

Государственное областное бюджетное
профессиональное образовательное учреждение
«Усманский многопрофильный колледж»

**МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ОРГАНИЗАЦИИ И
ПРОВЕДЕНИЮ ПРАКТИЧЕСКИХ РАБОТ**

по учебной дисциплине ОП.06. Основы алгоритмизации и программирования

Программы подготовки специалистов среднего звена (ППССЗ)

по специальности 09.02.04 Информационные системы (по отраслям)

по программе базовой подготовки

Усмань 2018

Методические рекомендации по организации и проведению практических работ по учебной дисциплине ОП.06.Основы алгоритмизации и программирования по специальности 09.02.04 Информационные системы (по отраслям).

Разработчики:

Мотин И.А. преподаватель информатики

Рассмотрены и утверждены на заседании предметно-цикловой комиссии естественнонаучных дисциплин

Протокол № 6 от 29.06.2018 г.

Председатель предметно-цикловой комиссии естественнонаучных дисциплин _____ Коровина Т.В.



УТВЕРЖДАЮ

Заместитель директора Думма Т.А.

по учебно-методической работе



Введение

Практические занятия, как вид учебных занятий, направлены на экспериментальное подтверждение теоретических положений и формирование учебных и профессиональных практических умений и составляют важную часть теоретической и профессиональной практической подготовки.

В процессе практического занятия обучающиеся выполняют одно или несколько практических заданий в соответствии с изучаемым содержанием учебного материала.

Содержание практических занятий по учебной дисциплине ОП.06. Основы алгоритмизации и программирования должно охватывать весь круг профессиональных умений, на подготовку к которым ориентирована данная дисциплина, а в совокупности охватывать всю профессиональную деятельность, к которой готовится специалист.

При разработке содержания практических занятий следует учитывать, что наряду с формированием умений и навыков в процессе практических занятий обобщаются, систематизируются, углубляются и конкретизируются теоретические знания, вырабатывается способность и готовность использовать теоретические знания на практике, развиваются интеллектуальные умения.

Выполнение обучающимися практических занятий проводится с целью:

- формирования практических умений в соответствии с требованиями к уровню подготовки обучающихся, установленными ФГОС и рабочей программой учебной дисциплины ОП.06. Основы алгоритмизации и программирования по конкретным разделам и темам дисциплины;
- обобщения, систематизации, углубления, закрепления полученных теоретических знаний;
- совершенствования умений применять полученные знания на практике, реализации единства интеллектуальной и практической деятельности;
- развития интеллектуальных умений у будущих специалистов:

аналитических, проектировочных, конструктивных и др.;

- выработки таких профессионально значимых качеств, как самостоятельность, ответственность, точность, творческая инициатива при решении поставленных задач при освоении общих и профессиональных компетенций.

Соответственно в процессе освоения учебной дисциплины ОП.06. Основы алгоритмизации и программирования обучающиеся должны овладеть:

умениями:

- использовать языки программирования;
- строить логически правильные и эффективные программы.

знаниями:

- общие принципы построения алгоритмов;
- основные алгоритмические конструкции;
- понятие системы программирования;
- основные элементы процедурного языка программирования;
- структуру программы, операции, управляющие структуры;
- структуры данных, файлы, кассы памяти;
- подпрограммы, составление библиотек программ;
- объектно-ориентированную модель программирования;
- понятие классов и объектов, их свойств и методов.

Вышеперечисленные умения и знания направлены на формирование следующих профессиональных и общих компетенций студентов:

Профессиональные компетенции:

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 2.2. Программировать в соответствии с требованиями технического задания.

ПК 2.3. Применять методики тестирования разрабатываемых приложений.

Общие компетенции:

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.

Данные методические рекомендации по организации и проведению практических работ составлены в соответствии с содержанием рабочей программы учебной дисциплины ОП.06. Основы алгоритмизации и программирования специальности 09.02.04 Информационные системы (по отраслям) по программе базовой подготовки.

Учебная дисциплина ОП.06. Основы алгоритмизации и программирования изучается в течение учебного семестра. Общий объем времени, отведенный на выполнение практической работы по учебной дисциплине ОП.06. Основы

алгоритмизации и программирования, составляет в соответствии с учебным планом и рабочей программой– 30 часов.

