МИНОБРНАУКИ РОССИИ
Глазовский инженерно-экономический институт (филиал)
федерального государственного бюджетного образовательного
учреждения высшего образования
«Ижевский государственный технический университет
имени М.Т.Калашникова»

Кафедра «Автоматизированных систем управления»

УТВЕРЖДЕН
на заседании кафедры
«10» мая 2018г., протокол № 5
Заведующий кафедрой

_____В.В. Беляев
(подпись)

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

по общепрофессиональной дисциплине

ОП.12 «WEB-ПРОГРАММИРОВАНИЕ»

программы подготовки специалистов среднего звена

по специальности

09.02.04 Информационные системы (по отраслям)

Глазов 2018

Оглавление

Формируемыекомпетенции 2

Критерии и шкала оценивания ответов на устные вопросы 3

Критерии и шкала оценивания результатов тестирования 4

Критерии и шкала оценивания рефератов 4

Критерии и шкала оценивания домашнего задания 5

Критерии и шкалы оценивания решения кейс-задач 5

Критерии и шкала оценивания контрольных работ 6

Критерии и шкала оценки презентаций 7

Типовые контрольные задания или иные материалы,
необходимые для оценки знаний, умений , навыков,
характеризующих этапы формирования компетенций 8

Формируемые компетенции

Код компетенции	Наименование результата обучения
ОК-1	Понимать сущность и социальную значимость своей будущей профессии, проявлять к ней устойчивый интерес.
ОК-2	Организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество.
ОК-3	Принимать решения в стандартных и нестандартных ситуациях и нести за них ответственность.
ОК-4	Осуществлять поиск и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития.
ОК-5	Использовать информационно-коммуникационные технологии в профессиональной деятельности.
ОК-6	Работать в коллективе и команде, эффективно общаться с коллегами, руководством, потребителями.
ОК-7	Брать на себя ответственность за работу членов команды (подчиненных), результат выполнения заданий.
ОК-8	Самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации.
ОК-9	Ориентироваться в условиях частой смены технологий в профессиональной деятельности.
ПК.1.1	Собирать данные для анализа использования и функционирования информационной системы, участвовать в составлении отчетной документации, принимать участие в разработке проектной документации на модификацию информационной системы.
ПК.1.2	Взаимодействовать со специалистами смежного профиля при разработке методов, средств и технологий применения объектов профессиональной деятельности.
ПК.1.3	Производить модификацию отдельных модулей информационной системы в соответствии с рабочим заданием, документировать произведенные изменения.
ПК.1.4	Участвовать в экспериментальном тестировании информационной системы на этапе опытной эксплуатации, фиксировать выявленные ошибки кодирования в разрабатываемых модулях информационной системы.

КРИТЕРИИ И ШКАЛА ОЦЕНИВАНИЯ ОТВЕТОВ НА УСТНЫЕ ВОПРОСЫ

№ п/п	критерии оценивания	количество баллов	оценка/зачет
1.	1) полно и аргументированно отвечает по содержанию задания; 2) обнаруживает понимание материала, может обосновать свои суждения, применить знания на практике, привести необходимые примеры не только по учебнику, но и самостоятельно составленные; 3) излагает материал последовательно и правильно.	10	отлично
2.	студент дает ответ, удовлетворяющий тем же требованиям, что и для оценки «5», но допускает 1-2 ошибки, которые сам же исправляет.	8	хорошо
3.	ставится, если студент обнаруживает знание и понимание основных положений данного задания, но: 1) излагает материал неполно и допускает неточности в определении понятий или формулировке правил; 2) не умеет достаточно глубоко и доказательно обосновать свои суждения и привести свои примеры; 3) излагает материал непоследовательно и допускает ошибки.	5	удовлетворительно
4.	студент обнаруживает незнание ответа на соответствующее задание, допускает ошибки в формулировке определений и правил, искажающие их смысл, беспорядочно и неуверенно излагает материал; отмечаются такие недостатки в подготовке студента, которые являются серьезным препятствием к успешному овладению последующим материалом.	0	неудовлетворительно

КРИТЕРИИ И ШКАЛА ОЦЕНИВАНИЯ РЕЗУЛЬТАТОВ ТЕСТИРОВАНИЯ

№ п/п	тестовые нормы:% правильных ответов	количество баллов	оценка/зачет
1	90-100 %	25-30	отлично
2	80-89%	19-24	хорошо
3	70-79%	13-18	удовлетворительно
4	60-69%	7-12	удовлетворительно
5	50-59%	1-6	неудовлетворительно
6	менее 50%	0	неудовлетворительно

КРИТЕРИИ И ШКАЛА ОЦЕНИВАНИЯ РЕФЕРАТОВ

№ п/п	критерии оценивания	количество баллов	оценка/зачет
1	выполнены все требования к написанию и защите реферата: обозначена проблема и обоснована её актуальность, сделан краткий анализ различных точек зрения на рассматриваемую проблему и логично изложена собственная позиция, сформулированы выводы, тема раскрыта полностью, выдержан объём, соблюдены требования к внешнему оформлению, даны правильные ответы на дополнительные вопросы.	*5 баллов*	Отлично
2	основные требования к реферату и его защите выполнены, но при этом допущены недочеты. В частности, имеются неточности в изложении материала; отсутствует логическая последовательность в суждениях; не выдержан объем реферата; имеются упущения в оформлении; на дополнительные вопросы при защите даны неполные ответы.	*4 балла*	Хорошо
3	имеются существенные отступления от требований к реферированию. В частности: тема освещена лишь частично; допущены фактические ошибки в содержании реферата или при	*3 балла*	Удовлетворительно

	ответе на дополнительные вопросы.		
4	тема освоена лишь частично; допущены грубые ошибки в содержании реферата или при ответе на дополнительные вопросы; во время защиты отсутствует вывод.	1-2 балла	Удовлетворительно
5	тема реферата не раскрыта, обнаруживается существенное непонимание проблемы.	*0 баллов*	неудовлетворительно

КРИТЕРИИ И ШКАЛА ОЦЕНИВАНИЯ ДОМАШНЕГО ЗАДАНИЯ

№ п/п	критерии оценивания	количество баллов
1	Задание выполнено полностью: цель домашнего задания успешно достигнута; основные понятия выделены; наличие схем, графическое выделение особо значимой информации; работа выполнена в полном объёме.	9-10
2	Задание выполнено: цель выполнения домашнего задания достигнута; наличие правильных эталонных ответов; однако работа выполнена не в полном объёме.	8-7
3	Задание выполнено частично: цель выполнения домашнего задания достигнута не полностью; многочисленные ошибки снижают качество выполненной работы.	6-5
4	Задание не выполнено, цель выполнения домашнего задания не достигнута.	менее 5

КРИТЕРИИ И ШКАЛЫ ОЦЕНИВАНИЯ РЕШЕНИЯ КЕЙС-ЗАДАЧ

№ п/п	критерии оценивания	количество баллов	оценка/зачет
1	умение анализировать ситуацию и находить оптимальное количества решений	1	
2	умение работать с информацией, в том числе умение затребовать дополнительную информацию, необходимую для уточнения ситуации	1	
3	умение моделировать решения в соответствии с заданием, представлять различные подходы к разработке планов действий, ориентированных на конечный результат	1	
4	умение принять правильное решение на	1	

	основе анализа ситуации;		
5	навыки четкого и точного изложения собственной точки зрения в устной и письменной форме, убедительного отстаивания своей точки зрения;	1	
6	навык критического оценивания различных точек зрения, осуществление самоанализа, самоконтроля и самооценки.	1	
7	адекватность и соответствие ответов специалиста современным тенденциям рынка, конструктивность.	1	
8	креативность, нестандартность предлагаемых решений*;*	1	
9	количество альтернативных вариантов решения задачи (версионность мышления);	1	
10	наличие необходимых навыков, их выраженность (в зависимости от *требований).*	1	
	Количество баллов в целом	**10**	

КРИТЕРИИ И ШКАЛА ОЦЕНИВАНИЯ КОНТРОЛЬНЫХ РАБОТ

№ п/п	критерии оценивания	количество баллов	оценка
1	исключительные знания, абсолютное понимание сути вопросов, безукоризненное знание основных понятий и положений, логически и лексически грамотно изложенные, содержательные, аргументированные и исчерпывающие ответы	28-30	отлично
2	глубокие знания материала, отличное понимание сути вопросов, твердое знание основных понятий и положений по вопросам, структурированные, последовательные, полные, правильные ответы	25-27	отлично
3	глубокие знания материала, правильное понимание сути вопросов, знание основных понятий и положений по вопросам, содержательные, полные и конкретные ответ на вопросы. Наличие	22-24	хорошо

	несущественных или технических ошибок		
4	твердые, достаточно полные знания, хорошее понимание сути вопросов, правильные ответы на вопросы, минимальное количество неточностей, небрежное оформление	19-21	хорошо
5	твердые, но недостаточно полные знания, по сути верное понимание вопросов, в целом правильные ответы на вопросы, наличие неточностей, небрежное оформление	16-18	удовлетворительно
6	общие знания, недостаточное понимание сути вопросов, наличие большого числа неточностей, небрежное оформление	13-15	удовлетворительно
7	относительные знания, наличие ошибок, небрежное оформление	10-12	удовлетворительно
8	поверхностные знания, наличие грубых ошибок, отсутствие логики изложения материала	7-9	неудовлетворительно
9	непонимание сути, большое количество грубых ошибок, отсутствие логики изложения материала	4-6	неудовлетворительно
10	не дан ответ на поставленные вопросы	1-3	неудовлетворительно
11	отсутствие ответа, дан ответ на другие вопросы, списывание в ходе выполнения работы, наличие на рабочем месте технических средств, в том числе телефона	0	неудовлетворительно

КРИТЕРИИ И ШКАЛА ОЦЕНКИ ПРЕЗЕНТАЦИЙ

№ п/п	критерии оценки	максимальное количество баллов
1	титульный слайд с заголовком	5
2	дизайн слайдов	10
3	использование дополнительных эффектов (смена слайдов, звук, графика, анимация)	5
4	список источников информации	5
5	широта кругозора	5
6	логика изложения материала	10
7	текст хорошо написан и сформированные идеи ясно	10

	изложены и структурированы	
8	слайды представлены в логической последовательности	5
9	грамотное создание и сохранение документов в папке рабочих материалов	5
10	слайды распечатаны в форме заметок	5
	средняя оценка:	

ТИПОВЫЕ КОНТРОЛЬНЫЕ ЗАДАНИЯ ИЛИ ИНЫЕ МАТЕРИАЛЫ, НАОБХОДИМЫЕ ДЛЯ ОЦЕНКИ ЗНАНИЙ, УМЕНИЙ , НАВЫКОВ, ХАРАКТЕРИЗУЮЩИХ ЭТАПЫ ФОРМИРОВАНИЯ КОМПЕТЕНЦИЙ

Введение в WEB-программирование на стороне клиента. Основы HTML

Темы лабораторных работ

1. Спроектировать структуру веб-сайта по теме вашего индивидуального проекта или по любой другой теме, сопоставимой (или большей) по объему с проектом.

2. Разработать эскиз оформления веб-сайта (использовать любой графический редактор).

3. Выполнить верстку макета страницы по разработанному эскизу.

4. Постройте таблицу следующего вида:

ТОП-10 Веб-студий по версии проекта «Рейтинга Рунета»

№	Компания	Проектов	Балл
1	Action Wunderman	123	3.26
2	Студия Артемия Лебедева	116	2.87
3	РБК СОФТ	84	2.85
4	Айтекс	777	2.10
5	Текард	176	1.78
6	Astronim	164	1.34
7	Группа МАХАОН	40	1.30
8	Quantum Art	41	1.17
9	АИСТ	46	0.96
10	Articul Media	33	0.94

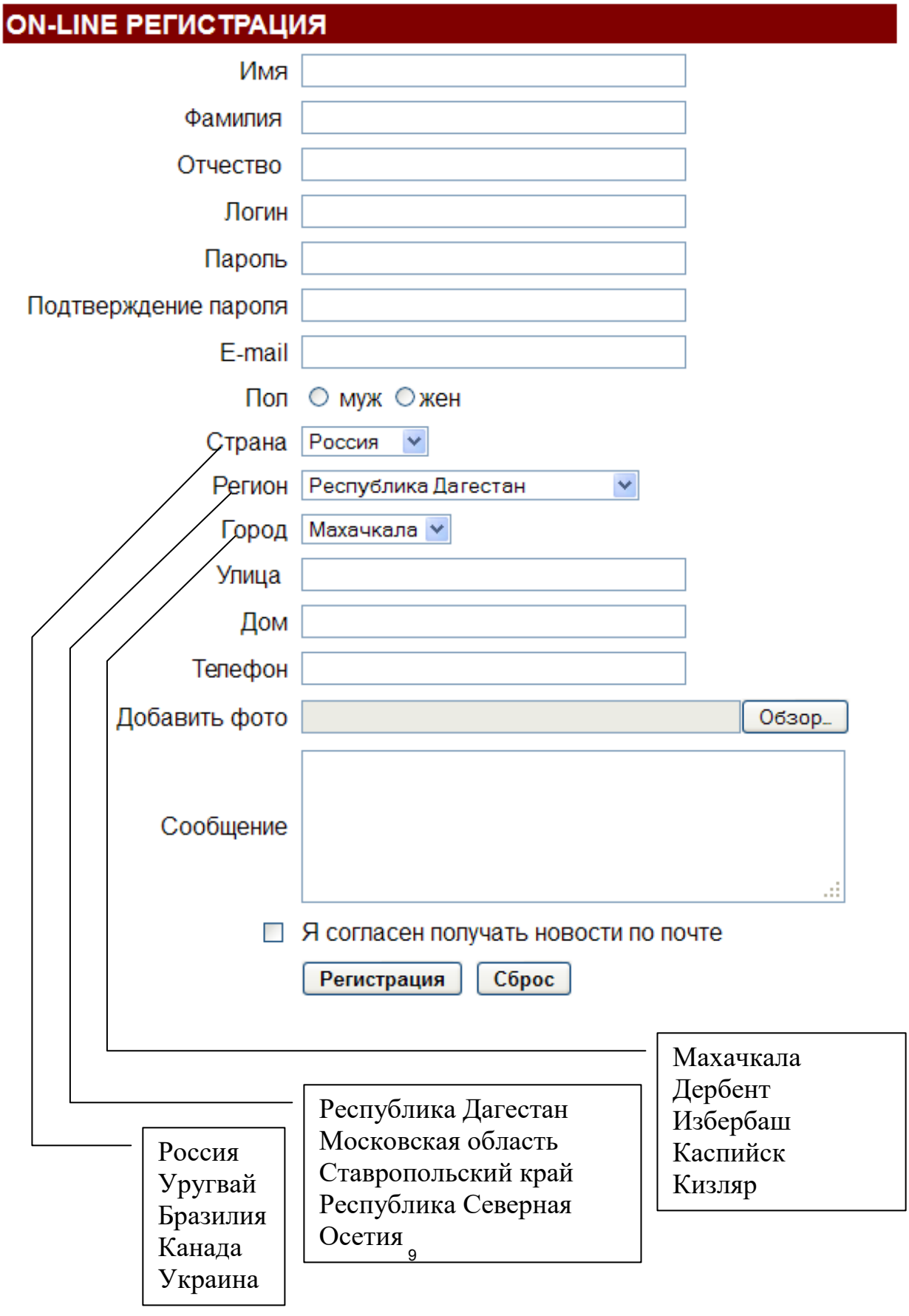
5. Постройте таблицу следующего вида:

<table>
<tr><td colspan="5"></td></tr>
<tr><td rowspan="2"></td><td></td><td rowspan="2"></td><td></td><td></td></tr>
<tr><td rowspan="2"></td><td rowspan="2"></td><td></td></tr>
<tr><td></td><td></td><td></td></tr>
</table>

6. Разработать форму on-line регистрации по образцу. Сохранить форму для вашего сайта.

7. Установить связи между страницами. Отобразить связи на странице «Карта сайта».

8. Создать изображение-карту для вашего сайта.



Тестовые задания

1) Компьютеры одной организации, связанные каналами передачи информации для совместного использования общих ресурсов и периферийных устройств и находящиеся в одном здании, называют сетью:

a) региональной
b) территориальной
c) локальной
d) глобальной

2) Компьютер, предоставляющий свои ресурсы другим компьютерам при совместной работе, называется:

a) коммутатором
b) сервером
c) модемом
d) адаптером

3) Скорость передачи информации по локальной сети обычно находится в диапазоне:

a) от 10 до 100 Мбит/с
b) от 10 до 100 Кбит/с
c) от 100 до 500 бит/с
d) от 10 до 100 бит/с

4) Сколько Кбайт будет передаваться за одну секунду по каналу с пропускной способностью 10 Мбит/с?

a) 1280
b) 10240
c) 160
d) 10000

5) Сколько Мбайт будет передаваться за одну минуту по каналу с пропускной способностью 100 Мбит/с?

a) 750
b) 12,5
c) 6000
d) 600

6) 1 Гбит/с равен:

a) 1024 Мбит/с
b) 1024 Мбайт/с
c) 1024 Кбит/с
d) 1024 байт/с

7) За сколько секунд будет передано 25 Мбайт информации по каналу с пропускной способностью 10 Мбит/с?

a) 20
b) 2,5
c) 40
d) 200

8) Вариант соединения компьютеров между собой, когда кабель проходит от одного компьютера к другому, последовательно соединяя компьютеры и периферийные устройства между собой – это:

a) линейная шина
b) соединение типа «звезда»
c) древовидная топология

9) Если к каждому компьютеру подходит отдельный кабель из одного центрального узла – это:

a) линейная шина
b) соединение типа «звезда»
c) древовидная топология

10) Выберите правильные ответы:

a) Каждый компьютер, подключенный к локальной сети, должен иметь сетевую карту
b) Одноранговые сети используются в том случае, если в локальной сети более 10 компьютеров
c) Сервер – это мощный компьютер, необходимый для более надежной работы локальной сети
d) Сеть на основе сервера – когда все компьютеры локальной сети равноправны

11) Выберите домен верхнего уровня в Интернете, принадлежащий России:

a) ra
b) ro
c) rus
d) ru

12) Интернет – это:

a) локальная сеть
b) корпоративная сеть
c) глобальная сеть
d) региональная сеть

13) Задан адрес сервера Интернета: www.mipkro.ru. Каково имя домена верхнего уровня?

a) www.mipkro.ru
b) mipkro.ru
c) ru
d) www

14) Для работы в сети через телефонный канал связи к компьютеру подключают:

a) адаптер
b) сервер
c) модем
d) коммутатор

15) Модем – это …, согласующее работу … и телефонной сети. Вместо каждого многоточия вставьте соответствующие слова:

a) устройство; программы
b) программа; компьютера
c) программное обеспечение; компьютера
d) устройство; дисковода
e) устройство; компьютера

16) Чтобы соединить два компьютера по телефонным линиям, необходимо иметь:

a) модем на одном из компьютеров
b) модем и специальное программное обеспечение на одном из компьютеров
c) по модему на каждом компьютере
d) по модему на каждом компьютере и специальное программное обеспечение
e) по два модема на каждом компьютере (настроенных, соответственно, на прием и передачу) и специальное программное обеспечение

17) Сети, объединяющие компьютеры в пределах одного региона:

a) локальные
b) региональные
c) корпоративные
d) почтовые

18) Сети, объединяющие компьютеры в пределах одной отрасли, корпорации:

a) локальные
b) региональные
c) корпоративные
d) почтовые

19) Компьютер, находящийся в состоянии постоянного подключения к сети:

a) хост-компьютер (узел)
b) провайдер
c) сервер
d) домен

20) Организация-владелец узла глобальной сети:

a) хост-компьютер (узел)
b) провайдер
c) сервер
d) домен

21) Выберите из предложенного списка IP-адрес:

a) 193.126.7.29
b) 34.89.45
c) 1.256.34.21
d) edurm.ru

22) Программное обеспечение, поддерживающее работу сети по протоколу TCP/IP:

a) базовое ПО
b) сервер-программа

c) клиент-программа

23) Программное обеспечение, занимающееся обслуживанием разнообразных информационных услуг сети:

a) базовое ПО
b) сервер-программа
c) клиент-программа

24) Internet Explorer – это:

a) базовое ПО
b) сервер-программа
c) клиент-программа

25) Согласно этому протоколу передаваемое сообщение разбивается на пакеты на отправляющем сервере и восстанавливается в исходном виде на принимающем сервере:

a) TCP
b) IP
c) HTTP
d) WWW

26) Доставку каждого отдельного пакета до места назначения выполняет протокол:

a) TCP
b) IP
c) HTTP
d) WWW

27) Обработка гиперссылок, поиск и передача документов клиенту – это назначение протокола:

a) TCP
b) IP
c) HTTP
d) WWW

28) Каждый отдельный документ, имеющий собственный адрес, называется:

a) Web-страницей
b) Web-сервером
c) Web-сайтом
d) Web-браузером

29) Компьютер, на котором работает сервер-программа WWW, называется:

a) Web-страницей
b) Web-сервером
c) Web-сайтом
d) Web-браузером

30) Web-сайт – это:

a) совокупность взаимосвязанных страниц, принадлежащих какому-то одному лицу или организации
b) сеть документов, связанных между собой гиперссылками
c) компьютер, на котором работает сервер-программа WWW
d) отдельный файл, имя которого имеет расширение .htm или .html

31) Web-браузер – это:

a) совокупность взаимосвязанных страниц, принадлежащих какому-то одному лицу или организации

b) сеть документов, связанных между собой гиперссылками

c) компьютер, на котором работает сервер-программа WWW

d) клиент-программа WWW, обеспечивающая пользователю доступ к информационным ресурсам Интернета

32) Режим связи с Web-сервером:

a) on-line режим

b) off-line режим

33) Автономный режим:

a) on-line режим

b) off-line режим

34) Если выбран режим сохранения документа «как текстовый файл». Тогда:

a) сохраняется только текст Web-страницы без каких-либо элементов оформления и форматирования

b) сохраняется текст со всеми элементами форматирования, не сохраняются встроенные объекты

c) сохраняется документ, в отдельной папке сохраняются файлы со всеми встроенными объектами

35) Если выбран режим сохранения документа «как документ HTML». Тогда:

a) сохраняется только текст Web-страницы без каких-либо элементов оформления и форматирования

b) сохраняется текст со всеми элементами форматирования, не сохраняются встроенные объекты

c) сохраняется документ, в отдельной папке сохраняются файлы со всеми встроенными объектами

36) Если выбран режим сохранения документа «как Web-страница полностью». Тогда:

a) сохраняется только текст Web-страницы без каких-либо элементов оформления и форматирования

b) сохраняется текст со всеми элементами форматирования, не сохраняются встроенные объекты

c) сохраняется документ, в отдельной папке сохраняются файлы со всеми встроенными объектами

37) Что означают буквы в URL-адресе Web-страницы: HTTP?

a) протокол, по которому браузер связывается с Web-сервером

b) имя пользователя в сети

c) адрес сервера в сети Internet

38) Что такое гиперссылка?

a) текст, выделенный жирным шрифтом

b) выделенный фрагмент текста

c) примечание к тексту

d) указатель на другой Web-документ

39) Назначение Web-серверов:

a) хранение гипертекстовых документов
b) подключение пользователей к сети Internet
c) хранение файловых архивов
d) общение по сети Internet

40) Web-страница имеет расширение:
a) .txt
b) .doc
c) .htm
d) .exe

41) В URL-адресе Web-страницы http://www.mipkro.ru/index.htm имя сервера - это:
a) http
b) www.mipkro.ru
c) index.htm
d) http://www.mipkro.ru/index.htm

