

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Якушин Владимир Андреевич
Должность: ректор, д.ю.н., профессор
Дата подписания: 02.11.2023
Уникальный программный ключ:
a5427c2559e1ff4b007ed9b1994671e27053e0dc

Министерство науки и высшего образования РФ
Образовательная автономная некоммерческая организация
высшего образования
«Волжский университет имени В.Н. Татищева» (институт)

УТВЕРЖДАЮ

Ректор Якушин В.А.

от 02.05.2023г. № 77/1

Рабочая программа

Пакеты и комплексы прикладных программ

Направление подготовки 09.03.02 Информационные системы и технологии

Квалификация (степень) выпускника – бакалавр

Форма обучения – очная, заочная, очно-заочная

Тольятти, 2023 г.

Рабочая программа **Пакеты и комплексы прикладных программ** составлена с требованиями ФГОС, ВО, ОПОП по направлению подготовки 09.03.02 Информационные системы и технологии (уровень высшего образования: бакалавриат) и учебного плана.

Программа обсуждена и рекомендована к использованию и (или) изданию решением кафедры на заседании кафедры «Информатика и системы управления»

протокол № 09 от 19.04.2023г.

Зав. кафедрой ИиСУ

к.п.н., доцент Е.Н. Горбачевская

Одобрено Учебно-методическим советом вуза

протокол № 4/23 от 27.04.2023г

Председатель УМС

к.п.н. И.И. Муртаева

1. ПЕРЕЧЕНЬ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

В результате освоения дисциплины у обучающихся должны быть сформированы следующие общепрофессиональные компетенции и профессиональные компетенции:

Наименование компетенции	Код компетенции
Способен применять естественнонаучные и общепрофессиональные знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности	ОПК-1
Способен осуществлять выбор платформ и инструментальных программно-аппаратных средств для реализации информационных систем	ОПК-7

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Данная учебная дисциплина относится к обязательной части образовательной программы 09.03.02 «Информационные системы и технологии».

В таблице 1 представлен перечень компетенций с указанием перечня дисциплин, формирующих эти компетенции согласно учебному плану ОПОП

Таблица 1

Код компетенции	Наименование компетенции, формируемой в рамках освоения дисциплины	Предшествующие дисциплины, формирующие указанную компетенцию	Последующие дисциплины, формирующие указанную компетенцию
ОПК-1	Способен применять естественнонаучные и общепрофессиональные знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности		Физика Информатика Математика Математическая логика и теория алгоритмов Операционные системы Информационные технологии Дискретная математика Инженерная и компьютерная графика Учебная практика. Ознакомительная практика Теория информационных процессов и систем Методы оптимизации Защита выпускной квалификационной работы, включая подготовку к процедуре защиты и процедуру защиты
ОПК-7	Способен осуществлять выбор платформ и инструментальных программно-аппаратных средств для реализации информационных систем		Операционные системы Учебная практика. Ознакомительная практика Теория информационных процессов и систем Электронно-

			вычислительные машины Защита выпускной квалификационной работы, включая подготовку к процедуре защиты и процедуру защиты
--	--	--	---

* в качестве этапа формирования компетенций используются номера семестров согласно учебного плана ОПОП

Перечень планируемых результатов обучения, соотнесенных с планируемыми результатами освоения основной профессиональной образовательной программы направления подготовки, представлен в таблице:

Код и наименование компетенции	Код и наименование индикатора достижения компетенции
ОПК-1. Способен применять естественнонаучные и общинженерные знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности;	ОПК-2.1. Осуществляет анализ современных информационных технологий и выбирает программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности. ОПК-2.2. Применяет современные информационных технологий и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности.
ОПК-7. Способен осуществлять выбор платформ и инструментальных программно-аппаратных средств для реализации информационных систем	ОПК-7.1. Анализирует платформы и инструментальные программно-аппаратные средства для реализации информационных систем ОПК-7.2. Использует основные платформы, технологии и инструментальные программно-аппаратные средства для реализации информационных систем ОПК-7.3. Применяет современные технологии для реализации информационных систем

