`, запишем ее в файл и закроем его.

```
$mytext =
```

```
'username='.$_POST['username'].'&password='.$_POST['password']
.'&email='.$_POST['email']. "\r\n";
fwrite($fp, $mytext);
fclose($fp);
```

Следующим шагом будет преобразование файла login.php. Теперь мы будем получать данные о пользователях не в самом сценарии, а из отдельного файла. Для этого в самом начале файла снова начнем сессию, подключим файл debug.php для отладки и делаем проверку, если существует пост запрос, то открываем файл user.txt, построчно считываем его содержимое и сравниваем с данными POST запроса. В случае успеха, добавляем пользователя в сессию и переходим на index.php.

```
<?php

session_start();
require_once('debug.php');
if ($_POST) {

    $fp = fopen('users.txt', 'r');
    while (!feof ($fp)) {
        $buffer = fgets($fp);

        preg_match('/username=([^&]*)&/', $buffer, $user);
        preg_match('/password=([^&]*)&/', $buffer, $pass);

        if ($_POST['username'] == $user[1] && $_POST['password'] ==
        $pass[1]) {

            $_SESSION['username'] = $_POST['username']; header('Location:
            index.php');
        }

    }
    fclose ($fp);
```

```

} else { ?> <html>
<head>

<title>Login page</title> </head>

<body>
<h3>Войти</h3>

<form method="POST" action="">
<p>Имя пользователя: <input type="text"
name="username"></p>
<p>Пароль: <input type="password" name="password"></p>

<p><input type="submit"></p> </form>

<?php
}
?>
</body>
</html>

```

Гостевая книга на файлах.

Цель занятия: создание гостевой книги на файлах.

Задание:

Создать гостевую книгу.

Ход работы.

Первоначально нужно определить функционал гостевой книги, т.е. как она будет работать. Сообщения могут оставлять только зарегистрированные пользователи, а пользователи – гости будут видеть только сами комментарии и надпись, что нужно зарегистрироваться. Гостевая книга будет находиться в файле index.php. Все данные будут записываться в файл guestbook.txt.

Для создание гостевой книги необходима форма, которая будет добавлять сообщения, поэтому разместим ее в файле после поля для логина.

```
<form method="POST">
<p> Тема поста: <input type="text" name="theme"></p>
<p>Текст поста:<textarea rows="10" cols="45"
name="text"></textarea></p>
<p><input type="submit"></p>
</form>
```

Затем запрос этой формы необходимо обработать. Поместим в начало файла после запуска сессии проверку, если существует POST запрос с параметрами theme, text

и пользователь находится в сессии, то открываем или создаем файл guestbook.txt, записываем туда данные, закрываем файл и создаем переменную с сообщением "Комментарий был успешно добавлен".

```
if      ($_POST['theme']      &&      $_POST['text']      &&
$_SESSION['username']) { $fp = fopen('guestbook.txt', 'a+');

if (!$fp) {
$fp = fopen('guestbook.txt', 'w+');
}

$mytext
'username='.$_SESSION['username'].'&text='.$_POST['text'].'&theme='
.$_POST[' theme'].'"&". "\r\n";

$stest = fwrite($fp, stripslashes($mytext)); fclose($fp);

$msg = "Комментарий был успешно добавлен";
}
```

Перед формой разместим условие, если \$msg существует, вывести его на экран, и если пользователь не в сессии то вместо формы выводим «Чтобы оставлять комментарии вы должны быть зарегистрированы».

```
<?php
if ($msg) {

?><p><?php echo $msg; ?></p><?php
}
if ($_SESSION['username']) {
?>
<form method="POST">

<p> Тема поста: <input type="text" name="theme"></p>
<p>Текст поста:<textarea rows="10" cols="45"
name="text"></textarea></p>
<p><input type="submit"></p>
</form>

<?php
} else {
?><p>Чтобы оставлять комментарии вы должны быть
зарегистрированы</p>
<?php
}

?>
```

Далее следуют сами комментарии. Выводить их будем в таблице вида

Комментарии		Таблица 1.
Имя пользователя	Заголовок	
Текст сообщения		

Для этого открываем файл guestbook.txt с параметром “r”, затем при помощи функций feof() и fgets() считываем построчно файл. feof() – проверяет, достигнут ли конец файла, fgets() –

возвращает строку из файла. Затем при помощи регулярных выражений вытаскиваем имя пользователя, заголовок, текст сообщения и заносим данные в таблицу.

```

<table class="table">
<?php

$fp = fopen('guestbook.txt', 'r');
while (!feof ($fp)) {
?>

<tr class="main">
<?php
$buffer = fgets($fp);

preg_match('/username=([^&]*)&/', $buffer, $user);
preg_match('/text=([^&]*)&/', $buffer, $text);
preg_match('/theme=([^&]*)&/', $buffer, $theme);
?>

<td>
<?php echo $user[1];?>
</td>

<td>
<?php echo $theme[1];?>
</td>
</tr>
<tr class="text">

<td colspan=2>
<?php echo $text[1]; ?>
</td>
</tr>
<?php
}

```

```
fclose ($fp);
?>
</table>
```

В принципе гостевая книга готова, но чтобы было удобнее работать, ее нужно оформить при помощи css. Объединим поля для регистрации и входа в один div с id=reg.

```
<div class="reg">

  <a href="registration.php">Регистрация</a>      <a
href="login.php">Войти</a>

  <p>Привет,
  <?php
  if ($_SESSION['username']) {
  echo $_SESSION['username'];
  ?>
  <a href="logout.php">Выйти</a>
  <?php

  } else {
  echo 'Гость';
  }
  ?>
  </p>

</div>
```

Создадим и подключим файл guestbook.css. Пример файла guestbook.css:

```
table {

width: 300px;

}
```

```
.main{
font-size: 120%;

font-family: Verdana, Arial, Helvetica, sans-serif;

color: #336;
border: 10px solid #666;

}
.text{
color: red;
border: 1px solid #666;
background: #eee;

padding: 5px;
}
.reg{
position: absolute;
right: 10px;

top: 10px;
width: 225px;
height: 180px;
background: #f0f0f0;
}
```

Перенос функционала с файлов на СУБД.

Цель занятия: научиться работать с базами данных.

Задание: перенос системы на файлах в базу данных.

Ход работы.

Система управления базами данных (СУБД) — совокупность программных и лингвистических средств общего или специального назначения, обеспечивающих управление созданием и использованием баз данных .

В состав денвера входит СУБД под названием phpmysqladmin. Чтобы начать работать с ней, нужно запустить денвер и в адресной строке набрать <http://localhost/Tools/phpMyAdmin/>. Изучите работу с phpmysqladmin.

Для работы с базой данных используйте расширение MySQLi. MySQLi является улучшенной версией старого драйвера PHP MySQL, предлагающего различные улучшения.

В файле для регистрации сразу после строчки с началом сессии подключитесь к базе данных. Это делается при помощи функции `mysqli_connect()`. Она создает соединение с MySQL сервером.

```
$link=
    mysqli_connect(
    'localhost',      /* Хост */
    'root',          /* пользователь */
    "",
    "");             /* База запросов*/
```

Создайте проверку на подключения к базе данных.

```
if (!$link) {

    printf("Невозможно подключиться к базе данных. Код ошибки:
    %s\n", mysqli_connect_error());
    exit;
}
```

Используя функцию `mysqli_query($link, «sql запрос»)` можно отправлять различные запросы в базу данных. Для этого используется язык запросов SQL. С синтаксисом языка можно ознакомиться на википедии. Создайте базу данных, если она не существует.

```
mysqli_query($link, 'CREATE DATABASE IF NOT EXISTS
my_db');
```

Теперь нужно подключиться к нужной базе. Это можно сделать функцией `mysqli_select_db()`.

```
mysqli_select_db($link,'my_db');
```

Удалите все функции для работы с файлами. После проверки паролей создайте таблицу с полями id, username, email, password с проверкой существования таблицы.

```
$query = "CREATE TABLE IF NOT EXISTS users (`id` INT( 11 ) NOT NULL AUTO_INCREMENT ,`username` VARCHAR( 150 ) NOT NULL ,`email` VARCHAR( 50 ) NOT NULL ,`password` VARCHAR( 50 ) NOT NULL ,PRIMARY KEY ( `id` ));";
```

```
mysqli_query($link, $query);
```

Вставьте в базу данные, которые пришли из POST запроса.

```
mysqli_query($link, 'INSERT INTO users(`username`, `email`, `password`) VALUES ("'.$_POST['username'].'", "'.$_POST['email'].'", "'.$_POST['password'].'")');
```

Если какой-то запрос не выполняется, всегда можно посмотреть в чем ошибка, используя функцию `mysqli_error($link)`. Она хранит последнюю ошибку.

В файле `index.php` после запуска сессии подключитесь к базе и попытайтесь создать базу данных как и в `registration.php`. Создаем таблицу с полями id, id_username, title, text. id_username хранит в себе id пользователя из таблицы users. После проверки POST запроса уберите все функции для работы с файлами и сделайте запрос в базу для получения id пользователя, присвойте его переменной \$id. При выполнении запроса SELECT, `mysqli_query` возвращает массив, который нужно обработать функцией `mysqli_fetch_assoc()`, на выходе которой получается ассоциативный массив.

```
$result = mysqli_query($link, 'SELECT id FROM `users` WHERE username="'.$_SESSION['username'].'"');
```

```
while($row = mysqli_fetch_assoc($result) ){ $id = $row['id'];
```

```
}
```

Измените код для генерации столбцов для комментариев. Для этого получите все данные из таблицы messages и по id_username получите имя пользователя из таблицы users.

```
<?php

$result = mysqli_query($link, 'SELECT * FROM `messages`');
while($row= mysqli_fetch_assoc($result) ){
    $id = $row['id_username'];
    $text = $row['text'];
    $theme = $row['title'];
    $username_results = mysqli_query($link, 'SELECT username
FROM `users` WHERE id="'. $id. "'");
    while($u_row=mysqli_fetch_assoc($username_results))
        { $username = $u_row['username'];
        }
    ?>
    <tr class="main">
    <td>

    <?php
    echo $username;
    ?>
    </td>
    <td>

    <?php
    echo $theme;
    ?>
    </td>
    </tr>
    <tr class="text">
    <td colspan=2>
    <?php
    echo $text;
```

```
?>
</td>
</tr>
<?php
}
?>
```

Список контрольных вопросов

1. В чем отличие php-страницы от html-страницы?
2. Какие типы переменных поддерживает язык PHP?
3. Как передать переменную в php-страницу?
4. Какие параметры существуют у функции date()?
5. Для чего используется функция isset()?
6. Каким образом происходит инициализация массивов в языке PHP?
7. Каким образом в языке PHP происходит обращение к элементам массивов и ассоциативных массивов?
8. Какие функции в языке PHP служат для добавления и удаления элементов массива?
9. Какие функции в языке PHP служат для преобразования массива в строку?
10. Какие функции для работы со строками существуют в языке PHP и каково их назначение?
11. С какой целью используются регулярные выражения в языке PHP? Приведите типовой пример задачи, где могут быть использованы регулярные выражения?
12. Объясните назначение и принцип работы оператора foreach в PHP.
13. С какой целью используется функция setcookie в языке PHP?
14. Рассмотрите синтаксис и назначение функций для открытия файла, чтения из файла, записи в файл, закрытия файла.
15. Каким образом можно установить указатель на начало файла в языке PHP?

16. Рассмотрите синтаксис и назначение функций для копирования файлов, переименования файлов, удаления файлов, определения размера файлов в PHP.

17. Объясните принцип аутентификации на основе механизма сессий?

18. Каким образом можно создать водяной знак с помощью языка PHP?

19. Рассмотрите синтаксис и назначение функций для создания нового изображения, работы с цветом, рисования линий и фигур в языке PHP.

20. Какие типы данных используются в СУБД MySQL?

21. Каким образом можно создать базу данных и таблицу в базе данных в СУБД MySQL?

22. Каким образом можно создать пользователя и назначить ему полномочия в СУБД MySQL?

23. Рассмотрите синтаксис и назначение команд, предназначенных для добавления данных в таблицу, обновления записей, удаления записей из таблицы в MySQL.

24. Как изменить структуру таблицы в MySQL?

25. Каким образом можно выбрать данные из нескольких таблиц в MySQL?

26. Объясните, каким образом можно осуществить доступ к базе данных в MySQL из PHP? Какие функции используются для установки соединения и выбора базы данных?

27. Объясните, каким образом происходит передача запросов к базе данных MySQL из PHP? Какая функция для этого используется?

28. Объясните, каким образом происходит обработка данных из базы MySQL в PHP-сценарии?