42) В URL-адресе Web-страницы http://www.mipkro.ru/index.htm имя файла - это:
a) http
b) www.mipkro.ru
c) index.htm
d) http://www.mipkro.ru/index.htm

43) Основой связи в сети Интернет является набор сетевых протоколов –
a) IP
b) TPI/PC
c) TCP
d) TCP/IP

44) Собственником Интернета является –
a) Правительство США
b) Министерство Обороны США
c) Консорциум всемирной паутины (W3C)
d) У Интернета нет собственника

45) Интернет-протокол это –
a) правила передачи данных между узлами компьютерной сети
b) «язык» используемый компьютерами для обмена данными при работе в сети
c) особая настройка сервера в Интернете
d) способ скачивания информации из сети

46) TCP/IP это –
a) сток протоколов
b) набор сетевых протоколов
c) стек протоколов
d) снэк протоколов

47) Протокол передачи гипертекста это –

a) FTP
b) HTTP
c) SMTP
d) DHCP

48) Протокол передачи файлов это –
a) FTP
b) DHCP
c) SMTP
d) HTTP

49) Для получения сообщений электронной почты с сервера используется протокол –
a) SMTP
b) POP
c) HTTP
d) POP3

50) Простой протокол передачи почты это –
a) SMTP
b) POP
c) ESMTP
d) POP3

51) Каждый компьютер в рамках сети Интернет имеет –
a) уникальный адрес
b) уникальную конфигурацию
c) уникальный маршрутизатор
d) уникальный серийный номер

52) Какой из этих ip-адресов не может существовать –
a) 195.250.32.116
b) 195.1.32.116
c) 195.1.32.255
d) 195.1.32.256

53) Телекоммуникационная технология, разработанная с целью предоставления универсальной беспроводной связи на больших расстояниях для широкого спектра устройств это –
a) WiFi
b) HiFi
c) WiMax
d) WiFax

54) WWW это –
a) это адрес сайта

b) это всемирная паутина
c) это аббревиатура World Wide Web
d) это аббревиатура Wild Wild West

55) Web-сайт это –

a) web-страницы объединенные общей темой, дизайном, а также связанные между собою ссылками
b) документы размещенные во всемирной паутине
c) web-страница которую закачали в Интернет
d) тоже самое что и web-сервер

56) Организация, разрабатывающая и внедряющая технологические стандарты для Всемирной паутины это –

a) W4C
b) W3C
c) W2C
d) W1C

57) Домен первого уровня это –

a) .ru
b) igumo.ru
c) org.ru
d) first.igumo.ru

58) Услуга хранения сайта на web-сервере называется -

a) Джостинг
b) Мостинг
c) Гостинг
d) Хостинг

59) Основными критерием качества работы поисковой машины является -

a) Релевантность
b) Адекватность
c) Скорость работы
d) Большая база

60) Интернет это –

a) Всемирная паутина
b) Большая сеть
c) Глобальное объединение сетей
d) Массив информации

61) Какой html-тег создает поле ввода?

a) <text>
b) <textfield>
c) <select>

d) <input>
e) <textbox>

62) Как сделать ширину таблицы на Полный экран?

a) <table width=”100%”> </table>
b) <table width=”fulscreen”></table>
c) <table size=”100”> </table>
d) <table width=”widescreen”> </table>

63) Какое значение атрибута type используется при создании кнопки очистки данных формы?

a) type=”submit”
b) type=”clear”
c) type=”reset”
d) type=”clearbutton”
e) type=”resetbutton”

64) В форме требуется поставить поле для пароля. Какой код для этой цели потребуется?

a) <input type=”pass”>
b) <password>
c) <input type=”password”>
d) <input type=”hidden”>
e) <input type=”text” display=”hidden”>

65) Что из перечисленного позволяет сделать тег <META>?

a) указать кодировку документа
b) импортировать скрипты из файла
c) указать дополнительную информацию о сайте
d) задать тип HTML документа

66) Какое расположение тегов является правильным?

a) Я изучаюHTML
b) Я изучаюHTML
c) Я изучаюHTML
d) Я изучаюHTML

67) Каким образом можно отметить нужную опцию в выпадающем списке (select) используя HTML? Укажите все валидные варианты.

a) <option selected="chosen">
b) <option selected="selected">
c) <option selected="true">
d) <option selected="select">
e) <option selected>

68) Выберите теги, которые могут использоваться при создании таблицы

a) <th>
b) <head>
c) <tr>
d) <table>
e) <td>
f) <cell>
g) <row>

69) Как выглядит тэг <form> на странице?

a) тэг <form> представляет из себя поле для ввода текста

b) тэг <form> не имеет собственного графического представления - это контейнер для других элементов

c) тэг <form> не имеет собственного графического представления - это контейнер для текста

d) тэг <form> представляет из себя выдающий список

70) Для чего используется тег <TITLE>

a) Определяет красную строку в тексте.

b) Определяет заголовок документа.

c) Определяет заголовок таблицы.

d) Определяет заголовок в тексте.

71) Выберите html-код, создающий checkbox.

a) <checkbox/>

b) <check/>

c) ☐

d) <select type="toggler"/>

e) <input type="check"/>

72) Что определяет параметр colspan тега <td>?

a) объединяет ячейки по вертикали

b) задает выравнивание внутри ячейки

c) задает отступ для колонок

d) определяет количество колонок в строке

e) объединяет ячейки по горизонтали

73) Укажите правильный вариант создания гипертекстовой ссылки в html.

a) quizful

b) quizful

c) quizful

d) <a>http://www.quizful.net

e) quizful

74) Какой атрибут тега <td> указывает количество строк, занимаемых ячейкой?

a) rowspan
b) colspan
c) cols
d) rows
e) vcells

75) Какой html-тег создает поле ввода?
a) <text>
b) <textfield>
c) <select>
d) <input>
e) <textbox>

76) Выберите HTML-тег, выводящий текст в виде верхнего индекса выше базовой линии текста
a) <up>
b) <top>
c) <upper>
d) <sup>
e) <base-up>

77) Какие записи являются корректными для ссылки, когда исходный документ хранится в подкаталоге корня, а ссылаемый в корне сайта? (Выберите все верные варианты)
a) Локальная ссылка
b) Локальная ссылка
c) Локальная ссылка
d) Локальная ссылка
e) Локальная ссылка

78) Содержимое одного и того же HTML-документа отображается одинаково во всех браузерах.
a) Да
b) Нет

79) Что такое якорь (anchor)?
a) Это любой статический текст на веб-странице.
b) Это локальная переменная в форме запроса
c) Это закладка с уникальным именем в определенном месте веб-страницы
d) Это ссылка на часто открываемую страницу на сайте

80) Какое значение по умолчанию имеет атрибут method у тега <form>?
a) GET
b) POST

c) Атрибут method является обязательным и не имеет значения по умолчанию

d) Значение по умолчанию зависит от браузера, поэтому следует всегда указывать его явно

81) Для чего используется атрибут target тега <a>?

a) задает адрес документа, по которому следует перейти

b) устанавливает имя якоря внутри документа

c) задает имя окна или фрейма, куда браузер будет загружать документ

d) добавляет всплывающую подсказку к тексту ссылки

e) атрибут target недопустим для тега <a>

82) Как создать поле для отправки файлов?

a) <input title="file" name ="resume" />

b) <input type="file" name ="resume" />

c) <input alt="file" name ="resume" />

d) <input form="file" name ="resume" />

e) <input type='text' item='file' name='resume' />

83) Как создать ссылку на E-Mail?

a) <mail>xxx@yyy</mail>

b) xxx@yyy

c) xxx@yyy

d) <mail href="xxx@yyy" />

e) <link email="xxx@yyy">xxx@yyy</link>

84) Атрибут size тега <input type="text"> устанавливает:

a) Размер поля ввода по горизонтали в символах

b) Максимальное значение вводимых символов

c) Размер поля ввода по горизонтали в пикселях

d) Размер поля ввода по вертикали в символах

e) Размер поля ввода по вертикали в пикселях

85) В каких случаях ссылка указывает на закладку (укажите все подходящие варианты)?

a) link

b) link

c) link

d) link

e) link

86) Какое значение атрибута target надо использовать при написании гиперссылки, чтобы по клику она загружалась в новом окне?

a) newwindow

b) top
c) self
d) blank

87) Выберите фрагмент HTML-кода, создающий ссылку со всплывающей подсказкой

a) текст ссылки
b) текст ссылки
c) текст ссылки
d) текст ссылки
e) текст ссылки

88) Для чего нужны ключевые слова, задаваемые через тег <meta>?

a) Для облегчения поиска по странице для браузера.
b) Для индексации слов и предложений текущего документа.
c) Для хранения в закладках браузера, чтобы быстрее находить нужный документ.
d) Для подсказок поисковым системам.
e) Для ускорения загрузки HTML-файла.

89) Какая из приведенных примеров ссылок является абсолютной?

a) Изучение HTML
b) Ссылка
c) Ссылка
d) Ссылка

90) Какая из приведенных примеров ссылок является относительной?

a) Ссылка
b) Изучение HTML
c) Ссылка
d) Ссылка

91) Выберите фрагмент HTML-кода, создающий ссылку, если исходный файл располагается в корне сайта, а файл, на который необходимо сделать ссылку - в папке.

a) Ссылка
b) Ссылка
c) Ссылка
d) Ссылка
e) Ссылка

92) Выберите фрагмент HTML-кода, создающий ссылку, если исходный файл находится в двух вложенных папках, и чтобы сослаться на документ в корне сайта.

a) Ссылка
b) Ссылка
c) Ссылка
d) Ссылка
e) Ссылка

93) Выберите фрагмент HTML-кода, создающий ссылку, если исходный документ хранится в одной папке, а ссылаемый в корне сайта
a) Ссылка
b) Ссылка
c) Ссылка
d) Ссылка
e) Ссылка

94) Выберите фрагмент HTML-кода, создающий ссылку, если необходимо сделать ссылку из исходного документа на ссылаемый.
a) Ссылка
b) Ссылка
c) Ссылка
d) Ссылка
e) Ссылка

95) Выберите фрагмент HTML-кода, создающий ссылку, если исходный файл располагается в корне сайта, а файл, на который необходимо сделать ссылку – внутри двух папок.
a) Ссылка
b) Ссылка
c) Ссылка
d) Ссылка
e) Ссылка

96) Элемент предназначенный для указания типа текущего документа.
a) <!DOCTYPE>
b) <HTML>
c) <HEAD>

97) Программа для проверки синтаксиса HTML и выявления ошибок в коде.
a) Текстовый редактор
b) Валидатор
c) Вьювер
d) Браузер

98) Какой атрибут определяет ширину промежутков между ячейками в пикселах.
a) Cellspacing

b) Valign
c) Cellpadding
d) Rules

99) Какой атрибут определяет ширину пустого пространства между содержимым ячейки и ее границами.

a) Cellspacing
b) Valign
c) Cellpadding
d) Rules

100) Какой атрибут определяет вид сетки между ячейками.

a) Cellspacing
b) Valign
c) Cellpadding
d) Rules

Назначение и применение CSS

Темы лабораторных работ

1. Создать внешние таблицы стилей (раздельные для устройств screen, print и handheld) для разрабатываемого вами сайта.
2. Подключить созданные таблицы к макету страницы.

Тестовые задания

1) Как изменить цвет ссылки на синий при наведении на нее курсором?

a) a:link {color:blue;}
b) a:active {color:blue;}
c) a:hover {color:blue;}

2) Выберите вариант CSS-кода, который устанавливает жирный шрифт для всех элементов <p>.

a) p { font-size: bold; }
b) p { font-weight: bold; }
c) p { font-style: bold; }
d) p { text-weight: bold }
e) p { text-style: bold; }

3) Каким свойством CSS можно задать шрифт для вывода текста?

a) font-face
b) font-family
c) font-name
d) font-type
e) font-glyph

4) Выберите фрагмент CSS, который не содержит синтаксических ошибок.

a) body { color: black }
b) body:color=black
c) {body:color=black}
d) {body;color:black}
e) body (color: black)

5) Как добавить цвет фона для всех элементов <h1>?

a) all.h1 {background-color:#FFFFFF}
b) h1.all {background-color:#FFFFFF}
c) h1 * {background-color:#FFFFFF}
d) h1 {background-color:#FFFFFF}
e) $$("h1").each(function(h1) { h1.style['background-color']="#FFFFFF"; });

6) Что в CSS делает свойство "font-style"?

a) определяет шрифт текста
b) определяет начертание шрифта
c) определяет размер шрифта
d) определяет толщину шрифта

7) Как с помощью CSS убрать подчеркивание со всех ссылок?

a) a { underline: none; }
b) a { text-decoration:none; }
c) a { decoration:none; }
d) a { decoration: no underline; }
e) a { text: no underline; }

8) Каким образом с помощью CSS можно центрировать inline контент блочного элемента (например,) по горизонтали?

a) halign:center;
b) content-align:center;
c) text-align:center;
d) align:center;
e) horizontal-align:center;

9) Какое значение свойства clear по умолчанию?

a) left
b) right
c) none
d) both

10) Каким свойством CSS можно задать высоту строки текста?

a) text-list-height
b) font-height
c) text-size

d) text-height
e) line-height

11) Каким свойством CSS задается внешний левый отступ от элемента?
a) margin-left
b) insets-left
c) text-indent
d) spacing-left
e) padding-left

12) Что такое таблица стилей?
a) Таблица стилей - это набор правил отображения, применяемых в документе, к которому присоединена соответствующая таблица стилейправляют внешним видом документа.
b) Таблица стилей - это набор таблиц, предназначеный для украшения сайта
c) Таблица стилей - это набор тегов

13) Выберите правильный вариант:
a) ТЭГ {свойство}
b) ТЭГ {свойство: значение}
c) ТЭГ {значение}
d) {свойство: значение} ТЭГ
e) Нет правильного варианта

14) Как можно задавать селекторы, свойства и их параметры
a) процентами
b) строчными буквами
c) прописными буквами
d) как строчными, так и прописными буквами, или в смешанном порядке
e) только цифрами.

15) Из каких элементов(а) состоит правило каскадных таблиц?
a) Селектор
b) Определение
c) Оператор
d) Аргумент
e) Селектор и определение

16) Что в приведенном примере является селектором?
a) H1
b) color
c) blue
d) B font-size
e) нет селектора

17) Таблица стилей представляет собой набор правил форматирования элементов:
   a) HTML
   b) JavaScript
   c) C++
   d) HTTP
   e) Неправильное утверждение
18) Что обозначает * в примере: * {font-weight: bold}?
   a) Что стилевое правило должно быть применено ко всем элементам документа.
   b) Такого не может быть
   c) Что элемент описывается позже
   d) Что стилево правило должно быть применено к элементу с именем *.
19) Для чего придумана в CSS дериктива @media?
   a) Для отображения документов на различных устройствах
   b) Синтеаторы речи
   c) Принтер
   d) Телевизоры
   e) Такой дериктивы не существует
20) Что делает в CSS псевдокласс hover?:
   a) Определяет, как отображать объект, когда на него попал курсор мышь
   b) Псевдоклассов не существует
   c) указывает, как отображать содержимое на некотором языке
   d) Определяет, как броузеру показать содержимое тега, ставшего объектом внимания.
   e) ределяют стиль отображения обычной ссылки
21) Как отделяется имя ID-класса от имени тега?
   a) :
   b) ID
   c) #
   d) -
   e) ID классов не существует.
22) Каким тегом определяются стили на уровне документа?
   a) Style
   b) Head
   c) link
   d) import.
23) Какой селектор состоит из простых селекторов, разделенных пробелом.
   a) Обычный селектор.
   b) Контекстный селектор.

c) Селектор ID.
d) Селектор CLASS.
e) Селектор параметров.
f) Универсальный селектор.

24) Определяет уникальное имя элемента, которое используется для изменения его стиля и обращения к нему через скрипты.

a) Обычный селектор.
b) Контекстный селектор.
c) Селектор ID.
d) Селектор CLASS.
e) Селектор параметров.
f) Универсальный селектор.

25) Применяются, когда необходимо определить стиль индивидуального элемента веб-страницы или задать разные стили для одного тега.

a) Обычный селектор.
b) Контекстный селектор.
c) Селектор ID.
d) Селектор CLASS.
e) Селектор параметров.
f) Универсальный селектор.

26) Устанавливает стиль для элемента, если задан специфичный атрибут тега.

a) Простой селектор атрибута
b) Атрибут со значением
c) Контекстный селектор
d) Универсальный селектор

27) Устанавливает стиль для элемента в том случае, если задано определенное значение специфичного атрибута.

a) Простой селектор атрибута
b) Атрибут со значением
c) Контекстный селектор
d) Универсальный селектор

28) Устанавливает одновременно один стиль для всех элементов веб-страницы.

a) Обычный селектор
b) Контекстный селектор
c) Селектор CLASS
d) Селектор параметров
e) Универсальный селектор

29) Что называется «селектором»?

a) Селектором называют имя стиля, для которого указаны параметры форматирования
b) Селектором называют тег с указанием параметров форматирования
c) Селектором называют часть html-кода, расположенная в контейнере <head>…</head>
d) Селектором называют имя файла, имеющего расширение .css

30) Как можно иначе записать следующий фрагмент:

```
p{color: green;}
h2{color: green; font-family: Courier;}
span{color: green;}
```

a) p h2 span {color: green;}
b) p, h2, span {color: green;}
c) p h2 span {color: green;} h2 {font-family: courier;}
d) p, h2, span {color: green;} h2 {font-family: courier;}

31) Для того, чтобы в списке отменить задание маркера, необходимо написать:
a) list-style-type: circle
b) list-style-type: none
c) list-style-type: inherit
d) list-style-type: decimal

32) Каким свойством можно задать цвет фона документа?
a) color
b) bgcolor
c) background
d) content

33) Как записываются комментарии в css?
a) << /комментарий >>
b) /* комментарий */
c) <<! комментарий >>
d) /*! комментарий */

34) Укажите пример группировки селекторов.
a) H1>h3 em {color: violet;}
b) H1 h3 em {color: violet;}
c) H1, h3, em {color: violet;}
d) H1+h3+em {color: violet;}

35) Каким свойством можно задать красную строку?
a) before
b) text-decoration
c) text-indent
d) text-align

36) Выберите верный комментарий к следующему примеру: strong+em, h2 {color: #ccfcff; font-size: 15pt;}

a) Для всех заголовков второго уровня и для всех абзацев установлен соответствующий цвет и размер шрифта.
b) Для всех заголовков второго уровня и содержимого всех контейнеров …, задан цвет и размер шрифта.
c) Для всех заголовков второго уровня и содержимого всех контейнеров … встречающихся внутри , задан цвет и размер шрифта.
d) Для всех заголовков второго уровня, а также содержимого всех контейнеров …, находящихся непосредственно после , задан цвет и размер шрифта.

37) Найдите пример, составленный без ошибок.

a) div {color: green; font-type: 10pt; font-family: arial;}
b) p {font-color: green; font-size: 10pt; font-family: arial;}
c) h2 {font-color: green; size: 10pt; font-family: arial;}
d) span {color: green; font-size: 10pt; font-family: arial;}

38) Найдите строку, где перечислены только псевдоэлементы.

a) after, before, first-letter, vertical-align
b) first-line, padding, visited, word-spacing
c) first-line, after, before, first-letter
d) after, before, visited, vertical-align

39) Укажите пример, где данный стиль применится ко всем элементам html-документа.

a) p.all {font-weight: bold; color: #453346;}
b) .all {font-weight: bold; color: #453346;}
c) *.all {font-weight: bold; color: #453346;}
d) *{font-weight: bold; color: #453346;}

40) В каком из перечисленных вариантов, стиль b>em{color:red;} применится к содержимому тега .

a) Жил на свете старичок, маленького роста. И смеялся старичок, чрезвычайно просто: хи-хи-хи, да хо-хо-хо …
b) Жил на свете старичок, маленького роста. И смеялся старичок, чрезвычайно просто: хи-хи-хи, да хо-хо-хо …
c) Жил на свете старичок, маленького роста. И смеялся старичок, чрезвычайно просто: хи-хи-хи, да хо-хо-хо …
d) Жил на свете старичок, маленького роста. И смеялся старичок, чрезвычайно просто: хи-хи-хи, да хо-хо-хо …

41) Какое расширение имеет файл с набором стилей (внешняя таблица стилей)?

a) html

b) xhtml
c) css
d) xml

42) Укажите верный способ подключения внешней таблицы стилей с именем Style.first.css к html-документу.

a) <link rel="stylesheet" type="text/css" href="Style.first.css" />
b) <link rel="style/sheet" type="text/css" href="Style.first.css" />
c) <link rel="stylesheet" type="text/html" href="Style.first.css" />
d) <link rel="style-sheet" type="text/css" href="Style.first.css" />

43) Выберите вариант, где указан стиль с использованием класса.

a) p, h2 {font-style: italic; background: yellow;}
b) h2.p {font-style: italic; background: yellow;}
c) p#h2 {font-style: italic; background: yellow;}
d) h2@p {font-style:italic; background: yellow;}

44) Что делает следующий стиль? OL LI:first-letter {color: red;}

a) Изменяет цвет первой буквы элемента маркированного списка.
b) Изменяет цвет первой буквы элемента нумерованного списка.
c) Изменяет цвет первой строки в маркированном списке.
d) Изменяет цвет первой строки в нумерованном списке.
e) Изменяет цвет текста всего списка.

45) Какое из CSS свойств используется для выставления размеров текста?

a) font-style
b) text-style
c) text-size
d) font-size

46) Как сделать жирными текст параграфов?

a) <p style=»text-size:bold;»>
b) <p style=»font-size:bold;»>
c) p {font-weight:bold;}
d) p {text-size:bold;}

47) Есть такой код: ul li:first-letter {font-size: 200%;}. Что он делает?

a) Делает первую букву у первого элемента в нумерованном списке размером 200%.
b) Ничего не делает, так как псевдоэлемента first-letter вообще не существует.
c) Делает первую букву у каждого элемента ненумерованного списка размером 200%.
d) Делает первую букву у первого элемента в ненумерованном списке размером 200%.

48) В каком из представленных ниже вариантов содержится явная ошибка:

a) p span#text (font-size: 150%;}
b) p text (font-size: 150%;}
c) p {font-size: 150%;}
d) p span {font-size: 150%;}

49) Есть такой HTML-код: Синий не синий . Какой CSS-код внутри тега сделает синий цвет:

a) Все представленные варианты подойдут
b) p span {color: blue;}
c) span {color: blue;}
d) .blue (color: blue;}

50) Увеличивает приоритет стиля

a) after
b) :first-child
c) :active
d) !important

Язык клиентских сценариев JavaScript

Темы лабораторных работ

1. Создайте страницу, которая спрашивает имя и выводит его.

2. Напишите скрипт, который сначала выводит на экран диалоговое окно, а затем, после нажатия кнопки "ОК", в окне браузера пишет фразу "Привет, Мир!".

3. Напишите скрипт, который запрашивает у пользователя информацию, а затем выводит ее в диалоговом окне.

4. Составьте документ так, чтобы диалоговое окно для ввода информации предлагалось только после наведения курсора мыши на ссылку, и введенная пользователем текстовая строка выводилась бы в виде диалогового окна, или в окно браузера.

5. Создайте программу для вычисления площади и периметра правильного n-угольника, описанного около окружности радиуса R.

6. Создайте программу для вычисления площади прямоугольника по введенным в диалоге двум сторонам.

7. Создайте программу для вычисления длины окружности и площади круга по указанному радиусу.

8. Создайте программу для вычисления длин высот треугольника, у которого длины сторон равны A, B, C.

9. Создайте программу для вычисления работы, совершенной при равномерном подъеме груза массой ***m*** [кг] на высоту ***h*** [м]. Ускорение свободного падения опишите как константу ***g*** = 9,81.

