

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Таныгин Максим Олегович

Должность: и.о. декана факультета фундаментальной и прикладной информатики

Дата подписания: 21.09.2023 13:06:21

Уникальный программный ключ:

65ab2aa0d384efe8480e6a4c688eddbc475e411a

МИНОБРНАУКИ РОССИИ

Юго-Западный государственный университет

УТВЕРЖДАЮ:

Заведующий кафедрой  
программной инженерии

*(наименование кафедры полностью)*



А.В. Малышев

*(подпись)*

«17» января 2022 г.

ОЦЕНОЧНЫЕ СРЕДСТВА

для текущего контроля успеваемости  
и промежуточной аттестации обучающихся  
по дисциплине

Web-программирование

*(наименование дисциплины)*

09.03.04 Программная инженерия

*(код и наименование ОПОП ВО)*

Курс – 2022

# 1. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ

## 1.1. ВОПРОСЫ ДЛЯ СОБЕСЕДОВАНИЯ

### *Тема 1 Современные технологии разработки web-приложений*

1. Основные принципы построения сети Интернет.
2. Объясните, что такое модель OSI.
3. Зачем применяются Интернет-протоколы,
4. На каких принципах построен протокол TCP/ IP?
5. Что такое порты ?
6. Для чего применяется протокол HTTP?
7. Объясните, что такое DNS,
8. Объясните, что такое URL.

### *Тема 2 Разработка статических web-страниц средствами HTML5*

1. Объясните, что такое статическая web-страница .
2. Объясните Структуру HTML-документа.
2. Какие тэги применяют для форматирования шрифта?
4. Какие тэги применяют для форматирования абзацев?
5. Какие тэги применяют для форматирования блоков?
6. Какие тэги применяют для построения таблиц?
7. Какие тэги применяют для создания внешние и внутренних гиперссылок?
8. Какие тэги применяют для работ с иллюстрациями?

### *Тема 3 Использование таблиц стилей CSS*

1. Объясните, что такое CSS.
2. Объясните базовые синтаксические правила CSS.
3. Объясните, как можно установить связь с файлом стилей.
4. Как задать свойства для цветового оформления?
5. Как задать свойства шрифтов?
6. Как задать свойства текста?
7. Объясните принцип наследования стилей.

8. Как создать рисунок с помощью CSS?

#### *Тема 4 Разработка динамических web-страниц средствами HTML5*

1. Объясните, что такое динамическая web-страница?

2. Объекты и методы форм

4. Какое событие возникает, когда элемент формы: `<button>`, `<input>`, `<label>`, `<select>`, `<textarea>` теряет фокус?

5. Как задать массив в JavaScript?

6. Как создать объект даты в JavaScript?

7. Как задать для ссылку на внешний сценарий под названием "abc.js"?

8. Что делает тег `<noscript>`?

#### *Тема 5 Разработка и ведение серверных баз данных*

1. Что такое база данных?

2. Какие модели данных используют для создания серверных баз данных?

3. Что такое запись таблицы базы данных?

4. Что такое поле таблицы базы данных?

5. Что такое Структура записи таблицы базы данных?

6. Что является основным объектом реляционной базы данных?

7. Перечислите серверные СУБД.

8. Объясните, что такое ведение базы данных.

9. Что такое ограничения целостности базы данных,

#### *Тема 6 Разработка серверных приложений на языке PHP*

1. Объясните, что такое PHP?

2. Объясните структуру кода на PHP/

3. Запишите оператор проверки условия в PHP

4. Запишите оператор цикла в PHP

5. Что такое GET в PHP?

6. Что такое `$_POST[]` в PHP?

7. Что такое `$_SESSION()` в PHP?

8. Перечислите суперглобальные переменные PHP.

---

**Шкала оценивания:** 10-балльная.

**Критерии оценивания:**

Каждый вопрос (задание) в тестовой форме оценивается по дихотомической шкале: выполнено – 1 балл, не выполнено – 0 баллов.

Применяется следующая шкала перевода баллов в оценку по 5-балльной шкале:

- 9 -10 баллов соответствуют оценке «отлично»;
- 7 -8 баллов – оценке «хорошо»;
- 6 -7 баллов – оценке «удовлетворительно»;
- 5 баллов и менее – оценке «неудовлетворительно».

## 2 ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ

### 2.1. БАНК ВОПРОСОВ И ЗАДАНИЙ В ТЕСТОВОЙ ФОРМЕ

1. JavaScript обрабатывает события встроенные в страницы HTML и может реагировать на пользовательские события

- щелчок мыши
- ввод информации в форму
- выключение компьютера
- закрытие окна браузера

2. Арифметическая операция в JavaScript

- =
- %
- >
- \*
- <>

3. Для включения JavaScript в HTML используется тег

- <javascript>
- <script language="JavaScript">
- <JS>
- <script type="text/JavaScript"

4. Как JavaScript хранит даты ?

- число миллисекунд начиная с 1 января 1970
- количество дней с 1 января 1900
- количество секунд с начала акции предлагают Netscape общественности.

5. Какие из перечисленных ниже имен идентификаторов не действительны в JavaScript?

- 2names
- first\_and\_last\_names

- FirstAndLast

6. Какое из перечисленных ниже действий не может быть сделано на стороне клиента JavaScript?

