

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования «Кабардино-Балкарский государственный  
университет им. Х.М. Бербекова» (КБГУ)

ИНСТИТУТ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА И ЦИФРОВЫХ ТЕХНОЛОГИЙ  
КАФЕДРА ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ

СОГЛАСОВАНО

Руководитель образовательной  
программы \_\_\_\_\_ М.М. Лафшьева

« 12 » 04 2023г.



Директор института  
\_\_\_\_\_ И.И. Писигов

04 2023г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ (ОЦЕНОЧНЫХ МАТЕРИАЛОВ)  
ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

«ПРОФЕССИОНАЛЬНАЯ РАЗРАБОТКА ПРОГРАММНОГО  
ОБЕСПЕЧЕНИЯ JavaScript»

02.03.02 Фундаментальная информатика и информационные технологии  
(код и наименование направления подготовки)

«Проектирование систем искусственного интеллекта»

(наименование профиля подготовки)

Бакалавр

Квалификация (степень) выпускника

Очная

Форма обучения

## Оглавление

1. Перечень компетенций и этапы их формирования .....	3
2. Методические материалы и типовые контрольные задания, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы.....	6
3. Перечень контрольных заданий и иных материалов, необходимых для оценки знаний, умений, навыков и опыта деятельности .....	6
4. Вопросы, выносимые на зачет по дисциплине «Профессиональная разработка программного обеспечения JavaScript» (контролируемые компетенции УК-1, ПКС-1).....	17

# 1. Перечень компетенций и этапы их формирования

## Карта компетенции

### Шифр и название компетенций:

профессиональных (ПКС):

Коды	Содержание компетенций
<b>УК-1</b>	Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач

профессиональных (ПКС):

Коды	Содержание компетенций
<b>ПКС-1</b>	Способен понимать, совершенствовать и применять современный математический аппарат

### Общая характеристика компетенции

**Тип компетенции:** общепрофессиональная компетенция выпускника образовательной программы по направлению подготовки высшего образования 02.03.02 Фундаментальная информатика и информационные технологии, уровень ВО бакалавр.

#### 1.1. Этапы формирования компетенций и средства оценивания

Результаты обучения (компетенции)	Индикаторы достижения компетенций	Основные показатели оценки результатов обучения	Виды оценочного материала, обеспечивающий формирование компетенций
<b>УК- 1.</b> Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	<b>УК-1.1.</b> Способен применять системный подход и методы анализа и синтеза в научно-познавательной деятельности	<p><b>Знать:</b> Принципы сбора, отбора, обобщения и систематизации информации, вероятные стратегии действий</p> <p><b>Уметь:</b> соотносить разнородные явления и систематизировать их в рамках проблемной ситуации в профессиональной деятельности.</p> <p><b>Владеть:</b> Опыт работы с информационными источниками, выработки стратегий действия</p>	<p>Типовые оценочные материалы для устного опроса (п. 5.1.1); типовые оценочные материалы для контрольной работы (п. 5.2.2);</p> <p>типовые оценочные материалы к зачету (п. 5.2.3.)</p>
	<b>УК-1.2.</b> Способен осуществлять поиск алгоритмов решения проблемной ситуации на основе доступных источников информации с применением	<p><b>Знать:</b> Принципы и методы системного подхода.</p> <p><b>Уметь:</b> отличать факты от мнений, интерпретаций, оценок и т.д. в рассуждениях других участников деятельности; применять принципы и методы</p>	