10. Создайте программу, вычисляющую периметр и площадь треугольника по введенным в диалоге длинам трех сторон, откомпилируйте ее и проверьте ее работу.

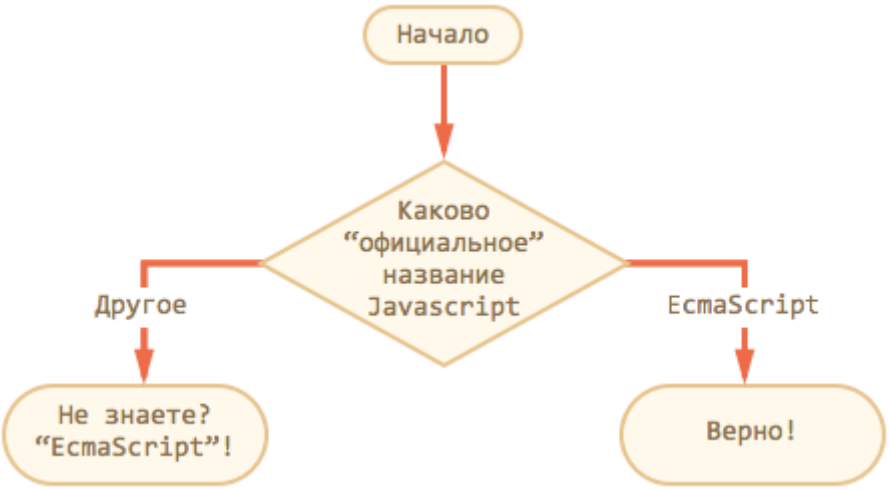
11. Создайте программу, вычисляющую по введенному значению радиуса длину окружности и ее площадь. Для вычисления значения числа $\pi \approx 3{,}141592...$ в JS используется метод Math.PI.

12. Создайте программу, вычисляющую периметр квадрата по указанному значению его площади.

13. Создайте программу, определяющую, сколько времени в минутах затратит школьник на дорогу от школы до стадиона, если известна длина этого расстояния ***s*** и средняя скорость движения школьника ***v*** [км/ч]? Значения ***s*** и ***v*** вводятся с клавиатуры.

14. Используя конструкцию if..else, напишите код, который будет спрашивать: «Каково «официальное» название JavaScript?». Если посетитель вводит «EcmaScript», то выводить «Верно!», если что-то другое — выводить «Не знаете? «EcmaScript»!».

Блок-схема:



15. Используя конструкцию if..else, напишите код, который получает значение prompt, а затем выводит alert:

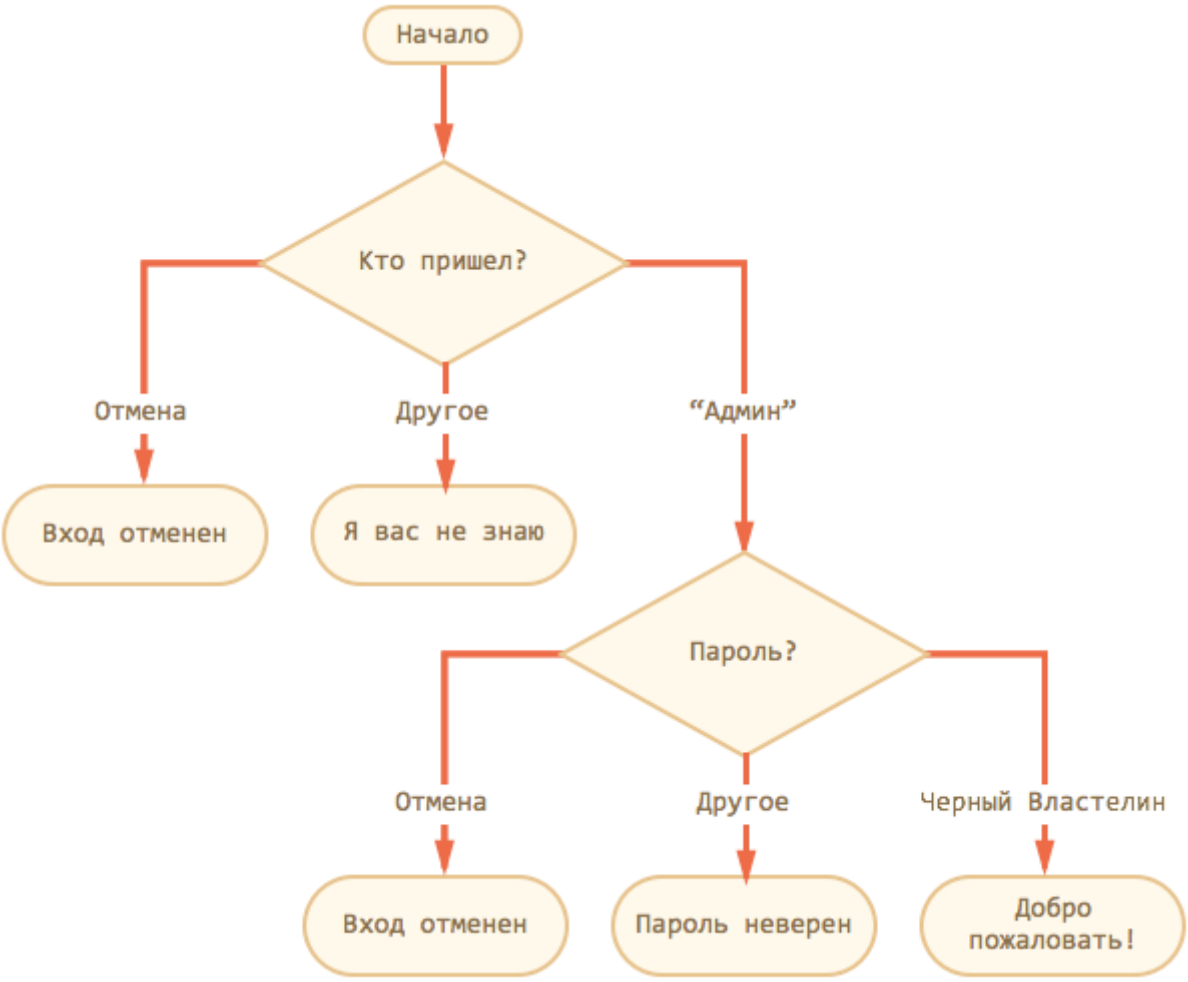
- 1, если значение больше нуля,
- -1, если значение меньше нуля,
- 0, если значение равно нулю.

16. Напишите код, который будет спрашивать логин (prompt).

Если посетитель вводит «Админ», то спрашивать пароль, если нажал отмена (escape) — выводить «Вход отменён», если вводит что-то другое — «Я вас не знаю».

Пароль проверять так. Если введён пароль «Чёрный Властелин», то выводить «Добро пожаловать!», иначе — «Пароль неверен», при отмене — «Вход отменён».

Блок-схема:



Для решения используйте вложенные блоки if. Обращайте внимание на стиль и читаемость кода.

17. Перепишите if с использованием оператора '?':

```
if (a + b < 4) {
  result = 'Мало';
} else {
  result = 'Много';
}
```

18. Перепишите if..else с использованием нескольких операторов '?'. Для читаемости — оформляйте код в несколько строк.

```
var message;

if (login == 'Вася') {
  message = 'Привет';
} else if (login == 'Директор') {
  message = 'Здравствуйте';
} else if (login == '') {
  message = 'Нет логина';
} else {
  message = '';
}
```

19. Написать программу, которая по паролю определяет уровень доступа сотрудника к секретной информации в базе данных. Доступ к базе данных имеют только 6 сотрудников, разбитых на три группы по уровням доступа. Они имеют следующие пароли: 9583, 1747 – доступны модули баз A, B и C; 3331, 7922 - доступны модули баз B и C; 9455, 8997 – доступен модуль базы C.

20. Дана последовательность слов. Напечатать все слова, предварительно выполнив преобразования по правилу: заменить в каждом слове первую встречную букву а буквой о, удалив все остальные (если в слове нет такой буквы, то ничего не делать).

21. Даны два вещественных положительных числа x и y. арифметические действия над ними пронумерованы (сложение – 1, вычитание – 2, умножение – 3, деление - 4). Составить программу, которая по введённому номеру выполняет то или иное действие над числами.

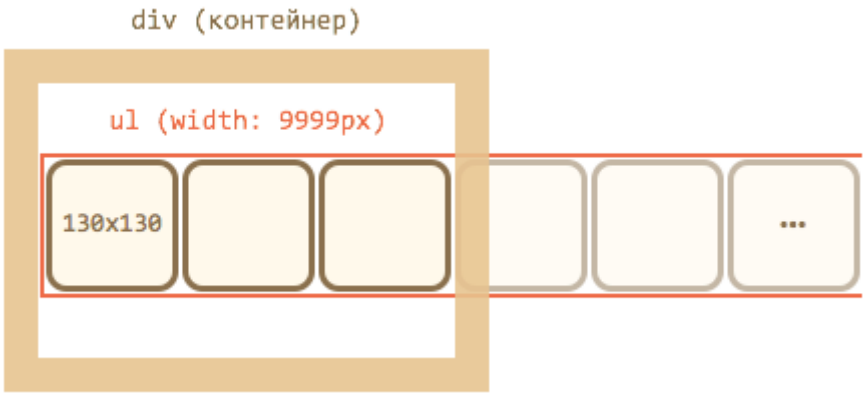
22. Создайте программу, которая для введенного целого числа *k* (от 1 до 99) напечатает фразу «Мне *k* лет», где *k* — введенное число, заменяя в нужных случаях слово «лет» словом «год» или «года». (Например, при *k* = 70 — «Мне 70 лет», при *k* = 15 — «Мне 15 лет», при *k* = 23 — «Мне 23 года» и т.п.)

23. Напишите программу-модель анализа работы пожарного датчика в помещении, которая выводит сообщение «Пожароопасная ситуация», если температура (в нашей модели она будет вводиться с клавиатуры) в комнате превысила 60 °C.

24. Напишите «Карусель» – ленту изображений, которую можно листать влево-вправо нажатием на стрелочки.

Лента изображений в разметке должна быть представлена как список <ul> тегов <img>.

Нужно расположить его внутри фиксированного размера, так чтобы в один момент была видна только нужная часть списка:

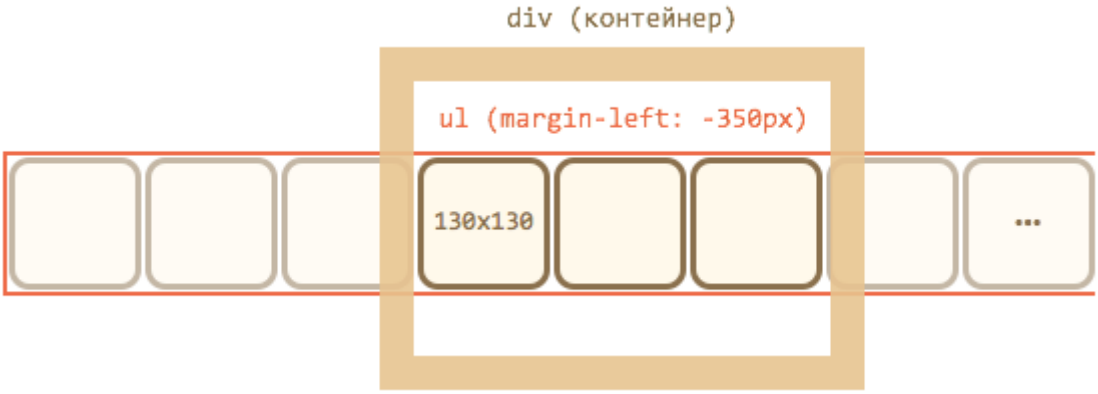


Чтобы список был длинный и элементы не переходили вниз, ему ставится width: 9999px, а элементам <li>, соответственно, float:left, либо для элементов используется display: inline-block, как в этом решении.

Главное – не использовать display:inline, так как такие элементы имеют дополнительные отступы снизу для возможных «хвостов букв».

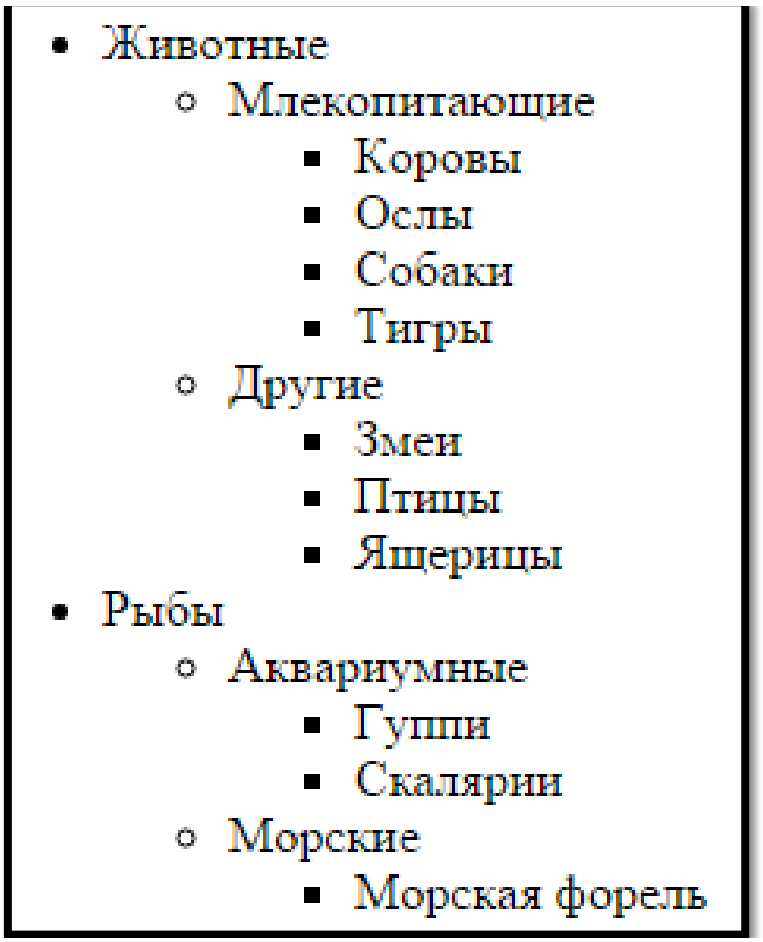
В частности, для <img> нужно поставить в стилях явно display:block, чтобы пространства под ними не оставалось.

Для «прокрутки» будем сдвигать <ul>. Это можно делать по-разному, например, назначением CSS-свойства transform: translateX() или margin-left:



У внешнего фиксированная ширина, поэтому «лишние» изображения обрезаются. Снаружи окошка находятся стрелки и внешний контейнер.

25. Создайте дерево, которое по клику на заголовок скрывает-показывает детей:



Требования:

- Использовать делегирование.
- Клик вне текста заголовка (на пустом месте) ничего делать не должен.
- При наведении на заголовок – он становится жирным, реализовать через CSS.

P.S. При необходимости HTML/CSS дерева можно изменить.

26. Для HTML-страницы содержащую форму добавьте скрипт, проверяющий следующие данные:

- заполнено ли поле имени,
- введен ли пароль и содержит ли он больше 4-х символов.

Используйте для этого свойство length данного поля,

- совпадают ли значения, введенные в оба поля для паролей,
- заполнено ли поле электронного адреса и содержит ли оно символ @,
- заполнено ли поле сообщения и содержит ли оно больше 10 символов,

При несоблюдении условий, курсор должен установиться в то поле, где пользователем введено неверное значение.

27. Разработать страницу для оформления заказа товаров.

28. Реализовать форму для ввода наименования и стоимости. При нажатии на кнопку «Добавить», добавить введенные значения в таблицу.

29. Сделать кнопку расчёта стоимости заказа.

30. Поставить обработчик на изменение фона ячеек таблицы со стоимостью в тот момент, когда над ними находится указатель мыши.

31. Подготовить страницу, содержащую таблицу товаров и их стоимости. На базе этой страницы обеспечить возможность выбора строк и добавления их в таблицу формирования заказа.

Тестовые задания

1) Какой оператор из этих - выполняет не только математические операции?
   a) *
   b) /
   c) +
   d) -
   e) >>>
2) Что из этого - не событие мыши?
   a) onmousescroll
   b) onclick
   c) onmouseover
   d) onmousemove
   e) onmousewheel
3) Какие конструкции для циклов есть в javascript?
   a) Только две: for и while.
   b) Только одна: for
   c) Три: for, while и do...while.
4) В каком случае из перечисленных событие не попадет на обработку javascript?
   a) Если в момент его наступления обрабатывается другое событие
   b) Если страничка просматривается локально, т.е offline
   c) Только если javascript отключен
5) Что делает код: break me;
   a) Ломает интерпретатор javascript
   b) Выходит из текущего блока цикла или switch на метку "me"
   c) Выдает ошибку
   d) В разных браузерах по-разному
6) Какое событие не вызывается кликом мыши?
   a) onfocus
   b) onclick
   c) onkeydown
   d) onmousedown
7) Какие из этих вариантов задают массив из элементов "a", "b" ?
   a) var a = new Array("a","b")
   b) var a = {"a", "b"}

c) var a = ("a", "b")
d) var a = ["a", "b"]
e) var a = "a,b".split(',')

8) Сколько параметров можно передать функции?
a) Ровно столько, сколько указано в определении функции
b) Сколько указано в определении функции или меньше
c) Сколько указано в определении функции или больше
d) Любое количество

9) Верно ли следующее утверждение: Содержимое тэга script нужно заключать в комментарии <!-- .. -->, чтобы браузеры, которые не поддерживают javascript, работали корректно?
a) Да
b) Нет

10) Какие варианты правильно объявляют переменную для f, возвращающей сумму двух аргументов?
a) var f = function(a,b) { return a+b }
b) var f = new Function('a,b', 'return a+b')
c) var f = new Function('a', 'b', 'return a+b')
d) Никакие.

11) Где в документе может располагаться тэг script по стандарту HTML?
a) Только в HEAD
b) Только в BODY
c) В HEAD или в BODY
d) Где угодно, главное чтобы был

12) Как правильно вывести приветствие через 5 секунд после запуска скрипта?
a) sleep(5); alert("Привет!");
b) sleep(5000); alert("Привет!");
c) setTimeout('alert("Привет!")', 5000);
d) setTimeout(function() {alert("Привет!")}, 5000);
e) setTimeout(function() {alert("Привет!")}, 5);

13) Можно ли через javascript подключить внешний js-файл, отсутствующий на странице?
a) Да, но только один раз
b) Да, но только до загрузки страницы
c) Да, сколько угодно файлов когда угодно

14) Можно ли в скрипте перевести посетителя на другую страницу сайта?
a) Да, куда угодно
b) Да, но только в рамках текущего сайта
c) Нет, нельзя

15) Какие обработчики событий сработают при клике на div?

div.onclick = function() { alert(1) }

div.onclick = function() { alert(2) }

a) Только первый
b) Только второй
c) Первый, затем второй

16) a = new Array(1,2,3)
b = new Array(1,2,3)
Правда ли что a == b ?

a) Правда
b) Неправда
c) Зависит от множества случайных обстоятельств.

17) Каких бинарных операторов НЕТ в javascript?

a) *
b) ^
c) %
d) #
e) &
f) >>
g) >>>
h) !

18) Какая арифметическая операция приводит к ошибке в javascript?

a) Деление на ноль
b) Умножение числа на строку
c) Корень из отрицательного числа
d) Никакая из вышеперечисленных

19) Какова длина a.length массива a?

```
var a = []
a[1] = 5
a[5] = 55
```

a) 1
b) 2
c) 3
d) 4
e) 5
f) 6
g) Другое

20) Какова длина *a.length* массива *a*:

```
1 var a = []
2 a[1] = 5
3 a[3] = 53
4 delete a[3]
```

a) 1

b) 2
c) 3
d) 4
e) Больше

21) Что делает оператор = = =?
a) Сравнивает по ссылке, а не по значению
b) Сравнивает без приведения типа
c) Нет такого оператора

22) Чему равна переменная name?

```
var name = "пупкин".replace("п","д")
```

a) дудкин
b) дупкин
c) пупкин
d) ляпкин-тяпкин

23) Выберите JavaScript команду позволяющую вывести текст на страницу.
a) write('Текст выведен с помощью JavaScript')
b) document.write('Текст выведен с помощью JavaScript')
c) text('Текст выведен с помощью JavaScript')

24) Выберите комментарий использующийся в JavaScript.
a) //Я являюсь комментарием
b) <! -- Я являюсь комментарием -->
c) <? Я являюсь комментарием?>

25) Чувствителен ли JavaScript к регистру символов?
a) Да
b) Нет

26) Выберите JavaScript команду создающую строковую переменную.
a) var str=new Array("Строковая переменная")
b) var str="Строковая переменная"
c) var str=new Object("Строковая переменная")

27) Какое событие позволяет выполнять код после щелчка мыши?
a) mouseout
b) mouseclick
c) onmouseclick
d) onclick

28) Выберите синтаксически корректную команду для создания объекта JavaScript.
a) var obj=create Object
b) var obj=call Object
c) var obj=new Object()
d) var obj=Object()

29) Выберите метод JavaScript позволяющий выполнять произвольный код через заданные промежутки времени.

a) callCode()
b) timer()
c) setInterval()
d) setTimeOut()

30) Выберите синтаксически корректную условную конструкцию.

a) if (a==3) then document.write('Привет')
b) if (a==3) document.write('Привет')
c) if (a==3) make document.write('Привет')

31) Укажите название встроенного JavaScript объекта для работы с датой и временем.

a) datetime
b) calendar
c) date

32) Логическая операция «не» в JavaScript:

a) !
b) $
c) #
d) &

33) Многострочный комментарий в JavaScript

a) /* */
b) / /
c) * *
d) */ /*

34) Ввод-вывод данных в JavaScript можно осуществить с помощью функции

a) prompt
b) eval
c) sum
d) alert

35) Однострочный комментарий в JavaScript

a) //
b) **
c) ++
d) <--

36) Шестнадцатеричное число в JavaScript выглядит как:

a) 0XF8f00
b) 454545
c) A15
d) ddfds

37) Управляющий код \n в JavaScript имеет значение:

a) перехода на новую строку
b) возврата каретки

c) двойной кавычки
d) одинарной кавычки

38) Сравнение без преобразования типов в JavaScript выполняет функция:
a) ===
b) ==
c) +=
d) =-

39) Для маскировки сценариев JavaScript используется:
a) заключение скрипта в html комментарий <!-- -->
b) заключение скрипта в круглые скобки ()
c) заключение скрипта в квадратные скобки []
d) заключение скрипта в фигурные скобки {}

40) Битовая целочисленная операция «НЕ» в JavaScript:
a) ~
b) *
c) +-
d) +=

41) Логическая операция «ИЛИ» в JavaScript:
a) ||
b) \\
c) //
d) /*

42) Язык для составления скриптов, встраиваемых в web-страницы для придания им более привлекательного вида и динамичности?
a) VBScript
b) JavaScript
c) Java
d) Object Pascal

43) Для включения программ на JavaScript в HTML-страницу необходимы тэги:
a) <SCRIPT> … </SCRIPT>
b) <SCRIPT Java> … </SCRIPT Java>
c) <Java> … </Java>
d) < JavaScript> … </ JavaScript>

44) Выберите верное утверждение:
a) В JavaScript тип данных переменной не указывается
b) В JavaScript тип данных переменной обязательно должен быть указан
c) В JavaScript переменные нельзя использовать без предварительного описания?
d) Нет верных утверждений

45) Как определить в каком браузере запущен сайт у пользователя?
a) Это невозможно средствами JavaScript
b) Используя объект navigator
c) Используя объект window

d) Используя объект document
e) Все ответы не верны

46) Переменные JavaScript, которые видны только в рамках одной функции?
a) Глобальные
b) Локальные
c) Логические
d) Строковые

47) Переменные JavaScript, описанные в рамках какого-либо тэга <SCRIPT> … </SCRIPT>, и видны для всех функций из любых тэгов <SCRIPT> … </SCRIPT> на данной странице, и существуют, пока загружена страница?
a) Логические
b) Строковые
c) Локальные
d) Глобальные

48) Какая функция JavaScript преобразует текстовый тип данных в целочисленный?
a) IntToStr ( )
b) ParseInt ( )
c) ParseFloat ( )
d) StrToInt ( )

49) Какая функция JavaScript преобразует текстовый тип данных в числовой с плавающей точкой?
a) IntToStr ( )
b) ParseInt ( )
c) ParseFloat ( )
d) StrToInt ( )

50) Специальный объект, который отправляется в обработчик событий при возникновении любого события:
a) alert
b) event
c) prompt
d) нет верных ответов

51) Метод объекта Window, т.е. объект Window, который может выдавать пользователю какое-либо предупреждение в окне сообщений:
a) alert
b) prompt
c) confirm
d) event

52) При использовании, какого метода JavaScript, в появляющемся на экране окне пользователь может сделать выбор между кнопками OK и Cancel?

a) event
b) confirm
c) prompt
d) alert

53) Метод JavaScript, который дает пользователю возможность вписать собственный ответ на вопрос, вместо того чтобы просто выводить на экран информацию или делать выбор между кнопками OK и Cancel:

a) alert
b) event
c) confirm
d) prompt

54) Объект – это …

a) структура данных
b) экземпляр некоторого класса
c) класс

55) Класс – это …

a) тип данных
b) описание объектов
c) структура данных

56) Какое значение выведется на экран alert'ом?

```
var a = new Array();
a['key'] = 'value';
alert(a.length);
```

a) 0
b) 1
c) Возникнет ошибка интерпретатора
d) Ничего из вышеперечисленного

57) Что будет выведено в консоль в результате выполнения интерпретатором кода?

a) console.log("Hello!' +"_"+ "World!')
b) Hello!_World!
c) Ошибка
d) Hello! World!
e) Hello!_
f) _World!