Список литературы

1. Дунаев В. В. Сценарии для Web-сайта: PHP и JavaScript. – 2-е изд., перераб. и доп. – СПб.: БХВ – Петербург, 2008. -576 с.
2. Колисниченко Д.Н. Современный сайт на PHP и JavaScript. - СПб.: Питер, 2009. – 176 с.
3. Дэвид Скляр. Изучаем PHP 7. Руководство по созданию интерактивных веб-сайтов. - Диалектика, 2017 год.- 464 стр.
4. Скляр Д., Трахтенбург А. PHP. Рецепты программирования. 3-е изд.- СПб.: Питер., 2015-784с.
5. Строганов А. С. Ваш первый сайт с использованием PHP-скриптов. – 2-е изд., испр. и дополн. – М.: Издательство Диалог-МИФИ, 2010. -288 с.
6. Дамашке Г. PHP и MySQL / Гизберт Дамашке: пер. с нем. В. Султанова. – М.: НТ Пресс, 2008. -314 с.
7. Робин Никсон. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. 4-е изд.- СПб.: Питер., 2016-768 с.
8. Прохоренок Н. А. HTML, JavaScript, PHP и MySQL. Джентменский набор Web-мастера – 4-е изд., перераб. и доп. / Н. А. Прохоренок, В. А. Дронов. – СПб.: БХВ – Петербург, 2015. -768 с.

УДК 004.56.5(076.5)

Составитель: А.Л. Ханис

Рецензент

Кандидат технических наук, доцент кафедры
информационной безопасности А.Г. Спеваков

Менеджер паролей: программа Password Commander :
методические указания по выполнению работ
для студентов укрупненной группы специальностей и направлений
подготовки 10.00.00 / Юго-Зап. гос. ун-т; сост.: А.Л. Ханис. -
Курск, 2021. - 16 с.: ил. 8, Библиогр.: с. 16.

Содержат краткие теоретические положения о методике хранения, генерации и шифрования паролей, автоматического заполнения форм Web-страниц с помощью программы Password Commander.

Предназначены для студентов укрупненной группы специальностей и направлений подготовки 10.00.00.

Текст печатается в авторской редакции

Подписано в печать . Формат 60x84 1/16.
Усл.печ. л. 0,93 Уч.-изд. л. 0,84. Тираж 100 экз. Заказ. Бесплатно.
Юго-Западный государственный университет.
305040, г. Курск, ул. 50 лет Октября, 94.

Менеджер паролей: программа Password Commander

Введение

Обычно за годы работы за компьютером у пользователей скапливается множество логинов и паролей (от электронных почтовых ящиков, интернет-магазинов, форумов, рассылок и т.д.). Конечно, можно просто все пароли выписывать на бумагу (тетрадку, блокнот и т. п.) или в обычный текстовый файл, а затем его хранить например в защищённом контейнере программы TrueCrypt. Но это гораздо менее удобно, а самое главное – менее безопасно по сравнению со специализированной программой для хранения паролей. Наиболее популярными программами "хранителями паролей" являются:

KeePass(<http://keepass.info>),
PasswordSafe(<http://passwordsafe.sourceforge.net>),
Password Commander
(<http://www.passwordcommander.com> и <http://pascom.ru>).

Все три программы бесплатны. Password Commander создана отечественными разработчиками, поэтому изначально имеет русский интерфейс. Для KeePass на её домашней странице можно скачать, а затем установить русский интерфейс и русскую справку помощи. Для программы PasswordSafe можно скачать файл помощи на русском языке.

Password Commander не только позволяет хранить и управлять паролями и логинами, но и имеет встроенный генератор случайных паролей, чтобы в случае необходимости вам не надо было их придумывать самому. Кроме того, Password Commander имеет очень удобную функцию автоматического заполнения веб-форм в браузере и различных текстовых полей.

Скачать Password Commander с русским интерфейсом можно со страницы <http://pascom.ru>. Для жителей бывшего СССР Password Commander распространяется бесплатно, для остальных пользователей предоставляется бесплатно только облегчённая (Lite) версия; полнофункциональный вариант (Professional edition) стоит \$37.95 (€29.95).

Краткие теоретические положения

Создание аккаунта

В Password Commander одновременно и независимо друг от друга могут хранить свои пароли сразу множество пользователей. Для этого каждый пользователь должен завести свой аккаунт в программе. Аккаунт – это база, где хранятся все пароли пользователя, доступ в которую возможен только после авторизации (рис. 1). Программа сразу после установки имеет один аккаунт под названием **Пример (Пароль = 123)**. Это учебный аккаунт, который содержит пример хранения логинов и паролей; если вы введёте пароль 123, то попадаете в него.



Рис. 1 Окно авторизации Password Commander

Для защиты от программ-кейлогеров Password Commander предоставляет возможность набора паролей на экранной клавиатуре.

Вам следует создать свой аккаунт. Для этого необходимо запустить Мастер создания нового аккаунта. Сделать это можно либо из окна авторизации (**Действия...→Создать новый аккаунт...**), либо из главного окна программы (**Файл →Создать новый аккаунт...**). Мастер создания нового аккаунта пошагово

поможет вам создать новый аккаунт. Если по ходу работы с мастером будет что-то не понятно, то всегда можно нажать кнопку **Справка** и почитать подробности.

В главном окне созданного аккаунта (рис. 2) можно создавать записи, однако записи должны обязательно находиться в какой-нибудь группе. Поэтому перед созданием первой записи нужно создать хотя бы одну группу. Для этого нужно вызвать **Редактор групп (Правка – Добавить группу...)**.

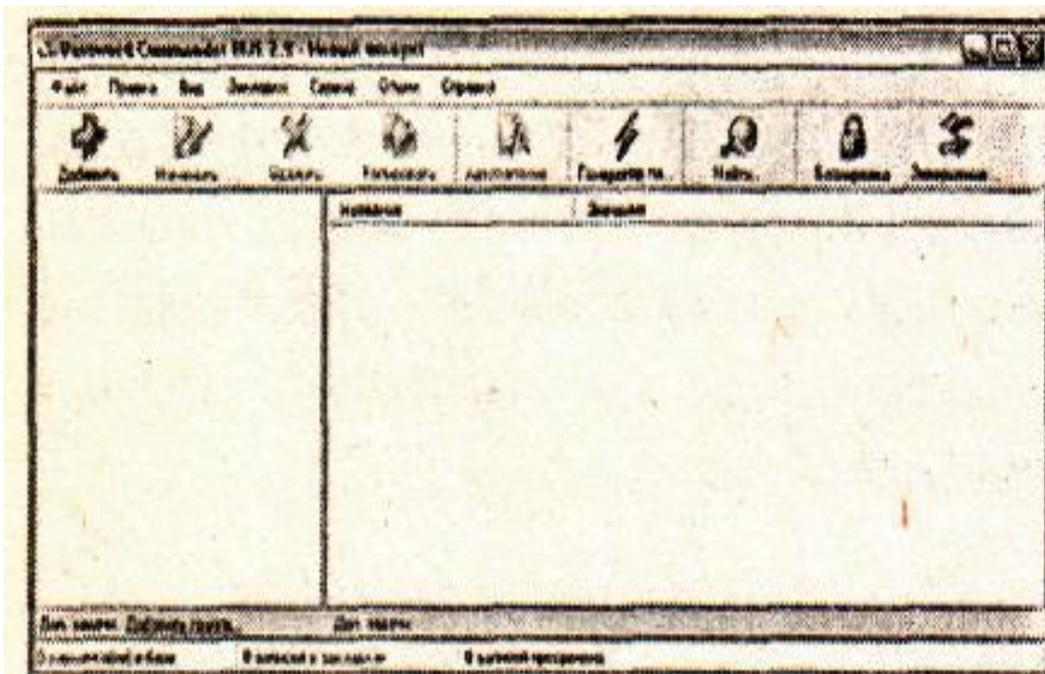


Рис.2. Главное окно созданного аккаунта

По умолчанию группа содержит два поля: **Login** и **Password** типов **Логин** и **Пароль** соответственно. Для большинства случаев этого достаточно, и после ввода названия группы можно просто нажать **ОК**, но Password Commander предоставляет возможность добавлять новые поля разных типов. Например, для почтового ящика можно добавить два дополнительных поля: **Адрес pop3 сервера** и **Адрес smtp сервера**, оба типа **Текст**. Для добавления полей существует кнопка **Добавить** в редакторе групп. В качестве примера создадим группу под именем "рассылка" (рис. 3).

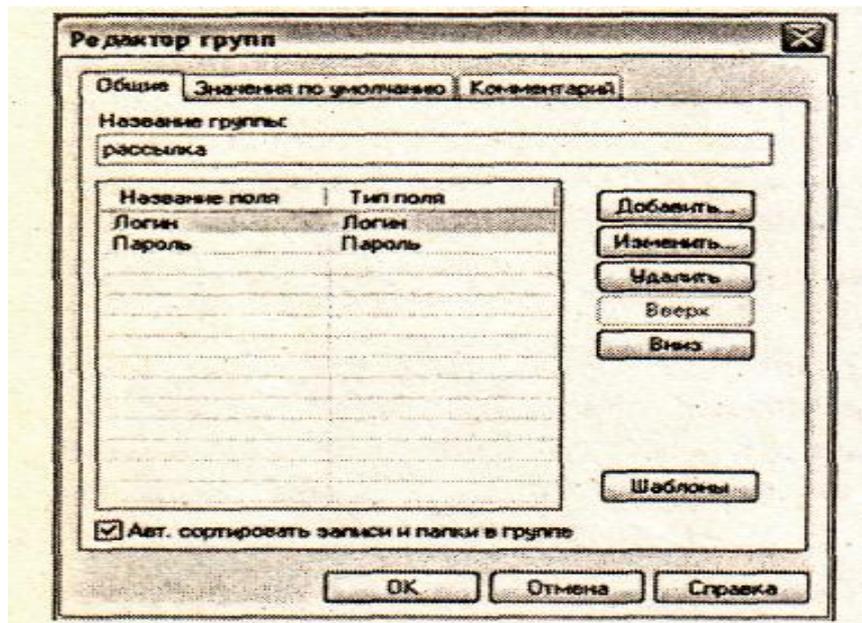


Рис. 3. Создание группы

Теперь в созданной группе можно создавать записи. Для этого нужно кликнуть по группе правой кнопкой мыши и выбрать в раскрывающемся меню пункт **Добавить запись...**. В появившемся **Редакторе записей** следует ввести данные в предложенные поля, затем нажать кнопку **ОК**. На рис. 4 показан пример создания записи под именем Subscribe.Ru.

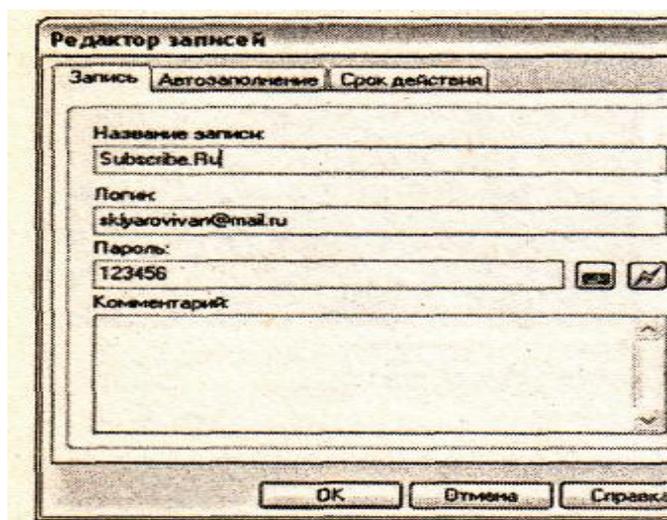


Рис 4. Создание записи

Записи в группе можно объединять в папки для облегчения навигации. Для этого нужно кликнуть по группе правой кнопкой мыши и выбрать в раскрывшемся меню пункт **Добавить подпапку**.

В появившемся диалоге следует ввести название новой папки. Далее можно перетаскивать мышью записи в нужную папку.

Все созданные элементы (группа папка, запись) вы можете изменять (удалять, перемещать, переименовывать и т.п.), для этого также имеются соответствующие пункты меню. На рис. 5 в качестве примера показаны записи произвольного аккаунта, также не забывайте обращаться за примерами к аккаунту **Пример (Пароль = 123)**.