Методические рекомендации призваны помочь студентам правильно организовать работу и рационально использовать свое время при овладении содержанием учебной дисциплины ОП.06. Основы алгоритмизации и программирования, закреплении теоретических знаний и практических умений.

Распределение часов на выполнение практической работы студентов по разделам и темам учебной дисциплины ОП.06. Основы алгоритмизации и программирования

Наименование раздела, темы	Количество часов на ПР
Раздел 2. Основные инструменты и подходы для создания web приложений на языке PHP	30
Тема 2.1. Работа сайта и сервера. Основы PHP.	4
Тема 2.2. Операторы. Циклы. Функции.	10
Тема 2.3. Встроенные возможности. Формы.	4
Тема 2.4. Базы данных	4
Тема 2.5. ООП	8

Перечень рекомендуемой литературы

Основные источники:

1. Лисьев Г.А, Романов П.Ю, АскеркоЮ.И. Программное обеспечение компьютерных сетей и web-серверов: учебное пособие - М: ИНФРА-М:2020.
2. Федорова Г. Н. Разработка, внедрение и адаптация программного обеспечения отраслевой направленности: учебное пособие. - М.: КУРС, 2020.
3. Ночка Е.И. Основы алгоритмизации и программирования. - М.: КУРС, 2017.

Дополнительные источники:

4. Виктор Зинченко. PHP-START - [Электронный ресурс] // URL: <https://php-start.com/course/php-start-theory>

Раздел 2. Основные инструменты и подходы для создания web приложений на языке PHP

Тема 2.1. Работа сайта и сервера. Основы PHP. (4 часа)

Практическая работа №1 «Основы»

Задачи обучающегося:

1. Научиться оставлять комментарии в коде.
2. Научиться именовать переменные
3. Изучить использование операторов

Опорные понятия: Основы PHP.

Планируемый результат:

Студент должен

Уметь оставлять комментарии в коде.

Знать назначение операторов.

Знать как использовать константы

Необходимое оборудование: ПК, IDE NetBeans, OpenServer

Алгоритм деятельности обучающегося:

Задача 1: Комментарии

Создать файл php. В начале файла добавить 2 комментария (однострочный и многострочный) с именем автора и датой выполнения. Напечатать приветствие при помощи echo.

Задача 2: Именованние переменных

- Придумайте и запишите имена переменных для таких сущностей: название телеканала, адрес производителя, цвет автомобиля, температура воды, модель телефона.
- Следуйте правилам именования переменных. Используйте lowerCamelCase для имен, составленных из двух и более слов.

Задача 3: Операторы

- Создайте 3 переменных с произвольными именами на свой выбор. Присвойте им значения 3, 5, 8 соответственно. Выведите значения этих переменных на страницу.
- Найдите сумму этих переменных и запишите ее в новую переменную. Выведите значение новой переменной.
- Найдите значение такого выражения: $2+6+2/5-1$. Запишите его в переменную с именем \$result и выведите на страницу.

Задача 4: Копирование значений переменных и копирование переменных по ссылке

Будем использовать переменные \$a, \$b, \$c, \$d.

- Задайте значения 1 и 2 переменным \$a, \$b соответственно.
- Выведите значения переменных \$a, \$b.
- Скопируйте значение переменной \$a в переменную \$c.
- Скопируйте переменную \$b в переменную \$d по ссылке.

- Распечатайте значения переменных \$c, \$d.
- Присвойте переменным \$a, \$b значения 3 и 4 соответственно.
- Распечатайте переменные \$a, \$b, \$c, \$d.
- Сделайте выводы относительно полученных результатов.

Задача 5: Константы

- Создайте 2 константы с произвольными именами со значениями 41 и 33.
- Найдите и выведите сумму этих констант.
- Попробуйте переопределить одну из констант. Внимательно прочтите подсказку об ошибке в Вашей IDE (если поддерживается)

Все задачи необходимо сделать в отдельных файлах (например: task1.php)

Оформить отчет с листингом кода

Раздел 2. Основные инструменты и подходы для создания web приложений на языке PHP

Тема 2.1. Работа сайта и сервера. Основы PHP. (4 часа)

Практическая работа №2

«Типы данных»

Задачи обучающегося:

1. Научиться определять типы переменных.
2. Научиться работать с переменными в строках
3. Научиться создавать простые массивы

Опорные понятия: Типы данных.