	современных информационных и коммуникационных средств и технологий	системного подхода для решения поставленных задач. <b>Владеть:</b> Практическими навыками выбора оптимальных способов решения задач, исходя из действующих правовых норм, имеющихся ресурсов и ограничений.	
<b>ПКС-1.</b> Способен понимать, совершенствовать и применять современный математический аппарат	<b>ПКС-1.1.</b> Способен владеть знаниями в области математических методов и методы исследования математических моделей объектов различной природы	<b>ПКС-1.1.</b> З-1. Знает основные принципы построения математических моделей сложных комплексных объектов и процессов и методики исследования этих моделей; современные технологии математического моделирования и вычислительного эксперимента <b>ПКС-1.1.</b> У-1. Умеет ставить задачи исследования и оптимизации сложных объектов на основе методов математического моделирования, <b>ПКС-1.1.</b> В-1. Владеет навыками выявлять общие закономерности исследуемых объектов, выбирать методы исследования математических моделей	Типовые оценочные материалы для устного опроса (п. 5.1.1); типовые оценочные материалы для контрольной работы (п. 5.2.2); типовые оценочные материалы к зачету (п. 5.2.3.)
	<b>ПКС-1.2.</b> Способен использовать методы проектирования и производства программного продукта, принципы построения, структуры и приемы работы с инструментальными средствами, поддерживающими создание программного продукта	<b>ПКС-1.2.</b> З-1. Знает методы и средства планирования и организации исследований и разработок; методы проведения экспериментов и наблюдений, обобщения и обработки информации в области прикладной математики и информатики <b>ПКС-1.2.</b> У-1. Умеет исследовать научные и технические проблемы с применением современных технологий математического моделирования и вычислительного эксперимента систематизировать результаты научно-исследовательских и опытно-конструкторских работ; применять методы анализа научно-технической информации	

		<b>ПКС-1.2. В-1.</b> Владеет навыками применения методов анализа научно-технической информации	
--	--	------------------------------------------------------------------------------------------------	--

## 1.2. Критерии формирования оценок на различных этапах их формирования

### Текущий и рубежный контроль

Этап (уровень)	Первый этап (уровень)	Второй этап (уровень)	Третий этап (уровень)
<b>Баллы</b>	36-50 баллов	51-60 баллов	61-70 баллов
<b>Характеристика</b>	Полное или частичное посещение аудиторных занятий. Частичное выполнение домашнего задания. Частичное выполнение заданий контрольных работ, тестовых заданий на оценку «удовлетворительно».	Полное или частичное посещение аудиторных занятий. Полное выполнение домашнего задания. Выполнение заданий на коллоквиуме на оценку «хорошо».	Полное посещение аудиторных занятий. Полное выполнение домашнего задания, заданий контрольных работ. Выполнение заданий на коллоквиуме на оценку «отлично».

На первом (начальном) этапе формирования компетенции формируются знания, умения и навыки, составляющие базовую основу компетенции, без которой невозможно ее дальнейшее развитие. Обучающийся воспроизводит термины, факты, методы, понятия, принципы и правила; решает учебные задачи по образцу.

На втором (основном) этапе формирования компетенции приобретает опыт деятельности, когда отдельные компоненты компетенции начинают «работать» в комплексе и происходит выработка индивидуального алгоритма продуктивных действий, направленных на достижение поставленной цели. На этом этапе обучающийся осваивает аналитические действия с предметными знаниями по конкретной дисциплине, способен самостоятельно решать учебные задачи, внося коррективы в алгоритм действий, осуществляя координирование хода работы, переносит знания и умения на новые условия.

Третий (завершающий) этап – это овладение компетенцией. Обучающийся способен использовать знания, умения, навыки при решении задач повышенной сложности и в нестандартных условиях. По результатам этого этапа обучающийся демонстрирует итоговый уровень сформированности компетенции.

### Промежуточная аттестация

#### 7 семестр-зачет

	<b>незачтено</b> (36-60 баллов)	<b>зачтено</b> (61-100 баллов)
--	------------------------------------	-----------------------------------

1	<p>Студент имеет 36-60 баллов по итогам текущего и рубежного контроля, на зачете не дал полного ответа ни на один вопрос, не сделал пример.</p> <p>Студент имеет 36-45 баллов по итогам текущего и рубежного контроля, на зачете дал полный ответ только на один вопрос, а пример сделан неправильно.</p>	<p>Студент имеет 61-100 баллов по итогам текущего и рубежного контроля, на зачете, как минимум, дал полный ответ на один вопроси частично (полностью) ответил на второй. Пример сделан верно.</p>
---	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

## 2. Методические материалы и типовые контрольные задания, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы

### Перечень оценочных средств

№	Наименование оценочного средства	Краткая характеристика оценочного средства	Представление оценочного средства в фонде
1.	Коллоквиум	Средство контроля усвоения учебного материала темы, раздела или разделов дисциплины, организованное как учебное занятие в виде собеседования преподавателя с обучающимися.	Вопросы по темам/разделам дисциплины
2.	Тест	Система стандартизированных заданий, позволяющая автоматизировать процедуру измерения уровня знаний и умений обучающегося.	Фонд тестовых заданий
3.	Контрольная работа	Средство проверки умений применять полученные знания для решения задач определенного типа по теме или разделу	Комплект контрольных заданий по вариантам