58) Дана функция: function func(a) { return "1"; } Какие из следующих вызовов этой функции допустимы?

a) func(new Object());
b) func(func());
c) func();
d) func(1, 2);
e) func("1");

59) Какими html-тегами можно подключить внешний javascript (укажите все подходящие варианты)?

a) <include>
b) <javascript>
c) <link>
d) <js>
e) <script>

60) Выберите все варианты правильно составленных условных операторов.

a) if i == 5 {}
b) if (i == 5) then {}
c) if i == 5 then {}
d) unless (i != 5) {}
e) if (i == 5) {}
f) if i eq 5 then {}

61) Дан код:

```
switch ("a") {
    case "b":
        console.log("b");
        break;
    default:
        console.log("d");
    case "c":
        console.log("c");
}
```

Что будет выведено на консоль.

a) Ошибка
b) d
c) d c
d) b d c
e) undefined

62) Сколько разных ключевых слов для описания циклов доступно в javascript?

a) Одно: for
b) Два: for и while
c) Три: for, while и do...while
d) Четыре: for, while, do...while и foreach

63) Дан код:

```
function a(value) {
    return true - value;
}

var b = a('4') + a('-4') + a(-'4') + a(4);
console.log(b);
```

Что будет выведено в консоль?

a) "-4true+4true+44"
b) "true-4true+4true+4true-4"
c) 4
d) "-3true+4true+43"
e) 0
f) "-344-3"

64) Укажите имя функции округления вверх?

a) ceil
b) math.ceil
c) Math.ceil
d) math.ceil()
e) ceil()

65) Какое количество раз выполнится этот цикл?

```
for (var Counter = 0; Counter < 10; Counter = Counter + 1) {
  Counter++;
}
```

a) 0
b) 1
c) 2
d) 5
e) 10

66) Каков будет результат выполнения функции?

```
(function(){
    return typeof arguments;
})();
```

a) "object"
b) null
c) "undefined"
d) "array"
e) "function"

67) Что выведет следующий код:

```
var a = "1";
a = a + 0;
a = a - 1;
a = a + 1;
alert(a);
```

a) 10
b) 1
c) 0

68) Укажите все возможные фрагменты javascript, выводящие модальное окно с сообщением "Quizful".

a) messageBox("Quizful");
b) print("Quizful");
c) alert("Quizful");
d) window("Quizful");
e) document.write("Quizful");

69) Как правильно написать IF конструкцию, чтобы выполнялся некоторый код, когда i не равно 5.

a) if (i <> 5)
b) if i <> 5
c) if i =! 5 then
d) if (i != 5)

70) Чему равно свойство length (в строках a, b) массива r.

```
var r = ['a', 'l', 'p', 'h'];

r.length = 10;
    //a. r.length = ?

delete r[0];
    //b. r.length = ?
```

a) a. 10 b. 10
b) a. 4 b. 3
c) a. 10 b. 9
d) a. undefiend b. undefiend

71) Выберите все вызовы parseFloat(), которые возвратят не число.

a) parseFloat("27.2 metres");
b) parseFloat("23,45");
c) parseFloat("$12.34");
d) parseFloat(".21");
e) parseFloat();
f) parseFloat("56,");

72) Аналогичны ли приведенные ниже две команды? alert('quizful'); ALERT('quizful');

a) Да

b) Нет

73) Какое значение будет присвоено переменной k?

```
var k;
k = parseInt('08', 8) + parseInt('05', 8);
```

a) 8
b) NaN
c) 5
d) Undefined
e) 13

74) Каких побитовых операторов НЕТ в JavaScript (выберите все подходящие варианты)?

a) ^
b) #
c) !
d) &
e) |
f) >>>

75) Выберите правильный вариант результата сортировки чисел в массиве. var n=[4, 6, 19, 34, 32]; n.sort();

a) //n is [4, 6, 19, 32, 34]
b) //n is [19, 32, 34, 4, 6]
c) //n is [4, 6, 19, 34, 32]
d) //n is [34, 32, 19, 6, 4]
e) //n is [4, 6, 32, 34, 19]

76) var s=new String(‘’); if (s) alert(true); else alert(false); Что выведет алерт?

a) true
b) false

77) Какие варианты ответов будут выводить: Wake up! каждые 10 минут?

a) setTimeout(“alert(‘Wake up!’);”, 600000);
b) setInterval(“alert(‘Wake up!’);”, 600000);
c) setInterval(“alert(‘Wake up!’);”, 10000);
d) setInterval(“alert(‘Wake up!’);”, 1000*60*10);

78) Каким будет окончательное значение переменной apt?

```
var apt = 2;
apt = apt << 2;
```

a) 0
b) 4
c) 8
d) 16
e) Возникает синтаксическая ошибка

79) Выберите все правильные варианты создания объекта Boolean со значением false.

a) var bool = new Boolean();
b) var bool = new Boolean(null);
c) var bool = new Boolean(NaN);
d) var bool = new Boolean(“Test”);
e) var bool = new Boolean(0);
f) var bool = new Boolean(“”);

80) Какие примеры кода объявляют функцию exec(string)? (Выберите все правильные варианты)

a) var exec = function(string){};
b) var exec = [{},string];
c) function exec(string) {}

81) Выберите все корректные варианты объявления функции в javascript.

a) function f() {}
b) function: f {}
c) f = function() {}
d) f() {}

82) Чему будет равно значение переменной C после выполнения данного фрагмента кода?

```
var a = 1;
var c = a+++1;
```

a) 1
b) 2
c) 3
d) возникает ошибка

83) window[‘undefined’] = = = underfiend

a) true
b) false

84) Что из перечисленного не является корректным JavaScript выражением, описывающим цикл?

a) var i = 0; while (true) { i = i; }
b) var i = 0; while (i <5) do { i ++; }
c) var i = 0; while (i <5) { i ++; }
d) var i = 0; do {i ++;} while (i < 5);

85) Как можно уведомить пользователя, что JavaScript отключен в его браузере?

a) Зависит от используемого пользователем браузера
b) Нет такого способа, чтобы обнаружить, что JavaScript отключен
c) Разместить внутри тега body следующий код: <script type=”text/javascript”> document.write(“У вас отключен JavaScript ”);

d) Разместить внутри тега body следующий код: <noscript>У вас отключен JavaScript …</ noscript>

86) Что вернёт данная функция при вызове?

```
function whatIsReturned() {
var total = “one”;
var TOTAL = “two”;
return total;
}
```

a) underfined
b) one
c) JavaScript выдаст ошибку о том, что переменная «total» уже определена
d) two

87) Дан следующий код:

```
for (i=5; i<10; i++){
for (j=I; j<i+1; j++){
alert(j); }}
```

a) 4
b) 5
c) 6
d) underfined

88) Какое сообщение покажет alert?

```
var i=5;
alert(++i);
```

a) 5
b) 6
c) 7
d) underfined

89) Какие объекты не являются прямыми потомками объекта Window?

a) Location
b) History
c) Forms
d) Select
e) Images

90) Являются ли свойства window и self синонимами? Можно ли применять любое из них по своему усмотрению, будет ли результат одинаков?

a) Да, результат будет одинаков
b) Нет, не являются синонимами
c) Да, только в некоторых случаях

91) Выберите правильный вариант синтаксиса метода open():

a) open ('адрес URL', 'имя_окна', config='параметр 1, параметр 2, ..., параметр n').
b) open ('адрес URL', 'имя_окна', target='параметр 1, параметр 2, ..., параметр n').
c) open ('имя_метода', 'адрес URL', target ='параметр 1, параметр 2, ..., параметр n').

92) Какой из следующих объектов содержит общую информацию о браузере пользователя:
a) Window
b) Navigator
c) Location
d) Screen

93) Какой из следующих объектов содержит информацию о различных параметрах экрана монитора пользователя:
a) Window
b) Navigator
c) Location
d) Screen
e) PrintScreen

94) Массив объектов, проиндексированный не только по числовым номерам элементов, но и по их именам, и имеющий свойства и методы.
a) список
b) коллекция
c) документ
d) ссылки
e) сценарий

95) Каким способом можно получить доступ к объектам HTML-документа:
a) используя коллекции и нумерацию объектов
b) присваивая каждому объекту уникальное имя

96) Какие из следующих утверждений верны:
a) Объект document является контейнером всех форм на странице
b) Объект Form и элементы управления именованы
c) Для того чтобы обратиться к отдельному элементу в конкретной форме, необходимо воспользоваться конструкцией document.ИмяЭлемента.

97) Это свойство отвечает за вызов программы CGI.
a) action
b) method
c) enctype
d) default

98) Какой из методов доступа задействуется по умолчанию и передает данные формы на указанный URL?
a) POST

b) GET
c) CGI
d) HTTP

99) Какие из следующих утверждений верны:

a) JavaScript – объектно-ориентированный язык программирования сценариев.

b) JavaScript используется в основном для создания встраиваемых в веб-страницы сценариев, позволяющих полностью управлять как самими веб-страницами, там и браузерами, в которых эти страницы открыты.

c) JavaScript загружается в браузер вместе с файлом.

d) Чтобы различить где кончается одно утверждение и начинается другое в JavaScript используется знак (;).

100) Как называются имена в JS, которые обозначают переменные, функции и объекты.

a) ключевые слова
b) идентификаторы
c) зарезервированные слова
d) литералы

Язык серверных сценариев PHP

Темы лабораторных работ

Задание 1. Установить набор дистрибутивов OpenServer.

Задание 2. Откройте в текстовом редакторе файл **lab2\vars.php** и выполните следующие задания:

1. Создайте переменную **name** и присвойте ей значение содержащее Ваше имя, например "Магомед" (обязательно в кавычках!).

2. Создайте переменную **age** и присвойте ей значение содержащее Ваш возраст, например 20.

3. Выведите с помощью **echo**(или **print**) фразу "Меня зовут: ваше_имя", например: "Меня зовут: Магомед" подставляя значение переменной **name.**

4. Выведите фразу "Мне ваш_возраст лет", например: "Мне 20 лет" подставляя значение переменной **age.**

5. Удалите переменную **age**.

Задание 3. Откройте в текстовом редакторе файл **lab2\types.php** и выполните следующие задания:

1. Создайте переменную **name** и присвойте ей значение содержащее Ваше имя, например "Магомед" (обязательно в кавычках!).

2. Создайте переменную **age** и присвойте ей значение содержащее Ваш возраст, например 20.

3. Выведите с помощью **echo**(или **print**) фразу "Меня зовут: ваше_имя", например: "Меня зовут: Игорь" подставляя значение переменной **name.**

4. Выведите фразу "Мне ваш_возраст лет", например: "Мне 20 лет" подставляя значение переменной **age.**

5. Измените код так, чтобы каждая фраза начиналась с новой строки.

6. Измените код так, чтобы каждая фраза начиналась с новой строки в исходном HTML-коде.

Задание 4. Откройте в текстовом редакторе файл **lab2\if.php** и выполните следующие задания:

1. Создайте переменную **age.**

2. Присвойте переменной **age** произвольное числовое значение.

1. Напишите конструкцию **if**, которая выводит фразу: "Вам ещё работать и работать" при условии, что значение переменной **age** попадает в диапазон чисел от 18 до 59 (включительно).

2. Расширьте конструкцию **if**, выводя фразу: "Вам пора на пенсию" при условии, что значение переменной **age** больше 59.

3. Расширьте конструкцию **if**, выводя фразу: "Вам ещё рано работать" при условии, что значение переменной **age** попадает в диапазон чисел от 1 до 17 (включительно).

4. Дополните конструкцию **if**, выводя фразу: "Неизвестный возраст" при условии, что значение переменной **age** не попадает в вышеописанные диапазоны чисел.

Задание 5. Откройте в текстовом редакторе файл **lab2\switch.php** и выполните следующие задания:

1. Откройте файл **lab2\switch.php.**

2. Создайте переменную **day** и присвойте ей произвольное числовое значение.

1. С помощью конструкции **switch** выведите фразу: "Это рабочий день", если значение переменной **day** попадает в диапазон чисел от 1 до 5 (включительно).

2. Выведите фразу: "Это выходной день", если значение переменной **day** равно числам 6 или 7.

3. Выведите фразу: "Неизвестный день", если значение переменной **day** не попадает в диапазон чисел от 1 до 7 (включительно).

Задание 6. Откройте в текстовом редакторе файл **lab2\array.php** и выполните следующие задания:

1. Создайте массив **bmw** с ячейками:

"model"

"speed"

"doors"

"year"

2. Заполните ячейки значениями: "X5", 120, 5, "2006".

3. Создайте массивы **toyota** и **opel** аналогичные массиву **bmw**.

4. Заполните массив **toyota** значениями: "Carina", 130, 4, "2007".

5. Заполните массив **opel** значениями: "Corsa", 140, 5, "2007".

6. Выведите значения всех трёх массивов в виде: name - model - speed - doors - year, например: bmw - 120 - 5 – 2006.

Задание 7. Откройте в текстовом редакторе файл **lab2\constant.php** и выполните следующие задания:

1. Создайте константу и присвойте ей значение.

2. Проверьте, существует ли константа, которую Вы хотите использовать.

3. Выведите значение созданной константы.

4. Попытайтесь изменить значение созданной константы.

Задание 8. Откройте в текстовом редакторе файл **lab2\for.php** и выполните следующее задание:

1. Используя цикл **for**, выведите в столбик Нечётные числа от 1 до 50.

Задание 9. Откройте в текстовом редакторе файл **lab2\while.php** и выполните следующее задание:

1. Используя цикл **while**, выведите в столбик Нечётные числа от 1 до 50.

Задание 10. Откройте в текстовом редакторе файл **lab2\table.php** и выполните следующие задания:

1. Создайте две числовые переменные **cols** и **rows.**

2. Присвойте созданным переменным произвольные значения в диапазоне от 1 до 10.

3. Используя циклы отрисуйте таблицу умножения в виде HTML-таблицы на следующих условиях:

a. Число столбцов должно быть равно значению переменной **cols.**

b. Число строк должно быть равно значению переменной **rows** c. Ячейки на пересечении столбцов и строк должны содержать значения, являющиеся произведением порядковых номеров столбца и строки.

4. Рекомендуется использовать циклы **for.**

5. Значения в ячейках первой строки и первого столбца должны быть отрисованы полужирным шрифтом и выровнены по центру ячейки.

6. Фоновый цвет ячеек первой строки и первого столбца должен быть отличным от фонового цвета таблицы.

Задание 11. Откройте в текстовом редакторе файл **lab2\menu.php** и выполните следующие задания:

1. Создайте ассоциативный массив **menu.**

2. Заполните массив, соблюдая следующие условия:

a. Название ячейки является пунктом меню, например: Home, About, Contact...

b. Значение ячейки является именем файла, на который будет указывать ссылка, например: index.php, about.php, contact.html...

3. Используя цикл **foreach**, отрисуйте вертикальное меню, структура которого описана в массиве **menu.**

Темы лабораторных работ

Задание 1. Откройте в текстовом редакторе файл **lab3\table.php** и выполните следующие задания:

1. Опишите функцию **getTable()**
2. Задайте для функции три аргумента: **cols**, **rows**, **color**
3. Откройте файл **lab3\table.php**
4. Скопируйте код, который отрисовывает таблицу умножения
5. Вставьте скопированный код в тело функции **getTable()**
6. Измените код таким образом, чтобы таблица отрисовывалась в зависимости от входящих параметров **cols**, **rows** и **color**
7. Отрисуйте таблицу умножения вызывая функцию **getTable()** с различными параметрами

Задание 2. Откройте в текстовом редакторе файл **lab3\menu.php** и выполните следующие задания:

1. Опишите функцию **getMenu()**
2. Задайте для функции первый аргумент **menu**, в него будет передаваться массив, содержащий структуру меню
3. Задайте для функции второй аргумент **vertical** со значением по умолчанию равным TRUE.

Данный параметр указывает, каким образом будет отрисовано меню – вертикально или горизонтально

4. Откройте файл **lab3\menu.php**
5. Скопируйте код, который создает массив **menu** и вставьте скопированный код в данный документ
6. Скопируйте код, который отрисовывает меню
7. Вставьте скопированный код в тело функции **getMenu()**
8. Измените код таким образом, чтобы меню отрисовывалась в зависимости от входящих параметров **menu** и **vertical**
9. Отрисуйте вертикальное меню вызывая функцию **getMenu()** с одним параметром
10. Отрисуйте горизонтальное меню вызывая функцию **getMenu()** со вторым параметром равным FALSE

Задание 3. Откройте в текстовом редакторе файл **lab3\strings.php** и выполните следующие задания:

1. Создайте строковую переменную **login** и присвойте ей значение "root"
2. Создайте строковую переменную **password** и присвойте ей значение "megaP@ssw0rd"
3. Создайте строковую переменную **email** и присвойте ей значение "ivan@petrov.ru"
4. Используя строковые функции, сделайте первый символ значения переменной **login** прописной (большой)
5. Используя строковые функции, сделайте первый символ значения переменной **password** прописной (большой)
6. Используя строковые функции проверьте, имеет ли значение переменной **email** символ "@"

Задание 4. Откройте в текстовом редакторе файл **lab3\date.php** и выполните следующие задания:

1. Создайте строковую переменную **now**
2. Создайте строковую переменную **birthday**
3. Присвойте переменной **now** значение метки времени актуальной даты (сегодня)
4. Присвойте переменной **birthday** значение метки времени Вашего дня рождения
5. Выведите фразу "До моего дня рождения осталось "
6. Выведите количество секунд, оставшееся до Вашего дня рождения
7. Закончите фразу " секунд"

Задание 5. Откройте в текстовом редакторе файл **lab3\site\index.php** и выполните следующие задания:

1. Подключите файл **lib.inc.php**
2. Подключите файл, содержащий код верхней части страницы (**top.inc.php**)
3. Подключите файл, содержащий код меню (**menu.inc.php**)
4. Подключите файл, содержащий код нижней части страницы (**bottom.inc.php**)
5. Запустите файл **lab3/site/index.php** в браузере

Задание 6. Откройте в текстовом редакторе файл **lab3\site\lib.inc.php** и выполните следующие задания:

1. Откройте файл **lab3\menu.php**
2. Скопируйте код функции **getMenu()**
3. Вставьте скопированный код в файл **lib.inc.php**
4. Откройте файл **lab3\table.php**

5. Скопируйте код функции **getTable()**

6. Вставьте скопированный код в файл **lib.inc.php**

Задание 7. Откройте в текстовом редакторе файл **lab3\site\menu.inc.php** и выполните следующие задания:

1. Создайте ассоциативный массив **menu**

2. Заполните массив следующими данными:

"Home"=>"index.php",

"Page1"=>"page1.php",

"Page2"=>"page2.php",

"Page3"=>"page3.php",

"Table"=>"table.php"

3. Отрисуйте меню вызывая функцию **getMenu().**

4. Запустите файл **lab3\site\index.php** в браузере

Темы лабораторных работ

В ходе работы необходимо разработать сайт под названием «MyTravelNotes», содержащий записи автора сайта о его путешествиях, а также базу данных (БД) под названием «MySiteDB», содержащую контент сайта. Данная задача включает в себя реализацию следующих функций web-проекта:

1. Возможность добавления записей автора и комментариев к ним (при этом все заметки и комментарии должны передаваться и храниться на сервере в БД);

2. Возможность модификации и удаления заметок и комментариев в БД посредством форм сайта;

3. Обеспечение защиты данных с помощью логина и пароля, разграничение доступа к данным с учетом установленного уровня доступа (администратор и пользователь);

4. Данные о пользователях и их уровне доступа должны храниться на сервере в БД;

5. Обеспечение корректного входа и выхода с сайта;

6. Реализацию возможности обратной связи посетителей сайта с автором блога;

7. Обеспечение дружественного пользовательского интерфейса и корректной организации навигации по разделам сайта.

Кроме того, при разработке данного программного проекта должны учитываться основные принципы web-дизайна (юзабилити) для удобства работы с сайтом конечного пользователя.

1. Создание статической основы web-страниц

В данном упражнении необходимо создать две первые страницы проекта, которые станут основой для дальнейшей разработки - **default.php** и **inform.php**. Страница **default.php** является первой страницей сайта, должна загружаться в браузере и содержать собственно заметки автора блога. Страница статистики **inform.php** будет вспомогательной страницей, содержащей статистическую информацию о размещенных на сайте заметках и комментариях.

На страницы необходимо добавить код HTML, который является основой для дальнейшей работы других технологий.

1. Создайте новую страницу **default.php**. На странице должно располагаться меню переходов между страницами и место размещения основного контента сайта (рисунок 1):



Рад приветствовать вас
на страницах моего сайта, посвященного путешествиям.