- Проверка формы
- Отправка формы содержимое по электронной почте
- Сохранение содержания формы в файл базы данных на сервере

7. Какое из следующих событий возникает, когда элемент формы: <button>, <input>, <label>, <select>, <textarea> теряет фокус?

- ONFOCUS
- ONBLUR
- OnClick
- ondblclick

8. Какой синтаксис соответствует оператору if JavaScript?

- if(условное выражение истинно) then execute this codeend if
- if (условное выражение истинно) execute this codeend if
- if (условное выражение истинно) {то выполнить этот код> ->}
- if (условное выражение истинно) then {выполнить этот код}

9. Какой способ правильно задает массив JavaScript?

- var ТХТ = новый массив (1: "Тим", 2: "Ким", 3: "Джим")
- var ТХТ = новый массив: 1 = ("Тим") 2 = ("КИМ") 3 = ("Джим")
- var ТХТ = новый массив ("Тим", "Ким", "Джим")
- var ТХТ = новый массив = "Тим", "Ким", "Джим"

10. Как создать объект даты в JavaScript?

- dateObjectName = новая дата ([параметры])
- dateObjectName.new Дата ([параметры])
- dateObjectName: = новая дата ([параметры])
- dateObjectName Дата ([параметры])

11. Когда пользователь просматривает страницу, содержащую программу JavaScript, какая на самом деле программа выполняет сценарий?

- у пользователя на компьютере работает веб-браузер
- веб-сервер
- центральная машина в корпоративных офисах компании Netscape
- компьютер провайдера Интернет

12. Логическая операция в JavaScript

- %%
- &&
- &
- ||
- and

13. Объекты встроенные в JavaScript

- Data
- Date
- Objekt
- Array
- One

14. Оператор ветвления в JavaScript

- do
- if
- like
- else
- case

15. Оператор ветвления в JavaScript \_\_\_\_\_

16. Оператор цикла в JavaScript

- do
- while

- if
- for
- else

17. Операторы JavaScript заканчиваются

- :
- ;
- "
- #

18. Укажите правильный синтаксис JavaScript, чтобы написать "Hello World"?

- System.out.println ("Hello World")
- println ("Hello World")
- document.write ("Hello World")
- Response.Write ("Hello World")

19. Укажите правильный синтаксис для ссылки на внешний сценарий под названием "abc.js"?

- <script href=" abc.js">
- <script name=" abc.js">
- <script src=" abc.js">
- Ни один из выше

20. Функция JavaScript преобразует строку в целое число

- Parse
- ParseFloat
- Int
- ParseInt
- Float

21. Функция JavaScript \_\_\_\_\_ преобразует строку в целое число

22. Что делает тег <noscript>?



- Используется для текста, который будет отображаться без JavaScript в браузере.
- Предотвращает сценарии на странице от исполнения.

23. Укажите элемент HTML, внутри которого мы ставим JavaScript?

- <js>
- <scripting>
- <script>
- <javascript>

24. Элемент HTML, внутри которого мы ставим JavaScript \_\_\_\_\_

25. Чтобы установить окна, чтобы захватить все события CLICK

- window.captureEvents (Event.CLICK);
- window.handleEvents (Event.CLICK);
- window.routeEvents (Event.CLICK);
- window.raiseEvents (Event.CLICK);

26. Что должно находиться в самом конце JavaScript?

Теги <script LANGUAGE="JavaScript">

- </ SCRIPT>
- <script>
- оператор END
- </html>

27. Что должно находиться в самом конце JavaScript? \_\_\_\_\_

28. Выберите Web-сервер.

- Crome
- IE
- IIS
- Firefox

29. Выберите идентификатор в PHP

- \$1stvar

- \$F\_Name
- Myvar
- \$&62##;

30. Выберите оператор условия в PHP

- >
- switch (выражение)
- ==
- !=

31. Оператор проверки условия в PHP

32. Выберите оператор цикла в PHP

- in
- make
- while

33. Оператор цикла в PHP

34. Выберите суперглобальные переменные PHP

- GET
- \$\_POST[]
- \$\_SESSION()

3. Выберите элемент массива в PHP

- a[1]
- \$members['FName']
- \$members["FName"]
- b{2}

36. Для чего используется объект document?

- Для представления в сценарии документа
- Для представления в сценарии загруженного документа
- Для представления в сценарии загруженного HTML - документа

- Для представления в сценарии загруженного изображения

37. Для представления в сценарии загруженного HTML - документа используется объект

38. Для чего используется строковый объект?

- Для работы с датами
- Для работы с числами
- Для работы со строками
- Для работы с символами

39. Для чего используется язык HTML?

- Для создания специальной разметки текста
- Для разметки текста
- Для программирования
- Для создания специальной разметки текста, которая интерпретируется браузером

40. Для создания специальной разметки текста, которая интерпретируется браузером используется язык \_\_\_\_\_

41. Для чего применяется объект хронологических дат?

- Для работы со временем
- Для работы с данными
- Для работы с числами
- Для работы с хронологическими датами

42. Как выполнить соединение в PHP с СУБД MySQL

- mysql.exe -h mysql.exe -h 172.23.30.98 -u 1408057 -p
- http -h 172.23.30.98 -u 1408057 -p
- mysql\_pconnect('172.23.30.98 ', ' 1408057 ', '1234567');

43. Какие объекты называются объектами ядра JavaScript?