## 3. Перечень контрольных заданий и иных материалов, необходимых для оценки знаний, умений, навыков и опыта деятельности

### 3.1. Контрольные задания «Профессиональная разработка программного обеспечения JavaScript» (контролируемые компетенции УК-1, ПКС-1)

- FizzBuzz.** Напишите программу, которая выводит через console.log все числа от 1 до 100, с двумя исключениями. Для чисел, нацело делящихся на 3, она должна выводить 'Fizz', а для чисел, делящихся на 5 (но не на 3) – 'Buzz'. Когда сумеете – исправьте её так, чтобы она выводила «FizzBuzz» для всех чисел, которые делятся и на 3 и на 5.
- Шахматная доска.** Напишите программу, создающую строку, содержащую решётку 8x8, в которой линии разделяются символами новой строки. На каждой позиции либо пробел, либо #. В результате должна получиться шахматная доска.
- Рекурсия.** Ноль чётный. Единича нечётная. У любого числа N чётность такая же, как у N-2. Напишите рекурсивную функцию isEven согласно этим правилам. Она должна принимать число и

- возвращать булевское значение. Потестируйте её на 50 и 75. Попробуйте задать ей -1. Почему она ведёт себя таким образом? Можно ли её как-то исправить?
- Считаем бобы.** Символ номер  $N$  строки можно получить, добавив к ней `.charAt(N)` ( “строка”.`charAt(5)` ) – схожим образом с получением длины строки при помощи `.length`. Возвращаемое значение будет строковым, состоящим из одного символа (к примеру, “к”). У первого символа строки позиция 0, что означает, что у последнего символа позиция будет `string.length - 1`. Другими словами, у строки из двух символов длина 2, а позиции её символов будут 0 и 1. Напишите функцию `countBs`, которая принимает строку в качестве аргумента, и возвращает количество символов “В”, содержащихся в строке. Затем напишите функцию `countChar`, которая работает примерно как `countBs`, только принимает второй параметр — символ, который мы будем искать в строке (вместо того, чтобы просто считать количество символов “В”). Для этого переделайте функцию `countBs`.
  - Сумма диапазона.** Напишите функцию `range`, принимающую два аргумента, начало и конец диапазона, и возвращающую массив, который содержит все числа из него, включая начальное и конечное. Затем напишите функцию `sum`, принимающую массив чисел и возвращающую их сумму. Запустите указанную выше инструкцию и убедитесь, что она возвращает 55. В качестве бонуса дополните функцию `range`, чтобы она могла принимать необязательный третий аргумент – шаг для построения массива. Если он не задан, шаг равен единице. Вызов функции `range(1, 10, 2)` должен вернуть `[1, 3, 5, 7, 9]`. Убедитесь, что она работает с отрицательным шагом так, что вызов `range(5, 2, -1)` возвращает `[5, 4, 3, 2]`.
  - Обращаем массив вспять.** Напишите две функции, `reverseArray` и `reverseArrayInPlace`. Первая получает массив как аргумент и выдаёт новый массив, с обратным порядком элементов. Вторая работает как оригинальный метод `reverse` – она меняет порядок элементов на обратный в том массиве, который был ей передан в качестве аргумента. Не используйте стандартный метод `reverse`.