Рисунок 1 - Пример страницы default.php

2. Визуальная разметка страницы (оформление) может быть произвольная.
3. Сохраните страницу в папке проекта.
4. Создайте страницу статистики **inform.php**. Схема страницы (рисунок 2):

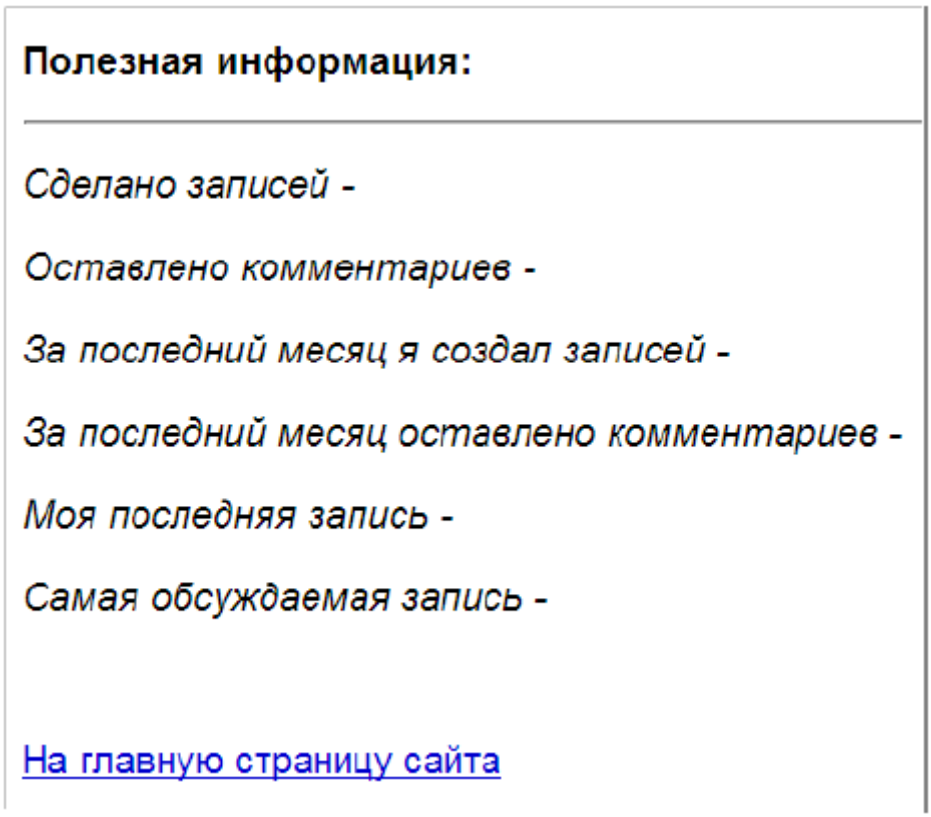


Рисунок 2 - Пример страницы inform.php

5. Сохраните страницу в папке проекта.
6. Свяжите гиперссылками созданные страницы.

В ходе выполнения данной лабораторной работы необходимо создать в MySQL новую базу данных с названием «***MySiteDB***» и добавить в нее три таблицы: **notes**, **comments** и **authors**. **Notes** содержит заметки блога; **comments** – комментарии к этим заметкам; **authors** – информацию о зарегистрированных пользователях. Схема данных (*рисунок 3*):



Рисунок 3 - Схема базы данных "MySiteDB"

2. Создание БД «MySiteDB»

В этом упражнении реализуется запрос на создание новой базы данных.

1. Создайте новый php документ, который будет называться **create_db.php**.

2. Создайте соединение с сервером localhost. Имя сервера ***localhost***, пользователь ***root***, пароля нет.

3. Создайте базу данных:

3.1. Сформируйте запрос на создание базы ***MySiteDB*** с использованием SQL;

3.2. Реализуйте запрос на создание БД с помощью функции mysqli_query().

4. Сохранить документ, выполнить запрос.

5. С помощью утилиты **PhpMyAdmin** убедитесь, что создана новая база данных.

6. Вторично выполните запрос, чтобы убедиться, что соединение есть, а база не создается (т.к. она была уже создана ранее, в ходе предыдущего выполнения скрипта).

7. Желательно добавить цикл ***if*** для обнаружения неполадок в работе.

Вариант реализации создания БД MySiteDB

```
<?php
//Создать соединение с сервером
$link = mysqli_connect ("localhost", "root", "");
if ($link) {
echo "Соединение с сервером установлено", "<br>";
} else {
echo "Нет соединения с сервером";
}
```

//Создать БД MySiteDB

//Сначала формирование запроса на создание

$db = "MySiteDB";

$query = "CREATE DATABASE $db";

//Затем реализация запроса на создание. Важна последовательность аргументов функции: соединение с сервером, SQL-запрос.

$create_db = mysqli_query($link, $query);

if ($create_db) {

echo "База данных $db успешно создана";

} else {

echo "База не создана";

}

?>

3. Создание пользователя admin

В этом упражнении необходимо создать нового пользователя базы данных с именем ***admin*** и паролем ***admin*** с правами администратора. Пользователей можно добавлять двумя способами:

• при помощи SQL-запроса GRANT

• в таблице назначения прав доступа MySQL ***Пользователи (Privileges)*** с помощью утилиты **PhpMyAdmin**.

Выберите один из двух приведенных далее способов.

Способ 1: создание нового пользователя с помощью SQL-запроса GRANT

1. Создайте новый php-документ, который будет называться **create_user.php**;

2. Создайте соединение с сервером;

3. Сформируйте SQL-запрос на создание нового пользователя базы данных:

*$query = "GRANT ALL PRIVILEGES ON *.* TO 'admin'@'localhost' IDENTIFIED BY 'admin'*

WITH GRANT OPTION";

//.* - глобальный уровень привилегий, применяется ко всем базам на сервере.*

4. Реализуйте запрос.

Проверка создания пользователя. С помощью утилиты **PhpMyAdmin** убедитесь, что создан новый пользователь. Для этого запустите утилиту **PhpMyAdmin** и перейдите на вкладку ***Пользователи (Privileges)***. Изучите список пользователей.

Способ 2: создание нового пользователя с помощью утилиты **PhpMyAdmin**

1. Запустите утилиту **PhpMyAdmin** и перейдите на вкладку ***Пользователи (Privileges)***. Нажмите кнопку *«Добавить пользователя (Add a new user)».*

2. Введите имя пользователя (***admin***), хост - локальный (***localhost***), пароль с подтверждением (***admin***). Предоставьте новому пользователю все права (***global privileges – Check All***).

3. Убедитесь, что новый пользователь создан корректно.

4. Все дальнейшие действия с базой данных будут проводиться под пользователем **admin** с паролем **admin** и соответствующими правами, если иное не указано в задании.

4. Создание таблицы notes

В данном упражнении будет продемонстрирован один из способов создания таблиц в ранее созданной базе данных на примере создания таблицы ***notes***. Таблица ***notes*** содержит заметки автора блога. Данная таблица будет создана средствами PHP. Информацию о полях таблицы см. в *Приложении 3.*

1. Создайте новый php-документ, который будет называться **create_tbl.php**;

2. Создайте соединение с сервером уже под созданным ранее пользователем ***admin*** с паролем ***admin***.

3. Подключитесь к базе данных MySiteDB.

4. Сформируйте запрос на создание таблицы **notes** с полями, указанными в *таблице ниже*.

Таблица 1*. Структура таблицы notes*

Имя поля	**Описание поля**	**Тип данных**	**Другое**
id	Идентификатор записи	SMALLINT(6)	Ключевое поле (AUTO_INCREMENT), INDEX – PRIMARY.
created	Дата создания заметки	DATE	
title	Заголовок заметки	VARCHAR (20)	Строка фиксированной длины 20 символов
article	Содержимое заметки	VARCHAR (255)	Строка фиксированной длины 255 символов

```
//Формирование запроса
$query = "CREATE TABLE notes
(id SMALLINT NOT NULL AUTO_INCREMENT,
PRIMARY KEY (id),
created DATE,
title VARCHAR (20),
article VARCHAR (255))";
```

5. Реализуйте запрос на создание таблицы.

6. С помощью утилиты **PhpMyAdmin** убедитесь, что создана новая таблица. Для этого запустите утилиту, перейдите к базе данных **MySiteDB** и просмотрите ее структуру. В ней должна появиться соответствующая таблица.

Вариант реализации создания таблицы notes

```
<?php
//Соединение с сервером
$link = mysqli_connect ('localhost', 'admin', 'admin');
//Выбор БД
$db = "mySiteDB";
$select = mysqli_select_db($link, $db);
if ($select){
echo "База успешно выбрана", "<br>";
} else {
echo "База не выбрана";
}
//Создание таблицы
//Формирование запроса
$query = "CREATE TABLE notes
(id SMALLINT NOT NULL AUTO_INCREMENT,
PRIMARY KEY (id),
created DATE,
title VARCHAR (20),
article VARCHAR (255))";
//Реализация запроса
$create_tbl = mysqli_query ($link, $query);
if ($create_tbl){
echo "Таблица успешно создана", "<br>";
} else {
echo "Таблица не создана";
}
?>
```

5. Создание таблицы comments

В данном упражнении будет продемонстрирован другой способ создания таблиц в ранее созданной базе данных на примере создания таблицы **comments**. Таблица **comments** содержит комментарии пользователей к заметкам автора блога. Таблица будет создана с помощью утилиты **PhpMyAdmin**.

***Таблица 2**. Структура таблицы comments*

Имя поля	Описание поля	Тип данных	Другое
id	Идентификатор записи	SMALLINT(6)	Ключевое поле (AUTO_INCREMENT), INDEX – PRIMARY.
created	Дата публикации комментария	DATE	
author_id	внешний ключ к таблице Authors	SMALLINT (6)	INDEX
comment	Содержимое комментария	VARCHAR (255)	Строка фиксированной длины 255 символов
art_id	Внешний ключ к таблице Notes	SMALLINT(6)	INDEX

1. Запустите браузер.

2. Запустите утилиту **phpMyAdmin**. В главном окне **PHPMyAdmin** выберите БД ***MySiteDB***.

3. В поле *«Создать таблицу (Create new table)»*, присвойте имя таблице – **comments**; количество полей - **5**, нажмите кнопку «Go» (*рисунок 4*).

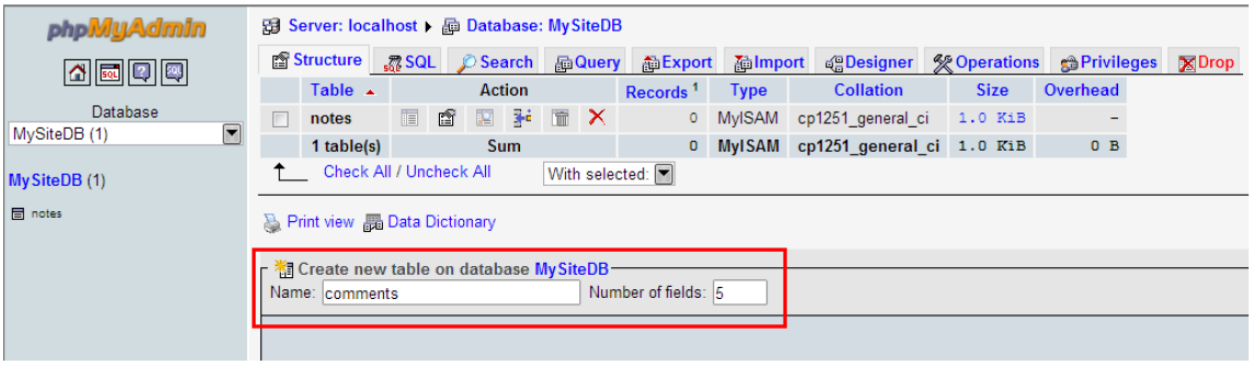


Рисунок 4 - Создание новой таблицы с помощью утилиты phpMyAdmin

4. Создание полей таблицы comments:

4.1. В открывшемся окне заполните необходимые поля таблицы (*рисунок 5*) и нажмите кнопку *«Сохранить (Save)»*.

4.2. Для поля **id** добавьте следующие атрибуты: обозначьте автоинкремент **A_I** и первичный ключ **PRIMARY** в поле со списком **INDEX**.

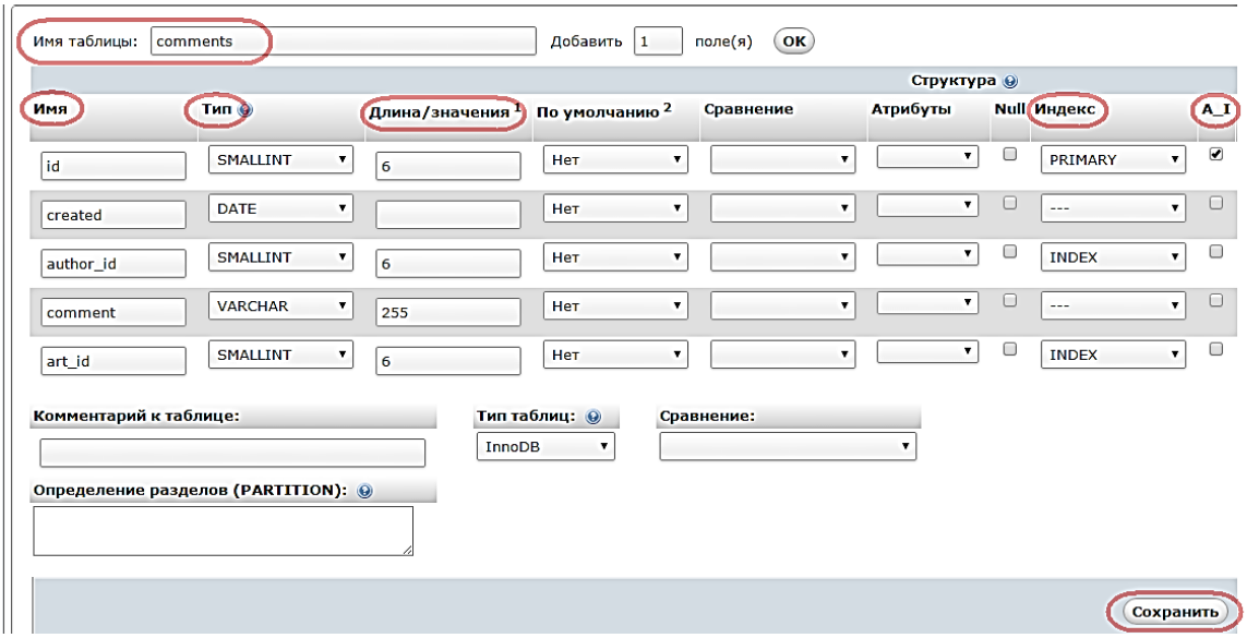


Рисунок 5 - Заполнение полей таблицы comments

5. Полученный результат должен выглядеть следующим образом (*рисунок 6*):



Рисунок 6 - Результат создания таблицы

6. Создание таблицы authors

Самостоятельно средствами PhpMyAdmin создайте таблицу authors.

***Таблица 3**. Структура таблицы authors*

Имя поля	**Описание поля**	**Тип данных**	**Другое**
id	Идентификатор записи	SMALLINT(6)	Ключевое поле (AUTO_INCREMENT), INDEX – PRIMARY.
login		VARCHAR (10)	Логин пользователя
password		VARCHAR (10)	Пароль пользователя
rights	Права доступа: u – user, a - administrator	ENUM(‘u’, ‘a’)	Права доступа пользователя к страницам сайта. Значение по умолчанию – u (пользователь)

В ходе создания таблицы поля для заполнения должны выглядеть следующим образом (рисунок 7):

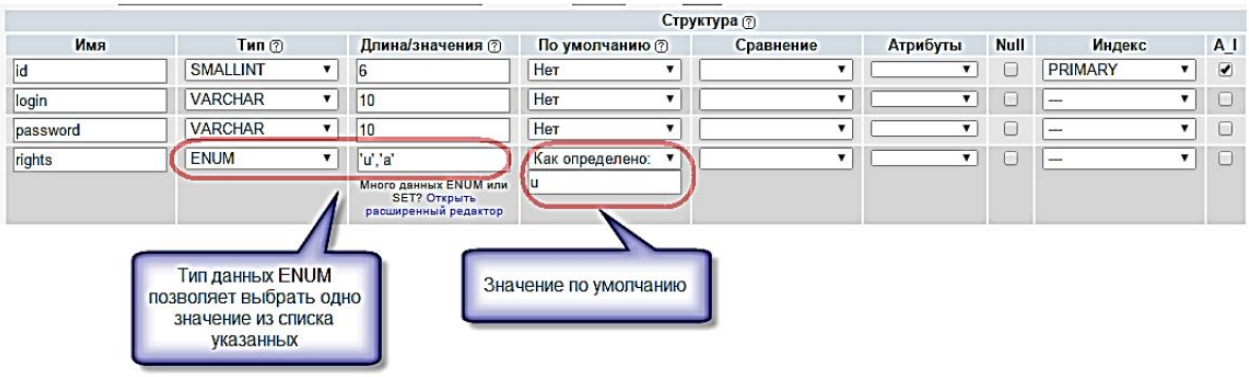


Рисунок 7 – создание таблицы authors

7. Создание межтабличных связей

В данном упражнении необходимо создать связи между таблицами для поддержания целостности данных web-приложения.

Внимание! При работе со сборкой **Open Server** инструмент **Designer** для утилиты **PHPMyAdmin** необходимо устанавливать и настраивать отдельно. См. дополнительную информацию к курсу и информацию на ресурсе *http://open-server.ru/* . Без данного инструмента связи между таблицами создаются через разделы таблиц **Связи**.

1. Для организации межтабличных связей выберите **БД MySiteDB**, вкладку **Designer**. Откроется окно схемы данных.

2. С помощью инструментов окна **Designer** создайте связь «один ко многим» (*рисунок 8*) сначала между таблицами **notes** и **comments**. При этом используйте стратегию каскадирования для поддержания ссылочной целостности (каскадное удаление, каскадное обновление). Для этого выберите команду «*Создать связь*»  - поле первичного ключа в главной таблице (**Notes**) – поле внешнего ключа в подчиненной таблице (**Comments**). В открывшемся окне выберите «On delete CASCADE», «On update CASCADE».

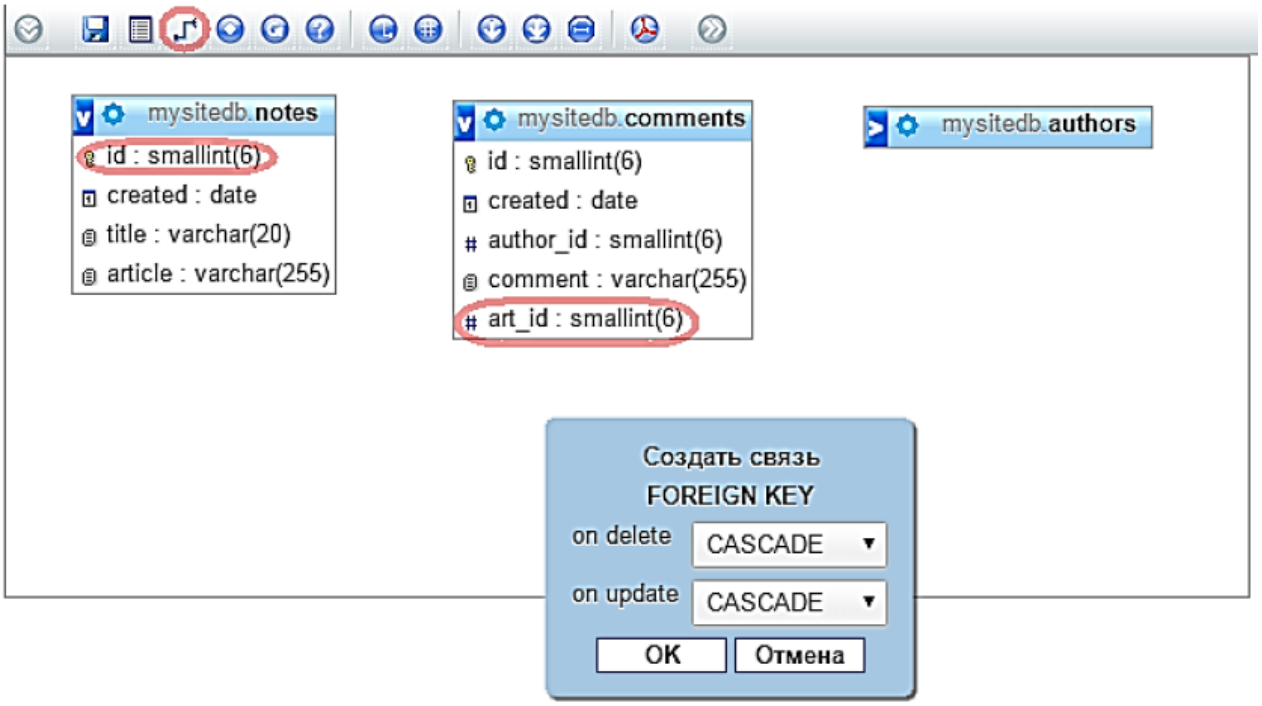


Рисунок 8 - Поле окна инструментов Designer

3. В результате успешного создания связи между таблицами **notes** и **comments** схема должна выглядеть как представлено на рисунке 9:

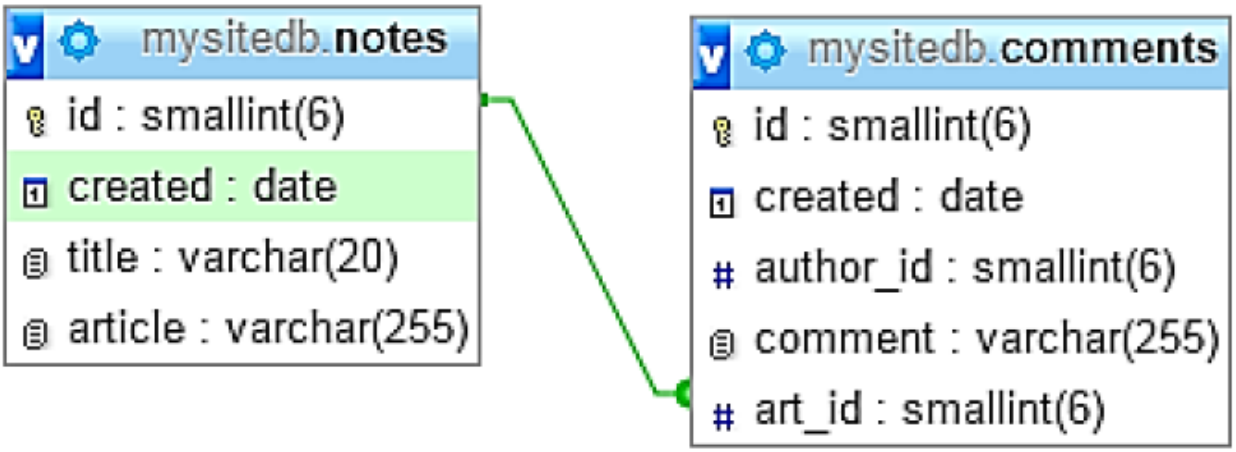


Рисунок 9 – Связь между таблицами Notes и Comments

4. Аналогичным образом создайте связь между таблицами **authors** и **comments**. Отличием от предыдущей связи будет являться выбранная стратегия поддержания ссылочной целостности (рисунок 10):

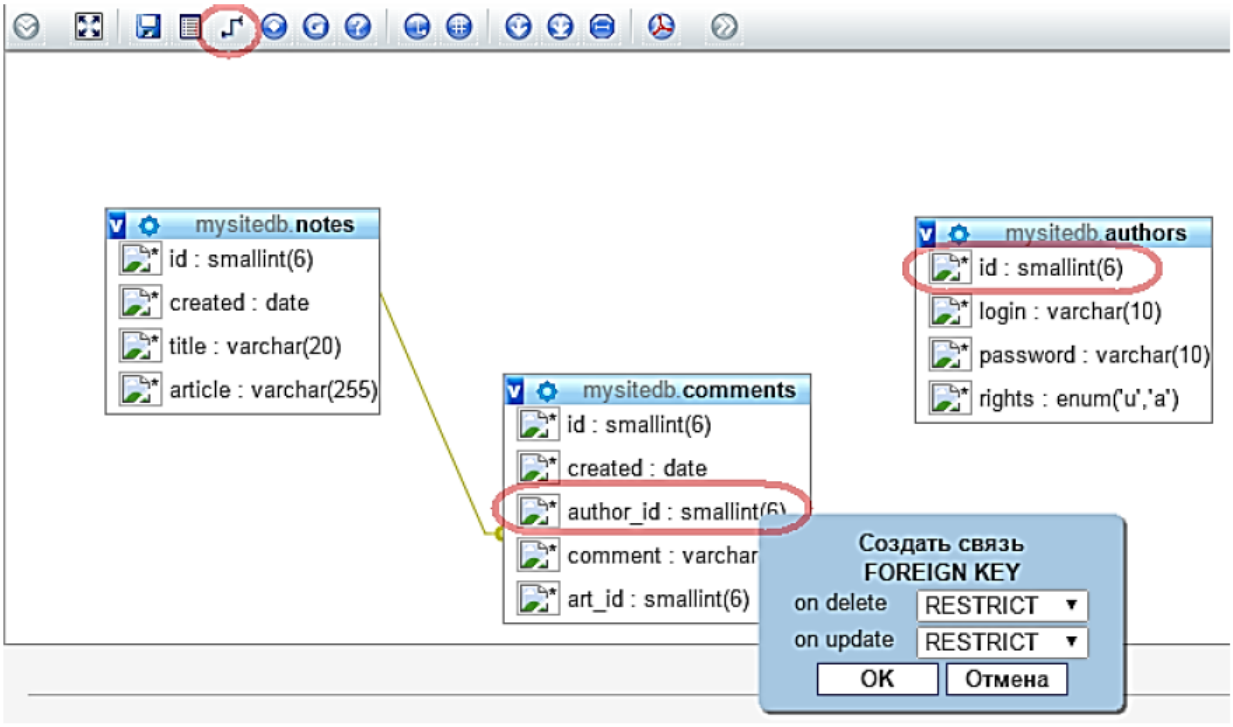


Рисунок 10 – Создание связи между таблицами authors и comments

5. В результате будет создана связь вида (см. рисунок 11):



Рисунок 11 – связь между таблицами authors и comments

8. Заполнение таблиц

Введите в созданные таблицы несколько записей – для проверки их работы и для использования на будущих серверных страницах сайта. Для этого выберите нужную таблицу и нажмите кнопку ***Вставить (Insert)*** на вкладках. После заполнения соответствующих полей таблицы нажмите кнопку **Ok**.