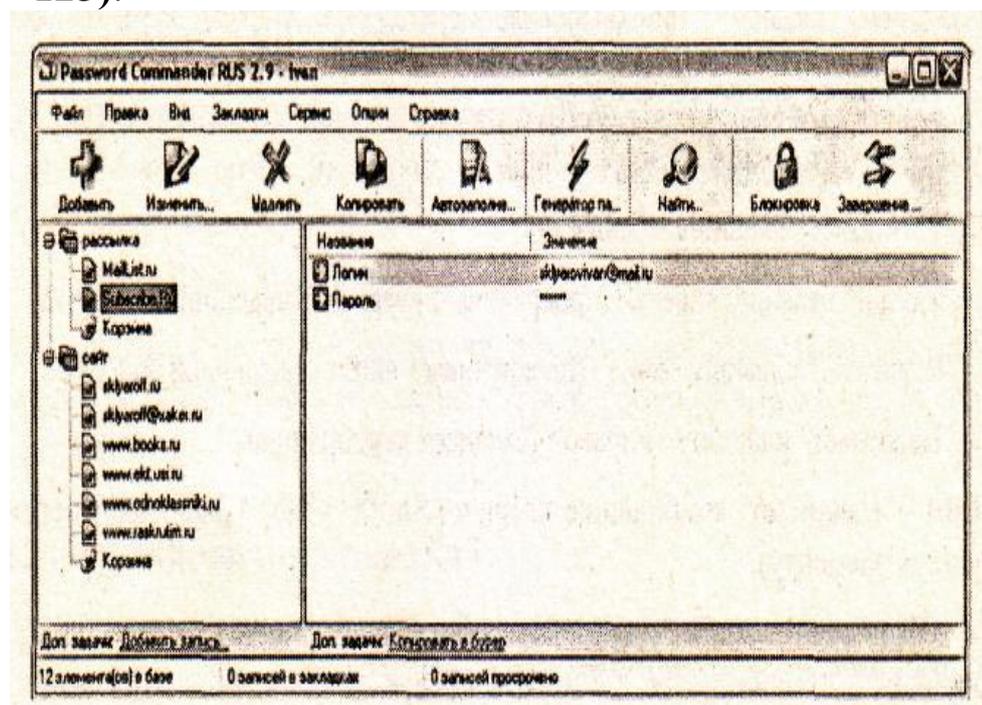


Рис 5. Записи произвольного аккаунта

Автозаполнение

Если вы хотите, чтобы Password Commander выполнял автозаполнение веб-форм и других текстовых полей, то необходимо сделать дополнительные настройки. С помощью функции автозаполнения вы можете вставить пароль (или другую информацию) в нужные поля всего одним щелчком мыши или нажатием "горячей клавиши".

Для этого надо в **Редакторе записей** перейти на вкладку **Автозаполнение** и нажать кнопку **Добавить**, затем, если вы собираетесь заполнять веб-формы на сайте, то следует установить переключатель на **Автозаполнение в веб-страницу**. В поле **URL** следует ввести адрес страницы, на которой расположены нужные поля для заполнения (веб-форма) (рис. 6)

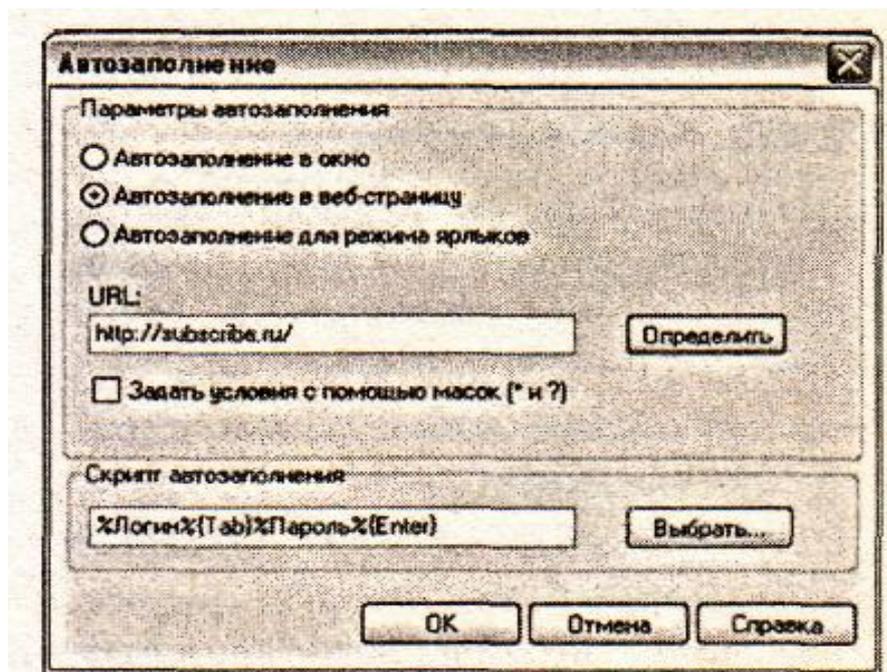


Рис. 6. Пример настройки автозаполнения в веб-страницу

В поле **Скрипт автозаполнения** вы можете видеть такую строку **%Логин%{Tab}%Пароль%{Enter}** это означает, что автоматически будет заполняться поле **Логин**, затем поле **Пароль**, затем виртуально будет нажата клавиша <Enter>. Этот скрипт подходит для большинства случаев, но вы можете настроить любой порядок заполнения полей для страницы. В **Справке** программы вы найдёте подробное объяснение по тому, как составляются скрипты автозаполнения. Выражение, заключённое в знак процента, означает, что Password Commander вставит в элемент ввода текста, на котором установлен курсор, содержимое поля, название которого обрамлено знаками процента. Выражения, заключённые в фигурные скобки, – это команды управления автозаполнением.

Список команд автозаполнения:

- {Clear} - Очищает содержимое текстового поля;
- {Tab} - "Нажимает" клавишу <Tab> (выполняет переход к следующему элементу формы);
- {Enter} - "Нажимает" клавишу <Enter>;
- {Esc} - "Нажимает" клавишу <Escape>;
- {Space} - "Нажимает" клавишу <Пробел>;

{Up} - "Нажимает" клавишу <вверх> (прокручивает вверх выпадающий список);

{Down} - "Нажимает" клавишу <вниз> (прокручивает вниз выпадающий список);

{Right} - "Нажимает" клавишу <вправо> (смещает курсор вправо);

{Shift+Tab} - "Нажимает" комбинацию клавиш <Shift>+<Tab> (Выполняет переход к предыдущему элементу).

{mTabN} - "Нажимает" клавишу <Tab> N раз (выполняет переход к следующему элементу формы);

{mUpN} - "Нажимает" клавишу <вверх> N раз (прокручивает вверх выпадающий список);

{mDownN} - "Нажимает" клавишу <вниз> N раз (прокручивает вниз выпадающий список);

{mRightN} - "Нажимает" клавишу <вправо> N раз (смещает курсор вправо);

{mShift+TabN} - "Нажимает" комбинацию клавиш <Shift>+<Tab> N раз (Выполняет переход к предыдущему элементу).

Генератор паролей

Разработчики серьезно подошли к проблеме генерации паролей и сделали генератор, который способен удовлетворить самые изысканные потребности пользователей.

Генератор паролей можно вызвать клавишей <F9> или меню **Сервис** —> **Генератор паролей** или соответствующей кнопкой на панели инструментов (рис. 7).

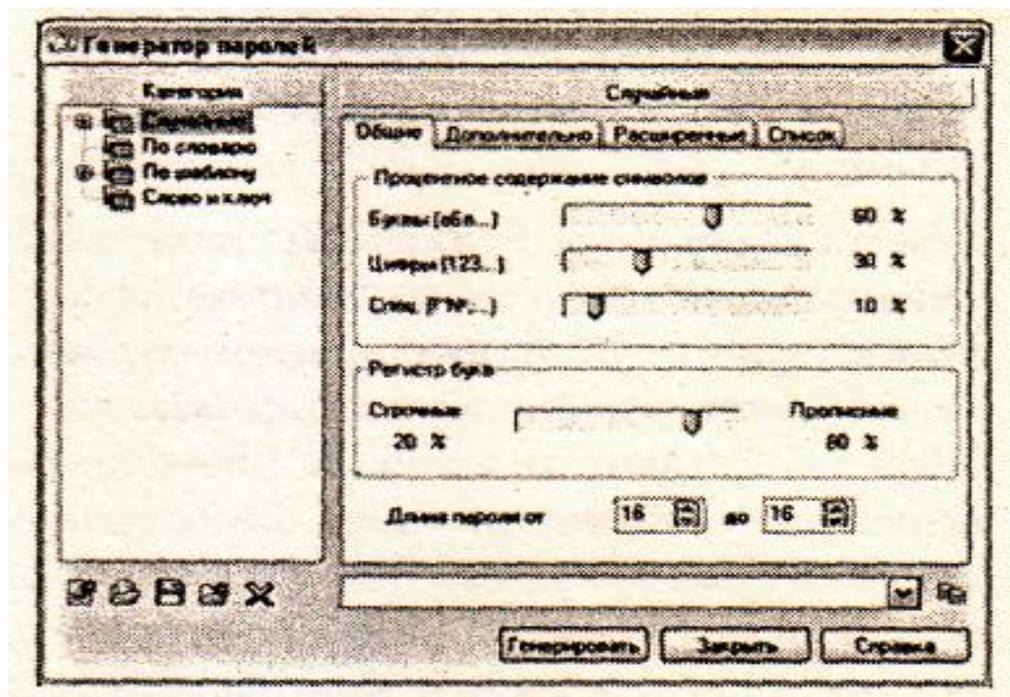


Рис. 7. Окно генератора паролей

С помощью генератора возможно создавать пароли четырёх типов:

Случайные пароли, устойчивые к взлому. Генератор Password Commander обладает широким спектром настроек генерации случайных паролей. Вы можете выбрать длину, процентное содержание тех или иных символов в пароле, регистр букв, добавить или исключить свои символы, создать список паролей и сохранить его на диск.

Легко запоминаемые пароли по словарю. Вы можете подгрузить любое количество дополнительных словарей, если словарь, поставляющийся с программой, вас не устроит, выбрать длину и тип генерации паролей, регистр букв в слове и т.д.

Генерация по шаблону. В программе представлено большое количество различных масок для генерации паролей. При создании паролей этого типа каждый специальный символ маски будет заменён по специальным правилам.

Генерация по слову и ключу. Специальная технология генерации: достаточно ввести слово и ключ, и вы получите сложный пароль, который потом может быть легко восстановлен, достаточно лишь снова запустить генератор и ввести слово и ключ.

Вам нужно выделить одну из категорий, при необходимости сделать настройки и нажать кнопку **Генерировать**. После чего сгенерированный пароль можно скопировать в буфер обмена.

Шифрование паролей

По умолчанию Password Commander хранит пароли пользователя в незашифрованном виде. Как сказано в **Справке** программы, данные пользователя сначала проходят этап обфускации (т.е. запутывания, от англ. obfuscate—запутывать, озадачивать, сбивать с толку, ставить в тупик), а потом сжимаются алгоритмом LHA. Такая схема позволяет обеспечить защиту данных, т.е. не дает в явном виде увидеть пароли, но принципиально не является шифрованием, а потому может быть легко расшифрована спецслужбами и заинтересованными лицами. Конечно, вы можете просто установить Password Commander с базами данных паролей в контейнер программы TrueCrypt и обеспечить, тем самым, надежную защиту базы паролей. Однако программа Password Commander позволяет подключать плагины для шифрования базы паролей. На странице <http://pascom.ru/download.html> можно скачать архив с одним из самых надежных алгоритмов шифрования Blowfish или целый пакет PC Plugins Powerpack, который содержит около 40 различных плагинов симметричного шифрования: 3Way, Blowfish, Cost, IDEA, TEA, Twofish, Cast 128, Cast 256, DES Triple 24byte, RC2, RC4, RC5, RC6, Rijndael и др.

Плагины представляют из себя dll-библиотеки, разработанные по определенной спецификации! Для установки плагина шифрования достаточно скопировать его в папку **{Папка с программой}\Plugins\Encryption** (обычно это C:\Program Files\Password Commander\Plugins\ Encryption), после чего плагин готов к использованию.

Для установки **PC Plugins Powerpack** скачайте и разархивируйте zip-архив, в нем окажется единственный exe-файл, который следует запустить. В появившемся мастере обычно ничего не требуется от вас, кроме как нажимать кнопку Next (Далее). Мастер должен самостоятельно определить папку, в которую необходимо установить плагины, так что от вас даже не потребуется копировать их самостоятельно.

Теперь вы можете найти в меню программы **Опции** —» **Настройки** и в открывшемся окне перейти в раздел **Аккаунт**. Установив опцию **Использовать плагин**, вы можете выбрать в списке алгоритм шифрования (рис. 8).