  - Список.** Объекты могут быть использованы для построения различных структур данных. Часто встречающаяся структура – список (не путайте с массивом). Список – связанный набор объектов, где первый объект содержит ссылку на второй, второй – на третий, и т.п. Списки удобны тем, что они могут делиться частью своей структуры. Например, можно сделать два списка, `{value: 0, rest: list}` и `{value: -1, rest: list}`, где `list` – это ссылка на ранее объявленную переменную. Это два независимых списка, при этом у них есть общая структура `list`, которая включает три последних элемента каждого из них. Кроме того, оригинальный список также сохраняет свои свойства как отдельный список из трёх элементов. Напишите функцию `arrayToList`, которая строит такую структуру, получая в качестве аргумента `[1, 2, 3]`, а также функцию `listToArray`, которая создаёт массив из списка. Также напишите вспомогательную функцию `prepend`, которая получает элемент и создаёт новый список, где этот элемент добавлен спереди к первоначальному списку, и функцию `nth`, которая в качестве аргументов принимает список и число, а возвращает элемент на заданной позиции в списке, или же `undefined` в случае отсутствия такого элемента.
  - Глубокое сравнение.** Оператор `==` сравнивает переменные объектов, проверяя, ссылаются ли они на один объект. Но иногда полезно было бы сравнить объекты по содержимому. Напишите функцию `deepEqual`, которая принимает два значения и возвращает `true`, только если это два одинаковых значения или это объекты, свойства которых имеют одинаковые значения, если их сравнивать рекурсивным вызовом `deepEqual`. Чтобы узнать, когда сравнивать величины через `===`, а когда – объекты по содержимому, используйте оператор `typeof`. Если он выдаёт “object” для обеих величин, значит нужно делать глубокое сравнение. Не забудьте об одном дурацком исключении, случившемся из-за исторических причин: “`typeof null`” тоже возвращает “object”.
  - Свертка.** Используйте метод `reduce` в комбинации с `concat` для свёртки массива массивов в один массив, у которого есть все элементы входных массивов.
  - Every и some.** У массивов есть ещё стандартные методы `every` и `some`. Они принимают как аргумент некую функцию, которая, будучи вызванной с элементом массива в качестве аргумента, возвращает `true` или `false`. Так же, как `&&` возвращает `true`, только если выражения с обеих сторон оператора возвращают `true`, метод `every` возвращает `true`, когда функция возвращает `true` для всех элементов массива. Соответственно, `some` возвращает `true`, когда заданная функция возвращает `true` при работе хотя бы с одним из элементов массива. Они не обрабатывают больше элементов, чем необходимо – например, если `some` получает `true` для первого элемента, он не обрабатывает оставшиеся. Напишите функции `every` и `some`, которые работают так же, как эти методы, только принимают массив в качестве аргумента.
  - Ещё одна ячейка.** Создайте тип ячейки `StretchCell(inner, width, height)`, соответствующий интерфейсу ячеек таблицы из этой главы. Он должен оборачивать другую ячейку (как делает `UnderlinedCell` - из