*Помните, что поля **id** в таблицах заполнять не надо – они заполняются автоматически.*

9. Файл подключения базы данных

В ходе выполнения данного упражнения необходимо создать файл, содержащий код подключения базы данных. В дальнейшем он будет использоваться на каждой странице, работающей с базой.

1. Создайте файл **MySiteDB.php** следующего вида:

```
<?php
$localhost = "localhost";
$db = "MySiteDB";
$user = "admin";
$password = "admin";
$link = mysqli_connect($localhost, $user, $password) or
trigger_error(mysql_error(),E_USER_ERROR);
//trigger_error выводит на страницу сообщение об ошибке. Первый параметр - сообщение об ошибке
//в строковом виде, в данном случае возвращается функция mysql_error(), второй - числовой код //ошибки(почти всегда используется значение константы E_USER_ERROR, равное 256)
//Следующие строки необходимы для того, чтобы MySQL воспринимал кириллицу.
//Параметры функции mysqli_query(): идентификатор соединения с сервером и запрос SQL
mysqli_query($link, "SET NAMES utf8;") or die(mysql_error());
mysqli_query($link, "SET CHARACTER SET utf8;") or die(mysql_error());
//Выбор базы данных на сервере localhost
mysqli_select_db($link, $db);
?>
```

2. Сохраните файл в папке проекта.

10. Простой вывод данных. Страницы default.php и comments.php.

1. Вывод данных из базы на страницу

В этом упражнении на главную страницу сайта необходимо вывести все заметки из таблицы БД **notes**.

1. Откройте страницу **default.php.**

2. Создайте соединение с сервером. Оно реализовано в файле **MySitDB.php** – данный файл надо включить с помощью функции ***require_once(),*** в качестве параметра передав ей путь к файлу («**MySiteBD.php**»):

```
<?php require_once ("MySiteDB.php"); ?>
```

3. Далее необходимо вывести записи (строки) на страницу сайта из таблицы notes. Сначала надо реализовать запрос на выборку. Для этого:

3.1. создайте SQL-запрос на выборку данных из таблицы (SELECT *fields* FROM *tableName*). Здесь SELECT – оператор выбора полей, FROM – оператор выбора таблицы-источника полей.

```
$query = "SELECT * FROM notes";
```

*Если вам необходимо выбрать все поля таблицы (как в данном случае), то запрос можно построить так: SELECT * FROM tablename, где символ «*» обозначает все поля таблицы.*

3.2. Реализуйте запрос на выборку с помощью функции mysqli_query:

$select_note = mysqli_query($link, $query);

4. Далее необходимо вывести запись на страницу сайта. Для этого используется функция ***mysqli_fetch_array()***. Параметром функции является переменная, содержащая результат выполнения запроса к БД (в данном случае – реализации запроса на выборку); собственно функция получает по одной записи из таблицы за один раз. Каждая запись возвращается в виде массива.

5. Для вывода информации из массива по отдельным элементам необходимо придерживаться следующего синтаксиса:

```
//Вывод элементов массива
echo $note ['id'], "<br>";
echo $note ['created'], "<br>";
echo $note ['title'], "<br>";
echo $note ['article'], "<br>";
```

6. Сейчас из таблицы с помощью функции ***mysqli_fetch_array()*** выводится только одна запись. С помощью цикла необходимо сделать так, чтобы выводились все записи из таблицы. Для этого необходимо изменить часть кода следующим образом:

```
//Использование цикла while
while ($note = mysqli_fetch_array($select_note))
{
echo $note ['id'], "<br>";
echo $note ['created'], "<br>";
echo $note ['title'], "<br>";
echo $note ['article'], "<br>";
}
```

Здесь переменной с именем *$select_note* присваивается результат выполнения запроса к БД *mysqli_query()*.

Вариант полного кода страницы

```
<?php require_once("MySiteDB.php"); ?>
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>Главная страница сайта</title>
</head>
```

```
<body>
<menu>
Вход
Новая заметка
Отправить сообщение
Добавить фото
Статистика
Администратору
Выход
</menu>
<p>Рад приветствовать вас на страницах моего сайта, посвященного путешествиям. </p>
<?php
$query = "SELECT * FROM notes";
$select_note = mysqli_query($link, $query);
while ($note = mysqli_fetch_array($select_note))
{
echo $note ['id'], "<br>";
echo $note ['created'], "<br>";
echo $note ['title'], "<br>";
echo $note ['article'], "<br>";
}
?>
<body>
</html>
```

2. Обмен данными между серверными страницами

Каждая заметка на главной странице блога может быть прокомментирована. Для реализации этой функции необходимо сделать из заголовка каждой заметки гиперссылку, перейдя по которой посетитель попадет на страницу со списком комментариев к выбранной заметке. Кроме того, на этой же странице должна отображаться сама выбранная для комментирования заметка.

Следовательно, необходимо реализовать механизм обмена данными между страницами таким образом, чтобы при переходе по гиперссылке передавалась информация о том, какая именно заметка была выбрана.

Для этого необходимо ввести некий идентификатор, значение которого будет совпадать с id комментируемой заметки, и который будет передаваться при переходе по гиперссылке.

1. Создание гиперссылки

1.1. Создайте новую страницу **comments.php**, которая будет содержать комментарии к выбранной заметке.

Реализуйте на ней соединение с сервером и подключение к БД.

<?php require_once("MySiteDB.php"); ?>

1.3. Для передачи идентификатора заметки введем аргумент **note**. В качестве значения он будет получать значение поля ***id*** таблицы **notes**.

1.4. На странице **default.php** найдите фрагмент кода, передающего заголовок заметки ***title (echo $note [‘title’];)***. Его необходимо отредактировать таким образом, чтобы он стал гиперссылкой на страницу комментариев **comments.php**, а также передавал ***id*** выбранной заметки:

```
while ($note = mysqli_fetch_array($select_note)){
echo $note['id'], "<br>";
?>
<a href="comments.php?note=<?php echo $note['id']; ?>">
<?php echo $note ['title'], "<br>";?></a>
<?php
echo $note ['created'], "<br>";
echo $note ['article'], "<br>";
}
```

Здесь мы создаем гиперссылку на страницу **comments.php** и в этой гиперссылке передаем идентификатор **note**, значение которого равно значению элемента массива ***$note[‘id’],*** т.е. значению ***id*** заметки.

2. Страница **comments.php**

2.1. Перейдите на страницу **comments.php**. На данной странице должны отображаться комментарии к выбранной записи, а также сама комментируемая запись (для удобства посетителя сайта).

2.2. Данную задачу можно выполнить по аналогии с выводом заметок на странице **default.php**. Основное отличие заключается в том, что вначале необходимо со станицы **default.php** получить переданный с помощью идентификатора **note** id заметки. Это делается с помощью метода $_GET:

```
//Переменной $note_id необходимо присвоить id заметки, переданной с помощью метода $_GET со страницы default.php
$note_id = $_GET['note'];
```

2.3. Далее необходимо вывести значения полей created, title, content из таблицы **notes** для заметки с полученным id. Для этого используется SQL запрос

SELECT… FROM… WHERE…

В нем с помощью оператора ***SELECT*** выбираем необходимые поля таблицы; с помощью ***FROM*** определяем таблицу-источник выборки; ***WHERE*** задает условие отбора, по которому выбираем заметку с выбранным ***id***:

//Формируем SQL-запрос на выборку с учетом переданного id заметки

$query = "SELECT created, title, article FROM notes WHERE id = $note_id";

2.4. После формирования SQL-запроса его необходимо реализовать с помощью функции ***mysqli_query()*** и вывести данные на страницу с помощью функции ***mysqli_fetch_array().***

2.5. Затем аналогичным образом выведите комментарии к выбранной заметке. Обратите внимание, что SQL-запрос на выборку комментариев должен строиться следующим образом:

*$query_comments = "SELECT * FROM comments WHERE art_id = $note_id";*

В условии WHERE мы реализуем поддержку связи между таблицами, которые связаны по полям ***id*** (таблица **notes**) и ***art_id*** (таблица **comments**).

В переменной **$note_id** содержится ***id*** выбранной заметки, следовательно, для выбора комментариев к этой заметке необходимо, чтобы значение поля ***art_id*** created также было равно **$note_id.**

3. Проверьте корректность данных между страницами **default.php** и **comments.php**. При переходе по ссылке с **default.php** на **comments.php** в адресной строке браузера должен отображаться ***id*** выбранной заметки, переданный с помощью идентификатора **note**.

4. Для того, чтобы выводились все комментарии, а не только первый – реализуйте цикл.

5. Если у заметки нет ни одного комментария – об этом надо сообщить. Самостоятельно реализуйте вывод надписи «*Эту запись еще никто не комментировал*» для тех заметок, к которым нет комментариев (например, используя функцию *mysqli_num_rows()* или другие решения).

Темы лабораторных работ

В ходе выполнения данной лабораторной работы рассматриваются принципы ввода и обработки информации при работе с формами HTML.

1. Отправка почты

Данное упражнение позволяет реализовать отправку сообщения через форму на сервер.

1. Создайте страницу **email.php**. Добавьте название страницы и пояснительный тест, форму с двумя текстовыми полями: **Тема сообщения** и **Текст сообщения**, кнопку **Отправить**, а также гиперссылку для возврата на главную страницу сайта.

2. Самостоятельно реализуйте обработку данных формы с помощью функции **mail().** «Получить» отправленное сообщение вы можете по локальному адресу:

C:\OpenServer\userdata\temp\email

3. Проверьте корректность работы, создайте гиперссылки с главной страницы сайта на страницу **email.php** и со страницы **email.php** на страницу **default.php**.

4. Самостоятельно реализуйте проверку заполнения всех полей формы для того, чтобы исключить отправку «пустого» письма.

2. Страница для добавления заметок

В этом упражнении будет проиллюстрировано создание страницы для добавления новых заметок – **newnote.php**.

1. Создайте новую страницу **newnote.php**, добавьте название и пояснительный текст.

2. Создать html-форму с именем «**newnote**», метод обработки данных – **POST**.

3. На форме разместите два поля: одно (типа *text*) для добавления заголовка заметки – с именем «**title**», другое (*textarea*) для добавления

самой заметки – с именем «**article**». Добавьте параметры размера элементов формы.

4. Также поместите на поле кнопку отправки с именем «**submit**».

*Не забывайте именовать html-форму и элементы html-формы (атрибут **name**). Эти имена важны при дальнейшей обработки данных, полученных через форму, в php-скриптах.*

5. Добавление даты создания заметки

5.1. В таблице **notes**, заполняемой через создаваемую нами форму, осталось незаполненным поле **art_id** (поле с датой создания заметки) – для него мы не создавали элемент формы. PHP позволяет получать текущую дату автоматически, с помощью функции ***date().*** Формат ее вызова: ***date(<формат>).*** MySQL требует формат даты <год>-<месяц>-<число>, при этом год – 4 цифры, месяц – 2 цифры, число – 2 цифры. Согласно шаблону, вид вызова функции: ***date("Y-m-d").*** Мы автоматизируем процесс получения текущей даты из формы.

5.2. Разместите на форме после второго текстового поля скрытое поле с именем «***created***».

5.3. Значение поле ***created*** будет получать через php- функцию **date().** Результат добавления поля:

```
<input type="hidden" name = "created" id = "created"
value ="<?php echo date("Y-m-d");?>"/>
```

Вариант реализации html-формы

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>Новая заметка</title>
</head>
<body>
<p>Добавить новую заметку: </p>
<form method="post">
<input type="text" name="title" size="20" maxlength="20"/>
<textarea name="article" cols="55" rows="10"></textarea>
<input type="hidden" name = "created" value ="<?php echo date("Y-m-d");?>"/>
<input type="submit" name="submit" value="Отправить" />
</form>
<a href="default.php">Возврат на главную страницу сайта</a>
</body>
```

</html>

6. Обработка html-формы. Вам необходимо создать php-скрипт, который выполнит два шага:

- Получит данные, введенные пользователем в поля созданной html-формы (т.е. новую заметку);
- Передаст эти данные в базу, где хранятся уже созданные ранее заметки.

6.1. Получение данных через форму. Для получения данных через форму необходимо:

6.1.1. Подключиться к серверу и выбрать базу данных;

6.1.2. Получить данные из полей формы. Данные мы получаем из элементов формы используя названия (атрибут **name**) этих элементов. Данные формы помещаются в массив $_POST, а затем присваиваются переменным php. Принцип получения:

$имя_переменной = $_POST *[‘АтрибутNameЭлементаФормы*’];

Таким образом информация, введенная пользователем в форму, «присваивается» в качестве значения для переменной php.

```
//Получение данных из формы
$title = $_POST['title'];
$created = $_POST['created'];
$article = $_POST[‘article’];
```

6.2. Передача данных в базу

6.2.1. Данные в базу передаются по обычному принципу: формирование SQL-запроса – реализация SQL-запроса . Формирование запроса: (в нем поле id получает свое значение автоматически):

```
//Формирование запроса
$query = "INSERT INTO notes (id, title, created, article)
VALUES (NULL, '$title', '$created', '$article’)";
```

В запросе используется SQL-инструкция INSERT. Синтаксис инструкции:

INSERT INTO tblName (tblField 1, tblField 2, … , tblField N)
VALUES (value 1, value 2, … , value N);

В ней *tblNamt* – имя таблицы, *tblField* – имя поля таблицы (перечисляются в том порядке, в котором располагаются в таблице), *value* – вставляемое значение поля таблицы (порядок должен соответствовать порядку имен полей).

6.2.2. Реализуйте запрос с помощью функции ***mysqli_query()***.

7. Проверьте корректность работы формы и обработки данных.

8. Самостоятельно программно исключите возможность передачи в базу данных пустой записи.

9. Добавьте гиперссылки между страницами **default.php** и **newnote.php**.

Вариант реализации кода страницы newnote.php

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>Новая заметка</title>
</head>
<body>
<p>Добавить новую заметку: </p>
<form method="post">
<input type="text" name="title" size="20" maxlength="20"/>
<textarea name="article" cols="55" rows="10"></textarea>
<input type="hidden" name = "created" value ="<?php echo date("Y-m-d");?>"/>
<input type="submit" name="submit" value="Отправить" />
</form>
<a href="default.php">Возврат на главную страницу сайта</a>
</body>
</html>
<?php
//Подключение к серверу
require_once ("MySiteDB.php");
//Получение данных из формы
$title = $_POST['title'];
$created = $_POST['created'];
$article = $_POST[‘article’];
if (($title)&&($created)&&($article))
{
//Формирование запроса
$query = "INSERT INTO notes (id, title, created, article)
VALUES (NULL, '$title', '$created', '$article’)";
//Реализация запроса
$result = mysqli_query ($link, $query);
}
?>
```

3. Обработка HTML-переключателей

1. На любой странице сайта (можно создать новую и дать ей произвольное имя, указав тип данных .php) создайте HTML-форму.

2. В созданной форме создайте группу переключателей (radio) на произвольную тему.

3. В том же файле создайте php-скрипт, обрабатывающий информацию о том, какой переключатель был выбран пользователем. Обратите внимание, что radio позволяет выбрать только один вариант ответа.

Вариант кода

```
<html>
<form action="ИмяВашегоФайла.php" method = "GET" enctype = "multipart/form-data">
<p> Выберите Ваш любимый город: </p><br />
<p><input type = "radio" name = "MyRadio" value = "Rome">Рим</p>
<p><input type = "radio" name = "MyRadio" value = "Paris">Париж</p>
<p><input type = "radio" name = "MyRadio" value = "Moscow">Москва</p>
<p><input type = "submit" name = "submit" value = “Отправить” />
</form>
</html>
<?php
$var = $_GET['MyRadio'];
switch($var)
{
case " Rome":
echo "You choose $var";
break;
case " Paris":
echo "You choose $var";
break;
case " Moscow":
echo "You choose $var";
break;
}
?>
```

4. Обработка HTML-флагов

1. Ниже на странице, созданной в предыдущем упражнении, создайте вторую HTML-форму.

2. В данной HTML-форме создайте группу флажков (checkbox) на произвольную тему.

3. В том же файле создайте php-скрипт, обрабатывающий информацию о том, какие варианты ответов были выбраны пользователем. Обратите внимание, что checkbox позволяет выбрать множество (от нуля до максимального количества) ответов.

Вариант кода

```
<html>
<form action="ИмяВашегоФайла.php" method = "GET" enctype = "multipart/form-data">
<p> Выберите Ваши любимые города: </p><br />
<p><input type = "checkbox" name = "MyCheckBox[]"
value = "Rome">Рим</p>
<p><input type = "checkbox" name = "MyCheckBox[]"
value = "Paris">Париж</p>
<p><input type = "checkbox" name = "MyCheckBox[]"
value = "Moscow">Москва</p>
<p><input type = "submit" name = "submit" value = “Отправить”/>
</form>
</html>
<?php$arr = $_GET['MyCheckBox'];
if(empty($arr))
{
echo "Вы не выбрали ни один вариант";
}
else
{
$count = count($arr);
echo "Вы выбрали:"."<br />";
for($i=0; $i<$count; $i++)
{
echo $arr[$i]."<br />";
}
} ?>
```

5. Модификация данных

В данной лабораторной работе рассматривается реализация редактирования уже существующих заметок, а также их удаление.

5.1. Страница для редактирования заметок

В этом упражнении необходимо создать страницу **editnote.php**, добавить название и пояснительный текст. Переход на эту страницу будет осуществляться со страницы **comments.php** (т.к. в начале этой страницы выводится текст комментируемой заметки).

1. Откройте страницу **comments.php**. Создайте между текстом комментируемой заметки и повторяющейся областью комментариев пустой абзац

и введите текст «*Изменить заметку*». Сделайте ее гиперссылкой для перехода на страницу **editnote.php**.

2. Гиперссылка на **editnote.php**. Для передачи информации на страницу **editnote.php** о том, какая именно заметка модифицируется (заметка с каким ***id***), необходимо передать идентификатор заметки со страницы **comments.php** в строке URL-адреса через гиперссылку.

3. При его получении на странице **editnote.php** используется метод GET (принцип работы аналогичен тому, что был использован при передачи идентификатора заметки со страницы **default.php** на страницу **comments.php**), *см. рисунок 5.1*:

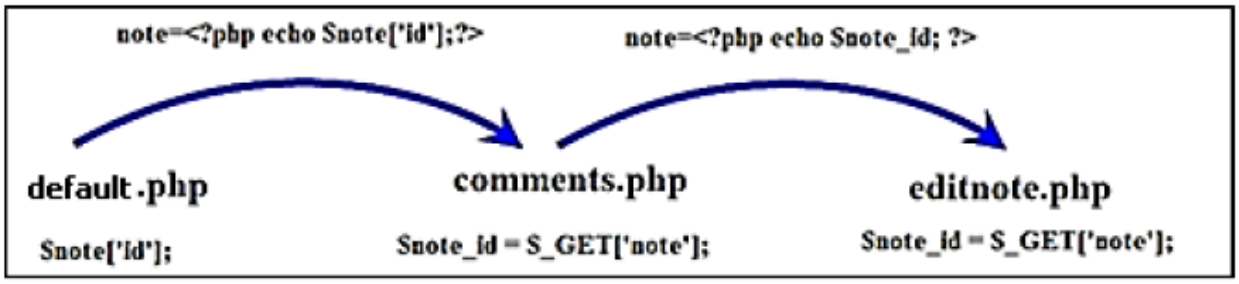


Рисунок 1. - Схема обмена данными методом GET

Дополните гиперссылку со страницы **comments.php** на страницу **editnote.php**:

*<a href="editnote.php?note=<?php echo $note_id; ?>">Изменить заметку *

4. Работа со страницей **editnote.php**

4.1. На странице **editnote.php** создайте html-форму с именем «***editnote***», метод обработки данных – POST.

4.2. На форме разместите два поля: одно (типа *text*) для изменения заголовка заметки – с именем «***title***», другое (*textarea*) для изменения самой заметки – с именем «***article***». Добавьте параметры размера элементов формы.

4.3. Также поместите на поле кнопку отправки с именем «***submit***».

4.4. Далее необходимо создать php-скрипт для обработки данных формы. Этот скрипт должен выполнять следующее:

- Отображать редактируемую заметку в полях формы (т.е. помещать данные из базы в поля формы);
- Получать измененные данные из формы;
- Передавать измененные данные в таблицу.

5. Заполнение полей формы

5.1. Введите переменную ***$note_id***, которая получит в качестве значения идентификатор обрабатываемой заметки. Это значение она должна получить через массив $_GET.

5.2. Реализуйте соединение с сервером.

5.3. Выберите базу данных.

5.4. Далее необходимо сформировать запрос на получение заметки с выбранным ***id*** из базы данных, для размещения ее в полях формы. Запрос реализуется с помощью оператора SELECT, условием запроса должно быть ***id*** выбранной заметки.

5.5. Реализуйте сформированный запрос.

5.6. С помощью функции ***mysqli_fetch_array()*** поместите результат выполнения запроса (т.е. полученную строку) в массив.

Вариант реализации кода:

```
<?php
//получение идентификатора
$note_id = $_GET['note'];
//Соединение с сервером и выбор базы данных
require_once ("MySiteDB.php");
//Запрос к БД на получение строки, содержащей заметку с выбранным id
$query = "SELECT * FROM notes WHERE id = $note_id";
//Реализация запроса к БД
$result = mysqli_query ($link, $query);
//Помещение выбранной строки в массив
$edit_note = mysqli_fetch_array ($result);
?>
```

6. Необходимо, чтобы записи полученной заметки отображались в соответствующих полях формы. Для этого:

6.1. Добавьте на html-форму скрытое поле с именем note (оно будет содержать id заметки).