Планируемый результат:

Студент должен

Уметь выводить значения переменных

Знать как работать с массивами.

Необходимое оборудование: ПК, IDE NetBeans, OpenServer

Алгоритм деятельности обучающегося:

Задача 1: Типы переменных

Определите типы следующих переменных и выведите их на экран:

```
$a = 152;
$b = '152';
$c = 'London';
$d = array(152);
$e = 15.2;
$f = false;
$g = true;
```

Задача 2: Работа со строками и переменными

Написать код, который будет формировать строку и выводить ее на экран, подставляя в нее значения переменных \$a и \$b.

Например:

Дано: a = 10, b = 5.

Результат: 5 из 10ти студентов посетили лекцию.

Условия

Строки в PHP можно формировать используя одинарные ' и двойные " кавычки. Решите эту задачу двумя способами с применением одинарных и двойных кавычек.

Подсказка:

В случае использования двойных кавычек будет полезной возможность экранирования переменных при помощи { }.

Задача 3: Работа со строками и переменными

Определить три переменных со значениями:

“Доброе утро”

“дамы”

“и господа”

- Вывести значения переменных в браузер.
- Сформировать строку "Доброе утро, дамы и господа" используя созданные переменные и комбинированный оператор склеивания.

Задача 4: Работа с массивами

- Создать 2 простых массива с количеством элементов 5.
- В первый массив добавить один элемент с индексом (!) element и произвольным значением.
- Из второго массива удалить элемент с индексом 0. Используйте функцию unset();
- Вывести на экран элементы под индексом 2 из первого и второго массива.
- Вывести на экран содержимое массивов полностью.
- Найти количество элементов в каждом массиве. Используйте функцию count(). Вывести результаты на экран.

Все задачи необходимо сделать в отдельных файлах (например: task1.php)

Оформить отчет с листингом кода

Раздел 2. Основные инструменты и подходы для создания web приложений на языке PHP

Тема 2.2. Операторы. Циклы. Функции. (6 часов)

Практическая работа №3 «Условный оператор»

Задачи обучающегося:

1. Изучить работу условного оператора

Опорные понятия: Условный оператор.

Планируемый результат:

Студент должен

Уметь составлять программы для решения уравнений

Уметь составлять программы которая проверяет вхождение переменной в определенный диапазон

Необходимое оборудование: ПК, IDE NetBeans, OpenServer

Алгоритм деятельности обучающегося:

Задача 1: Вхождение числа в диапазон

Диапазон значений, определяется константами $MIN = 10$, $MAX = 50$.

Написать программу, которая проверяет вхождение переменной x в диапазон (больше MIN , меньше MAX).

Если x входит в диапазон, вывести сообщение "+", иначе вывести сообщение "-".

Если x находится на границе диапазона ($x = 10$ или $x = 50$), вывести сообщение "+-".

Подсказка:

Подставляйте произвольные значения переменной x для проверки работы программы.

Задача 2: Квадратное уравнение

Решить квадратное уравнение.

Определить значения переменных a , b , c произвольными числовыми значениями (с ними и работать дальше).

По формуле найти дискриминант (D).

Используя условный оператор (if, elseif, else) сравнить значение D с 0:

- Если $D > 0$, вычислить и напечатать 2 корня уравнения (x_1 , x_2)
- Если $D = 0$, найти 1 корень (x)
- Если $D < 0$, вывести сообщение: "Нет корней"

Подсказки:

1) Вспомнить теорию и проверить работу своей программы можно на сайте <https://100formul.ru/kvadrat.php>

2) Арифметические операторы мы уже знаем. Для нахождения квадратного корня используется

функция `sqrt()`:

```
echosqrt(9); // Выведет "3"
```

Задача 3: Равенство чисел

Среди трех произвольно выбранных чисел найти среднее. Если среди чисел есть равные, вывести сообщение "Ошибка".

Например:

Дано: $a = 2$, $b = 4$, $c = 3$;

Результат: "Среднее число: $c = 3$ "

Все задачи необходимо сделать в отдельных файлах (например: task1.php)

Оформить отчет с листингом кода

Раздел 2. Основные инструменты и подходы для создания web приложений на языке PHP

Тема 2.2. Операторы. Циклы. Функции. (6 часов)

Практическая работа №4

«Циклы»

Задачи обучающегося:

1. Изучить работу циклов for, while, foreach

Опорные понятия: Циклы.