- Объекты Array, Math, Date, Location
- Объекты Array, Location, Date, History
- Объекты Array, Math, Date, String
- Объекты Location, Math, History, String

44. Какие существуют контейнеры текста?

- Существуют контейнеры текста: DIV и SPAN
- Существуют контейнеры текста: P и SPAN
- Существуют контейнеры текста: DIV, P и SPAN
- Существуют контейнеры текста: DIV и P

45. DIV и SPAN - контейнеры \_\_\_\_\_

Как изменять стили элементов страницы?

- С помощью спецификаций элементов
- С помощью спецификаций таблиц стилей
- С помощью таблиц стилей
- С помощью таблиц элементов

46. Какова роль объекта Server?

- Представляет в сценариях браузер
- Представляет в сценариях Интернет-сервер и браузер
- Представляет в сценариях Интернет-сервер и навигатор
- Представляет в сценариях Интернет-сервер

47. Какой метод создает массив?

- Метод new
- Метод slice
- Конструктор
- string

48. Как создается объект History?

- Создается браузером

- Создается с помощью конструктора
- Создается с помощью оператора new
- Создается с помощью метода new

49. Как создается объект Location?

- Создается с помощью конструктора
- Создается с помощью оператора new
- Создается с помощью метода href
- Создается браузером

50. Как создать массив?

- С помощью оператора array
- С помощью оператора string
- С помощью оператора try
- С помощью оператора new

51. Как создать нумерованный список?

- С помощью тэга <ol>
- С помощью тэга <num>
- С помощью тэга <td>
- С помощью тэга <ul>

52. Нумерованный список можно создать С помощью тэга \_\_\_\_\_

Как создать объект хронологических дат?

- С помощью оператора new
- С помощью оператора array
- С помощью оператора string
- С помощью оператора try

53. Как создать строковый объект?

- С помощью оператора array

- С помощью оператора new
- С помощью оператора string
- С помощью оператора try

54. Как создать форму в JavaScript объект?

- На основе класса
- На основе другого объекта
- На основе формы

55. Как создать фреймовую структуру?

- С помощью тэгов <frameset> и <table>
- С помощью тэгов <frameset> и <frame>
- С помощью тэгов <table> и <frame>
- С помощью тэгов <frameset> и <body>

56. Какую роль играет объект Window?

- Он представляет окно
- Он представляет окно сообщений
- Нужен для отображения страниц
- Он представляет в сценариях окно браузера

57. Можно в JavaScript наследовать объект?

- да
- нет

58. Можно в JavaScript создать свой объект?

- да
- нет

59. Можно ли использовать в клиентских сценариях объект Window?

- Нет
- При определенных условиях

- Да
- Всегда должен использоваться

60. Назначение файлов *cookies*

- Позволяет создать набор данных на сервере
- Позволяет создать набор данных Access
- Позволяет создать набор данных MySQL
- Предназначены для хранения информации и доступны из браузера

61. Оператор соединения строк и переменных в PHP

- . (точка)
- +
- &
- add

62. Проверка на наличие конца файла осуществляется при помощи функции

- end()
- file()
- feof()
- endfile()

63. Что делают методы математического объекта?

- Выполняют элементарные математические операции
- Выполняют любые математические функции
- Выполняют элементарные математические функции
- Выполняют математические преобразования

64. Что представляет собой таблица стилей?

- Текстовый файл, содержащий спецификации стилей
- Текстовый файл с расширением .js
- Файл с расширением .cs, содержащий спецификации стилей
- Текстовый файл с расширением .css, содержащий спецификации стилей

6. Что содержат свойства математического объекта?

- Основные математические функции
- Основные математические выражения
- Основные математические операции
- Основные математические константы

66. Что такое HTML - страница?

- Документ, созданный с использованием языка HTML
- Документ, созданный с использованием языка HTML, показанный в окне браузера
- Документ, показанный в окне браузера
- Документ, созданный с использованием языка JavaScript, показанный в окне браузера

67. Что такое массив?

- Набор однотипных значений в оперативной памяти
- Набор однотипных значений, занимающий в оперативной памяти непрерывный блок
- Набор разнотипных значений, занимающий в оперативной памяти непрерывный блок
- Набор разнотипных значений в оперативной памяти

68. Что такое математический объект?

- Объект String
- Объект Number
- Объект Array
- Объект Math

69. Что такое объект хронологических дат?

- Объект Math
- Объект Date
- Объект Array



- Объект String

70. Что такое серверные объекты JavaScript?

- Объекты, которые можно использовать в сценариях
- Объекты, которые можно использовать в сценариях, выполняемых на браузере
- Объекты, которые можно использовать в сценариях, выполняемых на сервере или браузере
- Объекты, которые можно использовать в сценариях, выполняемых на сервере

71. Что такое строковый объект?

- Объект языка JavaScript
- Объект String
- Объект String языка C++
- Объект String языка JavaScript

72. Является ли язык JavaScript полностью объектно-ориентированным?

- да
- нет

73. Определите последовательность выполнения действий при работе с логическими выражениями.