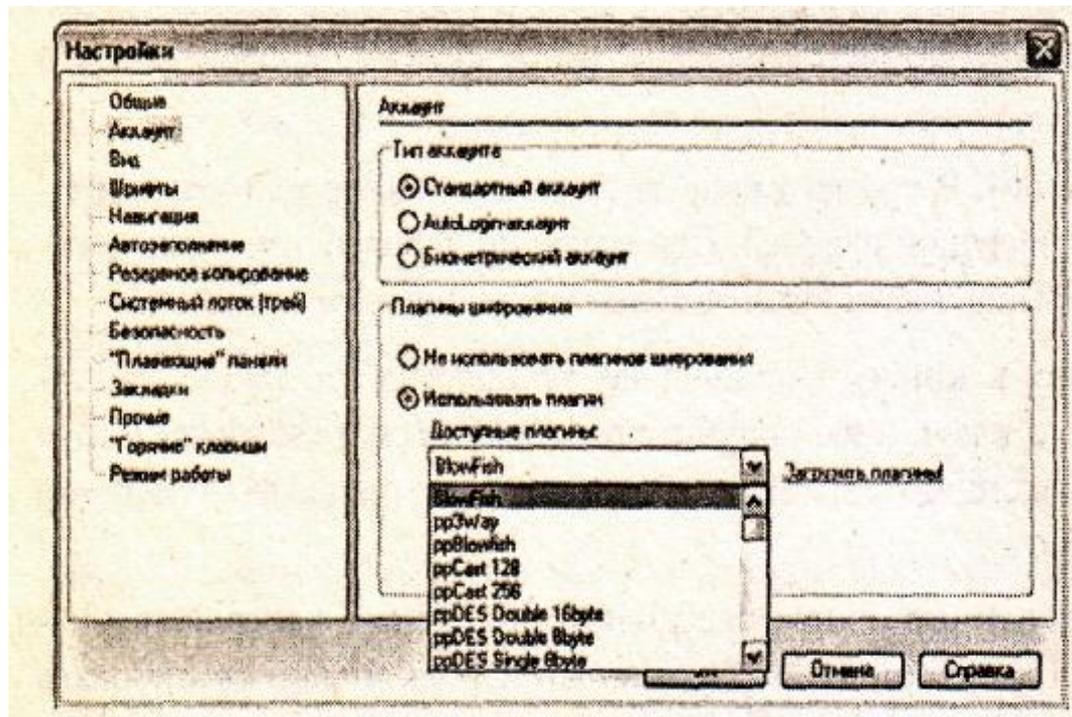


Рис.8 Выбор алгоритма шифрования паролей

После нажатия кнопки **ОК** плагин немедленно начнет работать, т.е. зашифрует ваши пароли выбранным алгоритмом. Теперь база данных паролей на диске будет постоянно в зашифрованном состоянии. Даже если во время дальнейшей вашей работы программа завершится аварийно, например, отключится компьютер, база паролей все равно останется зашифрованной.

Для программистов разработчики программы предоставляют возможность написать собственный плагин шифрования для того, чтобы быть уверенным в полной безопасности своих данных. Для этого на странице <http://pascom.ru/download.html> можно скачать "Пособие по написанию плагинов шифрования".

Паскарты

Паскарта (Passcard) представляет собой файл, в который может быть сохранена любая запись, папка или группа. Паскарта защищается паролем и может быть зашифрована с использованием

любого установленного плагина шифрования.

Пасскарта — это удобный способ безопасного переноса данных между разными аккаунтами. В Password Commander реализованы механизмы синхронизации, позволяющие импортировать данные даже из разнотипных групп.

Для сохранения данных (записи, папки, группы) в пасскарту нужно воспользоваться меню **Правка** —» **Сохранить в Пасскарту**.

Для извлечения данных из пасскарты воспользоваться меню **Файл** —» **Загрузить из Пасскарты**.

Практическое задание

Цель работы: изучить методику создания записей, генерации и шифрования паролей и автозаполнения форм web-страниц с помощью программы Password Commander.

Порядок выполнения работы:

- 1) Создайте папку с именем **Аккаунт**.
- 2) Рассмотрите демонстрационный пример в программе Password Commander. Обратите внимание, какие графические обозначения имеют группы и записи.
- 3) Создайте новый аккаунт в программе Password Commander под именем «**Электронная почта и сайты**» и сохраните его в папку с именем **Аккаунт**. При создании аккаунта используйте режим шифрования паролей с помощью алгоритма Blowfish.
- 4) В главном окне созданного аккаунта создайте новую группу с именем «**Электронная почта**».
- 5) В созданной группе создайте две записи с именами Ivan2010@mail.ru и Petr2010@mail.ru. При создании записи Ivan2010@mail.ru введите логин Ivan2010 и пароль **r123t235**. При создании записи Petr2010@mail.ru введите логин Petr2010 и пароль **s367n861**.
- 6) Создайте папку с именем «**Рассылка**» и перетащите мышью запись Ivan2010@mail.ru в данную папку.
- 7) Создайте новую группу с именем «**Сайты**».
- 8) В созданной группе создайте запись с именем **do.swsu.org**. При этом используйте логин и пароль, предназначенные для входа в ваш личный кабинет на сайте ЮЗГУ.

9) С помощью режима **«Автозаполнение»** автоматически заполните логин и пароль формы Web-страницы, предназначенной для входа в ваш личный кабинет на сайте ЮЗГУ. Разберем данный пункт более подробно:

- сделайте нужную запись активной, затем выберите команду **«Изменить»**;

- выберите вкладку **«Автозаполнение»**, затем нажмите кнопку **«Добавить»**;

- выберите режим **«Автозаполнение в Веб-страницу»**;

- через буфер обмена скопируйте URL адрес Web-страницы, предназначенной для входа в ваш личный кабинет на сайте ЮЗГУ;

- с помощью кнопки **«Выбрать»** укажите скрипт автозаполнения **%Логин%{Tab}%Пароль%{Enter}**;

- загрузите страницу с Web-формой, а сверху загрузите окно с программой Password Commander;

- сделайте запись активной и в главном меню выберите команду **«Автозаполнение»**;

- при этом автоматически заполняется логин, пароль и загружается нужная страница сайта, то есть вы входите в свой личный кабинет.

10) Сгенерируйте 4 пароля с помощью генератора паролей:

- случайный пароль, устойчивый к взлому;

- легко запоминаемый пароль по словарю;

- генерация по шаблону, для этого сначала надо задать шаблон, пользуясь кнопками;

- генерация по слову и ключу.

11) Сохраните запись Ivan2010@mail.ru в пасскарту. Для сохранения данных в пасскарту нужно использовать команду **Правка —» Сохранить в Пасскарту**. При этом надо указать путь, где располагается файл пасскарты и указать пароль к файлу пасскарты.

12) Удалите запись Ivan2010@mail.ru, а затем загрузите данную запись из пасскарты. Для извлечения данных из пасскарты использовать команду **Файл —» Загрузить из Пасскарты**.

Список контрольных вопросов

- 1) Приведите примеры программ, предназначенных для хранения паролей? Какие из них имеют русский интерфейс?
- 2) Объясните, что такое аккаунт в программе Password Commander?
- 3) Для какой цели используются группы в программе Password Commander?
- 4) Какие поля по умолчанию используются в записях? Можно ли добавить дополнительные поля в записях?
- 5) Каким образом происходит автозаполнение форм в Web-страницах?
- 6) Как будет выглядеть скрипт автозаполнения, если в Web-форме необходимо заполнить следующие поля: имя, фамилия, адрес e-mail, логин, пароль.
- 7) Какие типы паролей можно создавать с помощью генератора паролей? Дайте их краткую характеристику.
- 8) В каком виде хранятся пароли в программе Password Commander по умолчанию?
- 9) Как зашифровать пароли в программе Password Commander? Какие алгоритмы шифрования могут быть при этом использованы?
- 10) Объясните, что такое паскарта в программе Password Commander?

Список литературы

1. Нестеров С.А. Основы информационной безопасности [Электронный ресурс] : учебное пособие / С.А. Нестеров - СПб : Издательство Политехнического университета, 2014. - 322 с. // Режим доступа - <http://biblioclub.ru/index.php?page=book&id=363040>
2. Грибунин В. Г. Комплексная система защиты информации на предприятии [Текст] : учебное пособие / В. Г. Грибунин, В. В. Чудовский. – М.: Академия, 2009. - 416 с.
3. Тихонов В. А. Информационная безопасность: концептуальные, правовые, организационные и технические аспекты [текст]: учебное пособие / В. А. Тихонов, В. В. Райх. – М.: Гелиос АРВ, 2006. - 528 с.
4. Садердинов А. А. Информационная безопасность предприятия [Текст]: учебное пособие/ А. А. Садердинов, В. А. Трайнев, А. А. Федулов. 2-е изд. – М.: Дашков и К., 2004. - 336 с.
5. Игнатъев В. А. Защита информации в корпоративных информационно-вычислительных сетях [Текст]: монография.- Старый Оскол: ТНТ, 2005. – 552 с.
6. Безбогов А. А., Яковлев А. В., Шамкин В. Н. Методы и средства защиты компьютерной информации [электронный ресурс]: Учебное пособие. – Тамбов: Издательство ТГТУ, 2006.- 196 с. /Электронная библиотека «Единое окно доступа к образовательным ресурсам» - <http://window.edu.ru>
7. Шаньгин В. Ф. Защита компьютерной информации. Эффективные методы и средства / Шаньгин В. Ф. – М. : ДМК Пресс, 2010.-544 с.
8. Скляр И. С. Хакерские фишки. – М.: Лори, 2008. – 384 с.

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Юго-Западный государственный университет»
(ЮЗГУ)

Кафедра информационной безопасности

УТВЕРЖДАЮ
Проректор по учебной работе
О.Г. Локтионова
« 09 » _____ 2021 г.



ФАЕРВОЛ COMODO FIREWALL

Методические указания по выполнению лабораторных работ
для студентов укрупненной группы специальностей и
направлений подготовки 10.00.00

Курск 2021

УДК 004.56.5(076.5)

Составитель: А.Л. Ханис

Рецензент

Кандидат технических наук, доцент кафедры
информационной безопасности А.Л. Марухленко

Фаервол Comodo Firewall : методические указания по выполнению работ для студентов укрупненной группы специальностей и направлений подготовки 10.00.00 / Юго-Зап. гос. ун-т; сост.: А.Л. Ханис. - Курск, 2021. - 15 с.: ил. 8, Библиогр.: с. 15.

Содержат краткие теоретические положения о методике настройки и правилах эксплуатации фаервола Comodo Firewall.

Предназначены для студентов укрупненной группы специальностей и направлений подготовки 10.00.00.

Текст печатается в авторской редакции

Подписано в печать . Формат 60x84 1/16.
Усл.печ. л. 0,87 Уч.-изд. л. 0,78. Тираж 100 экз. Заказ. Бесплатно.
Юго-Западный государственный университет.
305040, г. Курск, ул. 50 лет Октября, 94.

Фаервол Comodo Firewoll

Введение

Межсетевой экран (МЭ) - это специализированный комплекс межсетевой защиты, называемый также брандмауэром или системой firewall. Межсетевой экран позволяет разделить общую сеть на две части или более и реализовать набор правил, определяющих условия прохождения пакетов с данными через границу из одной части общей сети в другую. Как правило, эта граница проводится между корпоративной (локальной) сетью предприятия и глобальной сетью Интернет. Обычно межсетевые экраны защищают внутреннюю сеть предприятия от вторжений из глобальной сети Интернет, хотя они могут использоваться и для защиты от нападений из корпоративной интрасети, к которой подключена локальная сеть предприятия. Для большинства организаций установка межсетевого экрана является необходимым условием обеспечения безопасности внутренней сети.

Для противодействия несанкционированному межсетевому доступу межсетевой экран МЭ должен располагаться между защищаемой сетью организации, являющейся внутренней, и потенциально враждебной внешней сетью (рис 1). При этом все взаимодействия между этими сетями должны осуществляться только через межсетевой экран. Организационно межсетевой экран входит в состав защищаемой сети.

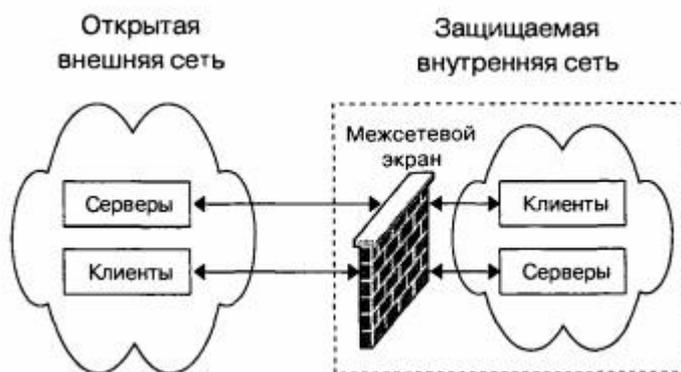


Рис. 1 – Схема подключения межсетевого экрана

Межсетевой экран, защищающий сразу множество узлов внутренней сети, призван решить две основные задачи. Первой

задачей является ограничение доступа внешних (по отношению к защищаемой сети) пользователей к внутренним ресурсам корпоративной сети. К таким пользователям могут быть отнесены партнеры, удаленные пользователи, хакеры и даже сотрудники самой компании, пытающиеся получить доступ к серверам баз данных, защищаемых межсетевым экраном. Вторая задача - разграничение доступа пользователей защищаемой сети к внешним ресурсам. Решение этой задачи позволяет, например, регулировать доступ к серверам, не требующимся для выполнения служебных обязанностей.

МЭ можно классифицировать по следующим основным признакам.

По функционированию на уровнях модели OSI: пакетный фильтр (экранирующий маршрутизатор – screening router), шлюз сеансового уровня (экранирующий транспорт), прикладной шлюз (application gateway), шлюз экспертного уровня (stateful inspection firewall).

