

ГОУ ВПО «ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ»
ФАКУЛЬТЕТ МАТЕМАТИКИ И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ
Кафедра прикладной механики и компьютерных технологий



УТВЕРЖДАЮ:

Проректор по научно-методической
и учебной работе

 Е.И. Скафа

«22» апреля 2020 г.

МП

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ
«РАЗРАБОТКА СЕРВЕРНЫХ WEB ПРИЛОЖЕНИЙ»

Направление подготовки:	09.04.04 Программная инженерия
Магистерская программа:	Программная инженерия
Образовательная программа:	академическая магистратура
Квалификация:	магистр
Форма обучения:	<u>очная</u> , очно-заочная, заочная нужное подчеркнуть

Донецк 2020

УТВЕРЖДАЮ:

Декан факультета математики
и информационных технологий

И. А. Моисеенко

«16» апреля 2020 г.

МП

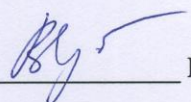
Программа составлена на основании Федерального государственного образовательного стандарта высшего образования (ФГОС ВО) направления подготовки 09.04.04 Программная инженерия, утвержденного приказом Министерства образования и науки Российской Федерации от «19» сентября 2017 г. № 932;

Порядка организации учебного процесса в образовательных организациях высшего профессионального образования Донецкой Народной Республики, утвержденного приказом Министерства образования и науки ДНР № 1171 от «10» ноября 2017 г.;

учебного плана и основной образовательной программы Программная инженерия, направления подготовки 09.04.04 Программная инженерия, разработанных в ГОУ ВПО «Донецкий национальный университет».


Разработчик:

доцент кафедры прикладной механики
и компьютерных технологий

 В.А. Цванг

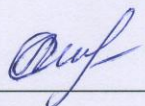
Программа учебной дисциплины утверждена на заседании кафедры прикладной механики и компьютерных технологий

Протокол № 11 от «02» апреля 2020 г.
Заведующий кафедрой

 А.С. Гольцев

Программа учебной дисциплины одобрена учебно-методической комиссией факультета математики и информационных технологий
Протокол № 8 от «15» апреля 2020 г.

Председатель учебно-методической
комиссии факультета

 Л.И. Селякова

1. ОБЛАСТЬ ПРИМЕНЕНИЯ И МЕСТО ДИСЦИПЛИНЫ В УЧЕБНОМ ПРОЦЕССЕ:

Учебная дисциплина «Разработка серверных Web приложений» относится к циклу Дисциплины (модули), Вариативная часть, Дисциплины по выбору.

Дисциплина «Разработка серверных Web приложений» является основой для Научно-исследовательской работы (НИР), Производственной (научно-исследовательской) практики и Производственной (преддипломной, подготовки ВКР: магистерской диссертации) практики, связанных с разработкой информационных систем. Поэтому умение разрабатывать современные Web приложения является обязательным для специалистов в области программной инженерии и информатики.

Содержание дисциплины является логическим продолжением содержания дисциплин:

- Операционные системы,
- Основы Интернет-технологий,
- Web-дизайн и Web-программирование
- Проектирование и разработка Web приложений

и формирует основу для прохождения следующих курсов и практик:

- Современные сетевые технологии.

2. СТРУКТУРА ДИСЦИПЛИНЫ

<i>Характеристика учебной дисциплины</i>		
Направление подготовки	09.04.04 Программная инженерия	
Магистерская программа	Программная инженерия	
Программа подготовки	академическая магистратура	
Квалификация	магистр	
Количество содержательных модулей	2	
Дисциплина базовой / вариативной части образовательной программы	Вариативная часть, дисциплины по выбору	
Формы контроля	1 модульный контроль и зачёт в 3 семестре	
Показатели	очная форма обучения	заочная форма обучения
	на базе высшего профессионального образования (ОП бакалавриат)	
Количество зачётных единиц (кредитов)	6	
Год подготовки	2	
Семестр	3	
Количество часов	216	
- лекционных	18	
- практических, семинарских	18	
- лабораторных	36	
- самостоятельной работы	144	
в т.ч. индивидуальное задание	—	
Недельное количество часов,	12	
в т.ч. аудиторных	4	

3. ОПИСАНИЕ ДИСЦИПЛИНЫ

Цели и задачи

Цель: обучение студентов методологии проектирования серверных приложений, взаимодействующих с СУБД для WEB.