Планируемый результат:

Студент должен

Уметь решать задачи применяя разные типы циклов

Уметь формировать меню для сайта на основе циклов

Необходимое оборудование: ПК, IDE NetBeans, OpenServer

Алгоритм деятельности обучающегося:

Задача 1: Сумма чисел

Найти и напечатать сумму чисел от 1 до 25.

Условие:

Решить задачу двумя способами: используя цикл for и цикл while.

Задача 2: Квадраты чисел

Найти и вывести все квадраты натуральных чисел (1,2,3, ...), не превосходящие данного числа N.

Число N задаем самостоятельно.

Задача 3: Меню на сайте

Цель задачи - сформировать html-код списка для отображению меню на сайте.

Сформируйте массив (10 элементов) со строками вида:

Кнопка 10

Кнопка 9

Кнопка 8

...

Кнопка 1

Отсортируйте массив в обратном порядке любым способом. Если возникают сложности,

Получить такой результат:

Кнопка 1

Кнопка 2

...

Кнопка 10

При помощи echo, операторов склеивания и цикла foreach получить и вывести на экран такой html-код:

```
<ul>
<li><a href="#">Кнопка 1</a></li>
<li><a href="#">Кнопка 2</a></li>
<li><a href="#">Кнопка 3</a></li>
...
<li><a href="#">Кнопка 10</a></li>
</ul>
```

Все задачи необходимо сделать в отдельных файлах (например: task1.php)

Оформить отчет с листингом кода

Раздел 2. Основные инструменты и подходы для создания web приложений на языке PHP

Тема 2.2. Операторы. Циклы. Функции. (6 часов)

**Практическая работа №5
«Пользовательские функции»**

Задачи обучающегося:

1. Научиться писать пользовательские функции исходя из поставленной задачи

Опорные понятия: Пользовательские функции.

Планируемый результат:

Студент должен

Уметь написать функцию под необходимую задачу

Необходимое оборудование: ПК, IDE NetBeans, OpenServer

Алгоритм деятельности обучающегося:

Задача 1: Информация о товарах в корзине

Написать функцию для получения итогов (общей информации) о корзине покупок.

Функция принимает массив с информацией о выбранных товарах примерно такого вида:

```
$products = array(
    array('name' => 'Телевизор', 'price' => '400', 'quantity' => 1),
    array('name' => 'Телефон', 'price' => '300', 'quantity' => 3),
    array('name' => 'Кроссовки', 'price' => '150', 'quantity' => 2),
);
```

Возвращает массив, который содержит:

- Общую сумму покупок
- Общее количество выбранных товаров

Задача 2: Квадратное уравнение

Написать функцию, которая решает квадратное уравнение. Функция принимает 3 аргумента (коэффициенты).

Возвращает:

- Массив с двумя корнями x_1 , x_2 , если $D > 0$
- Один корень x , если $D = 0$
- false, если $D < 0$

Подсказки:

Частично используйте код из прошлой практической работы.

Задача 3: Удаление отрицательных элементов из массива (вариант 1)

Есть массив с элементами (отрицательными и положительными). Нужно написать такую функцию `deleteNegtives()`, которая будет принимать массив, удалять из него элементы меньше 0 и возвращать этот массив.

Подсказки:

Можно использовать цикл `foreach` для обхода элементов массива.

Пример:

```
// Сейчас $digits содержит отрицательные и положительные числа
$digits = array(2,-10, -2, 4, 5, 1, 6, 200, 1.6, 95);
$digits = deleteNegtives($digits);
// Теперь $digits содержит только положительные числа
```

Задача 4: Удаление отрицательных элементов из массива (вариант 2)

Решить задачу №3 используя передачу аргумента по ссылке.

Пример:

```
$digits = array(2,-10, -2, 4, 5, 1, 6, 200, 1.6, 95);
// Сейчас $digits содержит отрицательные и положительные числа
deleteNegtives($digits);
// Теперь $digits содержит только положительные числа
```

*Все задачи необходимо сделать в отдельных файлах (например: task1.php)
Оформить отчет с листингом кода*

Раздел 2. Основные инструменты и подходы для создания web приложений на языке PHP

Тема 2.3. Встроенные возможности. Формы. (4 часа)

Практическая работа №6 «HTTP. Формы»

Задачи обучающегося:

1. Научиться создавать страницы с формами для ввода данных

Опорные понятия: пользовательские формы.