примера книги), и убеждаться, что результирующая ячейка имеет как минимум заданные ширину и высоту, даже если внутренняя ячейка была бы меньше.

12. **Интерфейс к последовательностям.** Разработайте интерфейс, абстрагирующий проход по набору значений. Объект с таким интерфейсом представляет собой последовательность, а интерфейс должен давать возможность в коде проходить по последовательности, работать со значениями, которые её составляют, и как-то сигнализировать о том, что мы достигли конца последовательности. Задав интерфейс, попробуйте сделать функцию `logFive`, которая принимает объект-последовательность и вызывает `console.log` для первых её пяти элементов – или для меньшего количества, если их меньше пяти. Затем создайте тип объекта `ArraySeq`, оборачивающий массив, и позволяющий проход по массиву с использованием разработанного вами интерфейса. Создайте другой тип объекта, `RangeSeq`, который проходит по диапазону чисел (его конструктор должен принимать аргументы `from` и `to`).
13. **Искусственный идиот.** Грустно, когда жители нашего мира вымирают за несколько минут. Чтобы справиться с этим, мы можем попробовать создать более умного поедателя растений. У наших травоядных есть несколько очевидных проблем. Во-первых, они жадные — поедают каждое растение, которое находят, пока полностью не уничтожат всю растительность. Во-вторых, их случайное движение (вспомните, что метод `view.find` возвращает случайное направление) заставляет их болтаться неэффективно и помирать с голоду, если рядом не окажется растений. И наконец, они слишком быстро размножаются, что делает циклы от изобилия к голоду слишком быстрыми. Напишите новый тип существа, который старается справиться с одним или несколькими проблемами и замените им старый тип `PlantEater` в мире долины. Следите за ними. Выполните необходимые подстройки.
14. **Повтор.** Допустим, у вас есть функция `primitiveMultiply`, которая в 50% случаев перемножает 2 числа, а в остальных случаях выбрасывает исключение типа `MultiplicatorUnitFailure`. Напишите функцию, обёртывающую эту, и просто вызывающую её до тех пор, пока не будет получен успешный результат. Убедитесь, что вы обрабатываете только нужные вам исключения.
15. **Кавычки в тексте.** Допустим, вы написали рассказ, и везде для обозначения диалогов использовали одинарные кавычки. Теперь вы хотите заменить кавычки диалогов на двойные, и оставить одинарные в сокращениях слов типа `aren't`. Придумайте шаблон, различающий два этих использования кавычек, и напишите вызов метода `replace`, который производит замену.
16. **Круговые зависимости.** Запутанная тема в управлении зависимостями – круговые зависимости, когда модуль А зависит от Б, а Б зависит от А. Многие системы модулей это просто запрещают. Модули `CommonJS` допускают ограниченный вариант: это работает, пока модули не заменяют объект `exports`, существующий по-умолчанию, другим значением, и начинают использовать интерфейсы друг друга только после окончания загрузки. Можете ли вы придумать способ, который позволил бы воплотить систему поддержки таких зависимостей? Посмотрите на определение `require` и подумайте, что нужно сделать этой функции для этого.
17. **Чиним область видимости.** Сейчас мы можем присвоить переменной значение только через `define`. Эта конструкция работает как при присвоении старым переменным, так и при создании новых. Эта неоднозначность приводит к проблемам. Если вы пытаетесь присвоить новое значение нелокальной переменной, вместо этого вы определяете локальную с таким же именем. (Некоторые языки так и делают, но мне это всегда казалось дурацким способом работы с областью видимости). Добавьте форму `set`, схожую с `define`, которая присваивает переменной новое значение, обновляя переменную во внешней области видимости, если она не задана в локальной. Если переменная вообще не задана, швыряйте `ReferenceError` (ещё один стандартный тип ошибки). Техника представления областей видимости в виде простых объектов, до сего момента бывшая удобной, теперь будет вам мешать. Вам может понадобиться функция `Object.getPrototypeOf`, возвращающая прототип объекта. Также помните, что область видимости не наследуется от `Object.prototype`, поэтому если вам надо вызвать на них `hasOwnProperty`.
18. **Шляпа кота.** Расширьте анимацию кота, чтобы и кот и его шляпа `` летали по противоположным сторонам эллипса. Или пусть шляпа летает вокруг кота. Или ещё что-нибудь интересное придумайте. Чтобы упростить расположение множества объектов, неплохо будет переключиться на абсолютное



позиционирование. Тогда top и left будут считаться относительно левого верхнего угла документа. Чтобы не использовать отрицательные координаты, вы можете добавить заданное число пикселей к значениям position.

19. **Закладки.** Интерфейс закладок встречается часто. Он позволяет вам выбирать панель интерфейса, выбирая одну из нескольких торчащих закладок над элементом. В упражнении вам нужно сделать простой интерфейс закладок. Напишите функцию asTabs, которая принимает узел DOM, и создаёт закладочный интерфейс, показывая дочерние элементы этого узла. Ей нужно вставлять список элементов <button> вверху узла, по одному на каждый дочерний элемент, содержащих текст, полученный из атрибута data-tabname. Все, кроме одного из дочерних элементов, должны быть скрыты (при помощи display style none), а текущий видимый узел можно выбирать нажатием кнопки. Когда оно заработает, расширьте функционал, чтобы у текущей активной кнопки был свой стиль.
20. **Верстак JavaScript.** Сделайте интерфейс, позволяющий писать и исполнять кусочки кода JavaScript. Сделайте кнопку рядом с <textarea>, по нажатию которой конструктор Function из главы 10 будет обёртывать введённый текст в функцию и вызывать его. Преобразуйте значение, возвращаемое функцией, или любую её ошибку, в строку, и выведите её после текстового поля.
21. **И снова согласование содержания.** В главе 17 первое упражнение было посвящено созданию запросов к eloquentjavascript.net/author, спрашивавших разные типы содержимого путём передачи разных заголовков Ассерпт. Сделайте это снова, используя функцию Node http.request. Запросите, по крайней мере, типы text/plain, text/html и application/json. Помните, что заголовки запроса можно передавать как объект в свойстве headers, первым аргументом http.request. Выведите содержимое каждого ответа.