Задачи:

- расширение и углубление теоретических знаний в области разработки баз данных;
- овладение специальными языками программирования, обеспечивающими возможность разработки приложений баз данных для WEB;
- изучение наиболее типичных примеров создания динамических WEB-страниц, взаимодействующих с СУБД;
- формирование у студентов устойчивых навыков программирования на PHP;
- расширение знаний в области информационных технологий;
- выработка у обучающихся практических навыков по созданию приложений электронной коммерции.

Требования к результатам освоения дисциплины: Процесс изучения дисциплины «Разработка серверных Web приложений» направлен на формирование элементов следующих компетенций в соответствии с ФГОС ВО РФ направления подготовки 09.04.04 Программная инженерия и основной образовательной программы высшего профессионального образования направления подготовки 09.04.04 Программная инженерия (магистерская программа: Программная инженерия):

а) универсальных (УК):

- **УК 2** – способен управлять проектом на всех этапах его жизненного цикла;

б) общепрофессиональных (ОПК):

• **ОПК-2** – способен разрабатывать оригинальные алгоритмы и программные средства, в том числе с использованием современных интеллектуальных технологий, для решения профессиональных задач;

• **ОПК-5** – способен разрабатывать и модернизировать программное и аппаратное обеспечение информационных и автоматизированных систем;

• **ОПК-6** – способен самостоятельно приобретать с помощью информационных технологий и использовать в практической деятельности новые знания и умения, в том числе в новых областях знаний, непосредственно не связанных со сферой деятельности;

• **ОПК-7** – способен применять при решении профессиональных задач методы и средства получения, хранения, переработки и трансляции информации посредством современных компьютерных технологий, в том числе, в глобальных компьютерных сетях;

• **ОПК-8** – способен осуществлять эффективное управление разработкой программных средств и проектов;

в) профессиональных (ПК):

• **ПК-9** – способностью проектировать трансляторы и интерпретаторы языков программирования;

• **ПК-10** – способностью проектировать сетевые службы;

• **ПК-11** – способностью проектировать основные компоненты операционных систем;

• **ПК-12** – способностью проектировать вспомогательные и специализированные языки программирования и языки представления данных;

• **ПК-16** – владением навыками создания трансляторов и интерпретаторов языков программирования;

• **ПК-17** – владением навыками создания служб сетевых протоколов;

• **ПК-18** – владением навыками создания компонент операционных систем и систем реального времени;

В результате изучения учебной дисциплины студент должен

Знать:

- классификацию и архитектуру информационно-вычислительных сетей; уровни управления в модели систем;
- интерфейсы и протоколы физического уровня;
- организацию информационного обмена в сети Internet;
- понятие общего шлюзового интерфейса CGI;
- методы передачи данных между браузером и сценарием;

Уметь:

- пользоваться сетевыми интерфейсами и протоколами;
- управлять потоками в вычислительной сети;
- работать с HTTP и WWW;
- работать с сетевыми функциями;
- управлять интерпретатором.

Владеть:

- навыками управления в модели взаимодействия открытых систем;
- навыками синхронизации и асинхронизации передачи данных;
- принципами IP-маршрутизации;
- навыками передачи данных между браузером и сценарием;
- навыками работы с HTTP и WWW;
- навыками обработки ошибок и исключений;
- навыками работы с файлами.

4. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ И ФОРМЫ ОРГАНИЗАЦИИ УЧЕБНОГО ПРОЦЕССА

Порядковый номер и тема	Краткое содержание темы
Содержательный модуль 1	
Тема 1	Принципы управления вычислительных сетях.
Тема 2	Интерфейс CGI.
Тема 3	Основные элементы PHP.
Тема 4	Основные конструкции PHP.
Содержательный модуль 2	
Тема 5	Функции и область видимости.
Тема 6	Стандартные функции PHP. Работа с изображениями в PHP.
Тема 7	Принципы программирования на PHP
Тема 8	Объектно- ориентированное программирование на PHP.

Тематический план

Содержательный модуль 1												
Названия содержательных модулей и тем	Количество часов											
	Очная форма обучения						Заочная форма обучения					
	всего	в т.ч.					всего	в т.ч.				
		лекции	практические	лабораторные	самостоятельная работа	индивидуальная работа		лекции	практические	лабораторные	самостоятельная работа	индивидуальная работа
Тема 1. Принципы управления вычислительных сетях.	20	2	2	4	12							
Тема 2. Интерфейс CGI.	22	2	2	4	14							
Тема 3. Основные элементы РНР.	28	2	2	4	20							
Тема 4. Основные конструкции РНР.	32	3	3	6	20							
Итого по содержательному модулю 1	102	9	9	18	66							
Содержательный модуль 2												
Тема 5. Функции и область видимости.	26	2	2	4	18							
Тема 6. Стандартные функции РНР. Работа с изображениями в РНР.	29	2	2	5	20							
Тема 7. Принципы программирования на РНР	29	2	2	5	20							
Тема 8. Объектно-ориентированное программирование на РНР.	30	3	3	4	20							
Итого по содержательному модулю 2	114	9	9	18	78							
Всего часов по модулю	216	18	18	36	144							

5. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ДЛЯ ПРОВЕДЕНИЯ ЛЕКЦИОННЫХ, ПРАКТИЧЕСКИХ И ЛАБОРАТОРНЫХ ЗАНЯТИЙ

Темы лекционных занятий

<i>№ n/n</i>	<i>Название темы</i>	<i>Количество часов</i>
1	Принципы управления вычислительных сетях.	2
2	Интерфейс CGI.	2
3	Основные элементы PHP.	2
4	Основные конструкции PHP.	3
5	Функции и область видимости.	2
6	Стандартные функции PHP. Работа с изображениями в PHP.	2
7	Принципы программирования на PHP	2
8	Объектно- ориентированное программирование на PHP.	3
	ВСЕГО	18

Темы практических занятий

<i>№ n/n</i>	<i>Название темы</i>	<i>Количество часов</i>
1	Принципы управления вычислительных сетях.	2
2	Интерфейс CGI.	2
3	Основные элементы PHP.	2
4	Основные конструкции PHP.	3
5	Функции и область видимости.	2
6	Стандартные функции PHP. Работа с изображениями в PHP.	2
7	Принципы программирования на PHP	2
8	Объектно- ориентированное программирование на PHP.	3
	ВСЕГО	18

Темы лабораторных занятий

<i>№ n/n</i>	<i>Название темы</i>	<i>Количество часов</i>
1	Принципы управления вычислительных сетях. Установка и работа с VirtualBox. Установка Ubuntu Server 18.04.	4
2	Установка веб-сервера Apache, MySQL, PHP и phpMyAdmin.	4
3	Серверные приложения. Язык PHP. Библиотечные функции PHP	4
4	Основные конструкции PHP.	6
5	Взаимодействие с пользователем. HTML-формы и элементы управления.	4
6	Базы данных. Разработка веб-интерфейса к базе данных. Совместное использование PHP и MySQL.	5
7	Работа с сессиями. Организация работы с учётными записями.	5
8	Обработка данных ввода пользователя сервером. Обработка файлов. Управление веб-сайтом. Файл htaccess.	4
	ВСЕГО	36

6. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ОРГАНИЗАЦИИ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