3. ОБЪЕМ ДИСЦИПЛИНЫ И ВИДЫ УЧЕБНОЙ РАБОТЫ
ОЧНАЯ ФОРМА ОБУЧЕНИЯ

Вид учебной работы	Всего	Семестр
		1
Общая трудоёмкость дисциплины	144 час 4 з.е.	144 час 4 з.е.
Контактная работа с преподавателем (всего)	64	64
В том числе:		
Лекции	32	32
Практические / семинарские занятия	-	-
Лабораторные занятия	32	32
Консультации	-	-
Самостоятельная работа (всего)	80	80
В том числе (если есть):		
Курсовой проект / работа	-	-
Расчетно-графическая работа	-	-
Контрольная работа	-	-
Реферат / эссе / доклад	-	-
Иное	80	80
Вид промежуточной аттестации (зачет, экзамен)	Зачет с оценкой	Зачет с оценкой

ЗАОЧНАЯ ФОРМА ОБУЧЕНИЯ

Вид учебной работы	Всего	Семестр
		1
Общая трудоёмкость дисциплины	144 час 4 з.е.	144 час 4 з.е.
Контактная работа с преподавателем (всего)	12	12
В том числе:		
Лекции	6	6
Практические / семинарские занятия	-	-
Лабораторные занятия	6	6
Консультации	-	-
Самостоятельная работа (всего)	132	132
В том числе (если есть):		
Курсовой проект / работа	-	-
Расчетно-графическая работа	-	-
Контрольная работа	-	-
Реферат / эссе / доклад	-	-
Иное	132	132

Вид учебной работы	Всего	Семестр
		1
Вид промежуточной аттестации (зачет, экзамен)	Зачет с оценкой	Зачет с оценкой

ОЧНО-ЗАОЧНАЯ ФОРМА обучения

Вид учебной работы	Всего	Семестр
		1
Общая трудоёмкость дисциплины	144 час 4 з.е.	144 час 4 з.е.
Контактная работа с преподавателем (всего)	24	24
В том числе:		
Лекции	12	12
Практические / семинарские занятия	-	-
Лабораторные занятия	12	12
Консультации	-	-
Самостоятельная работа (всего)	120	120
В том числе (если есть):		
Курсовой проект / работа	-	-
Расчетно-графическая работа	-	-
Контрольная работа	-	-
Реферат / эссе / доклад	-	-
Иное	120	120
Вид промежуточной аттестации (зачет, экзамен)	Зачет с оценкой	Зачет с оценкой

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

4.1. КАЛЕНДАРНО-ТЕМАТИЧЕСКОЕ ПЛАНИРОВАНИЕ

ОЧНАЯ ФОРМА ОБУЧЕНИЯ

№ п/п	Тема	Количество часов на			
		лекции	практические /семинарские занятия	лабораторные занятия	самостоятельную работу
		Семестр 1			
1	Введение. Основные концепции ООП	2			6
2	Инструментальные средства JDK	2		2	6

3	Синтаксис языка Java	2		4	6
4	Стандартная библиотека классов	2		4	6
5	Использование легковесных процессов	2		2	6
6	Обобщенное программирование	2		4	8
7	Коллекции	2		2	8
8	Лямбда-выражения. Пакет java.util.concurrent.	2			8
9	Работа с потоками ввода/вывода	4		2	8
10	Сериализация объектов..	4		4	8
11	Графический интерфейс пользователя	4		4	8
12	Интернационализация и локализация	4		4	10
	Итого 1 семестру	32		32	80