Планируемый результат:

Студент должен

Уметь создавать формы различных типов

Необходимое оборудование: ПК, IDE NetBeans, OpenServer

Алгоритм деятельности обучающегося:

Задание 1: Request&Response

Посетить свои любимые сайты, посмотреть на запросы браузера (обратить внимание на использование методов GET и POST, отправляемые заголовки) и ответы сервера (статус ответа, заголовки, тело сообщения).

Подсказка: Используйте инструменты для разработчиков Вашего браузера.

Задача 1: Использование форм

Создать страницу с формой на 7 полей для текстового ввода (input, type=text) и кнопкой отправки (input, type=submit).

Данные, отправленные формой приводятся к типу integer: используем функцию intval().

Далее из 7-ми введенных в форму значений вычисляются:

- максимальное значение
- минимальное значение
- среднее арифметическое значение.

Результаты выводятся после отправки формы на этой же странице.

Подсказки:

Данные можно записать в массив:

```
$a = intval($_POST['a']);  
...  
$arr = array($a, $b, ...);
```

И далее применить функции `max()`, `min()` для нахождения максимального и минимального значений.

Задача 2: Использование форм

Создать страницу с формой, состоящей из:

- текстового поля (имя)
- поля выбора пола (как в примере в видео - `inputtyperadio`).

В зависимости от отправленных данных (значения поля "пол" - М или Ж) выводить приветствие.

Пример:

```
"Добро пожаловать, мистер Смит!"  
или  
"Добро пожаловать, миссис Смит!"
```

*Все задачи необходимо сделать в отдельных файлах (например: `task1.php`)
Оформить отчет с листингом кода*

Раздел 2. Основные инструменты и подходы для создания web приложений на языке PHP

Тема 2.3. Встроенные возможности. Формы. (4 часа)

Практическая работа №7

«Cookie. Сессии»

Задачи обучающегося:

1. Научиться создавать страницы с формами для ввода данных

Опорные понятия: пользовательские формы.

Планируемый результат:

Студент должен

Уметь создавать формы различных типов

Необходимое оборудование: ПК, IDE NetBeans, OpenServer

Алгоритм деятельности обучающегося:

Задача 1: Время жизни cookie

Пользователь приходит на сайт. Используя cookie сделать так, чтобы б впервые пришедший пользователь видел фразу:

"Добро пожаловать, новичок!"

Если пользователь уже посещал сайт в течении последних 10-ти часов, выводить фразу:

"С возвращением, дружище!"

Подсказка: Используйте инструменты для разработчиков Вашего браузера для просмотра и очистки текущих значений cookie.

Задача 2: Дата и время последнего посещения

Используя cookie реализовать вывод на страницу сообщения с датой и временем последнего визита.

Задача 3: Счетчик посещений

Используя cookie реализовать вывод на страницу сообщения с количеством посещений страницы.

Задача 4: Многостраничный тест

Представим себе сайт с прохождением теста: на каждой странице находится вопрос, варианты ответа и кнопка "Далее". Тест содержит некоторое количество страниц (допустим 3). На последней, 4-й странице пользователь должен получить результат теста.

Задача – создать подобный тест.

Подсказки:

- Для создания страниц вопросов используйте формы (с элементом `type="radio"`), которые будут отправлять результаты на следующую страницу с вопросом или результатами.
- Ответы пользователя на вопросы на предыдущих страницах можно (и нужно) сохранять в сессии
- Для расчета результатов теста сравните ответы пользователя (хранятся в сессии) с заранее определенными правильными ответами (могут храниться, например, в массиве или переменных).
- После окончания текста сессию следует очистить

Все задачи необходимо сделать в отдельных файлах (например: `task1.php`)

Оформить отчет с листингом кода

Раздел 2. Основные инструменты и подходы для создания web приложений на языке PHP

Тема 2.4. Базы данных. (4 часа)

Практическая работа №8 «Базы данных»

Задачи обучающегося:

1. Научиться создавать базы данных, создавать, удалять, изменять записи

Опорные понятия: пользовательские формы.

Планируемый результат:

Студент должен

Уметь создавать создавать базы данных, создавать, удалять, изменять записи

Необходимое оборудование: ПК, IDENetBeans, OpenServer

Алгоритм деятельности обучающегося:

Создайте базу данных со структурой 4 таблицы с заданными полями. Обратите внимание на типы полей, добавьте первичный ключ в каждую таблицу и AutoIncrement для первичного ключа.