Организация самостоятельной работы студентов

№ n/n	Название темы	Количество часов
1	Принципы управления вычислительных сетях.	12
2	Интерфейс CGI.	14
3	Основные элементы PHP.	20
4	Основные конструкции PHP.	20
5	Функции и область видимости.	18
6	Стандартные функции PHP. Работа с изображениями в PHP.	20
7	Принципы программирования на PHP	20
8	Объектно- ориентированное программирование на PHP.	20
	ВСЕГО	144

7. ИНДИВИДУАЛЬНЫЕ ЗАДАНИЯ

Индивидуальная работа № 1 БАЗЫ ДАННЫХ

Цель: В каждом варианте надо написать скрипт для создания таблиц базы, наполнения базы данными и удаления таблиц базы. Заодно нужно написать страницу согласно заданию. Каждая таблица в базе должна содержать нормальный первичный ключ, который генерируется автоматически. Если таблица ссылается на другую, следует прописать нормальный внешний ключ.

Задания:

Вариант №1

1. Необходимо создать две таблицы. Первая хранит в себе сведения о фирмах, а именно название фирмы (до 250 символов). Вторая – сведения об автомобилях, производимых фирмами: название автомобиля (до 150 символов), его стоимость (целое число) и ссылка на фирму. Далее вывести сведения о фирмах и автомобилях в виде двухуровневого списка. Первый уровень содержит названия фирм в алфавитном порядке. На втором расположены сведения об автомобилях для данной фирмы, отсортированные по названиям автомобилей в алфавитном порядке. Все данные страница должна извлечь из базы одним запросом, т. е. нельзя сделать запрос чисто для фирм, а потом для каждой фирмы извлекать дополнительным запросом сведения об автомобилях. Выглядеть страница должна так:

```

1. Ford.
  1. Name - extended. Cost - 2500.
  2. Name - T-model. Cost - 2700.
2. Mercedes.
  1. Name - advanced. Cost - 2800.
  2. Name - Basic. Cost - 2900.

```


Содержание отчёта

1. Цель работы.
2. Содержание варианта задания.
3. Скрипты для базы.
4. Код программы на PHP.
5. Скриншоты результатов выполнения страниц.
6. Выводы.

Индивидуальная работа № 2 РАЗРАБОТКА ВЕБ-ИНТЕРФЕЙСА К БАЗЕ ДАННЫХ. СОВМЕСТНОЕ ИСПОЛЬЗОВАНИЕ PHP И MYSQL

Цель: В ходе каждого задания нужно будет выполнить следующее.

1. Написать скрипт создания необходимых таблиц и взаимосвязей. Рабочее название скрипта – `schema.sql`. Первичные ключи должны быть автоинкрементными, т. е. при вставке новой записи в базу первичный ключ не обязательно должен указываться.
2. Написать скрипт заполнения таблиц тестовыми данными. Рабочее название скрипта – `data.sql`. Скрипт строится на предположении, что схема уже создана. При заполнении данными можно указывать первичные ключи явно (чтобы правильно потом задать взаимосвязи с дочерними таблицами), однако схема должна поддерживать возможность добавления записей с автоматической генерацией ключа.
3. Написать скрипт очистки таблиц от текстовых данных. Рабочее название скрипта – `flush.sql`. Скрипт должен соблюдать порядок удаления данных, т. е. если таблица *A* ссылается на таблицу *B*, то сначала данные должны быть удалены из таблицы *A* и только затем из таблицы *B*, поскольку иначе может возникнуть нарушение целостности (т. е. этот запрос сгенерирует исключение).
4. Написать скрипт удаления созданных таблиц из базы. Рабочее название скрипта – `drop.sql`. Скрипт должен соблюдать порядок удаления сущностей, т. е. если таблица *A* ссылается на таблицу *B*, то сначала надо удалить таблицу *A* и только затем таблицу *B* (причины те же, что и для предыдущего пункта).
5. Написать страницу `display.php`, которая будет подключаться к базе (если доступа нет, то страница должна прекратить генерацию), извлекать из нее данные и отображать их в требуемом формате.