ЗАОЧНАЯ ФОРМА ОБУЧЕНИЯ

№ п/п	Тема	Количество часов на			самостоятельную работу
		лекции	практические /семинарские занятия	лабораторные занятия	
		Семестр 1			
1	Введение. Основные концепции ООП	0,5			10
2	Инструментальные средства JDK	0,5			10
3	Синтаксис языка Java	0,5			10
4	Стандартная библиотека классов	0,5			10
5	Использование легковесных процессов	0,5			10
6	Обобщенное программирование	0,5		1	10
7	Коллекции	0,5		1	10
8	Лямбда-выражения. Пакет java.util.concurrent.	0,5			10
9	Работа с потоками ввода/вывода	0,5			11
10	Сериализация объектов..	0,5		2	13
11	Графический интерфейс пользователя	0,5		2	14
12	Интернационализация и локализация	0,5			14
	Итого 1 семестру	6		6	132

№ п/п	Тема	Количество часов на			самосто ятельну ю работу
		лекции	практические /семинарские занятия	лабор аторн ые занят ия	
		Семестр 1			
1	Введение. Основные концепции ООП	1			10
2	Инструментальные средства JDK	1			10
3	Синтаксис языка Java	1			10
4	Стандартная библиотека классов	1			10
5	Использование легковесных процессов	1			10
6	Обобщенное программирование	1		2	10
7	Коллекции	1		2	10
8	Лямбда-выражения. Пакет java.util.concurrent.	1			10
9	Работа с потоками вводавывода	1			10
10	Сериализация объектов..	1		4	10
11	Графический интерфейс пользователя	1		4	10
12	Интернационализация и локализация	1			10
	Итого 1 семестру	12		12	120

4.2. КРАТКОЕ СОДЕРЖАНИЕ ЛЕКЦИОННОГО КУРСА

Тема 1. Введение. Основные концепции ООП

Язык программирования Java. Объекты и классы. Конструкторы. Сообщения. Инкапсуляция. Наследование. Полиморфизм. Интерфейсы. Вложенные, локальные и анонимные классы. Принципы проектирования классов.

Тема 2. Инструментальные средства JDK

Javac. Java. Javadoc. Appletviewer. Интегрированные среды разработки.

Тема 3. Синтаксис языка Java

Приложение Hello, World! Апплет Hello, World! Класс Applet. Безопасность при работе с апплетами. Лексический разбор исходного кода. Идентификаторы и ключевые слова. Типы данных и литералы. Переменные. Области видимости. Операторы. Управляющие конструкции. Модификаторы. Аннотации. Перечисляемые типы. Использование объектов. Метод finalize()

Тема 4. Стандартная библиотека классов

Пакеты, входящие в JDK . Пакет java.lang. Класс Object. Класс Class. Класс System . Класс Math. Классы-оболочки. Автоупаковка и автораспаковка. Класс String. Классы StringBuffer и StringBuilder. Использование String и StringBuffer. Класс java.util.StringTokenizer. Класс Exception. Класс RuntimeException. Класс Error. Множественная обработка исключений. Проброс исключений более узких типов. Выражение try-with-resources. Пакет java.util.

Тема 5. Использование легковесных процессов

Реализация потока. Состояние потока. Распределение приоритета между потоками. Класс `java.lang.ThreadGroup`. Методы класса `java.lang.Thread`. Взаимодействие и синхронизация потоков. Модификатор `volatile`.

Тема 6. Обобщенное программирование.

Шаблоны. Описание типов с шаблонами. Описание методов с шаблонами. Формальные параметры типа. Шаблоны с групповой подстановкой.

Тема 7. Коллекции.

Интерфейс `Iterator`. Интерфейс `Collection`. Интерфейсы коллекций. Коллекции общего назначения. Специальные коллекции. Сортировка элементов коллекции. Класс `Collections`

Тема 8. Лямбда-выражения. Пакет `java.util.concurrent`

Исполнители (`Executors`). Очереди (`Queues`). Потокбезопасные коллекции (`Concurrent Collections`). Синхронизаторы (`Synchronizers`). Блокировки (`Locks`). Атомарные объекты (`Atomic objects`).