- Действия в скобках Ответ 1
- Инверсия Ответ 2
- Конъюнкция Ответ 3
- Дизъюнкция Ответ 4
- Импликация Ответ 5
- Эквивалентность Ответ 6

74. Установите соответствие обозначений и определений

НЕ	/
И	&
ИЛИ	V

75. Установите соответствие понятий и определений

- Запись таблицы базы данных      Совокупность данных в одной строке таблицы базы данных
- Поле таблицы базы данных      Совокупность данных в одном столбце таблицы базы данных
- Структура записи таблицы      Совокупность имен столбцов и типов данных
- Файл базы данных      Совокупность Структуры записи таблицы и данных

76. Определите последовательность действий для создания запроса :

- выбрать таблицу
- выбрать столбцы
- задать условие отбора
- выполнить запрос

77. Основным объектом базы данных является

- Таблица
- Запрос
- Отчет
- Форма

78. База данных - это:

- совокупность сведений, характеризующих объекты, процессы или явления реального мира.
- пакет пользовательских программ;
- совокупность файлов на жестком диске;

79. Соотнесите тип связи между объектами и его определение

- Один экземпляр одного объекта связан с единственным экземпляром другого объекта    Ответ 1
- Один-ко-многим    Ответ 2
- Многие-ко-многим    Ответ 3

80. Реляционная модель данных основана:

- на таблицах;
- на древовидных структурах;
- на иерархических списках;
- на объектах;

81. Определите последовательность действий для создания базы данных :

физическое проектирование

Логическое проектирование

концептуальное проектирование

Нормализация отношений

разработка сценария

82. Система управления базами данных (СУБД) – это

- специальный программный комплекс для обеспечения доступа к данным и управления ими;
- система средств администрирования банка данных
- система средств архивирования и резервного копирования банка данных;
- система средств хранения данных и управления транзакциями.

83. В основе реляционной модели лежит понятие: \_\_\_\_\_

84. Идентификатор некоторого ресурса сети Интернет имеет следующий вид:

**http://www.ftp.ru/index.html**. Какая часть этого идентификатора указывает на протокол, используемый для передачи ресурса? \_\_\_\_\_

85. Если задан адрес электронной почты в сети Internet user\_name@int.glasnet.ru, то имя владельца электронного адреса... \_\_\_\_\_

86. Определите НЕверный пример записи адреса

- moon.math.msu.edu.ru
- 123.32.0.76
- http://www.city.ru
- nntp://news.microsoft.com
- 

87. Укажите какой топологии компьютерной сети не существует:

- Звезда.
- Кольцо.
- Дерево.
- Снежинка.
- Морской узел.
- 
- Шина.

88. Уникальный адрес компьютера в Сети называется

- IP - адрес
- TCP - адрес
- e-mail
- 
- 

89. Что не является сетевым протоколом?

- UDP
- HTML

- TCP/IP

- ICMP

•

90. Выбор последовательности компьютеров - узлов сети Интернет, через которые пройдет информация, называется \_\_\_\_\_

91.

Авторизация - это ...

- ввод информации на сайте
- выбор фотографии клиента на сайте
- введение логина и пароля
- совершение определенных действий для получения расширенных полномочий при работе с сайтом

92. совершение определенных действий для получения расширенных полномочий при работе с сайтом — это \_\_\_\_\_.

93. Баннер - banner - изображение или текстовый блок ...

- предназначен для доступа в режиме администратора
- данные администратора Web-портала
- рекламного характера, представляющий собой гиперссылку на страницы с подробным описанием продукта или услуги
- гиперссылка на картинку с рекламой

94. Группа компьютеров или других устройств в Сети, управляемых как единое целое в рамках общих правил и процедур

- триггер
- сервер
- домен
- domen

95. Группа компьютеров или других устройств в Сети, управляемых как единое целое в рамках общих правил и процедур — это \_\_\_\_\_

96. Место, где физически расположены файлы веб-страниц, базы данных и пр., а также серверные программы, взаимодействующие с обращающимися к ним браузерами и организующие, по соответствующему запросу браузера, компоновку требуемой веб-страницы.

- Web-сервер
- виртуальный каталог IIS
- SQL-сервер

- База данных

97. Объект веб-страницы (текст или изображение), устанавливающий связь с другим объектом в сети Интернет

- Гиперссылка
- URL
- изображение
- нумерованный список

98. Объект веб-страницы (текст или изображение), устанавливающий связь с другим объектом в сети Интернет — это \_\_\_\_\_

99. Программа навигации по сети Интернет и просмотра веб-ресурсов

- апплет
- браузер
- IE
- утилита

100. Распределенная иерархическая база данных, содержащая информацию об именах серверов Интернета и позволяющая по имени системы определить ее IP-адрес.

- SQL-сервер
- DNS
- операционная система
- парсер
- Domain Name Service

101. Сервис сети Интернет, позволяющий получать доступ к массивам информации, размещенным в глобальной сети — это \_\_\_\_\_

**Шкала оценивания результатов тестирования:** в соответствии с действующей в университете балльно-рейтинговой системой оценивание результатов промежуточной аттестации обучающихся осуществляется в рамках 100-балльной шкалы, при этом максимальный балл по промежуточной аттестации обучающихся по очной форме обучения составляет 36 баллов, по очно-заочной и заочной формам обучения – 60 баллов (установлено положением П 02.016).

Максимальный балл за тестирование представляет собой разность двух чисел: максимального балла по промежуточной аттестации для данной формы обучения (36 или 60) максимального балла за решение компетентностно-ориентированной задачи (6).

Балл, полученный обучающимся за тестирование, суммируется с баллом, выставленным ему за решение компетентностно-ориентированной задачи.