Задания:

Вариант №1

В базе должны быть производители и автомобили. У каждого производителя есть название, у каждого автомобиля – название, производитель и цвет (цвета должны содержаться в отдельной таблице и на них должна быть ссылка). Сведения нужно отображать в виде двухуровневого списка. На первом уровне – производители (отсортированные в алфавитном порядке), на втором – автомобили для данного производителя (также отсортированные в алфавитном порядке). На втором уровне данные нужно представлять в виде строки "[Название марки авто]. Цвет – [название цвета]".

Содержание отчёта

1. Цель работы.
2. Вариант задания.
3. Исходный текст скриптов для работы с базой данных.
4. Исходный текст программы на PHP.
5. Скриншот сгенерированной страницы.
6. Выводы.

Индивидуальная работа № 3 РАБОТА С СЕССИЯМИ

Цель: В каждом варианте заданий необходимо написать скрипт для создания таблиц и скрипт для заполнения этих таблиц данными и реализовать страницы, поддерживающие указанный контракт.

Задания:

Вариант №1

Необходимо реализовать в базе две таблицы. Первая хранит в себе данные о пользователях – логин пользователя и пароль (для хранения пароля использовать MD5 без соли), вторая – сведения о страницах, посещённых данным пользователем: внешний ключ со ссылкой на пользователя, название страницы, которую он посетил, а также время посещения.

Необходимо реализовать три страницы. Первая содержит форму ввода логина и пароля, а также кнопку отправки. При попытке отправки данных пользователя переводят на эту же страницу, данные сопоставляются с базой, после чего следует начать сессию для этого пользователя (если данные верны).

Вторая страница проверяет содержимое сессии. Если пользователь еще не авторизовался, она сразу автоматически переводит его на страницу авторизации. Если пользователь авторизовался, то страница на основе данных сессии обращается к базе и обновляет сведения о том, сколько раз данный пользователь посещал эту страницу. После чего она выдает пользователю сообщение вида "Вы посетили данную страницу *N* раз".

Третья страница содержит статистические сведения в виде маркированного списка и сведения о последних 10 заходах на предыдущую страницу пользователей вообще. Каждый элемент маркированного списка отражает сведения вида "Пользователь *username* заходил на страницу *pagename* в последний раз *date*".

Содержание отчёта

1. Цель работы.
2. Вариант задания.
3. Исходный текст программы на PHP.
4. Скриншоты сгенерированных страниц.
5. Выводы.

Индивидуальная работа № 4 Работа с сессиями

Цель: Задания для данной лабораторной работы являются логическим развитием предыдущей лабораторной. Страницы добавления/редактирования сущностей должны позволять редактировать все поля записи (за исключением первичного ключа). Ссылки (внешние ключи) представляются с помощью выпадающих списков, все остальное – с помощью текстовых полей. Страница редактирования должна хранить в себе первичный ключ сущности (в виде скрытого поля). Страницы добавления/редактирования должны производить валидацию и в случае ошибки оставлять пользователя на той же странице. Каждый табличный список содержит в качестве первого столбца "№ п/п".

Задания:

Вариант №1

Страница просмотра списка фирм отображает в виде таблицы со столбцами "Название", "Просмотр". В первом – название фирмы. Во втором – ссылка, которая ведет на страницу просмотра сведений о фирме (описана детально ниже). Внизу страницы просмотра списка фирм имеется кнопка "Добавить", при нажатии на которую пользователь переходит на страницу добавления сведений о фирмах.

Страница просмотра отдельной фирмы отображает следующие сведения: название фирмы, маркированный список автомобилей, где каждый элемент списка – это ссылка с названием автомобиля. Нажав на эту ссылку, пользователь попадает на страницу просмотра сведений об автомобиле. Внизу страницы есть кнопки "Редактировать" и "Удалить" (эта кнопка отключена, если у фирмы есть автомобили).