6.2. В html-форме задаем значение value для всех элементов из массива:

```
<!-- $edit_note - это имя массива, в который помещается результат выполния функции mysqli_fetch_array(); -->
<form method="post">
<p>Заголовок заметки: <input type="text" name="title"
value = "<?php echo $edit_note['title'];?>" /></p>
<p>Текст заметки: <textarea name="article">
<?php echo $edit_note['article'];?></textarea></p>
<input type="hidden" name = "note" value="<?php echo $edit_note['id']?>" />
<input type="submit" name="submit" value="Изменить" />
```

</form>

7. Получение данных из формы после изменения. Принцип реализации похож на добавление новой заметки:

7.1. Получите из формы измененные данные с помощью метода $_POST;

7.2. Передайте данные в таблицу с помощью SQL-запроса. Разница заключается только в SQL-запросе - при добавлении используется INSERT, а при обновлении UPDATE.

Оператор UPDATE обновляет поля таблицы в соответствии с их новыми значениями в строках. Синтаксис запроса на обновление:

UPDATE tblName SET fieldName1 = expr1, fieldName2 = expr2, ... ,

fieldName N = expr N WHERE ...

где *tblName* – имя таблицы, *fieldName = expr* - указывается, какие именно поля надо изменить и какими должны быть их новые значения.

Вариант кода получения и передачи данных из формы

```
<?php
//Собственно обновление данных
//Получение обновленных значений из формы
$title = $_POST['title'];
$article = $_POST[‘article’];
//Создание запроса на обновление
$update_query = "UPDATE notes SET title = '$title', article = '$article’
WHERE id = $note_id";
//Реализация запроса на обновление
$update_result = mysqli_query ($link, $update_query);
?>
```

8. Проверьте корректность работы скриптов.

9. Создайте гиперссылку для возврата на страницу комментариев.

Вариант полной реализации editnote.php кода

```
<?php
$note_id = $_GET['note'];
require_once ("MySiteDB.php");
$query = "SELECT * FROM notes WHERE id = $note_id";
$result = mysqli_query ($link, $query);
$edit_note = mysqli_fetch_array ($result);
?>
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
```

```
</head>
<body>
<form method="post">
<p>Заголовок заметки: <input type="text" name="title"
value = "<?php echo $edit_note['title'];?>" /></p>
<p>Текст заметки: <textarea name="article">
<?php echo $edit_note['article'];?></textarea></p>
<input type="submit" name="submit" value="Изменить" />
</form>
<a href = "default.php"> На главную </a>
</body>
</html>
<?php
$title = $_POST['title'];
$article = $_POST['article'];
$update_query = "UPDATE notes SET title = '$title', article = '$article'
WHERE id = $note_id";
$update_result = mysqli_query ($link, $update_query);
?>
```

5.2. Создание страницы удаления заметок

Самостоятельно создайте страницу для удаления заметки deletenote.php. Переход на эту страницу также должен осуществляться со страницы comments.php.

Для реализации удаления записи из БД используется SQL- оператор DELETE. Синтаксис оператора DELETE:

DELETE FROM tblName WHERE …

где tblName – имя таблицы.

Т.к. при создании связи между таблицами **notes** и **comments** было реализовано каскадное удаление, то при удалении заметки из таблицы **notes** автоматически должны удаляться все комментарии к ней из таблицы **comments**. Убедитесь в этом при проверке работы оператора DELETE.

Вариант кода

```
<?php
require_once("mysitedb.php");
$note_id = $_GET['note'];
mysqli_select_db($link, $db);
$query = "DELETE FROM notes WHERE id = $note_id";
$res = mysqli_query($link, $query);
//Работа с заголовками (см. документацию php)
```

//header("Location: default.php");

header("refresh:5;url = default.php");

echo 'Your note was deleted. You\'ll be redirected in about 5 secs.

If not, click here.';

?>

Темы лабораторных работ

1. Работа с заметками

1.1. Работа со страницей default.php

1. Необходимо сделать так, чтобы последняя добавленная заметка отображалась в самом верху списка заметок. Для этого отредактируйте код SQL-запроса к БД таким образом, чтобы данные передавались в необходимом порядке. С этой целью используется следующий синтаксис:

SELECT *fieldName* FROM t*blName* ORDER BY *fieldName order*

fieldName – имя поля (полей) таблицы,

t*blName* – имя таблицы – источника,

order – порядок следования записей. Он может быть **ASC** – по возрастанию, **DESC** – по убыванию, **RAND** – в случайном порядке.

Например:

*SELECT * FROM table ORDER BY name ASC*

т.е. необходимо выбрать все поля из таблицы ***table*** *и расположить их в порядке возрастания значений поля* ***name*** *(т.е. в алфавитном порядке, если поле строкового типа).*

1.2. Работа с комментариями к заметкам

Самостоятельно создайте страницу для добавления комментариев к заметкам.

2. Страница статистики inform.php

В ходе выполнения лабораторной работы будет организована работа и выведены на страницу статистики следующие данные web-сайта:

1. Сколько всего было сделано записей в блоге;
2. Сколько комментариев было добавлено;
3. Сколько записей было сделано за последний месяц;
4. Сколько комментариев было оставлено за последний месяц;
5. Какая заметка была сделана последней;
6. Какую заметку больше всего комментировали.

2.1. Общее количество заметок и общее количество комментариев

1. Установите подключение к серверу и выберите базу данных.

2. С помощью SQL-запроса необходимо вычислить общее количество заметок в блоге. Для этого используется SQL-функция COUNT(). Данная функция возвращает количество строк, которые соответствуют определенным критериям. Синтаксис функции:

***SELECT** COUNT (fieldName) **FROM** tblName*

Данная функция является **агрегатной**, т.е. позволяет выполнять различные действия сразу над многими записями.

Вариант реализации кода

//Вычисление количества заметок

$query_allnotes = "SELECT COUNT(id) AS allnotes FROM notes";

$allnotes = mysqli_query ($link, $query_allnotes) or die (mysqli_error());

//mysqli_error()возвращает строку ошибки последней операции с MySQL

$row_allnotes = mysqli_fetch_assoc ($allnotes);

$allnotes_num = $row_allnotes['allnotes'];

mysqli_free_result ($allnotes);

//mysqli_free_result() освобождает память от результата запроса

3. Аналогичным образом реализуйте подсчет общего количества комментариев.

2.2. Подсчет количества заметок и комментариев за последний месяц

В ходе выполнения этого упражнения необходимо реализовать следующий алгоритм работы:

- Вычислить начальную и конечную даты текущего месяца;
- Подставить результаты этих вычислений в условие фильтрации SQL-запроса.

1. Работа с датой

//Функция getdate() возвращает массив, содержащий информацию о различных составляющих текущей даты (чтобы дальше работать с ними "по частям"). В массив помещаются: секунды, минуты, часы, порядковый номер дня, порядковый номер месяца, порядковый номер года, название дня недели, название месяца, количество секунд с начала эпохи Unix.

$date_array = getdate();

//Вычисление начальной даты текущего месяца

//Функция mktime() возвращает объединенное значение времени. Аргументы: кол-во часов, минут, секунд, № месяца, число, год.

$begin_date = date ("Y-m-d", mktime(0,0,0, $date_array['mon'],1, $date_array['year']));

//Т.к. время в данном случае не нужно - поставлены нули.

//Возвращенное функцией mktime() значение приведено к воспринимаемому MySQL параметру даты "Y-m-d".

//Вычисление конечной даты текущего месяца

$end_date = date ("Y-m-d", mktime(0,0,0, $date_array['mon'] + 1,0, $date_array['year']));

//Здесь все аналогично, кроме того, что мы вводим число месяца, равное нулю; на основании этого функция mktime(), встретив дату с нулевым числом, вернет последнее число предыдущего месяца (28, 29, 30 или 31).

2. Запрос на получение заметок за последний месяц

```
$query_lmnotes = "SELECT COUNT(id) AS lmnotes FROM notes
WHERE created>='$begin_date' AND created<='$end_date'";
$lmnotes = mysqli_query ($link, $query_lmnotes)or die (mysqli_error());
$row_lmnotes = mysqli_fetch_assoc ($lmnotes);
$lmnotes_num = $row_lmnotes['lmnotes'];
mysqli_free_result ($lmnotes);
//$begin_date – первое число месяца,
//$end_date – последнее число месяца.
```

3. Аналогичным образом вычислите количество комментариев за последний месяц.

2.3. Последняя добавленная заметка

1. Для вывода последней добавленной заметки необходимо использовать оператор LIMIT в конструкции SELECT.

- Выражение LIMIT используется для ограничения количества строк, возвращенных командой SELECT. LIMIT принимает один или два числовых аргумента. Эти аргументы должны быть целочисленными константами. Если заданы два аргумента, то первый указывает на начало первой возвращаемой строки, а второй задает максимальное количество возвращаемых строк. При этом смещение начальной строки равно 0, а не 1 (т.к. первый элемент массива строк имеет индекс 0).

Например:

SELECT * FROM table LIMIT 5,10; // возвращает строки 6-15

Вариант реализации кода

//Последняя добавленная заметка

//Таблица notes сортируется по дате публикации заметки по убыванию, а затем из нее берется только самая первая запись (LIMIT 0,1) - "начиная с нулевой записи выбрать одну запись"

```
$query_last_note = "SELECT id, title FROM notes
```

ORDER BY created DESC LIMIT 0,1";
$lastnote = mysqli_query ($link, $query_last_note) or die (mysqli_error());
$row_lastnote = mysqli_fetch_assoc ($lastnote);
mysqli_free_result ($lastnote);

2.4. Самая комментируемая заметка

В ходе выполнения этого упражнения необходимо реализовать следующий алгоритм работы:

- Связать таблицы **notes** и **comments** по полям ***id*** и ***art_id*** соответственно, чтобы затем вычислить количество комментариев для каждой заметки;
- Выполнить группировку таблицы **comments** по идентификатору заметки.
- Вычислить количество комментариев для каждой заметки.
- Отсортировать результат по количеству комментариев для каждой заметки по убыванию.
- Вывести первую запись из получившегося набора записей.

1. Построение SQL-запроса
$query_mcnote = "SELECT notes.id, notes.title FROM comments, notes
WHERE comments.art_id=notes.id
GROUP BY notes.id
ORDER BY COUNT(comments.id) DESC LIMIT 0,1";
В тексте запроса:

GROUP BY – оператор группировки. Группировка – это объединение записей в группы по какому-либо критерию (т.н. критерию группировки), который записывается сразу после оператора (в данном случае группировка по полю id таблицы notes).

ORDER BY COUNT (comments.id) DESC LIMIT 0,1 – осуществление сортировки по убыванию результатов выполнения агрегирующей функции COUNT() по id комментариев и вывод первой записи (т.е. записи с самым большим количеством комментариев).

Реализуйте данный запрос и поместите его результат в массив.

2.5. Размещение данных на странице

С помощью php-сценариев и оператора echo вывести результаты на страницу сайта. Ниже представлен вариант реализации кода вывода информации:

<html>
<body>
*Сделано записей - <?php echo $allnotes_num; ?>
*
*Оставлено комментариев - <?php echo $allcomments_num; ?>
*

*За последний месяц я создал записей - <?php echo $row_lmnotes['lmnotes'];?>
*

*За последний месяц оставлено комментариев - <?php echo $row_lmcomments['lmcomments'];?>
*

Моя последняя запись -

<a href="comments.php?note=<?php echo $row_lastnote['id'];?>">

*<?php echo $row_lastnote['title'];?>
*

Самая обсуждаемая запись -

<a href="comments.php?note=<?php echo $row_mcnote['id'];?>">

*<?php echo $row_mcnote['title'];?>

*

* Возврат на главную страницу сайта *

</body>

</html>

В представленном коде:

$allnotes_num, $allcomments_num, $row_imnotes, $row_lmcomments, $row_lastnote, $row_mcnote – массивы, в которые помещаются результаты выполнения функций mysqli_fetch_array(), вызываемых в ранее созданном коде для получения и хранения соответствующих данных.

Ниже представлен возможный вариант реализации всего кода страницы inform.php.

<?php require_once ("MySiteDB.php");

//Вычисление количества заметок

$query_allnotes = "SELECT COUNT(id) AS allnotes FROM notes";

$allnotes = mysqli_query ($link, $query_allnotes) or die (mysqli_error());

$row_allnotes = mysqli_fetch_assoc ($allnotes);

$allnotes_num = $row_allnotes['allnotes'];

mysqli_free_result ($allnotes);

//Вычисление количества комментариев

$query_allcomments = "SELECT COUNT(id) AS allcomments FROM comments";

$allcomments = mysqli_query ($link, $query_allcomments) or die (mysqli_error());

$row_allcomments = mysqli_fetch_assoc ($allcomments);

$allcomments_num = $row_allcomments['allcomments'];

mysqli_free_result ($allcomments);

//Работа с датой

$date_array = getdate();

$begin_date = date ("Y-m-d", mktime(0,0,0, $date_array['mon'],1, $date_array['year']));

```
$end_date = date ("Y-m-d", mktime(0,0,0, $date_array['mon'] + 1,0, $date_array['year']));
//Заметки за последний месяц
$query_lmnotes = "SELECT COUNT(id) AS lmnotes FROM notes
WHERE created>='$begin_date' AND created<='$end_date'";
$lmnotes = mysqli_query ($link, $query_lmnotes)or die (mysqli_error());
$row_lmnotes = mysqli_fetch_assoc ($lmnotes);
$lmnotes_num = $row_lmnotes['lmnotes'];
mysqli_free_result ($lmnotes);
//Комментарии за последний месяц
$query_lmcomments = "SELECT COUNT(id) AS lmcomments FROM comments WHERE created >= '$begin_date' AND created <= '$end_date'";
$lmcomments = mysqli_query ($link, $query_lmcomments)or die (mysqli_error());
$row_lmcomments = mysqli_fetch_assoc ($lmcomments);
$lmcomments_num = $row_lmcomments['lmcomments'];
mysqli_free_result ($lmcomments);
//Последняя добавленная заметка
$query_last_note = "SELECT id, title FROM notes
ORDER BY created DESC LIMIT 0,1";
$lastnote = mysqli_query ($link, $query_last_note) or die (mysqli_error());
$row_lastnote = mysqli_fetch_assoc ($lastnote);
mysqli_free_result ($lastnote);
//Самая комментируемая заметка
$query_mcnote = "SELECT notes.id, notes.title FROM comments, notes
WHERE comments.art_id=notes.id
GROUP BY notes.id
ORDER BY COUNT(comments.id) DESC LIMIT 0,1";
$mcnote = mysqli_query($link, $query_mcnote)or die (mysqli_error());
$row_mcnote = mysqli_fetch_assoc($mcnote);
mysqli_free_result ($mcnote);
?>
<html>
<body>
Сделано записей - <?php echo $allnotes_num; ?><br>
Оставлено комментариев - <?php echo $allcomments_num; ?><br>
За последний месяц я создал записей - <?php echo $row_lmnotes['lmnotes'];?><br>
За последний месяц оставлено комментариев - <?php echo $row_lmcomments['lmcomments'];?><br>
```

```
Моя последняя запись -
<a href="comments.php?note=<?php echo $row_lastnote['id'];?>">
<?php echo $row_lastnote['title'];?></a><br>
Самая обсуждаемая запись -
<a href="comments.php?note=<?php echo $row_mcnote['id'];?>">
<?php echo $row_mcnote['title'];?> </a><br><br>
<p><a href="default.php">Возврат на главную страницу сайта </a></p>
</body>
</html>
```

Темы лабораторных работ

В ходе выполнения данной лабораторной работы будут изучены основные функции работы со строками, а также поиск информации по web-сайту (по одному и нескольким словам поискового запроса).

1. Реализация поиска по сайту

В ходе выполнения данного упражнения с использованием функций работы со строками необходимо реализовать возможность поиска по ключе-вому слову заметки на главной странице сайта (возможны два варианта реа-лизации: поиск по одному слову и поиск по фразе).

1. Поиск по одному ключевому слову: Для реализации поиска по одному слову можно использовать оператор LIKE и заменители символов %.

Вариант реализации кода:

```
//Поиск по одному слову
$user_search = $_GET['usersearch'];
if (!empty($user_search))
{
$query_usersearch = "SELECT * FROM notes
WHERE title LIKE '%$user_search%'
OR article LIKE '%$user_search%'";
$result_usersearch = mysqli_query($link, $query_usersearch);
while ($array_usersearch = mysqli_fetch_array($result_usersearch))
{
echo $array_usersearch['id'];
echo $array_usersearch['title'];
echo $array_usersearch['article'];
}
}
```

2. Реализация поиска по фразе:

2.1. Фразу надо разбить на отдельные слова (подстроки) и поместить в массив подстрок с помощью функции ***explode();***

```
$search_query = “SELECT * FROM tableName”
$where_clause = ‘ ‘; // условие поиска
$user_search = $_GET['usersearch']; // получаем данные из поля поиска
$search_words = explode(' ', $user_search);
foreach($search_words as $word)
{
// Формируем условие поиска
$where_clause .= ” fieldName LIKE ‘%$word%’ OR “
}
if (!empty($where_clause))
{
$search_query .= “ WHERE $where_clause “;
}
```

2.2. Для того, чтобы в конце строки запроса не была оператора OR, можно использовать функцию ***implode()***, создающую строку из массива подстрок, переданного ей в качестве аргумента. Представленный ранее код можно изменить следующим образом:

```
$search_query = “SELECT * FROM tableName”
$where_list = array();
$user_search = $_GET['usersearch'];
$search_words = explode(' ', $user_search);
foreach($search_words as $word)
{
//В конец массива добавляется новый элемент
$where_list[] = "article LIKE '%$word%'";
}
$where_clause = implode (' OR ', $where_list);
if (!empty($where_clause))
{
$query_usersearch .=" WHERE $where_clause";
}
```

Вариант реализации кода:

```
//Поиск по фразе (по содержанию заметки)
```

```
$user_search = $_GET['usersearch'];
$where_list = array();
$query_usersearch = "SELECT * FROM notes";
$search_words = explode(' ', $user_search);
foreach($search_words as $word)
{
$where_list[] = " article LIKE '%$word%'";
}
$where_clause = implode (' OR ', $where_list);
if (!empty($where_clause))
{
$query_usersearch .=" WHERE $where_clause";
}
$res_query = mysqli_query($link, $query_usersearch);
while ($res_array = mysqli_fetch_array($res_query))
{
echo $res_array['id'], "<br>";
echo $res_array[‘article’], "<br>", "<hr>", "<br>";
}
```

2. Обработка строки поиска

Строка поиска должна содержать несколько слов, разделенных одним пробелом. Но надо учитывать, что пользователь может вводить слова поискового запроса через запятую (например, «заметка, моя, новая»). Такую строку необходимо перед передачей в запрос в базе данных обработать и привести к необходимому виду. Минимальная обработка осуществляется в два этапа:

- Замена запятых на пробелы;
- Удаление лишних пробелов между словами строки.

1. Замена запятых на пробелы осуществляется с помощью функции ***str_replace().*** Эта функция заменяет строку поиска на строку замены. Обязательными являются три аргумента: что заменить, чем заменить, где заменить. Следовательно, в данном случае вызов функции будет выглядеть следующим образом:

```
str_replace (‘ , ’, ‘ ‘, $user_search);
```

2. Удаление лишних пробелов между словами строки поискового запроса: в том случае, когда запятые были заменены на пробелы, появились лишние пробелы (т.е. более одного) между словами строки запроса. Если их не удалить, то в запросе они будут рассматриваться как пустые элементы массива, на

основании которого формируется запрос к базе данных. Следовательно, при таком запросе будут выдаваться все записи базы данных. Для удаления пустых элементов массива можно сделать следующее:

- Создать новый массив, в котором будут сохраняться только действительные (непустые) критерии поиска. На основании этого массива будет строиться запрос к базе данных.
- Для создания этого массива можно пройти в цикле *foreach* все элементы уже существующего созданного ранее массива, используя управляющую конструкцию *if* найти все непустые элементы и скопировать их в новый массив.

2.1. В новый массив ***$final_search_words*** помещаются непустые элементы уже существующего массива ***$search_words***.

```
//Извлечение критериев поиска в массив
//Замена запятых на пробелы
$clean_search = str_replace(',', ' ', $user_search);
$search_words = explode(' ', $user_search);
//Создаем еще один массив с окончательными результатами
$final_search_words = array();
//Проходим в цикле по каждому элементу массива $search_words.
//Каждый непустой элемент добавляем в массив с названием
//$final_search_words
if (count($search_words) > 0)
{
foreach($search_words as $word)
{
if (!empty($word))
{
$final_search_words[] = $word;
}
}
}
```

2.2. Далее реализация работы с массивом такая же, разница – в используемом массиве (работа происходит с новым массивом ***$final_search_words***).

```
foreach ($final_search_words as $word)
{
$where_list[] = " article LIKE '%$word%'";
}
```

```
$where_clause = implode (' OR ', $where_list);
if (!empty($where_clause))
{
$query_usersearch .=" WHERE $where_clause";
}
```
Вариант реализации кода:
```
//Поиск по фразе (по содержанию заметки)
$user_search = $_GET['usersearch'];
$where_list = array();
$query_usersearch = "SELECT * FROM notes";
$clean_search = str_replace(',', ' ', $user_search);
$search_words = explode(' ', $user_search);
//Создаем еще один массив с окончательными результатами
$final_search_words = array();
//Проходим в цикле по каждому элементу массива $search_words.
//Каждый непустой элемент добавляем в массив $final_search_words
if (count($search_words) > 0)
{
foreach($search_words as $word)
{
if (!empty($word))
{
$final_search_words[] = $word;
}
}
}
//работа с использованием массива $final_search_words
foreach ($final_search_words as $word)
{
$where_list[] = " article LIKE '%$word%'";
}
$where_clause = implode (' OR ', $where_list);
if (!empty($where_clause))
{
$query_usersearch .=" WHERE $where_clause";
}
$res_query = mysqli_query($link, $query_usersearch);
while ($res_array = mysqli_fetch_array($res_query))
{
```

```
echo $res_array['id'], "<br>";
echo $res_array[‘article’], "<br>", "<hr>", "<br>";
}
?>
```

Темы лабораторных работ

В данной лабораторной работе будет изучены основные возможности PHP для реализации передачи файлов на сервер.

1. Вывод списка файлов

1. Создайте папку **photo** для размещения изображений. Поместите в эту папку несколько изображений.

2. Создайте страницу **photo.php**, разместите на ней поясняющий текст, ниже поясняющего текста – две горизонтальные линии (между этими линиями будет выводиться список имеющихся на web-узле изображений и ссылками на эти изображениями), см. *рисунок 1.*

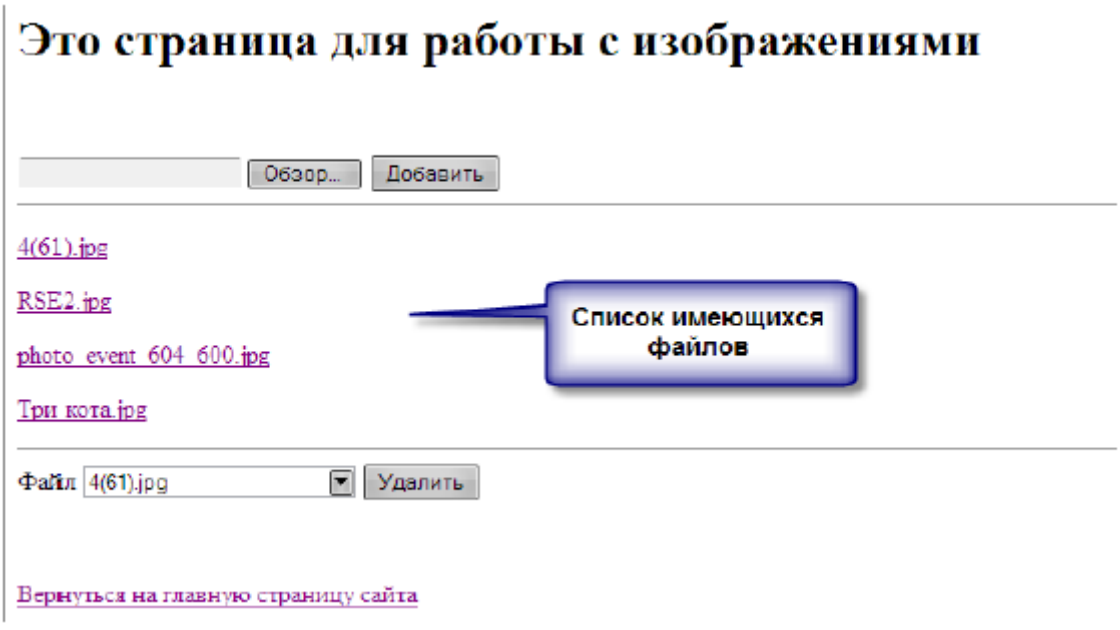


Рисунок 1 - Схема страницы photo.php

3. Для вывода списка файлов, уже имеющихся на сервере, необходимо создать два сценария:

• первый будет формировать список файлов с изображениями,

• второй - выводить этот список на страницу.