Общий балл по промежуточной аттестации суммируется с баллами, полученными обучающимся по результатам текущего контроля успеваемости в течение семестра; сумма баллов переводится в оценку по шкале (указать нужное: по 5-балльной шкале или

дихотомической шкале) следующим образом (привести одну из двух нижеследующих таблиц):

Соответствие 100-балльной и 5-балльной шкал

<i>Сумма баллов по 100-балльной шкале</i>	<i>Оценка по 5-балльной шкале</i>
100-85	отлично
84-70	хорошо
69-50	удовлетворительно
49 и менее	неудовлетворительно

***Критерии оценивания результатов тестирования:***

Каждый вопрос (задание) в тестовой форме оценивается по дихотомической шкале:  
выполнено – **2 балла**, не выполнено – **0 баллов**.

### 3. КОМПЕТЕНТНОСТНО-ОРИЕНТИРОВАННЫЕ ЗАДАЧИ

1. Напишите программу на PHP.

Выведите на экран одна за другой четыре строки: 'Всем ', 'привет ', 'от ', 'PHP!<br><br>', используя для этого только один оператор echo. После данной инструкции напишите два однострочных комментария: 'Для вывода слов на разных строках мы используем элемент "br", т.е разрыв строки' и 'Переменные выводимые оператором echo перечисляются через запятую'. Далее измените код первой части так, чтобы слова после вывода на экран находились на разных строках. В самом конце нашего кода напишите многострочный комментарий: 'После каждой инструкции мы будем ставить точку с запятой даже, если инструкция является последней в блоке php'

2. Напишите программу на PHP.

Создайте переменную, присвойте ей число 5 и выведите его на экран. Создайте вторую переменную, присвойте ей число 10 и выведите его на экран. Выведите на экран сумму, разность, произведение и частное от деления значения первой переменной на значение второй переменной. Присвойте сумму значений этих двух переменных третьей и выведите итог на экран. Теперь увеличьте текущее значение третьей переменной на величину значения второй переменной и выведите его на экран. И, наконец, увеличьте текущее значение третьей переменной в два раза и выведите его на экран.

3. Восстановите по комментариям php-код условия.

```
<?php  
  
/* Присвоение переменной по значению. */  
  
//Присвоили переменной $a число 15  
  
//Присвоили переменной $b переменную $a, теперь $b==$a==15  
  
//Изменили значение переменной $b на 'new', при этом  
//значение переменной $a не изменилось ($a==15)  
  
//Опять присвоили переменной $b переменную $a, теперь $b==15  
  
//Изменили значение переменной $a на 'sos', при этом  
//значение переменной $b не изменилось ($b==15)  
  
/* Таким образом, переменная, присвоенная по значению, не зависит от  
переменной, которой она была присвоена */  
  
?>
```

4. Создайте переменную \$a и присвойте ей значение 20. Создайте переменную \$b и присвойте ей значение 10. Теперь создайте переменную \$car и присвойте ей строку с именем первой переменной. Выведите на экран \${\$car}. Создайте еще одну переменную \$track и присвойте ей строку с именем второй переменной. Выведите на экран \${\$track}.

5. Создайте переменную \$a и присвойте ей строку 'number'. Теперь попробуйте вывести на экран строку '\$a'. Что нужно сделать, чтобы переменная, находящаяся внутри строки была

правильно распознана интерпретатором PHP? Исправьте код и выведите строку на экран.

6. Посмотрите внимательно на программу, в которой сделана попытка вывода значений переменных на экран. При попытке вывода каких из переменных интерпретатором будет выдана ошибка и почему?

```
<?php
//Локальные и глобальные переменные

$a=1;

//Объявили новую пользовательскую функцию m_func()
function m_func(){
    $b=2;
    echo $a;
}

$c=3;

echo $a.<br>;
echo $b.<br>;
echo $c;
m_func();

?>
```

7. Имена каких переменных заданы неверно? Почему?

```
<?php
//Правила записи имен
$a=1;
$B=2;
$b=3;
$first.number=4;
$firstNumberOf=5;

function m_func(){
    $_local=1;
    $locAl=2;
    $local_var=3;
    $10_local_var=4;
    $local_Var=5;
    $local Var=6;
}

function 5_func(){
    $__a_A=10;
}

?>
```

8. Дана программа. При попытке вывода каких элементов массива будут выведены ошибки и почему?

```
<?php
```



```

$m_3=[4=>56, 33];

$m_3[]=30;
$m_3[8]=80;
$m_3[]=100;

echo $m_3[3]."<br>";
echo $m_3[4]."<br>";
echo $m_3[5]."<br>";
echo $m_3[6]."<br>";
echo $m_3[7]."<br>";
echo $m_3[8]."<br>";
echo $m_3[9]."<br>";

?>

```

9. Создайте три переменные и присвойте им числовые значения. Выведите на экран их произведение. Теперь создайте пользовательскую функцию, принимающую два аргумента по ссылке и один аргумент по значению. Функция должна присваивать переменным другие числовые значения. Вызовите функцию, после чего снова выведите на экран произведение всех переменных. Объясните результат.