При нажатии на кнопку "Редактировать" пользователь попадает на страницу редактирования сведений о фирме, которая описана ниже, при нажатии на кнопку "Удалить" – на страницу удаления записи о фирме, описанной ниже. Страница просмотра ожидает в качестве параметра GET-запроса параметр **id**, который хранит идентификатор фирмы. Если идентификатор не задан или такой фирмы в базе нет, то страница должна остановить выполнение и выдать сообщение об ошибке. Внизу страницы имеется ссылка "Назад", ведущая на страницу просмотра списка фирм.

Страница просмотра сведений об автомобиле отображает следующие сведения: название фирмы, название автомобиля и цвет. Никаких опций добавления/редактирования/удаления не нужно. Внизу страницы имеется ссылка "Назад к фирме", которая будет переводить пользователя назад на страницу просмотра сведений о фирме производителе данного автомобиля. Страница просмотра сведений ожидает в качестве параметра GET-запроса параметр **id**, который хранит идентификатор автомобиля. Если идентификатор не задан или такого автомобиля в базе нет, то страница должна остановить выполнение и выдать сообщение об ошибке.

Страница добавления новой фирмы содержит заголовок "Добавление новой фирмы", одно текстовое поле, в которое пользователь введет название. Внизу страницы имеются две кнопки "Добавить" и "Отмена", нажатие на них возвращает назад на эту же страницу, но обрабатывать эти события страница должна по-разному. Если нажата кнопка "Отмена", то происходит принудительный возврат пользователя на общую страницу просмотра списка фирм. Если нажата кнопка "Добавить", то производится валидация. Если она пройдена успешно, то фирма добавляется в базу и осуществляется принудительный переход на страницу просмотра сведений для этой фирмы. Валидация не проходит в следующих случаях: название фирмы пустое, название фирмы превышает допустимую длину текстового поля в базе, фирма с таким названием уже есть. Если валидация не проходит, то пользователь остается на странице добавления и под текстовым полем ему выводится красным цветом сообщение об ошибке – что именно неправильно.

Страница редактирования сведений о фирме выглядит практически так же, как предыдущая. Отличия следующие: заголовок "Редактирование сведений о фирме [Название фирмы]", текстовое поле будет изначально не пустым (в нем название фирмы, извлеченное из базы). При нажатии на кнопку "Отмена" пользователь возвращается на страницу просмотра сведений о фирме. Вместо кнопки "Добавить" будет кнопка "Сохранить". Страница ожидает в качестве параметра POST-запроса параметр **id**, который хранит идентификатор фирмы. Если идентификатор не задан или такой фирмы в базе нет, то страница останавливает выполнение и выдает сообщение об ошибке. Этот параметр должен сохраняться в форме в виде скрытого поля, чтобы при отправке данных из формы это поле снова присутствовало в POST-запросе.

Страница удаления фирмы ожидает в качестве параметра POST-запроса параметр **id**, который хранит идентификатор фирмы. Если идентификатор не задан или такой фирмы в базе нет или у этой фирмы есть автомобили, то страница останавливает выполнение и выдает сообщение об ошибке. В противном случае страница удаляет записи из базы и выдает сообщение "Фирма [Название фирмы] удалена успешно". Внизу страницы имеется ссылка "Назад", ведущая на страницу просмотра списка фирм.

Содержание отчёта

6. Цель работы.
7. Вариант задания.
8. Исходный текст программы на PHP.
9. Скриншоты сгенерированных страниц.
10. Выводы.

8. КОНТРОЛЬНЫЕ ВОПРОСЫ К ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

1. Общее устройство сети интернет.
2. Понятие домена и управление доменами.
3. Протоколы интернет.
4. Выбор технологий web-разработки.
5. Web-приложения и их разновидности.
6. Необходимость программирования сервера.
7. Логика действия PHP.
8. Установка и настройка PHP.
9. Синтаксис «встраивания» PHP.
10. Выражения и операции в PHP.
11. Типы данных в PHP.
12. Функции в PHP.
13. Сессии в PHP.
14. Передача и приём параметров в скрипт PHP.
15. Обработка форм с помощью PHP.
16. Структура web-приложения.
17. Авторизация пользователей в web-приложениях.
18. Обмен информацией между модулями в web-приложении.
19. Использование внешних данных в web-приложении.
20. Понятие и назначение языка SQL.
21. Установка MySQL и доступ к базам данных.
22. Использование MySQL в веб-приложении на PHP.
23. Основные виды запросов в MySQL.
24. Общая методика разработки web-сайта.
25. Методика развёртывания web-сайта.
26. Проектная документация при web-разработке.