Добавить в каждую таблицу 5-10 записей с правдоподобными данными:

- Несколько категорий новостей: спорт, политика, ...
- Пользователи: имена, фамилии, логины, пароли, ... (последний визит не заполняйте)

Для новостей скопируйте тексты с новостных сайтов, укажите даты (с интервалом в несколько дней), поле preview оставить пустым, значение в category_id соответствует значению id из таблицы news_category, аналогично author_id соответствует user.id

Комментарии: текст, дата, другие поля аналогично.

Подсказка: используйте phpMyAdmin

Задача 1: INSERT

Добавить 3 записи в таблицу user с любыми данными одним запросом INSERT.

Задача 2: DELETE; AND и OR в условиях WHERE

- 1) Написать запрос для удаления пользователей (таблица user) с именем "Томас".
- 2) Написать запрос для удаления пользователей с именем "Томас" и фамилией "Смит".
- 3) Написать запрос для удаления пользователей с именем "Томас" или именем "Джон".

Подсказка:(на английском языке. Если сложно читать - просто посмотреть примеры запросов)http://www.techonthenet.com/mysql/and_or.php

Задача 3: UPDATE

- 1) Изменить имя (на "Тимофей") и фамилию (на "Опель") пользователю с идентификатором "3".
- 2) Изменить фамилию троим пользователям с наибольшими значениями id на "Смит".

Подсказка: используйте ORDER BY, LIMIT.

Задача 4: SELECT

- 1) Выбрать 3 последних новости из категории с идентификатором 2.
- 2) Выбрать всех пользователей с именем "Владислав" или "Елена".
- 3) В категории новости подсчитать количество новостей со статусом "1" и "0".

Подсказка: используйте GROUP BY, COUNT().

Задача 5: SELECT + PHP

Вывести список всех новостей со статусом "1" в виде списка:

<h1>Заголовок</h1>

<p>Короткое описание</p>

Читать далее

Подсказка: используйте запрос с оператором SELECT и цикл foreach

Раздел 2. Основные инструменты и подходы для создания web приложений на языке PHP

Тема 2.4. ООП. (8 часов)

Практическая работа №9 «Объектно-ориентированное программирование»

Задачи обучающегося:

1. Научиться создавать базы данных, создавать, удалять, изменять записи

Опорные понятия: классы и объекты.

Планируемый результат:

Студент должен

Уметь создавать функции используя принципы ООП

Необходимое оборудование: ПК, IDE NetBeans, OpenServer

Используйте класс Car из занятия.

- 1) Добавить классу 3 свойства на свое усмотрение (например: количество дверей, стоимость, ...). Одному из свойств присвоить значение по умолчанию.
- 2) Создать 4 объекта класса Car.
- 3) Для двух объектов задать значения свойств, используя обращение к свойству (например, \$car1->price).
- 4) Для двух оставшихся задать свойства используя конструктор (написать конструктор для инициализации объекта).
- 5) Написать функцию fuelConsumption() для расчета количества топлива, затраченного на поездку на заданную дистанцию. Используйте свойство объекта \$fuel (расход топлива на 100 км).

- 6) Добавить в класс 3 числовых константы (со значениями, например, 2,5,12). Распечатать значения констант в коде программы (вне класса).
- 7) Добавить статический метод getMaxConstant(). Этот метод находит наибольшую из констант класса и возвращает её значение. Не забывайте, что статические методы принадлежат классу, и вызываются из контекста класса.

Практическая работа №10 «Объектно-ориентированное программирование»

Задачи обучающегося:

1. Научиться создавать базы данных, создавать, удалять, изменять записи

Опорные понятия: классы и объекты.