10. Создайте две переменные `$sum` и `$product` и присвойте им стартовые значения, равные 0 и 1. Создайте пользовательскую функцию, принимающую два аргумента по ссылке, а все остальные по значению в массив переменной длины. Функция должна вычислять сумму всех аргументов, переданных в массив переменной длины, и присваивать ее первому аргументу. Второму аргументу она должна присваивать произведение всех аргументов, переданных в массив переменной длины. Возвращать функция должна массив, состоящий из первых двух аргументов, переданных функции (т.е. сумму и произведение). Используйте для доступа к элементам массива цикл `foreach`. Вызовите функцию, передав ей в качестве первых двух аргументов наши переменные, а также три числа. Выведите на экран значения возвращенного массива, а также значения наших переменных. Объясните результат.

11. Создайте пользовательскую функцию, которая будет выводить на экран сумму первых двух переданных ей аргументов. Количество передаваемых функции аргументов не указывайте. Если функция будет получать более двух аргументов, она должна выдавать соответствующее предупреждение и не выводить сумму. Вызовите функцию, передав ей в качестве аргументов массив из двух чисел и развернув его в простые аргументы при помощи `...`. Повторите тоже самое для массива из трех чисел. Напомним, что количество переданных функции элементов можно узнать при помощи функции `func_num_args()`, а получить доступ к ним можно при помощи функции `func_get_arg(n)`.

12. Создайте класс, в котором задайте константу, общедоступные свойство и метод класса. Метод класса должен выводить какое-нибудь сообщение. Создайте экземпляр класса. Вызовите метод класса, а также выведите на экран сумму значений константы класса и его свойства. Не забудьте, что для доступа к константе класса нужно использовать двойное двоеточие `::`, а не объектный оператор `→`.

13. Дана программа. Закомментируйте неверные инструкции.

```

<?php

class parent_class{

```

```

        protected $p_1='Защищенное свойство ';

        private $p_2='Закрытое свойство ';

        public function parent_func(){
            echo $this->p_2.'<br>';
            echo $this->p_1.'<br>';
        }
    }
}

```

```

class descendant_class extends parent_class{
    private $p_1='p_var ';
    private $p_2='p_var ';

    function child_func(){
        echo $this->p_1.'<br>';
        echo $this->p_2.'<br>';
    }
}

```

```

$obj=new parent_class();
$obj_2=new descendant_class();

```

```

$obj->parent_func();
echo $obj->p_2;
echo $obj_2->p_2;
$obj_2->child_func();
echo $obj->p_1;

```

?>

<?php

```

class parent_class{

    protected $p_1='Защищенное свойство ';

    private $p_2='Закрытое свойство ';

    public function parent_func(){
        echo $this->p_2.'<br>';
        echo $this->p_1.'<br>';
    }
}

```

```

class descendant_class extends parent_class{
    private $p_1='p_var ';
    private $p_2='p_var ';

    function child_func(){
        echo $this->p_1.'<br>';
        echo $this->p_2.'<br>';
    }
}

```

```
$obj=new parent_class();  
$obj_2=new descendant_class();
```

```
$obj->parent_func();  
echo $obj->p_2;  
echo $obj_2->p_2;  
$obj_2->child_func();  
echo $obj->p_1;
```

```
?>
```

14. Создайте абстрактный класс, который будет содержать два абстрактных защищенных метода `return_car_name()` и `return_car_price()`, а также один обычный общедоступный метод `return_year($year)`, принимающий год выпуска автомобиля, а возвращающий строку "Год выпуска авто: {\$year}". После оформления абстрактного класса, создайте два класса-потомка `bmw_car` и `ford_car`. Абстрактный метод `return_car_name()` после реализации должен возвращать строку с маркой автомобиля, а `return_car_price()`, соответственно, его цену в виде числа. Создайте экземпляр первого класса и вызовите его методы, а затем повторите тоже самое со вторым классом.

15. Задайте режим строгой типизации (используйте инструкцию `declare(strict types=1);`), после чего объявите класс `A`. В нем создайте общедоступный метод, который будет просто возвращать свой числовой аргумент. Затем создайте еще один общедоступный статический метод, который будет принимать два числовых аргумента, а затем выводить на экран произведение вызова первого метода, которому будет передаваться в качестве аргумента первый аргумент данного статического метода, на второй аргумент метода. На первый метод сошлитесь при помощи конструкции `self::`. Создайте класс-потомок `B`, в котором переопределите первый метод таким образом, чтобы он возвращал не сам аргумент, а его квадрат. После оформления вызовите из второго класса первый метод. Объясните результат. Замените в скрипте конструкцию `self::` на `static::` и вызовите метод повторно. Объясните изменение результата.

16. Напишите программу на PHP.

Выведите на экран одна за другой четыре строки: 'Всем ', 'привет ', 'от ', 'PHP!<br><br>', используя для этого только один оператор `echo`. После данной инструкции напишите два однострочных комментария: 'Для вывода слов на разных строках мы используем элемент "br"', т.е. разрыв строки' и 'Переменные выводимые оператором `echo` перечисляются через запятую'. Далее измените код первой части так, чтобы слова после вывода на экран находились на разных строках. В самом конце нашего кода напишите многострочный комментарий: 'После каждой инструкции мы будем ставить точку с запятой даже, если инструкция является последней в блоке `php`'

17. Напишите программу на PHP.

Создайте переменную, присвойте ей число 5 и выведите его на экран. Создайте вторую переменную, присвойте ей число 10 и выведите его на экран. Выведите на экран сумму, разность, произведение и частное от деления значения первой переменной на значение второй переменной. Присвойте сумму значений этих двух переменных третьей и выведите итог на экран. Теперь увеличьте текущее значение третьей переменной на величину значения второй переменной и выведите его на экран. И, наконец, увеличьте текущее значение третьей переменной в два раза и выведите его на экран.