8. ОБРАЗЕЦ МОДУЛЬНОГО КОНТРОЛЯ

ГОУ ВПО «ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ»

Факультет математики и информационных технологий

<i>Направление подготовки:</i>	09.04.04 Программная инженерия
<i>Магистерская программа:</i>	Программная инженерия
<i>Программа подготовки:</i>	академическая магистратура
<i>Семестр</i>	3
<i>Учебная дисциплина</i>	Разработка серверных Web приложений

МОДУЛЬНАЯ КОНТРОЛЬНАЯ РАБОТА

ВАРИАНТ №1

1. Какие парадигмы программирования поддерживает PHP?
2. В чем состоят функции сервера?
3. Как можно получить и вывести на экран список всех методов класса, представителем которого является объект?

Утверждено на заседании кафедры прикладной механики и компьютерных технологий, протокол № ____ от « ____ » _____ 20 ____ г.

Заведующий кафедрой
Преподаватель

А.С.Гольцев
В.А.Цванг

Критерии оценивания модульного контроля

<i>Номер задания</i>	<i>Количество баллов</i>
1	10
2	10
3	10
<i>Всего</i>	<i>30</i>

10. ОБРАЗЕЦ ТЕСТОВОГО ЗАДАНИЯ

Какие из операторов switch записаны правильно с точки зрения синтаксиса?		
а) switch (\$par){ "1": echo "1"; "2": echo 2; } 	б) switch (\$par){ case "1": echo "1"; case 2: echo 2; default: echo 3; } 	в) switch (\$par): case "1": echo "1"; break; case 2: echo 2; break; endswitch;

11. КРИТЕРИИ ОЦЕНИВАНИЯ

По курсу предполагается проведение промежуточной аттестации в виде модульного контроля, выполнение индивидуальных заданий и зачёта.

Распределение баллов, которые могут получить студенты в процессе изучения дисциплины

Организационно учебная работа студента	СРС			Всего
	Индивидуальная работа	Модульный контроль	Индивидуальная творческая работа	
Max 30 баллов	max 30 баллов	max 30 баллов	max 10 баллов	100 баллов
Активность на практических и лабораторных занятиях	Выполнение индивидуальных заданий	Выполнение модульной контрольной работы	Разработка доклада на студенческую научную конференцию	

Шкала соответствия баллов национальной шкале

Оценка по шкале ECTS	Оценка по 100-балльной шкале	Оценка по государственной шкале (экзамен, дифференцированный зачет)	Оценка по государственной шкале (зачет)
A	90-100	5 (отлично)	зачтено
B	80-89	4 (хорошо)	зачтено
C	75-79	4 (хорошо)	зачтено
D	70-74	3 (удовлетворительно)	зачтено
E	60-69	3 (удовлетворительно)	зачтено

FX	35-59	2 (неудовлетворительно) с возможностью повторной сдачи	не зачтено
F	0-34	2 (неудовлетворительно) с возможностью повторной сдачи при условии обязательного набора дополнительных баллов	не зачтено

12. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ УЧЕБНОГО ПРОЦЕССА

Лекционные, практические и лабораторные занятия проводятся в компьютерном классе, оборудованном компьютерами с лицензионным программным обеспечением, доступом к сети Интернет, столами, доской.