3.1. Формирование списка файлов. Для формирования списка файлов необходимо получить путь к целевой папке, где хранятся необходимые файлы, а затем создать массив, в который будет помещаться необходимая информация о файлах (имена, пути к ним);

Внимание! Представленные в листинге пути к файлам написаны для работы в папке *localhost\photo* (для Open Server) или *localhost\www\photo* (для Denwer).

Внимательно отнеситесь к прописыванию путей к файлам. Например, если Ваша папка находится по следующему адресу *C:\OpenServer\domains\localhost\MyTravelNotes\photo*, то Вам необходимо это учитывать в пути к файлам и, например, вместо

$_SERVER['DOCUMENT_ROOT'] . "/photo";

писать

*$_SERVER['DOCUMENT_ROOT'] . "/**mytravelnotes**/photo";*

Фрагменты, содержащие пути, на которые необходимо обратить внимание, выделены в листинге.

Вариант реализации кода

```
<?php
//Получаем полный путь к папке, где хранятся графические файлы
$image_dir_path = $_SERVER['DOCUMENT_ROOT'] . "/photo";
//Запускаем просмотр папки. Функция opendir() возвращает идентификатор
//папки
$image_dir_id = opendir($image_dir_path);
//$array_files - массив, в который будут помещаться все найденные файлы
$array_files = null;
//Служебная переменная, используемая для вычисления индекса следующего
//элемента массива $array_files
$i = 0;
//Запускаем цикл просмотра
while(($path_to_file = readdir($image_dir_id)) !== false)
//Функция readdir() возвращает полный путь к очередному файлу,
хранящемуся //в папке, идентификатор которой был возвращен функцией
opendir() и передан //в качестве параметра.
//$path_to_file получает полный путь к файлу для дальнейшей обработки.
Если в папке нет непросмотренных файлов - возвращается логическое значение
false
{
if(($path_to_file !=".") && ($path_to_file !=".."))
//Точки обозначают вложенные файлы: одна точка - текущая папка, две
точки // - папка, в которую вложена текущая папка.
{
$array_files[$i] = basename($path_to_file);
$i++;
//Помещаем имя найденного файла в массив $array_files. Функция basename()
//позволяет получить имя файла из полного пути к нему.
}
```

```
}
closedir($image_dir_id);
//closedir() удаляет из памяти переданный ей идентификатор папки, таким //образом завершая просмотр.
?>
```

3.2. Вывод списка файлов на страницу.

3.1.1. Найдите код, создающий две горизонтальные линии. Создайте следующий сценарий вывода списка файлов:

```
<?php
//Получаем количество элементов массива $array_files, т.е. количество //найденных файлов.
$array_files_count = count($array_files);
if ($array_files_count)
{
?>
<hr />
<?php
sort($array_files);
for ($i=0; $i<$array_files_count; $i++)
{
//Выводим мена хранящихся в массиве файлов на страницу
?>
<p><a href="/photo/<?php echo $array_files[$i]; ?>" target="_blank">
<?php echo $array_files[$i]; ?></a></p>
<?php
}
?>
<hr />
<?php
}
?>
```

3.1.2. Создайте гиперссылку возврата на главную страницу и гиперссылку с главной страницы на страницу **photo.php** («Фото» в наборе гиперссылок).

4. Проверьте работу сценариев.

2. Загрузка файлов на сервер

Отправка файлов на web-сайт состоит из двух этапов:

• создание необходимой формы для отправки файлов на странице сайта;

• написание сценария PHP для получения отправленного посетителем сайта файла.

1. Создание формы. Стандарт HTML предусматривает т.н. ***поле выбора файла***, в которое посетитель должен будет ввести имя отправляемого файла. От обычного поля ввода оно отличается тем, что позволяет работать с окном открытия файлов Windows, а также оно отправляет серверной программе не введенное в него имя файла, а сам файл. Поле выбора файла создается с помощью тэга INPUT, атрибут type которого имеет значение “file”. Также в форму с данным полем необходимо добавить скрытое поле, задающее максимальный размер отправляемого файла, с именем MAX_FILE_SIZE. Значение этого поля (т.е. максимально возможный размер файла) определяется в байтах.

Форма, из которой будет отправляться файл, должна кодировать данные по методу multipart/form-data и передавать данные только методом POST.

```
<!-- Форма для отправки файла на сервер -->
<form name = "file_upload" action="photo.php"
enctype="multipart/form-data" method="post">
<input type="file" name="file_upload" />
<input type="hidden" name="MAX_FILE_SIZE" value="65536" />
<input type="submit" name="submit" value="Добавить" />
</form>
```

В данном случае максимальный размер файла равен 65536 байта, т.е. 64 Кбайта. Данный размер при необходимости может быть увеличен.

2. Получение отправленного файла. Все принятые файлы помещаются интерпретатором PHP в особую служебную папку, которая не является частью сайта (т.н. «буферная» папка). Сведения обо всех принятых и помещенных в буферную папку файлах хранятся во встроенном массиве PHP $_FILES. Каждый элемент массива соответствует принятому файлу и представляет собой вложенный массив, содержащий различные сведения о файле. В начало страницы поместите следующий код:

```
<?php
//Сценарий отправки файла на сервер
//Проверяем, была ли выполнена отправка файла. Далее реализуем сценарий.
if (isset($_POST["MAX_FILE_SIZE"]))
{
$tmp_file_name = $_FILES["file_upload"]["tmp_name"];
$dest_file_name = $_SERVER['DOCUMENT_ROOT'].
"/photo/".$_FILES["file_upload"]["name"];
move_uploaded_file($tmp_file_name, $dest_file_name);
}
```

?>

Функция ***move_uploaded_file (string filename, string destination)*** проверяет, является ли файл *filename* загруженным на сервер (переданным по протоколу HTTP POST). Если файл действительно загружен на сервер, он будет перемещён в место, указанное в аргументе *destination*.

3. Удаление файла с сервера

Для удаления файла надо выбрать необходимый файл. Это делается с помощью формы, в которую помещается список для выбора удаляемого файла из имеющихся на сервере файлов. Для заполнения этого списка необходимо использовать уже созданный массив $array_files.

Реализация удаления происходит в два этапа:

- Создание формы для удаления файла;
- Создание сценария удаления файла.

1. Создание формы для выбора удаляемого файла

```
<!-- Форма для удаления файла с сервера -->
<form name="file_delete" action="photo.php" method="post"
enctype=" multipart/form-data ">
Файл <select name = "file_delete" size="1">
<option><option></select>
<input type="submit" name="submit" value="Удалить" />
</form>
```

Парные тэги SELECT создает список. Пункты списка создаются тэгами OPTION. Сам список имеющихся на сервере файлов необходимо получить из созданного ранее массива $array_files.

```
<!-- Форма для удаления файла с сервера -->
<form name="file_delete" action="photo.php" method="post"
enctype=" multipart/form-data ">
Файл <select name = "file_delete" size="1">
<?php for ($i=0; $i<$array_files_count; $i++)
{ ?>
<option><?php echo $array_files[$i]; ?></option>
<?php } ?>
</select>
<input type="submit" name="submit" value="Удалить" />
</form>
```

Во включенном в форму цикле создается столько пунктов (тэгов <option>), сколько элементов присутствует в массиве $array_files.

2. Создание сценария удаления файла.

```
<?php
```

```
//Сценарий удаления файла
//Сначала проверяем, было ли запущено удаление файла
if (isset($_POST["file_delete"]))
{
//Формируем полное имя файла
$file_name = $_SERVER['DOCUMENT_ROOT'] . "/photo/". $_POST["file_delete"];
//Функция unlink() удаляет файл
unlink($file_name);
}
?>
```

3. Сохраните изменения, проверьте корректность работы.

4. Работа с содержимым файла

1. С помощью **PhpMyAdmin** откройте базу данных **MySiteDb** и импортируйте данные из таблицы **authors** в файл **authors.xml**. Для этого выберите таблицу **authors** – Вкладка **Экспорт** – Способ экспорта «*Быстрый – отображать минимум настроек*» - Вывод «*Сохранить на сервере в каталоге c:/openserver/userdata/temp/*» - Формат **XML** – **ОК** (см. рисунок 2)

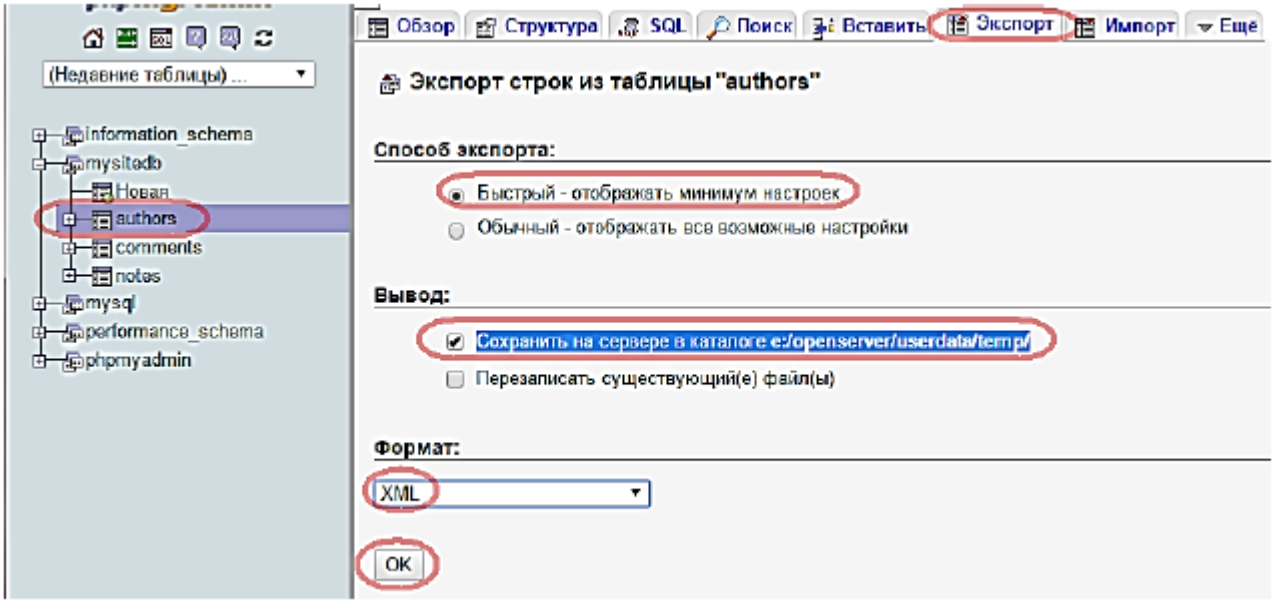


Рисунок 2 – Экспорт данных из таблицы

2. Скопируйте созданный в результате операции экспорта файл **au-thors.xml** в рабочую папку проекта. Изучите его.

3. В редакторе создайте новый файл **files.php**. В нем создайте код открытия и чтения данных из файла:

```
<?php
```

$f = fopen("authors.xml", "a+");

echo fread($f, filesize("authors.xml"));

?>

4. Убедитесь в работоспособности кода, запустив данный файл на выполнение (через адресную строку браузера).

5. Измените код на построчное чтение данных из файла:

<?php

$f = fopen("authors.xml", "a+");

echo fgets($f);

?>

6. Убедитесь в работоспособности кода. Т.к. в начале созданного xml файла много строк служебной информации, возможно вызов функции ***fgets()*** необходимо будет сделать несколько раз.

7. Закомментируйте весь только что созданный код. Напишите код, считывающий данные построчно из файла и помещающий его в массив. Выведите информацию из массива на страницы сайта:

$array = file("authors.xml");

print_r($array);

8. Закомментируйте код выше. Считайте данные из файла одной строкой:

$get_cont = file_get_contents("authors.xml");

echo $get_cont;

9. Измените данные в файле **authors.xml**, дополнив их произвольной информацией, например:

$put_cont = file_put_contents("authors.xml", "My new contents", FILE_APPEND);

echo $put_cont;

Темы лабораторных работ

В данной лабораторной работе будут реализованы основы реализации разграничения прав доступа к контенту сайта с использованием механизма сессий.

1. Создание страницы входа на сайт

1. Создайте новый файл **login.html**, содержащий форму, передающую методом **POST** файлу **login.php** (он будет создан на следующем шаге) на обработку логин и пароль посетителя сайта.

2. Создайте файл **login.php**, принимающий данные, поступившие из html-формы файла **login.html**. Далее необходимо сравнить полученные данные с теми данными, которые имеются в базе данных, и если они совпадают (т.е.

пользователь с таким логином и паролем существует), то для него необходимо запустить сессию.

Вариант реализации кода

```
<?php
require_once("MySiteDB.php");
$login = $_POST['login'];
$password = $_POST['password'];
//echo $login, $password, "<br>";
if(($login) &&($password))
{
$query = "SELECT * FROM authors WHERE login = '$login' AND password = '$password'";
$send_query = mysqli_query($link, $query);
$user_array = mysqli_fetch_array($send_query);
$login = $user_array['login'];
$rights = $user_array['rights'];
$count = mysqli_num_rows($send_query);
if ($count >0)
{
session_start();
$_SESSION['login'] = $login;
$_SESSION['rights'] = $rights;
header( "refresh:3;url = default.php" );
echo “Вход на сайт автоматически осуществится через 3 секунды или нажмите <a href="default.php">сюда</a>.';
}
else
{
echo "Извините, Вы не зарегистрированы.";
}
}
?>
```

3. Организуйте в проекте соответствующие гиперссылки для перехода между страницами.

2. Организация доступа к страницами сайта

В данном упражнении необходимо разграничить права доступа ко всем страницам сайта, согласно следующей информации:

Страница	**Права доступа**

default.php	Все пользователи
newnote.php	a
deletenote.php	a
editnote.php	a
email.php	u, a
inform.php	a
newcomment.php	u, a
photo.php	u, a

1. Возможный вариант реализации представлен на примере страницы **newnote.php**:

```
<?php
session_start();
if ($_SESSION['rights']=='a')
{
echo "hello, ".$_SESSION['login'];
?>
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>Новая заметка</title>
</head>
<body>
<!—Здесь код формы для добавления новой заметки -->
</body>
</html>
<?php
require_once ("MySiteDB.php");
$title = $_POST['title'];
$created = $_POST['created'];
$article = $_POST['article'];
if (($title)&&($created)&&($article))
{
$query = "INSERT INTO notes (id, title, created, article) VALUES (NULL, '$title', '$created', '$article')";
$result = mysqli_query ($link, $query);
}
}
```

```
else
{
echo "Извините, у Вас нет доступа";
echo "<a href = \"default.php\">На главную</a>";
}
?>
```

2. Аналогичным образом организуйте разграничение прав доступа к другим страницам сайта. При желании можно необходимый код вынести в отдельный файл и подключать его к страницам, чтобы не дублировать информацию.

3. Создание страницы выхода с сайта

Самостоятельно создайте страницу **loguot.php**, позволяющую завершать сессию и таким образом осуществлять выход пользователя с сайта. Используйте функции ***session_destroy()*** и ***session_unset()***.

4. Страница администрирования

В данном упражнении необходимо создать страницу с информацией для администратора сайта (**admin.php**), на которую будет выводиться информация о зарегистрированных пользователях, а также с помощью которой можно будет добавлять новых пользователей (для этого также необходимо будет создать страницу **newUser.php**).

1. Создайте новый файл **admin.php**. В нем (можно в табличной форме) выведите информацию на страницу сайта о зарегистрированных пользователях (т.е. записи из таблицы **authors**).

2. Проверьте корректность вывода информации.

3. В том же файле создайте форму для добавления нового пользователя. Данные должны передаваться методом **POST** на обработку в файл **newUser.php**, который будет создан на следующем шаге.

4. Создайте новый файл **newUser.php**. В нем реализуйте возможность добавления нового пользователя в таблицу зарегистрированных пользователей. Вариант реализации файла **newUser.php**:

```
<?php
require_once('mysitedb.php');
$login = $_POST['login'];
$password = $_POST['password'];
$rights = $_POST['rights'];
if(($login)&&($password)&&($rights))
{
```

```
$query = "INSERT INTO authors VALUES (NULL, '$login', '$password', '$rights')";
mysqli_query($link, $query);
header( "refresh:1;url=admin.php" );
}
?>
```

Темы лабораторных работ

В рамках этой работы, необходимо разработать программу позволяющий организовать электронный магазин.

Задание: Написать скрипт, позволяющий организовать электронный магазин. Список товаров хранится в базе данных на стороне сервера. Покупатель должен иметь возможность просмотреть все имеющиеся в наличии товары и сделать заказ. Покупатель должен иметь возможность сделать запрос, например, указав интервал цен, который его устраивает или какие-либо другие данные. До тех пор, пока покупатель выбирает отдельные товары, его заказ хранится на стороне клиента в виде cookie. После того как покупатель сформировал заказ, заказ отсылается на сторону сервера, где покупка товара учитывается в базе данных.

Вариант 1. В базе данных содержится информация о книгах: автор, название, изображение обложки, издательство, год выпуска, цена.

Вариант 2. В базе данных содержится информация об автомобилях: модель, изображение автомобиля, год выпуска, тип кузова, мощность двигателя, цвет, цена.

Вариант 3. В базе данных содержится информация о туристических поездках: страна, город, изображение городской достопримечательности, количество дней, дата поездки, класс отеля, цена.

Вариант 4. В базе данных содержится информация о журналах: название, изображение обложки, год выпуска, номер, издательство, число страниц, цена.

Вариант 5. В базе данных содержится информация о местах в отеле: название отеля, класс номера, изображение номера, количество мест в номере, цена.

Содержание отчета:

− титульный лист;

− задание;

− краткое описание использованных средств и методов;

− описание базы данных;

− исходные тексты;

− выводы по работе.

Темы лабораторных работ

Задание 1. Откройте в текстовом редакторе файл **oop\users.php** и выполните следующее задания:

1. Создайте класс **User** со свойствами **name**, **login** и **password**

2. Создайте три объекта, экземпляра класса **User**

3. Задайте произвольные значения свойств **name**, **login** и **password** для каждого из Объектов.

Задание 2.

1. В классе **User** опишите метод **showInfo**()

2. Метод **showInfo**() должен выводить значения свойств объектов

3. Вызовите метод **showInfo**() для каждого объекта.

Задание 3.

1. В классе **User** опишите конструктор

2. Конструктор должен задавать начальные значения свойств **name**, **login** и **password**

3. Создайте заново три объекта, экземпляра класса **User.**

Задание 4.

1. В классе **User** опишите метод **__clone**()

2. Метод **__clone**() должен задавать начальные значения свойств по умолчанию при копировании объектов

3. Значения свойств по умолчанию: **name** = "Guest", **login** = "guest", **password** = "qwerty"

4. Создайте четвёртый объект скопировав один из имеющихся объектов.

Задание 5.

1. Опишите класс **SuperUser** наследованный от класса **User**

2. В классе **SuperUser** опишите свойство **role** и создайте объект, экземпляр класса **SuperUser**

3. Задайте значение свойству **role** = "admin"

4. Вызовите метод **showInfo**() для созданного объекта

5. Отдельно от метода **showInfo**() выведите значение свойства **role.**

Задание 6.

1. Опишите конструктор класса **SuperUser**, который будет задавать начальные значения свойств

2. В классе **SuperUser** опишите метод **showInfo**(), который будет выводить на экран значения всех свойств

3. Создайте заново объект класса **SuperUser** и вызовите метод **showInfo**().

Задание 7.

1. Измените конструктор класса **SuperUser**, вызвав родительский конструктор

2. Передайте родительскому конструктору необходимые значения

3. Измените метод **showInfo**() класса **SuperUser**, вызвав родительский метод **showInfo**()

4. Передайте родительскому методу необходимые значения.

Задание 8.

1. Сделайте все параметры конструктора класса **User** параметрами по умолчанию со значениями «пустая строка» ("")

2. В конструкторе класса **User** генерируйте исключение, если введены не все данные

3. Опишите перехват исключения и выводите в браузер сообщение об ошибке

4. Попробуйте создать экземпляр класса **User** без какого-либо параметра (-ов).

Задание 9.

1. Создайте константу класса **User INFO_TITLE**

2. Присвойте константе **INFO_TITLE** строковое значение «Данные пользователя:»

3. Обратитесь к константе **INFO_TITLE** перед вызовами метода **showInfo**()

4. Запустите код и проверьте его работоспособность

5. Создайте метод **showTitle**() в классе **User**

6. Опишите метод **showTitle**(), чтобы он выводил в браузер значение константы **INFO_TITLE**

7. Обратитесь к метод **showTitle**() перед вызовами метода **showInfo**().

Задание 10.

1. Создайте абстрактный класс **AUser**

2. В абстрактном классе **AUser** объявите абстрактный метод **showInfo**()

3. Обновите класс **User**, унаследовав его от абстрактного класса **AUser**

4. Внесите в класс **User** необходимые изменения.

Задание 11.

1. Создайте интерфейс **ISuperUser**

2. В интерфейс **ISuperUser** объявите метод **getInfo**()

3. Опишите метод **getInfo**() в классе **SuperUser**

4. Метод **getInfo**() должен возвращать ассоциативный массив, в котором именами ячеек являются имена свойств объекта, а значениями ячеек – значения свойств объекта

6. Вызовите метод **getInfo**() для экземпляра класса **SuperUser**

7. В цикле выведите данные, полученные с помощью метода **getInfo**().

Задание 12.

1. Опишите в классах **User** и **SuperUser** статические свойства для подсчета количества созданных объектов

2. Присвойте этим свойствам начальные значения (0)

3. В конструкторах инкрементируйте данные свойства

4. После создания экземпляров классов **User** и **SuperUser** выведите на экран количество тех и других объектов.

Задание 13.

1. Опишите функцию **checkObject**(), которая принимает в качестве входящего параметра объект

2. Проверьте Ваш объект, используя следующие условия:

a. Если объект является экземпляром класса SuperUser, выводите сообщение, что данный пользователь обладает правами администратора

b. Если объект является экземпляром класса User, выводите сообщение, что данный пользователь является обычным пользователем

c. Если объект не является ни тем, ни другим, выводите сообщение, что данный пользователь - неизвестный пользователь.

Задание 14.

1. В директории «oop» создайте файл «User.class.php»

2. Перенесите описание класса **User** из файла «users.php» в файл «User.class.php»

3. В директории «oop» создайте файл «SuperUser.class.php»

4. Перенесите описание класса **SuperUser** из файла «users.php» в файл «SuperUser.class.php»

5. Посторите те же действия для класса **AUser** и интерфейса **ISuperUser**

6. В файле «users.php» опишите функцию **__autoload**(), которая автоматически присоединяет файлы с описанием классов к файлу «users.php».