18. Восстановите по комментариям `php`-код условия.

```

<?php

/* Присвоение переменной по значению. */

//Присвоили переменной $a число 15

//Присвоили переменной $b переменную $a, теперь $b==$a==15

//Изменили значение переменной $b на 'new', при этом
//значение переменной $a не изменилось ($a==15)

//Опять присвоили переменной $b переменную $a, теперь $b==15

//Изменили значение переменной $a на 'sos', при этом
//значение переменной $b не изменилось ($b==15)

/* Таким образом, переменная, присвоенная по значению, не зависит от
переменной, которой она была присвоена */

?>

```

19. Создайте переменную \$a и присвойте ей значение 20. Создайте переменную \$b и присвойте ей значение 10. Теперь создайте переменную \$car и присвойте ей строку с именем первой переменной. Выведите на экран `$$car`. Создайте еще одну переменную \$track и присвойте ей строку с именем второй переменной. Выведите на экран `$$track`.

20. Создайте переменную \$a и присвойте ей строку 'number'. Теперь попробуйте вывести на экран строку '\$a'. Что нужно сделать, чтобы переменная, находящаяся внутри строки была правильно распознана интерпретатором PHP? Исправьте код и выведите строку на экран.

21. Посмотрите внимательно на программу, в которой сделана попытка вывода значений переменных на экран. При попытке вывода каких из переменных интерпретатором будет выдана ошибка и почему?

```

<?php

//Локальные и глобальные переменные

$a=1;

//Объявили новую пользовательскую функцию m_func()
function m_func(){
    $b=2;
    echo $a;
}

$c=3;

echo $a.<br>;
echo $b.<br>;
echo $c;
m_func();

```

?>

22. Имена каких переменных заданы неверно? Почему?

```
<?php
```

```
//Правила записи имен
```

```
$a=1;
```

```
$B=2;
```

```
$b=3;
```

```
$first.number=4;
```

```
$firstNumberOf=5;
```

```
function m_func(){
```

```
    $_local=1;
```

```
    $locAl=2;
```

```
    $local_var=3;
```

```
    $10_local_var=4;
```

```
    $local_Var=5;
```

```
    $local Var=6;
```

```
}
```

```
function 5_func(){
```

```
    $__a_A=10;
```

```
}
```

```
?>
```

23 . Дана программа. При попытке вывода каких элементов массива будут выведены ошибки и почему?

```
<?php
```

```
$m_3=[4=>56, 33];
```

```
$m_3[]=30;
```

```
$m_3[8]=80;
```

```
$m_3[]=100;
```

```
echo $m_3[3]."<br>";
```

```
echo $m_3[4]."<br>";
```

```
echo $m_3[5]."<br>";
```

```
echo $m_3[6]."<br>";
```

```
echo $m_3[7]."<br>";
```

```
echo $m_3[8]."<br>";
```

```
echo $m_3[9]."<br>";
```

```
?>
```

24. Создайте три переменные и присвойте им числовые значения. Выведите на экран их произведение. Теперь создайте пользовательскую функцию, принимающую два аргумента по ссылке и один аргумент по значению. Функция должна присваивать переменным другие числовые значения. Вызовите функцию, после чего снова выведите на экран произведение всех переменных. Объясните результат.

25. Создайте две переменные \$sum и \$product и присвойте им стартовые значения, равные 0 и 1.

Создайте пользовательскую функцию, принимающую два аргумента по ссылке, а все остальные по значению в массив переменной длины. Функция должна вычислять сумму всех аргументов, переданных в массив переменной длины, и присваивать ее первому аргументу. Второму аргументу она должна присваивать произведение всех аргументов, переданных в массив переменной длины. Возвращать функция должна массив, состоящий из первых двух аргументов, переданных функции (т.е. сумму и произведение). Используйте для доступа к элементам массива цикл `foreach`. Вызовите функцию, передав ей в качестве первых двух аргументов наши переменные, а также три числа. Выведите на экран значения возвращенного массива, а также значения наших переменных. Объясните результат.

26. Создайте пользовательскую функцию, которая будет выводить на экран сумму первых двух переданных ей аргументов. Количество передаваемых функции аргументов не указывайте. Если функция будет получать более двух аргументов, она должна выдавать соответствующее предупреждение и не выводить сумму. Вызовите функцию, передав ей в качестве аргументов массив из двух чисел и развернув его в простые аргументы при помощи `...`. Повторите тоже самое для массива из трех чисел. Напомним, что количество переданных функции элементов можно узнать при помощи функции `func_num_args()`, а получить доступ к ним можно при помощи функции `func_get_arg(n)`.

27. Создайте класс, в котором задайте константу, общедоступные свойство и метод класса. Метод класса должен выводить какое-нибудь сообщение. Создайте экземпляр класса. Вызовите метод класса, а также выведите на экран сумму значений константы класса и его свойства. Не забудьте, что для доступа к константе класса нужно использовать двойное двоеточие `::`, а не объектный оператор `→`.