13. РЕКОМЕНДОВАННАЯ ЛИТЕРАТУРА

№ п/п	Наименование	Кол-во экземпляров в библиотеке ДонНУ	Наличие электронной версии в ЭБС
Основная литература			
1.	Проектирование и разработка Web приложений: учебное пособие / Сост.: В.А. Цванг. – Донецк: ДонНУ, 2019. – 119 с.	-	+
2.	Практикум по разработке приложений на языке JavaScript: учебно-методическое пособие / Сост.: В.А. Цванг. – Донецк: ДонНУ, 2019. – 185 с.	-	+
3.	Практикум по разработке серверных приложений: учебно-методическое пособие / Сост.: В.А. Цванг. – Донецк: ДонНУ, 2019. – 181 с.	-	+
4.	Прохоренок, Н. А. HTML, JavaScript, PHP и MySQL : джентльменский набор Web-мастера / Николай Прохоренок. - 3-е изд. - Санкт-Петербург: БХВ-Петербург, 2010. - 890 с. + электрон. опт. диск (CD-ROM).	1	-
Дополнительная литература			
5.	Прийменко С. А. Компьютерные сети: учеб. пособие / С. А. Прийменко, Р. Н. Нескородев, Я. А. Арчаков. - Донецк: ДонНУ, 2013. - 97 с. 2004. - 592 с.	12	-
6.	Колисниченко, Д. Н. PHP 5/6 и MySQL 6 : разработка Web-приложений / Д. Н. Колисниченко. - 2-е изд. - Санкт-Петербург: БХВ-Петербург, 2010. - 540 с. + 1 электрон. опт. диск (CD-ROM).	3	-
7.	Кузнецов, М. В. PHP 5. Практика создания Web-сайтов : ["хитрости" PHP, система администрирования контента сайта, разработ. динамичес. WEB-приложений, работа с графикой, FLASH, PDF-док., работа с базами	3	-

	данных] / Максим Кузнецов, Игорь Симдянов, Сергей Голышев. - СПб.: БХВ-Петербург, 2007. - XII, 948 с.		
--	---	--	--

14. ИНФОРМАЦИОННЫЕ РЕСУРСЫ

1. Htmlbook.ru. Самоучитель HTML. Справочник по HTML. XHTML. HTML5. [Электронный ресурс]. – URL:<http://htmlbook.ru/>.
2. Wisdomweb.ru – учебники для веб-разработчиков. [Электронный ресурс] – URL: <http://www.wisdomweb.ru/>.
3. Javascript.ru/ Здесь живет профессиональный javascript. [Электронный ресурс] – URL: <http://learn.javascript.ru/>.
4. <https://phpclub.ru/>. PHPClub - клуб разработчиков PHP. [Электронный ресурс] – URL: <https://phpclub.ru/>.

15. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

Редакторы Sublime Text 3, Visual Studio Code, браузер Google Chrome, серверная платформа Open Server, операционная система Ubuntu Server.

Рабочая программа рассмотрена и переутверждена на заседании кафедры прикладной механики и компьютерных технологий с изменениями (без изменений) на 20____ год.
 Протокол № ____ от “ ____ ” _____ 20__ г.
 Заведующий кафедрой _____

Рабочая программа рассмотрена и переутверждена на заседании кафедры прикладной механики и компьютерных технологий с изменениями (без изменений) на 20____ год.
 Протокол № ____ от “ ____ ” _____ 20__ г.
 Заведующий кафедрой _____

Рабочая программа рассмотрена и переутверждена на заседании кафедры прикладной механики и компьютерных технологий с изменениями (без изменений) на 20____ год.
 Протокол № ____ от “ ____ ” _____ 20__ г.
 Заведующий кафедрой _____

Рабочая программа рассмотрена и переутверждена на заседании кафедры прикладной механики и компьютерных технологий с изменениями (без изменений) на 20____ год.
 Протокол № ____ от “ ____ ” _____ 20__ г.
 Заведующий кафедрой _____

Рабочая программа рассмотрена и переутверждена на заседании кафедры прикладной механики и компьютерных технологий с изменениями (без изменений) на 20____ год.
 Протокол № ____ от “ ____ ” _____ 20__ г.
 Заведующий кафедрой _____