#### ***Критерии формирования оценок по точкам (контрольные работы)***

**5 баллов** - правильно выполнены все задания, продемонстрирован высокий уровень владения материалом, проявлены превосходные способности применять знания и умения к выполнению конкретных заданий.

**4 балла** - правильно выполнена большая часть заданий, присутствуют незначительные ошибки, продемонстрирован хороший уровень владения материалом, проявлены средние способности применять знания и умения к выполнению конкретных заданий.

**2 балла** - задания выполнены менее чем наполовину, продемонстрирован неудовлетворительный уровень владения материалом, проявлены недостаточные способности применять знания и умения к выполнению конкретных заданий.

**1 балл** - дан неполный ответ, представляющий собой разрозненные знания по теме вопроса существенными ошибками в определениях.

**0 баллов** - при полном несоответствии всем критериям и отсутствию ответа.

### ***3.2. Типовые тестовые задания по дисциплине «Профессиональная разработка программного обеспечения JavaScript» (контролируемые компетенции УК-1, ПКС-1)***

*Отметьте свойства языка JavaScript которые могут быть источниками возможных ошибок программирования:*

- a) наличие встроенных типов данных;
- b) возможность создания абстрактных классов;
- c) наличие указателей;
- d) возможность динамического распределения памяти;
- e) возможность определения символических констант;
- f) возможность преобразования типов.

*Класс В наследован от класса А. Отметьте верное для класса В:*

- a) объект класса В может использоваться как объект базового класса;
  - b) класс В должен быть определен с ключевым словом `derived`;
  - c) класс В может непосредственно обращаться к внутренним атрибутам базового класса;
  - d) класс В наследует все операторы базового класса.
- c) `class A { virtual int f(); }`.

*В чем заключается принцип полиморфизма?*

- a) в наличии виртуальных методов;
- b) в наличии множественного наследования;
- c) в использовании виртуального наследования.

*Какая операция позволяет получить значение, записанное по адресу, который содержится в указателе?*

- a) \*;
- b) ? ;
- c) ^ ;
- d) &.

*Операция "." обозначает*

- a) что атрибут объекта, следующий за этим оператором, будет изменен;
- b) обращение к атрибуту объекта, используемое в специальных случаях;
- c) обращение к атрибуту объекта.

*Прототип функции задает*

- a) тип функции, включая количество и тип аргументов и тип результата;
- b) возможность выполнения этой функции из программ на других языках программирования;
- c) имя функции и минимальное количество параметров.

*Как называется функция, которая вызывает саму себя?*

- a) конструктором;
- b) деструктором;
- c) подставляемой;
- d) рекурсивной.

*В программе на языке JavaScript обязательно имеется функция:*

- a) `head`;
- b) `start`;
- c) `prime`;
- d) `main`;
- e) `finish`.

*Ключевое слово `void` обозначает что функция...*

- a) возвращает число с плавающей запятой;
- b) возвращает целое число;
- c) ничего не возвращает;
- d) является главной.

*Укажите зарезервированное ключевое слово для динамического выделения памяти*

- a) `malloc`;
- b) `new`;

- c) create;
- d) value.

*Выберите наиболее подходящее определение класса:*

- a) тип, содержащий набор функций;
- b) тип, который отображает состояние некоторого объекта;
- c) тип, описывающий поведение некоторой сущности;
- d) тип, описывающий характеристики и поведение объекта.

*Как называется способность объекта скрывать свои данные и реализацию от других объектов системы?*

- a) полиморфизм;
- b) инкапсуляция;
- c) абстракция;
- d) наследование.

*Конструктор класса - это метод, который вызывается при создании объекта для...*

- a) выделения памяти под динамические атрибуты класса;
- b) выделения памяти под статические атрибуты класса;
- c) инициализации атрибутов объекта;
- d) загрузки методов класса в память.

Где верно указано имя переменной?

```
var num_1;  
var num;  
var num-1;  
var 2num;  
ver num;
```

Где можно использовать JavaScript?