По используемой технологии: контроль состояния протокола (stateful inspection), на основе модулей посредников (proxy).

По исполнению: программно-аппаратный и программный.

По схеме подключения: схема единой защиты сети, схема с защищаемым закрытым и не защищаемым открытым сегментами сети, схема с отдельной защитой закрытого и открытого сегментов сети.

Основной функцией МЭ является фильтрация трафика. Фильтрация осуществляется на основе выбора предварительно загруженных в межсетевой экран правил, соответствующих принятой политике безопасности. Политика работы межсетевого экрана может быть реализована на одном из двух принципов:

- запрещено все, что явно не разрешено;
- разрешено все, что явно не запрещено.

Принцип «запрещено все, что явно не разрешено» является лучшим с точки зрения информационной безопасности. При использовании принципа «разрешено все, что явно не запрещено» повышается использование сетевых сервисов со стороны пользователя, но снижается безопасность межсетевого взаимодействия.

Рассмотрим дополнительные функции МЭ. Межсетевые экраны могут выполнять идентификацию и аутентификацию пользователей, которые желают получить доступ к внешним или внутренним сетевым ресурсам, разделяемым МЭ. Межсетевые экраны выполняют еще одну важную функцию – трансляцию сетевых адресов. Данная функция реализуется ко всем пакетам, следующим из внутренней сети во внешнюю. Для этих пакетов выполняется автоматическое преобразование IP – адресов компьютеров-отправителей в один «надежный» IP-адрес. Это позволяет предотвратить многие атаки злоумышленников, при которых хакеру надо знать адрес своей жертвы. Также важными функциями МЭ являются регистрация событий, реагирования на события, анализ зарегистрированной информации и составление отчетов.

Таким образом, правильная эксплуатация МЭ является важной задачей защиты информации в корпоративных сетях.

Программно-аппаратные и программные варианты МЭ имеют определенные преимущества и недостатки. Преимущества программно-аппаратных МЭ: относительная простота развертывания и использования, меньшие размеры и энергопотребление, более высокие производительность и надежность. Преимущества программных межсетевых экранов: более низкая стоимость, возможность разграничения сегментов локальной сети без выделения подсетей, возможность развертывания на существующих серверах, расширенный функционал. В настоящее время существуют хорошие бесплатные программные МЭ, которые по своим функциональным возможностям мало в чем уступают коммерческим аналогам.

Результаты тестирования говорят о том, что фаервол Sygate Personal Firewall хорошо контролирует приложения и надежно прикрывает компьютер от посягательств из сети. Правила, как для фильтрации пакетов, так и для приложений, достаточно гибки в настройках и могут решить практически любую задачу по ограничению доступа. Возможность ограничить действие правила по времени в сумме с защитой настроек и закрытия фаервола паролем, поможет, например, ограничить доступ в интернет для ребенка в то время, когда отсутствуют родители. Sygate Personal

Firewall решает любые задачи по фильтрации трафика, например, по публикации в сети только определённых сервисов, работающих на компьютере, и сокрытия всей остальной информации о нём. По качеству исполнения и количеству функций фаервол легко может конкурировать с платными аналогами, иногда даже превосходя их в чём-то. Всё это позволяет рекомендовать Sygate Personal Firewall тем, кто использует антивирус, поставляемый в виде отдельного продукта, и хотел бы использовать легальный, бесплатный и качественный фаервол.

Если пользователю важен русскоязычный, интуитивно понятный интерфейс и простота управления, то можно остановить свой выбор на бесплатном фаерволе Comodo Firewall. Программа в процессе функционирования наглядно демонстрирует пользователю, какие процессы запущены в тот или иной момент, и какие приложения используются системой. Программа ведёт полный учёт и контроль программ, которые в определённый момент работают с подключением к Интернету. База данных программы постоянно обновляется. Поэтому, обновление, если таковое имеется, будет предложено вам в виде всплывающего сообщения. Данный фаервол распознаёт довольно большое количество троянов, шпионских программ или вредоносных кодов. Тесты показывают, что Comodo Firewall обеспечивает высокую информационную безопасность при блокировании сетевых атак.

Краткие теоретические положения

Чтобы установить Comodo Firewall, скачайте сначала установочный пакет с сайта <https://personalfirewall.comodo.com/>. Для этого нужно нажать на главной странице кнопку Download Free Firewall и на следующей странице в открывшемся списке выбрать язык **Russia** (если конечно хотите, чтобы у программы был русский интерфейс). Должно появиться две ссылки, первая предназначена для скачивания полной версии Comodo Firewall с русским интерфейсом, вторая — для скачивания только одного языкового пакета, чтобы потом установить его поверх уже установленного Comodo Firewall. Если у вас ещё не установлен Comodo Firewall, то нужно выбрать первый вариант.

Скаченный файл нужно запустить и следовать указаниям мастера. Перед началом установки появится предупреждение о том, что если в системе уже установлен какой-нибудь фаервол, то его следует удалить во избежание конфликтных ситуаций с Comodo Firewall (Рис.2). Нажмите **Да** для продолжения установки, если в вашей системе не работают другие фаерволы (в том числе встроенный фаервол Windows). Окна мастера будут на английском языке, но от вас ничего не потребуется, кроме как нажимать кнопку **Next (Далее)**, а также принять лицензионное соглашение кнопкой **Yes**. Для окончания установки потребуется перезагрузка компьютера.

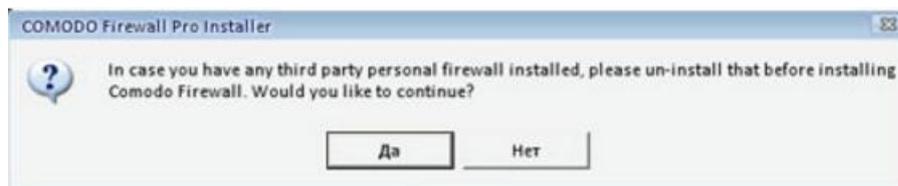


Рис. 2 - Comodo Firewall предупреждает о том, что в системе не должны работать другие фаерволы

Сразу после установки Comodo Firewall будет готов к защите вашего компьютера с установками по умолчанию. Основная работа с фаерволом сводится к тому, что он будет вам задавать вопросы об активности программ, которые хотят использовать сеть. А от вас требуется решить запретить или нет конкретной программе работу с сетью. Для этого Comodo Firewall будет выводить в правом нижнем углу экрана информационные окна (Рис. 3).

При нажатии кнопки **Разрешить** или **Запретить** фаервол однократно пропустит или не пропустит программу в интернет. В случае повторной попытки этой же программы выйти в интернет Comodo Firewall вновь выдаст окно. Если вы не хотите каждый раз отвечать на один и тот же вопрос, можете перед нажатием **Разрешить** или **Запретить** поставить галочку **Запомнить мой ответ для этого приложения**.

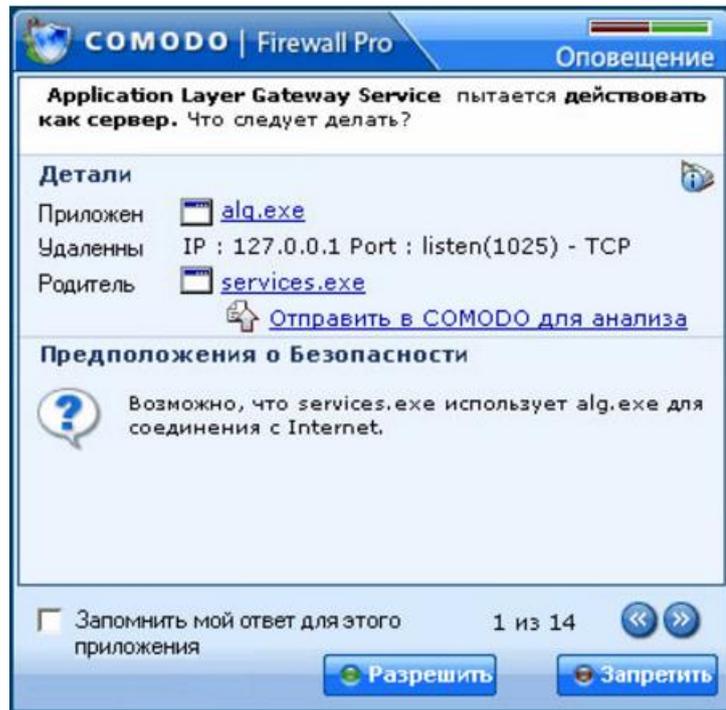


Рис. 3 – Comodo Firewall выявил программу, использующую сеть

Обычно сразу после установки фаервол будет выдавать сообщения о сетевой активности системных служб svchost.exe, alg.exe и др. Им следует разрешить работать с сетью, иначе потом будет невозможна работа в интернете. Но не нужно разрешать доступ в сеть всем программам подряд, т.к. в этом случае весь смысл фаервола теряется. Всем подозрительным программам, а также программам, которые вы не хотите, чтобы они работали с сетью, необходимо запрещать доступ в сеть.

К подозрительным программам относятся те, о происхождении которых вам ничего неизвестно — почти наверняка это может оказаться зловредное ПО, которое пытается выслать ваши пароли куда-то на неизвестный адрес. Иногда зловредные программы имеют нетипичные имена, например: save, 123124, tzsdg, trojan и т. п. Вообще придерживайтесь принципа: "лучше запретить неизвестному приложению доступ в сеть, чем разрешить".

Если по ошибке вы запретите доступ в сеть легальному приложению, и у вас после этого возникнут какие-нибудь проблемы в работе с сетью, то это легко исправить в настройках Comodo Firewall. Для этого нужно дважды щелкнуть на изображении маленького щита возле часов в трее. Откроется

главное окно программы (Рис. 4).



Рис. 4 – Главное окно Comodo Firewall

Вверху окна осуществляется выбор между тремя вкладками **Сводка**, **Защита**, **Активность**. Сведения обо всех разрешенных и запрещенных вами приложениях находятся на вкладке **Защита** — панель **Монитор Приложений**. Вы можете просто удалить из списка программу, которую вы ошибочно "разрешили" или "запретили", тогда при повторном обращении программы к сети, Comodo Firewall снова выведет окно подобное тому, что показано на Рис. 3. Вы также можете дважды щелкнуть на любой программе в списке и произвести более тонкую ее настройку в открывшемся окне (Рис. 5), в том числе выбрать действие "Разрешать" или "Блокировать" доступ программе в сеть.

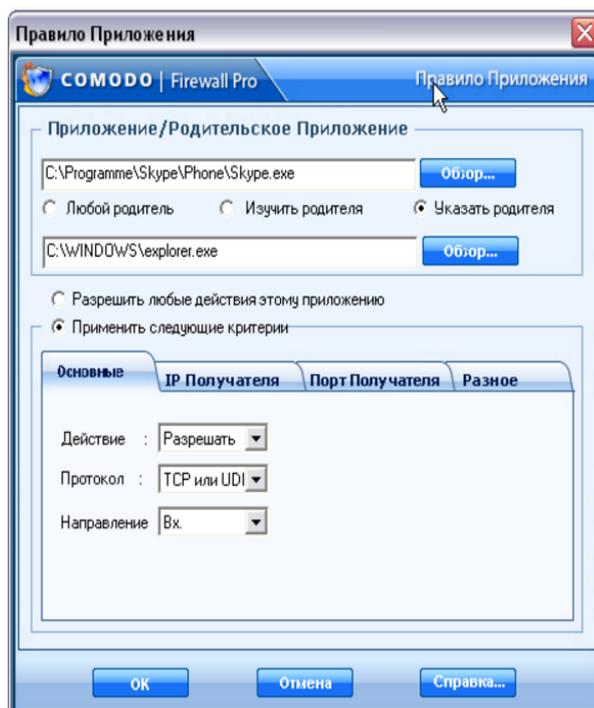


Рис. 5 – Тонкая настройка приложений через фаервол

Большинство настроек фаервола интуитивно понятны, поэтому не будем их подробно рассматривать (рекомендую вам самостоятельно посмотреть и оценить возможности программы), остановимся лишь на некоторых особенностях.

Для упрощения работы с фаерволом есть смысл в самом начале работы выбрать на вкладке **Защита** в окне **Задачи** опцию **Поиск известных приложений**. В итоге Comodo Firewall автоматически настроит правила почти для всех имеющихся приложений, которым необходима работа в сети. По утверждению разработчиков, встроенная база данных включает описания более 10 тыс. различных программ, так что вероятность того что она опознает большую часть из установленных на вашем компьютере, достаточно высока.

Кроме того, в начале работы рекомендуется выполнить обновление фаервола, чтобы он защищал от самого современного зловредного ПО. Для этого в правом верхнем углу нужно нажать кнопку **Обновление**. В дальнейшем ручное обновление делать не понадобится, т.к. Comodo Firewall настроен на автоматическое обновление, которое будет периодически выполняться в фоновом режиме, пока вы работаете в интернете (эта настройка расположена на вкладке **Защита** — панель **Дополнительно** — раздел **Разное** —

кнопка **Настроить** — опция "Автоматически проверять наличие обновлений"). Стоит еще особо обратить внимание на **сетевой монитор** (Рис. 6).