Планируемый результат:

Студент должен

Уметь создавать функции используя принципы ООП

Необходимое оборудование: ПК, IDE NetBeans, OpenServer

- 1) Создать класс Figure - плоская геометрическая фигура. У этого класса есть такие свойства: площадь, цвет. И константа: количество сторон.
- 2) Для класса Figure написать метод infoAbout(). Этот метод возвращает сообщение: "Это геометрическая фигура!".
- 3) От класса Figure унаследовать классы: Rectangle, Triangle, Square.
- 4) Добавить для Rectangle приватные свойства a, b - длины сторон.
- 5) Добавить для Square приватное свойство a - длина стороны.
- 6) Добавить для Triangle приватные свойства a, b, c - длины сторон.
- 7) Для каждого из классов Rectangle, Triangle, Square определить значение константы: количество сторон. Например, для квадрата: const SIDES_COUNT = 4;
- 8) Создать конструкторы для классов Rectangle, Triangle, Square для инициализации значений длин сторон.
- 9) Для каждого из классов Rectangle, Triangle, Square реализовать метод getArea() - подсчет площади. Методы возвращают значение площади. Подсказки ниже.
- 10) Для каждого из классов Rectangle, Triangle, Square переопределить метод infoAbout() так, что бы он возвращал строку такого содержания: (пример для квадрата):
"Это класс квадрата. У него 4 стороны".

Аналогично для других классов.

Подсказка: использовать константу класса для вывода количество сторон.

11) Сделать методы infoAbout() финальными.

10) Для каждого класса Rectangle, Triangle, Square создать по 2 объекта (с передачей значений длин сторон в конструктор).

11) Вызвать для всех объектов методы getArea(), вывести результаты.

Подсказки по математике:

1) Площадь прямоугольника:

$$S = a * b;$$

2) Площадь квадрата:

$$S = a * a;$$

3) Площадь треугольника:

<http://www-formula.ru/index.php/2011-09-19-02-39-24/2011-09-19-03-26-12>

Практическая работа №11 «Объектно-ориентированное программирование»

Задачи обучающегося:

1. Научиться создавать базы данных, создавать, удалять, изменять записи

Опорные понятия: классы и объекты.

Планируемый результат:

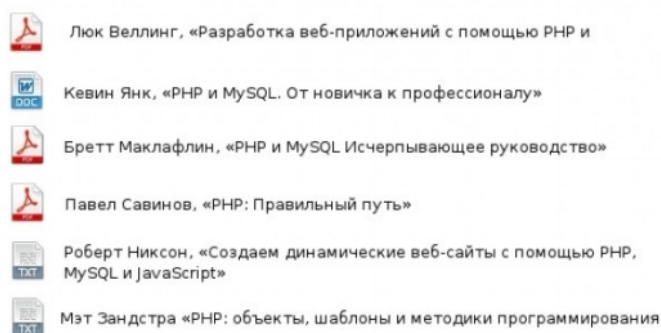
Студент должен

Уметь создавать функции используя принципы ООП

Необходимое оборудование: ПК, IDE NetBeans, OpenServer

В таблице books хранятся записи о книгах.

Задача - получить эти записи и вывести так:



Нужно получить примерно такой html код:

```
<p>
<imgsrc="/images/BookPdf.jpg" alt="" />
<a href="/files/webappsdev.pdf">ЛюкВеллинг,Разработкавебприложений</a>
</p>
<p>
<imgsrc="/images/BookDoc.jpg" alt="" />
<a href="/files/phpmysql.doc">КевинЯнк, PHP MySQL</a>
</p>
```

1. Решить задачу нужно, используя полиморфизм.

2. Виды книг отличаются иконками.

3. Составляйте план решения задачи:

- определите логические части программы
- создайте файловую структуру
- реализуйте необходимые классы, методы

Структура таблицы book:

- id
- name
- author
- file_path - путь к файлу на диске в папке /files/.
- type - тип записи: BookTxt, BookPdf, BookDoc.
- sort_order - порядковый номер книги в списке, учитывается при выборке записей из базы данных (ORDER BY)

2. Создать структуру файлов: books.php, classes.php, data.php.

3. Реализовать класс Book - базовый класс книги.

От класса Book унаследовать классы: BookPdf, BookTxt, BookDoc.

Для классов BookPdf, BookTxt, BookDoc реализовать метод printInfo() – печатает необходимый блок для каждой книги.

4. Осуществить вывод списка книг подобно выводу публикаций в уроке.

5. Файлы хранить в папке /files/ сайта. При выводе книг подставлять значение из поля file_path в атрибут href="#" ссылок. Тем самым, по нажатию на книгу мы сможем её скачивать.

6. Иконки сохранить в папке /images/ (найти любые иконки маленького размера из интернета). При выводе книг подставлять в путь картинки значение из поля type.