28. Дана программа. Закомментируйте неверные инструкции.

```
<?php

class parent_class{

    protected $p_1='Защищенное свойство ';

    private $p_2='Закрытое свойство ';

    public function parent_func(){
        echo $this->p_2.<br>';
        echo $this->p_1.<br>';
    }
}

class descendant_class extends parent_class{
    private $p_1='p_var ';
    private $p_2='p_var ';

    function child_func(){
        echo $this->p_1.<br>';
        echo $this->p_2.<br>';
    }
}

$obj=new parent_class();
$obj_2=new descendant_class();
```

```

$obj->parent_func();
echo $obj->p_2;
echo $obj_2->p_2;
$obj_2->child_func();
echo $obj->p_1;

?>

<?php

class parent_class{

    protected $p_1='Защищенное свойство ';

    private $p_2='Закрытое свойство ';

    public function parent_func(){
        echo $this->p_2.<br>;
        echo $this->p_1.<br>;
    }
}

class descendant_class extends parent_class{
    private $p_1='p_var ';
    private $p_2='p_var ';

    function child_func(){
        echo $this->p_1.<br>;
        echo $this->p_2.<br>;
    }
}

$obj=new parent_class();
$obj_2=new descendant_class();

$obj->parent_func();
echo $obj->p_2;
echo $obj_2->p_2;
$obj_2->child_func();
echo $obj->p_1;

?>

```

29. Создайте абстрактный класс, который будет содержать два абстрактных защищенных метода `return_car_name()` и `return_car_price()`, а также один обычный общедоступный метод `return_year($year)`, принимающий год выпуска автомобиля, а возвращающий строку "Год выпуска авто: {\$year}". После оформления абстрактного класса, создайте два класса-потомка `bmw_car` и `ford_car`. Абстрактный метод `return_car_name()` после реализации должен возвращать строку с маркой автомобиля, а `return_car_price()`, соответственно, его цену в виде числа. Создайте экземпляр первого класса и вызовите его методы, а затем повторите тоже самое со вторым классом.

30. Задайте режим строгой типизации (используйте инструкцию `declare(strict_types=1);`), после чего объявите класс `A`. В нем создайте общедоступный метод, который будет просто возвращать свой числовой аргумент. Затем создайте еще один общедоступный статический метод, который будет принимать два числовых аргумента, а затем выводить на экран

произведение вызова первого метода, которому будет передаваться в качестве аргумента первый аргумент данного статического метода, на второй аргумент метода. На первый метод сошлитесь при помощи конструкции `self::`. Создайте класс-потомок В, в котором переопределите первый метод таким образом, чтобы он возвращал не сам аргумент, а его квадрат. После оформления вызовите из второго класса первый метод. Объясните результат. Замените в скрипте конструкцию `self::` на `static::` и вызовите метод повторно. Объясните изменение результата.

**Шкала оценивания решения компетентностно-ориентированной задачи:** в соответствии с действующей в университете балльно-рейтинговой системой оценивание результатов промежуточной аттестации обучающихся осуществляется в рамках 100-балльной шкалы, при этом максимальный балл по промежуточной аттестации обучающихся по очной форме обучения составляет 36 баллов, по очно-заочной и заочной формам обучения – 60 (установлено положением П 02.016).

Максимальное количество баллов за решение компетентностно-ориентированной задачи – 6 баллов.

Балл, полученный обучающимся за решение компетентностно-ориентированной задачи, суммируется с баллом, выставленным ему по результатам тестирования.

Общий балл промежуточной аттестации суммируется с баллами, полученными обучающимся по результатам текущего контроля успеваемости в течение семестра; сумма баллов переводится в оценку по шкале (*указать нужное: по 5-балльной шкале или дихотомической шкале*) следующим образом (*привести одну из двух нижеследующих таблиц*):

Соответствие 100-балльной и 5-балльной шкал

<i>Сумма баллов по 100-балльной шкале</i>	<i>Оценка по 5-балльной шкале</i>
100-85	отлично
84-70	хорошо
69-50	удовлетворительно
49 и менее	неудовлетворительно

**Критерии оценивания решения компетентностно-ориентированной задачи** (*нижеследующие критерии оценки являются примерными и могут корректироваться*):

**6-5 баллов** выставляется обучающемуся, если решение задачи демонстрирует глубокое понимание обучающимся предложенной проблемы и разностороннее ее рассмотрение; свободно конструируемая работа представляет собой логичное, ясное и при этом краткое, точное описание хода решения задачи (последовательности (или выполнения) необходимых трудовых действий) и формулировку доказанного, правильного вывода (ответа); при этом обучающимся предложено несколько Компетентностно-ориентированная задача №ов решения или оригинальное, нестандартное решение (или наиболее эффективное, или наиболее рациональное, или оптимальное, или единственно правильное решение); задача решена в установленное преподавателем время или с опережением времени.

**4-3 балла** выставляется обучающемуся, если решение задачи демонстрирует понимание обучающимся предложенной проблемы; задача решена типовым способом в установленное преподавателем время; имеют место общие фразы и (или) несущественные недочеты в описании хода решения и (или) вывода (ответа).

**2-1 балла** выставляется обучающемуся, если решение задачи демонстрирует поверхностное понимание обучающимся предложенной проблемы; осуществлена попытка шаблонного решения задачи, но при ее решении допущены ошибки и (или) превышено установленное преподавателем время.

**0 баллов** выставляется обучающемуся, если решение задачи демонстрирует



непонимание обучающимся предложенной проблемы, и (или) значительное место занимают общие фразы и голословные рассуждения, и (или) задача не решена.