Рис.6 – Панель сетевой монитор

На панели можно задать более тонкие настройки параметров фильтрации фаерволом передачи данных по адресам и портам. Здесь Важен порядок следования правил. Comodo Firewall выполняет правила сверху вниз. С помощью кнопок **Вверх** и **Вниз** можно менять размещение правил в списке. Например, чтобы закрыть 137 порт нажмите кнопку **Добавить** и в появившемся окне выберите действие **Блокировать** укажите на закладке **Порт источника** "один порт" и пропишите номер порта (Рис. 7). После нажатия кнопки ОК, новое правило появится в списке.

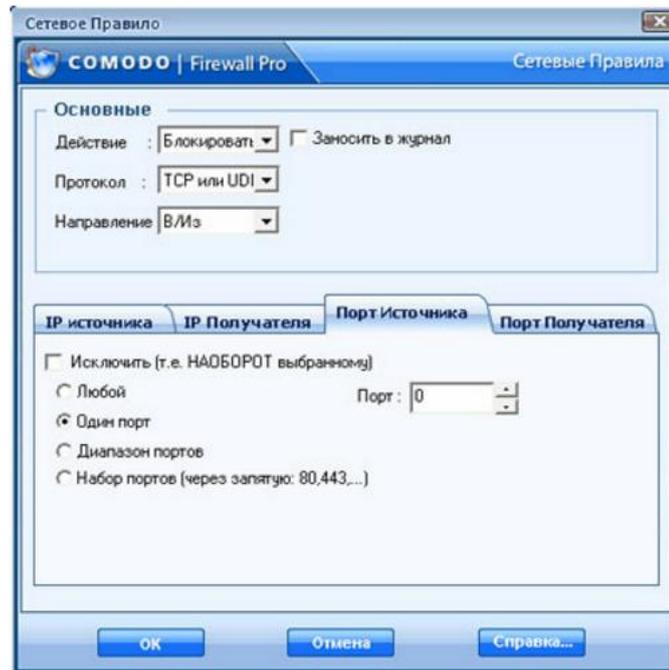


Рис. 7 – Блокирование порта

Это правило нужно ставить самым первым в списке, т.к. самое первое стандартное правило разрешает исходящие TCP и UDP соединения на любой порт источника и любой порт получателя. На вкладке **Активность** (Рис. 8) расположены две панели: **Соединения** и **Журнал**.

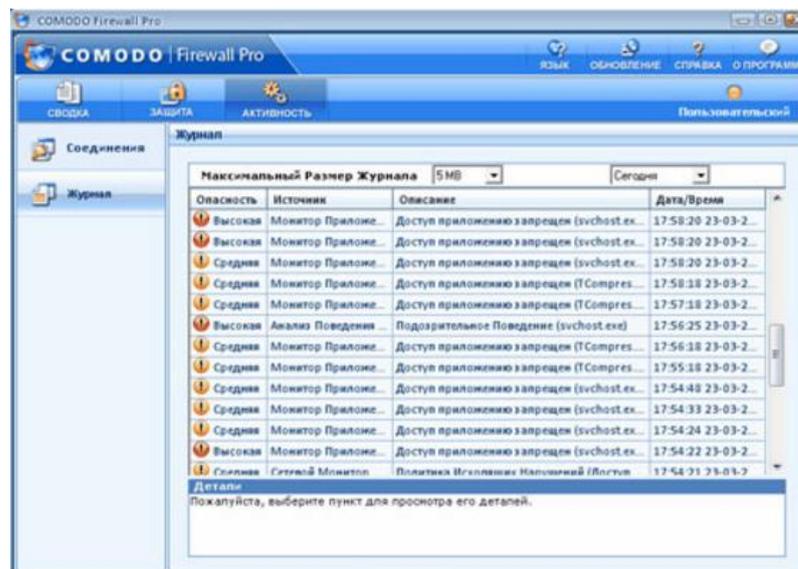


Рис. 8 – Вкладка активность

В **Соединениях** отображается список программ, которые в данный момент работают с сетью. Здесь можно также посмотреть,

объем переданных/полученных данных.

В **Журнале** ведется хронологическая запись важных событий. К нему можно всегда обратиться при анализе действий какой-то из программ.

Если Comodo Firewall вас будет "доставать" своими информационными окнами, то вы можете изменить некоторые настройки, чтобы уменьшить их количество. Например, на вкладке **Защита**-> панель **Дополнительно**-> раздел **Анализ Поведения Приложений**-> кнопка **Настроить** можно отключить анализатор поведения приложений, который часто реагирует на легальные приложения. Кроме того, на той же панели **Дополнительно**-> раздел **Разное**-> кнопка **Настроить** можно изменить уровень частоты оповещения, установив соответствующий рычажок на самый низкий уровень. Но только ни в коем случае не отключайте фаервол и внимательно читайте все его сообщения!

Практическое задание

Цель работы: изучить методику настройки фаервола Comodo Firewall.

Порядок выполнения работы:

- 1) Установите фаервол Comodo Firewall на ЭВМ.
- 2) Выполните обновление фаервола Comodo Firewall.
- 3) Настройте правила разрешения и запрета программ для выхода в сеть Интернет в ответ на запросы Comodo Firewall об активности программ, которые хотят использовать сеть.
- 4) Настройте автоматически правила для разрешения выхода приложений в сеть Интернет.
- 5) Выполните блокирование порта № 137.
- 6) Просмотрите список программ, которые в данный момент работают с сетью.
- 7) Просмотрите журнал регистрации событий.

Список контрольных вопросов

- 1) Дайте определение межсетевого экрана.
- 2) Перечислите основные функции межсетевых экранов.
- 3) Перечислите основные схемы подключения межсетевых экранов.

- 4) Перечислите типы межсетевых экранов, функционирующих на отдельных уровнях модели OSI.
- 5) Дайте классификацию межсетевых экранов.
- 6) Существуют две политики работы межсетевого экрана: «запрещено все, что явно не разрешено», «разрешено все, что явно не запрещено». Объясните, каковы их плюсы и минусы.
- 7) Приведите примеры программных межсетевых экранов.
- 8) Поддерживает ли фаервол Comodo Firewall русский язык?
- 9) Возможна ли конфликтная ситуация между Comodo Firewall и другими фаерволами?
- 10) Каким образом в фаерволе Comodo Firewall можно ограничить доступ программ в сеть Интернет?
- 11) Каким образом можно выполнить блокирование порта с определенным номером с помощью фаервола Comodo Firewall?
- 12) Каким образом можно уменьшить количество информационных сообщений с помощью настроек Comodo Firewall?

Список литературы

1. Нестеров С.А. Основы информационной безопасности [Электронный ресурс]: учебное пособие / С.А. Нестеров - СПб: Издательство Политехнического университета, 2014. - 322 с. // Режим доступа -<http://biblioclub.ru/index.php?page=book&id=363040>
2. Грибунин В. Г. Комплексная система защиты информации на предприятии [Текст]: учебное пособие / В. Г. Грибунин, В. В. Чудовский. – М.: Академия, 2009. - 416 с.
3. Садердинов А. А. Информационная безопасность предприятия [Текст]: учебное пособие / А. А. Садердинов, В. А. Трайнев, А. А. Федулов. 2-е изд. – М.: Дашков и К., 2004. - 336 с.
4. Игнатъев В. А. Защита информации в корпоративных информационно-вычислительных сетях [Текст]: монография. - Старый Оскол: ТНТ, 2005. – 552 с.
5. Безбогов А. А., Яковлев А. В., Шамкин В. Н. Методы и средства защиты компьютерной информации [электронный ресурс]: Учебное пособие. – Тамбов: Издательство ТГТУ, 2006. - 196 с. /Электронная библиотека «Единое окно доступа к образовательным ресурсам» - <http://window.edu.ru>
6. Шаньгин В. Ф. Защита компьютерной информации. Эффективные методы и средства / Шаньгин В. Ф. – М.: ДМК Пресс, 2010. -544 с.
7. Информационная безопасность и защита информации [Текст]: учебное пособие / Ю. Ю. Громов [и др.] – Старый Оскол: ТНТ, 2013. -384 с.
8. Технологии защиты информации в компьютерных сетях. Межсетевые экраны и интернет-маршрутизаторы [Текст]: учебное пособие / Е. А. Богданова [и др.]. - М.: Национальный Открытый Университет "ИНТУИТ", 2013. - 743 с.
9. Заика А. Компьютерная безопасность [Электронный ресурс] / А. Заика. - М.: РИПОЛ классик, 2013. - 160 с. // Режим доступа – <http://biblioclub.ru/index.php?page=book&id=227317>

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Юго-Западный государственный университет»
(ЮЗГУ)

Кафедра информационной безопасности

УТВЕРЖДАЮ
Проректор по учебной работе
О.Г. Локтионова
« 09 » 2021 г.



Антивирусная программа: Kaspersky Internet Security

Методические указания по выполнению лабораторных работ
для студентов укрупненной группы специальностей и
направлений подготовки 10.00.00

Курск 2021

УДК 004.56.5(076.5)

Составитель: А.Л. Ханис

Рецензент

Кандидат технических наук, доцент кафедры
информационной безопасности А.Л. Марухленко

Антивирусная программа: Kaspersky Internet Security :
методические указания по выполнению работ
для студентов укрупненной группы специальностей и направлений
подготовки 10.00.00 / Юго-Зап. гос. ун-т; сост.: А.Л. Ханис. -
Курск, 2021. - 14 с.: ил. 8, Библиогр.: с. 14.

Содержат краткие теоретические положения о методике
настройки и правилах эксплуатации антивирусной программы:
Kaspersky Internet Security.

Предназначены для студентов укрупненной группы
специальностей и направлений подготовки 10.00.00.

Текст печатается в авторской редакции

Подписано в печать . Формат 60x84 1/16.
Усл.печ. л. 0,81 Уч.-изд. л. 0,73. Тираж 100 экз. Заказ. Бесплатно.
Юго-Западный государственный университет.
305040, г. Курск, ул. 50 лет Октября, 94.

Антивирусная программа: Kaspersky Internet Security

Введение

"Лаборатория Касперского" — российская компания-разработчик антивирусных средств защиты. Первый свой продукт, прототип нынешнего Антивируса Касперского, компания выпустила в 1994 году. Разработка сразу же привлекла к себе внимание рынка средств информационной защиты, опередив на международном тестировании в показателях обнаружения и нейтрализации вирусов другие программные продукты. С тех пор антивирусные продукты "Лаборатории Касперского" постоянно занимают высокие места в рейтингах международных исследований антивирусного программного обеспечения.

Для домашнего использования "Лаборатория Касперского" в настоящее время представляет два пакета, осуществляющие защиту компьютеров: Антивирус Касперского и Kaspersky Internet Security, а также продукт для защиты смартфонов — Kaspersky Mobile Security. Базовым решением обеспечения антивирусной безопасности является Антивирус Касперского. Он обеспечивает безопасность компьютера при работе в Интернете и защиту электронной почты. Используемые им технологии позволяют защищать систему от неизвестных угроз, блокировать доступ к зараженным и опасным веб-сайтам, проверять на вирусы ICQ-сообщения, а также надежно защищать сам антивирус от попыток его отключения вредоносными программами. К дополнительным возможностям Антивируса Касперского относится проверка операционной системы компьютера и программного обеспечения на присутствие уязвимостей и настройка их безопасности, средства для восстановления работоспособности операционной системы и возможность безопасного ввода логинов, паролей и другой конфиденциальной информации при работе в Интернете. Совместно с Антивирусом Касперского для полноценной защиты компьютера компания-производитель рекомендует использовать брандмауэр. Пакет Kaspersky Internet Security обладает более широкими возможностями информационной защиты. Он осуществляет контроль за работой приложений операционной

системы и ограничивает их доступ к системным областям и личным данным пользователя, в том числе к логинам и паролям, включает в себя специальную технологию "безопасной среды" для запуска и открытия в ней подозрительных файлов и сайтов и интеллектуальный метод эффективной фильтрации нежелательных сообщений. В числе других полезных возможностей Kaspersky Internet Security инструмент для анализа работы сети, блокирование рекламы на веб-сайтах и средство, предназначенное для компьютеров, используемых всей семьей, которое позволяет регулировать применение Интернета детьми.