Веб-приложения  
Мобильные приложения  
Прикладное программное обеспечение  
Можно во всех перечисленных  
Серверные приложения

Какая переменная записана неверно?

```
var number = 12,5;  
var isDone = 0;  
var num = "STRING";  
var b = false;
```

Что будет записано в переменную test?

```
var a = 5;  
var test = 5 != a ? "Yes" : "No";  
"No"
```

Будет ошибка

```
a  
5
```

Где верно указано имя переменной?

```
var 2num;
```

```
ver num;  
var num  
var num-1;  
var num_1;
```

Где можно использовать JavaScript?

Мобильные приложения

Веб-приложения

Серверные приложения

Прикладное программное обеспечение

Можно во всех перечисленных

Какая переменная записана неверно?

```
var num = "STRING";
```

```
var isDone = 0;
```

```
var b = false;
```

```
var number = 12,5;
```

Что будет записано в переменную test?

```
var a = 5;
```

```
var test = 5 != a ? "Yes" : "No";
```

```
"Yes"
```

```
a
```

```
5
```

Будет ошибка

```
"No"
```

Почему код ниже не будет работать?

```
<script type="javascript/text">
```

```
  console.log("Hi!")
```

```
</script>
```

Необходима точка с запятой после console.log("Hi!")

Запись console.log необходимо прописывать лишь в отдельных файлах

Неверно записан атрибут type

Какие циклы есть в языке JavaScript?

for, forMap, foreach, while

for, forMap, foreach, while, do while

for, while, do while, foreach

for, while, do while

Какие значения можно хранить в переменных?

Только числа и строки

Строки, числа с точкой и простые числа

Строки, числа с точкой, простые числа и булевы выражения

Где верно указан вывод данных?

```
documentWrite("Hello");
```

```
print(Hello);
```

```
prompt("Hello")
```

```
write("Hello");
```

```
console.log("Hello");
```

Что такое условный оператор?

Конструкция, что выполняет код несколько раз

Конструкция для создания определенной переменной

Оператор сравнения значений

Где верно указано имя переменной?

var 2num;

ver num;

var num

var num-1;

var num\_1;

Какая переменная записана неверно?

var num = "STRING";

var isDone = 0;

var b = false;

var number = 12,5;

Что будет записано в переменную test?

var a = 5;

var test = 5 != a ? "Yes" : "No";

"Yes"

a

5

Будет ошибка

"No"

Почему код ниже не будет работать?

```
<script type="javascript/text">
```

```
  console.log("Hi!")
```

```
</script>
```

Необходима точка с запятой после console.log("Hi!")

Запись console.log необходимо прописывать лишь в отдельных файлах

Неверно записан атрибут type

Какие циклы есть в языке JavaScript?

for, forMap, foreach, while

for, forMap, foreach, while, do while

for, while, do while, foreach

for, while, do while

Какие значения можно хранить в переменных?

Только числа и строки

Строки, числа с точкой и простые числа

Строки, числа с точкой, простые числа и булевы выражения

Где верно указан вывод данных?

```
documentWrite("Hello");
```

```
print(Hello);
```

```
prompt("Hello")
```

```
write("Hello");
```

```
console.log("Hello");
```

Что такое условный оператор?

Конструкция, что выполняет код несколько раз

Конструкция для создания определенной переменной

Оператор сравнения значений

В чем разница между confirm и prompt?

Они ничем не отличаются

confirm вызывает диалоговое окно с полем для ввода, prompt - окно с подтверждением

prompt вызывает диалоговое окно с полем для ввода, confirm - окно с подтверждением

Какое количество сообщений будет выведено в консоль?

```
for(var i = 10; i < 35; i += 5) {  
  console.log(i);  
}
```

25

Такой цикл работать не будет

6

15

5

Где верно указан запуск всплывающего окна?

`new alert("Hi")`

`info("Hi")`

Нет верных вариантов

`alert("Hi")`

Какие функции выполняет JS?

\* В вопросе не идет речь про платформу Node JS

Выполняет работу с сервером

Создает разметку на странице сайта

Создает стилевое оформление сайта

Отвечает за работу с базами данных

Отвечает за функции на стороне клиента

Что будет выведено в консоль?

```
var b = 10;  
if (b % 3 == 0) {  
  var i = 10;  
}
```

`console.log(i);`

Ошибку

Число 10

Значение Nan

Значение пустой строки

Значение undefined

В чем отличие между локальной и глобальной переменной?