Kaspersky Mobile Security — средство защиты для мобильных платформ. С помощью KasperskyMobileSecurity вы можете защитить свой мобильный телефон от проникновения вирусов, хакерских атак, нежелательных звонков и SMS, а также защитить устройство и информацию, хранящуюся на нем, от нежелательного использования. Для осуществления последней функции в Kaspersky Mobile Security встроена система, называемая "Анти-Вор". Она предназначена для случаев потери смартфона или его кражи, и включает следующие средства защиты:

SMS-Block — инструмент блокировки смартфона и хранящихся на нем данных. Для включения блокировки необходимо отправить на его номер SMS с заданным вами заранее паролем. При нахождении телефона разблокировать его можно с помощью введения другого пароля, также ранее заданным вами. SMS-Find — средство для определения местонахождения потерянного или украденного смартфона. Путем отправки SMS с паролем на номер мобильного устройства вы имеете возможность узнать координаты его нахождения в системе картографического сервиса Google Maps — интернет-сервиса, представляющего собой спутниковую карту мира. Средство SMS-Find может использоваться только в смартфонах с поддержкой GPS-навигатора.

SMS-Clean позволяет с помощью отправки SMS удалить всю хранящуюся на смартфоне информацию. Например, в случае невозможности вернуть похищенный телефон.

SIM Watch — инструмент защиты извлечения из смартфона SIM-карты. При попытке извлечь SIM-карту из телефона SIM

Watch автоматически блокирует телефон. При установке новой SIM-карты телефон отправляет вам сообщение, содержащее его новый номер.

Также для защиты данных телефона Kaspersky Mobile Security содержит функцию их шифрования. Для хранения зашифрованных данных используется специальная папка на карте памяти телефона, доступ к которой можно получить только введением задаваемого вами пароля. Даже если карта памяти будет вставлена в другое устройство. Непосредственно для защиты от вирусов Kaspersky Mobile Security содержит антивирусный компонент и сетевой экран. Антивирус обеспечивает постоянную защиту устройства, имеет антивирусный сканер и функцию постоянного обновления вирусных баз. Сетевой экран следит за сетевыми соединениями с целью предупреждения нежелательного проникновения извне. Для защиты от нежелательных звонков и SMS в Kaspersky Mobile Security существует возможность создания "черных" и "белых" списков абонентов. Блокировку SMS можно осуществлять не только по номеру отправителя, но и по ключевым фразам, которые содержатся в сообщении.

Краткие теоретические положения

Пакет Kaspersky Internet Security является решением, предназначенным для комплексной защиты вашего компьютера. В нем имеются как средства для защиты от компьютерных вирусов, троянов и червей, так и средства для защиты от несанкционированного проникновения в сеть, средства защиты от сомнительных сайтов и многое другое.

Скачать дистрибутив KIS 2016 можно на сайте www.kaspersky.ru. Вам будет доступна бесплатная 30-дневная полнофункциональная версия продукта. Перед началом установки вам необходимо ознакомиться с требованиями к оборудованию и программному обеспечению, предоставляемыми разработчиком — Лабораторией Касперского для эффективной работы пакета. Необходимо соблюдать эти требования, т. к. иначе, если вы будете использовать машину с меньшим количеством оперативной памяти или более слабым процессором, после установки антивируса компьютер станет работать существенно медленнее.

Сама по себе установка не вызывает каких-либо трудностей. По заявлениям разработчиков, KIS 2016 при установке автоматически удаляет другие антивирусы. Однако если у вас на компьютере до установки KIS 2016 уже использовался какой-либо антивирусный продукт, лучше все же удалить его вручную, во избежание возможных проблем при установке пакета KIS 2016.

После установки KIS 2016 обязательно должен обновить антивирусные базы через Интернет, так что вам необходимо предоставить программе доступ в глобальную сеть.

Для открытия консоли KIS 2016 нажмите на клавиатуре клавишу или соответствующий значок на рабочем столе. Далее выберите All Programs | Kaspersky Internet Security | Kaspersky Internet Security. Откроется рабочее окно антивируса Kaspersky Internet Security 2016 (рис.1).

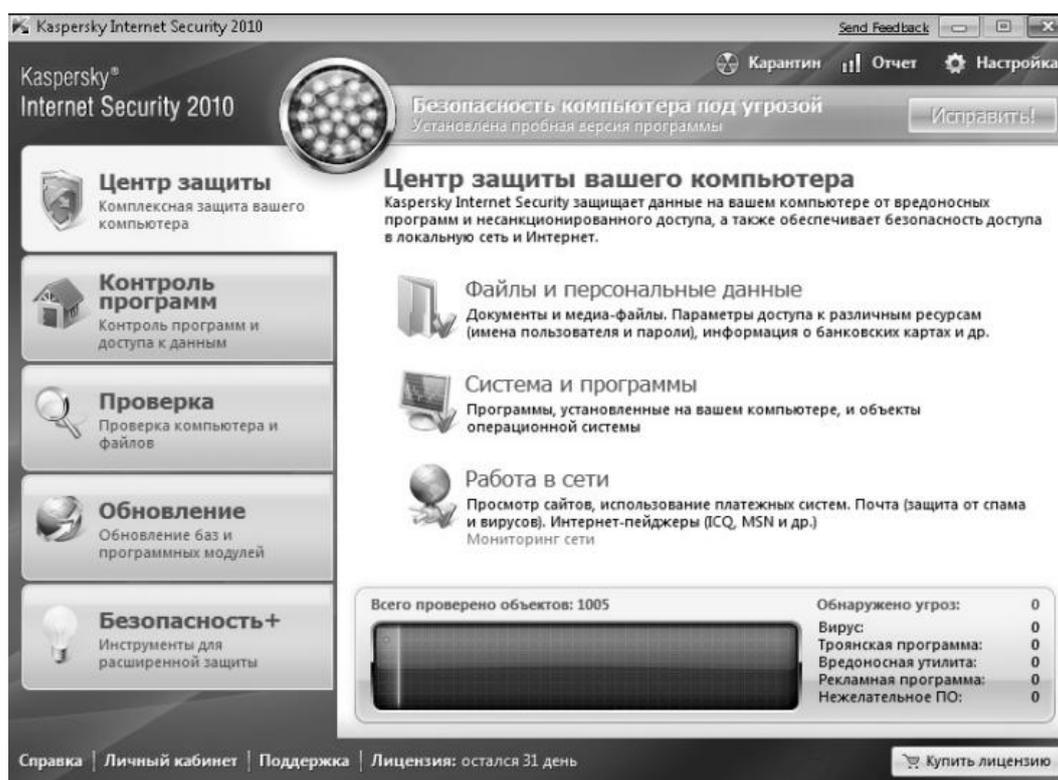


Рис. 1. Центр защиты антивируса KIS 2016

Итак, на рис. 1 перед нами предстает рабочее окно KIS 2016, открытое на вкладке Центр защиты. В верхней части окна находится сигнал светофора, показывающий статус системы на настоящий момент. Статус защиты желтый, т. е. безопасность компьютера под угрозой если используется испытательная версия

программы. В случае если бы не были установлены обновления, сигнал светофора был бы красный. А если бы была установлена коммерческая версия KIS 2016 и последние версии антивирусных баз, то сигнал был бы зеленый. Для того чтобы исправить существующие проблемы, можно воспользоваться кнопкой **Исправить**, в правой верхней части экрана. В нашем случае откроется окно с предложением приобрести лицензию на использование KIS 2016. В случае если используются просроченные антивирусные базы, будет произведено принудительное обновление баз. Также в этом окне вы можете наблюдать за состоянием системы, обнаруженными вредоносными программами, количеством проверенных объектов и т. д. В центральной части окна показан статус защиты различных компонент системы, таких как **Файлы и персональные данные, Система и программы, Работа в сети**. Обратите внимание на то, что около каждого из этих элементов должна стоять зеленая галочка. Это говорит о том, что защита данной компоненты включена.

На второй вкладке **Контроль программ** (рис. 2) производится контроль и предотвращение выполнения программами каких-либо вредоносных действий.

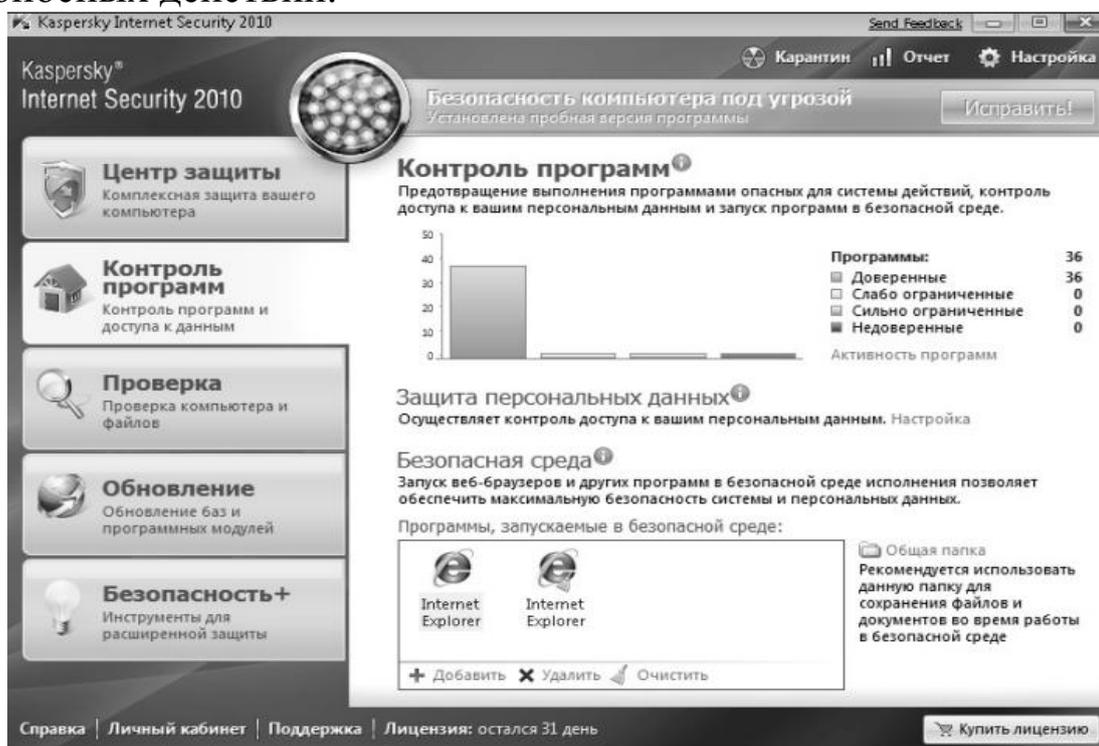


Рис. 2. Вкладка контроля программ и доступа к данным

На вкладке контроля программ графически представлена активность различных приложений на вашем компьютере. Также в этой вкладке можно настроить защиту персональных данных.

Еще одним новым средством защиты в KIS 2016 является безопасная среда. В нее можно помещать различные приложения, например веб-браузер или же клиент электронной почты. Работа в безопасной среде позволяет оградить работающее приложение от основной среды, и в случае проникновения вредоносного кода в данное приложение, например при заражении веб-браузера, злоумышленник не сможет проникнуть в другие приложения и использовать их ресурсы.

Во вкладке **Проверка** вы можете произвести полную проверку системы или же произвести выборочное сканирование отдельных дисков компьютера (рис. 3). Также здесь можно открыть окно поиска уязвимостей.

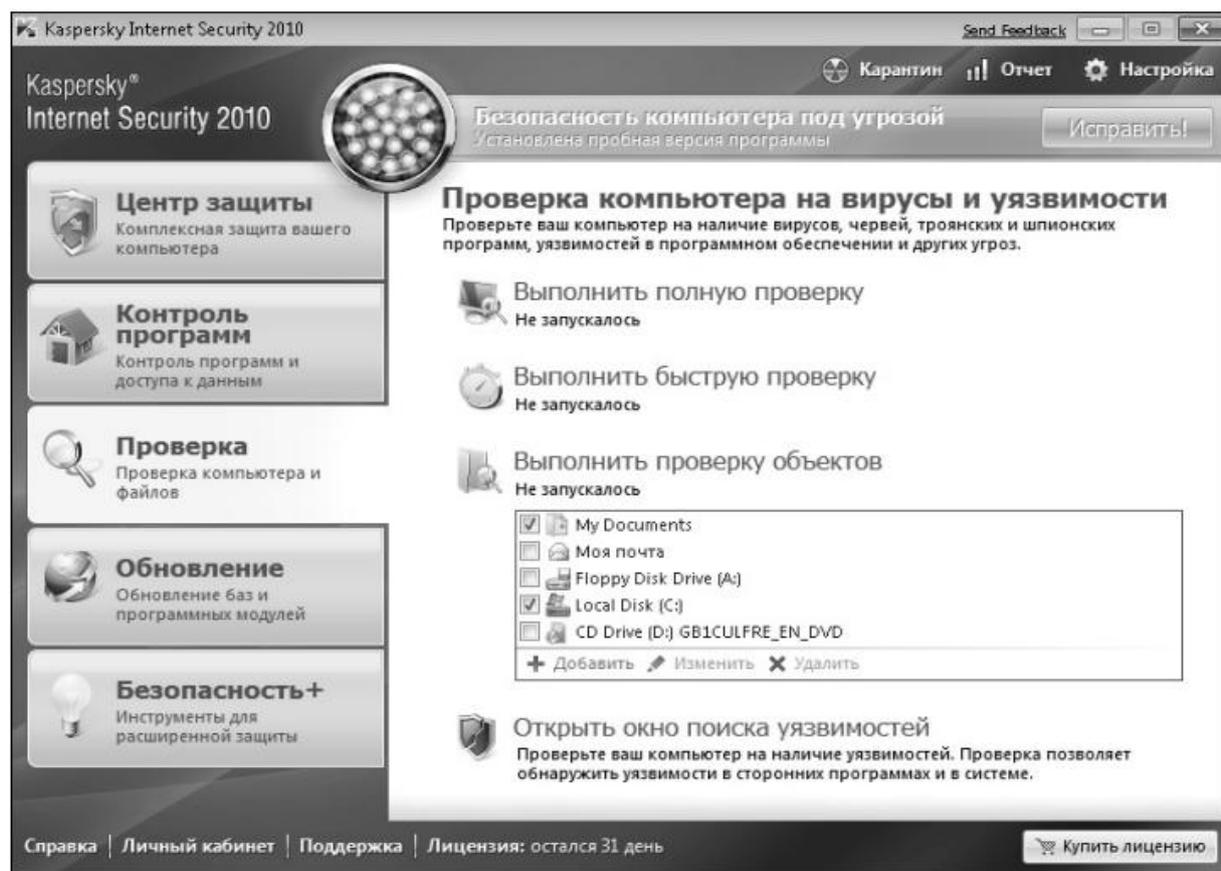


Рис. 3. Вкладка проверки компьютера и файлов

На вкладке **Обновление** показан статус всех баз, используемых KIS 2016.