Отличий нет

Локальные видны повсюду, глобальные только в функциях

Глобальные можно переопределять, локальные нельзя

Локальные можно переопределять, глобальные нельзя  
Глобальные видны повсюду, локальные только в функциях

Где встраивается javascript?

внутри <head>  
внутри <html>  
внутри <body>  
внутри <img/>

Как можно вызвать функцию myFunction в javascript ?

call function myFunction(...)  
funcall myFunction(...)  
myFunction(...)  
call myFunction(...)

Как можно использовать "While" в Javascript ?

(while ...){...}  
while ... ...  
while(...){...}  
do while (...) {...} then {...}

Выберите правильный способ использования комментариев в Javascript ?

{# ... #}  
<!-- ... --!>  
// ...  
\\...

Как найти максимальное значение(max) для x и y?

ceil(x, y)  
max(x, y)  
Math.max(x, y)  
top(x, y)

Какое значение имеет "variable" ?

var variable = 0;  
var i = variable + 10;

for( i < 20; i++){  
variable += i + 10;  
}

undefined  
200  
245  
2452

Является ли Javascript back-end языком программирования ?

Да  
Нет

Как можно сделать изменения в таблице базы данных с помощью Javascript ?

modify [dbName].[TableName] ...

change [dbName].[TableName] ...

никак

Как объявить новый date в Javascript ?

var date = Date()

var date = date('now')

var date = new Date()

var date = date().current()

Выберите правильный вариант.

i+=1

i+=1

i=++1

+i+

Javascript не может изменить html-тег.

Да

Нет

Не верно

С помощью javascript можно установить любой стиль для html-тега

Да

Нет

Как объявить объект с помощью Javascript ?

var variable = new Object()

var variable = Object()

var variable = { }

var variable = new myFunction()

Как открыть новое окно с помощью Javascript ?

window.open(...)

window.new(...)

open(new window())

window.open\_new(...)

Какая функция используется для загрузки следующего URL в списке посещений?

window.history.next()

window.history.load\_next()

window.history.forward()

window.history.load\_forward()

*Как открыть confirm window в Javascript ?*

confirm()

location.confirm()

window.open\_confirm()

window.new\_confirm()

*Как получить cookies в Javascript ?*

window.cookies

location.cookies



document.cookie  
document.cookies

*Как можно написать что-либо на веб-странице с помощью Javascript ?*

window.write(...)  
document.write(...)  
window.page.write(...)  
document.page.write(...)

*Выберите неопределенную переменную*

```
variable1 = '14';
```

```
if ( variable2 ) {  
delete variable2;  
}  
else if ( variable1 ) {  
delete variable1;  
}
```

никакая не определена  
обе определены  
только variable1  
только variable2

#### ***Критерии формирования оценок по тестовым заданиям:***

По итогам выполнения тестовых заданий оценка производится по пятибалльной шкале. При правильных ответах на:

- 89-100% заданий – «5» (баллов);
- 70-88% заданий – «4» баллов);
- 50-69% заданий – «3» (балла);
- 30-49% заданий – «2» (балла);
- 10-29% заданий – «1» (балл);
- менее 10% заданий – «0» (баллов).

#### **4. Вопросы, выносимые на зачет по дисциплине «Профессиональная разработка программного обеспечения JavaScript» (контролируемые компетенции УК-1, ПКС-1)**

1. Представление вещественных чисел. Покажите стандартные вещественные типы языка JavaScript.
2. Представление символьных данных. Покажите стандартный тип CHAR.
3. Каким образом реализуется простое ветвление в программе?
4. Каким образом реализуется множественное ветвление в программе?
5. Каким образом реализуется цикл с предусловием в программе?
6. Каким образом реализуется цикл с постусловием в программе?
7. Каким образом реализуется цикл с заданным числом повторений?
8. Для чего и как выполняется связывание физического файла с логическим файлом?
9. Перечислите основные файловые типы.
10. Как выполнить разыменованье?
11. Как организовать линейный односвязный список?
12. Что такое рекурсия, как ее организовать?

13. Какой метод класса называется конструктором, как его записать?
14. Какой метод класса называется деструктором, как его записать?
15. Понятие виртуальных методов в объектно-ориентированном программировании.
16. Понятие компонента и принцип компонентного программирования.
17. Что называется, свойством компонента?
18. Понятие события компонента. Обработчик события.