Рис. 4. Вкладка управления обновлениями программы

Здесь мы можем видеть количество сигнатур для различных угроз, а также даты выпуска этих баз. При необходимости можно выполнить принудительное обновление, щелкнув ссылку **Выполнить обновление**. На вкладке **Безопасность+** находятся дополнительные инструменты и сервисы для обеспечения безопасности вашего компьютера и оптимизации выполнения различных задач (рис. 5).



Рис. 5. Вкладка инструментов расширенной защиты
Например, с помощью виртуальной клавиатуры вы можете

защититься от клавиатурных перехватчиков. С помощью ссылки **Родительский контроль** ограничить доступ пользователей к определенным веб-ресурсам. Также здесь имеются различные средства для восстановления системы. Вернемся к уже упоминавшемуся средству по поиску уязвимостей. Для того чтобы воспользоваться этим средством, необходимо открыть вкладку **Проверка основного окна Kaspersky Internet Security 2016** и затем выбрать ссылку **Открыть окно поиска уязвимостей**. Откроется окно **Поиск уязвимостей** (рис. 6).

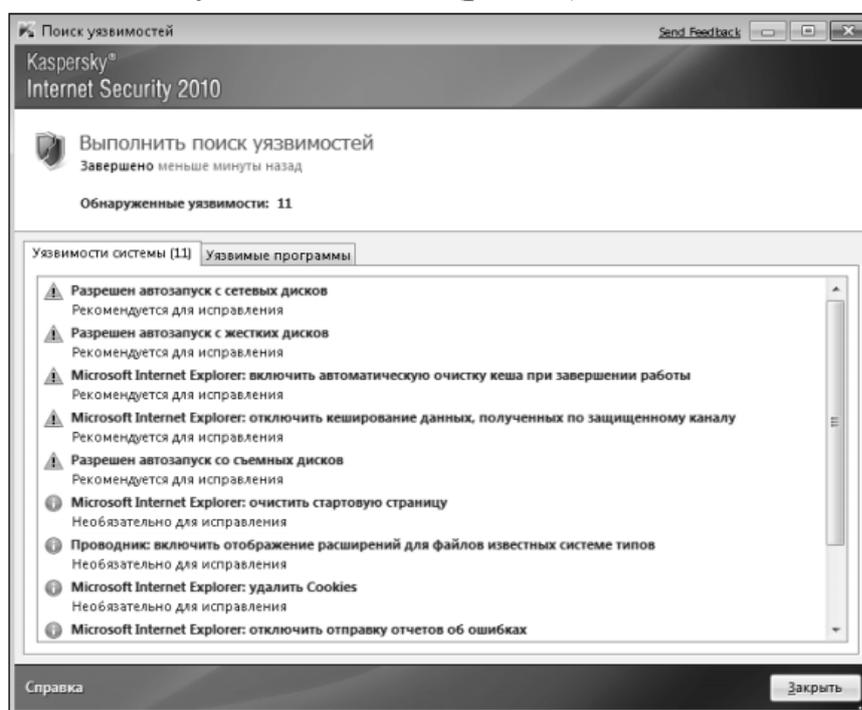


Рис. 6. Окно поиска уязвимостей

Здесь вы можете проверить ваш компьютер на наличие уязвимостей. Программа KIS 2016 содержит сведения об известных уязвимостях в операционной системе и установленных приложениях. Еще одним интересным средством является **Мастер восстановления системы** (рис. 7). С помощью данного средства вы сможете восстановить систему после воздействия вредоносного кода, а также устранить последствия некорректной настройки отдельных компонентов системы. Общие настройки KIS 2016, в которых содержатся параметры всех ранее описанных компонент, и многое другое открываются щелчком на ссылке **Настройка** в главном окне Kaspersky Internet Security 2016 (рис. 8). Здесь вы можете найти настройки любого элемента KIS 2016 и произвести

соответствующие изменения.

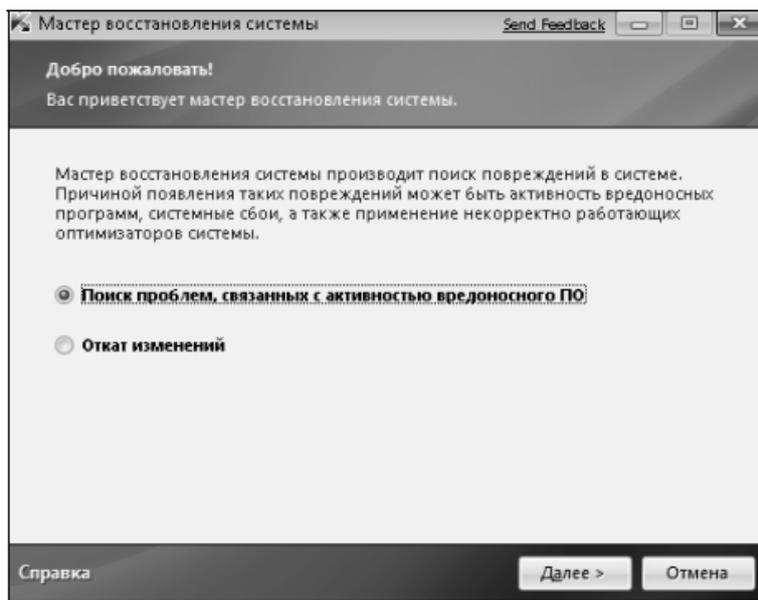


Рис. 7. Мастер восстановления системы

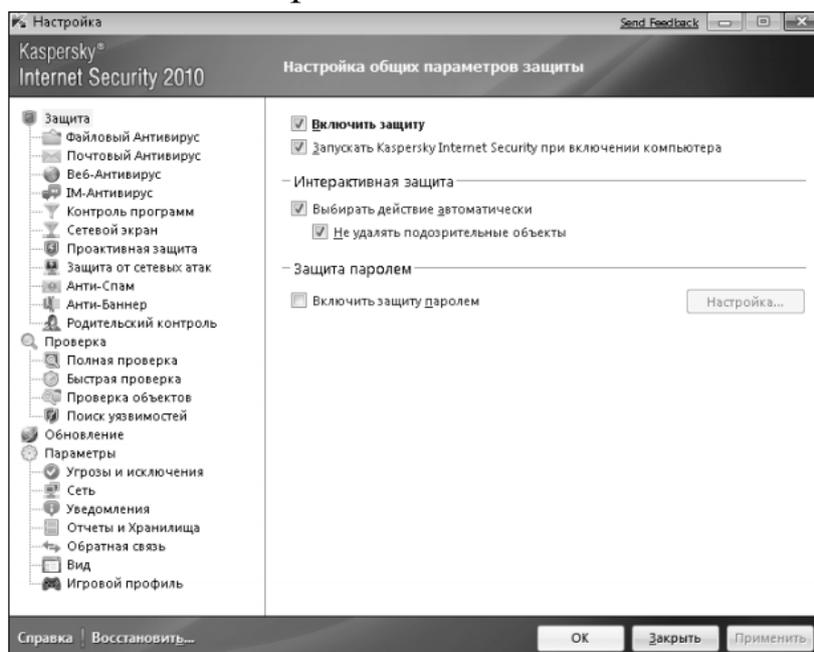


Рис. 8. Окно настройки параметров защиты

Практическое задание

Цель работы: изучить интерфейс, методику эксплуатации и настройки программы **Kaspersky Internet Security**.

Порядок выполнения работы:

- 1) Проверьте статус защиты компонент системы: **Файлы и персональные данные, Система и программы, Работа в сети.**
- 2) Обновите антивирусные базы Kaspersky Internet Security.

3) Проверьте активность приложений на вашем компьютере. Проанализируйте, являются ли все приложения доверенными.

4) Настройте защиту персональных данных.

5) Установите Web-браузер в безопасную среду.

6) Проведите сканирование диска D.

7) Воспользуйтесь виртуальной клавиатурой для набора абзаца текста.

8) С помощью ссылки **Родительский контроль** ограничьте доступ пользователя к Web-сайту знакомств.

9) Проверьте операционную систему и установленные приложения на наличие уязвимостей.

10) Оптимизируйте настройку системы с помощью **Мастера восстановления системы**.

11) Настройте межсетевой экран.

12) Настройте анти-спам.

13) Настройте анти-баннер.

14) Настройте защиту от сетевых атак.

15) Выполните резервное копирование информации.

Список контрольных вопросов

1. Дайте классификацию компьютерных вирусов.

2. В чем основное отличие вирусов-сценариев от файловых вирусов?

3. Существование каких вирусов зависит от конкретной программы?

4. В чем основное отличие троянской программы от вируса. Приведите пример троянской программы.

5. Дайте классификацию компьютерных червей. Приведите примеры компьютерных червей.

6. Перечислите методы обнаружения вирусов.

7. Какой метод выявления вирусов позволяет обнаруживать только известные вирусы?

8. В чем сущность метода обнаружения вирусов, основанного на сигнатурах?

9. В чем сущность метода выявления вирусов – обнаружение программ подозрительного поведения?

10. В чем сущность метода обнаружения вирусов при помощи “белого списка”?

11. В чем сущность обнаружения вирусов при помощи эмуляции работы программы?

12. В чем сущность метода выявления вируса - эвристический анализ?

13. Почему не рекомендуется на одной ЭВМ использовать одновременно несколько антивирусов?

14. Какая антивирусная программа не конфликтует с другими антивирусами?

15. Приведите примеры бесплатных антивирусов.

16. Дайте общую характеристику возможностей программы Kaspersky Internet Security.

Список литературы

1. Нестеров С.А. Основы информационной безопасности [Электронный ресурс] : учебное пособие / С.А. Нестеров - СПб : Издательство Политехнического университета, 2014. - 322 с. // Режим доступа -<http://biblioclub.ru/index.php?page=book&id=363040>
2. Грибунин В. Г. Комплексная система защиты информации на предприятии [Текст] : учебное пособие / В. Г. Грибунин, В. В. Чудовский. – М.: Академия, 2009. - 416 с.
3. Садердинов А. А. Информационная безопасность предприятия[Текст]: учебное пособие/ А. А. Садердинов, В. А. Трайнев, А. А. Федулов. 2-е изд. – М.: Дашков и К., 2004. - 336 с.
4. Игнатъев В. А. Защита информации в корпоративных информационно-вычислительных сетях [Текст]: монография.- Старый Оскол: ТНТ, 2005. – 552 с.
5. Безбогов А. А., Яковлев А. В., Шамкин В. Н. Методы и средства защиты компьютерной информации [электронный ресурс]: Учебное пособие. – Тамбов: Издательство ТГТУ, 2006.- 196 с. /Электронная библиотека «Единое окно доступа к образовательным ресурсам» - <http://window.edu.ru>
6. Шаньгин В. Ф. Защита компьютерной информации. Эффективные методы и средства / Шаньгин В. Ф. – М. : ДМК Пресс, 2010.-544 с.
7. Жадаев А. Г. Антивирусная защита ПК: от “чайника” к пользователю.
8. Технологии защиты информации в компьютерных сетях. Межсетевые экраны и интернет-маршрутизаторы [Текст] : учебное пособие / Е. А. Богданова [и др.]. - М. : Национальный Открытый Университет "ИНТУИТ", 2013. - 743 с.
9. Заика А. Компьютерная безопасность [Электронный ресурс] / А. Заика. - М. : РИПОЛ классик, 2013. - 160 с. // Режим доступа – <http://biblioclub.ru/index.php?page=book&id=227317>