Тема 9. Работа с потоками ввода-вывода

Иерархия потоков в Java. Класс `InputStream`. Класс `OutputStream`. Класс `Reader`. Класс `Writer`. Специализированные потоки Преобразующие потоки. Стандартные потоки ввода-вывода

Тема 10. Сериализация объектов

Интерфейс `java.io.Serializable`. Класс `java.io.ObjectOutputStream`. Класс `java.io.ObjectInputStream`. Пример сериализации и восстановления объекта. Интерфейс `java.io.Externalizable`. Контроль версий сериализуемого класса

Тема 11. Графический интерфейс пользователя

AWT — `Abstract Window Toolkit`. Компоненты. Кнопка (`Button`). Переключатели (`Checkbox`). Группы переключателей (`CheckboxGroup`). Списки (`List`). Выпадающие списки (`Choice`). Полоса прокрутки (`Scrollbar`). Надписи (`Label`). Холст (`Canvas`). Текстовые Компоненты (`TextComponent`). Контейнеры. Панель (`Panel`). Скроллирующие панели (`ScrollPane`). Окно приложения (`Frame`). Менеджеры компоновки. Меню (`Menu`). Модель обработки событий. Интерфейс слушателя. Слушатель события. Событие. Класс `AWTEventMulticaster`. Основные события AWT. JFC - `Java Foundation Classes`. `Swing`. Архитектура MVC и модель `Swing`. Внешний вид компонент (`Look & Feel`)

Тема 12. Интернационализация и локализация

Интернационализация. Локализация. Класс `Locale`. Класс `ResourceBundle`. Класс `ListResourceBundle`. Класс `PropertyResourceBundle`. Иерархия классов `java.text`. Класс `NumberFormat`. Класс `DecimalFormat`. Класс `DecimalFormatSymbols`. Класс `DateFormat`.

Класс `SimpleDateFormat`. Класс `DateFormatSymbols`. Класс `MessageFormat`. Класс `ChoiceFormat`. Класс `Collator`. Класс `RuleBasedCollator`.

3.1. ТЕМАТИКА ЛАБОРАТОРНЫХ ЗАНЯТИЙ

ЛАБОРАТОРНАЯ РАБОТА №1. СОЗДАНИЕ ПРОГРАММ НА ЯЗЫКЕ JAVA

ЛАБОРАТОРНАЯ РАБОТА №2. СОЗДАНИЕ СОБСТВЕННЫХ КЛАССОВ

ЛАБОРАТОРНАЯ РАБОТА №3. ГРАФИКА В JAVA. ПАКЕТ JAVA.AWT

ЛАБОРАТОРНАЯ РАБОТА №4. ОБРАБОТКА СОБЫТИЙ В JAVA

ЛАБОРАТОРНАЯ РАБОТА №5. ПАКЕТ `java.io`. СЕРИАЛИЗАЦИЯ ОБЪЕКТОВ

ЛАБОРАТОРНАЯ РАБОТА №6. ПАКЕТ `javax.swing`. ГРАФИЧЕСКИЙ ИНТЕРФЕЙС

4. ПЕРЕЧЕНЬ УЧЕБНО-МЕТОДИЧЕСКОГО ОБЕСПЕЧЕНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ

5.1 Основная литература

Соколова, В. В. Вычислительная техника и информационные технологии. Разработка мобильных приложений: учебное пособие для вузов / В. В. Соколова. — Москва: Издательство Юрайт, 2022. — 175 с. — (Высшее образование). — ISBN 978-5-9916-6525-4. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/490305>

Полуэктова, Н. Р. Разработка веб-приложений: учебное пособие для вузов / Н. Р. Полуэктова. — Москва: Издательство Юрайт, 2022. — 204 с. — (Высшее образование). — ISBN 978-5-534-13715-6. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/496682>

5.2 Дополнительная литература

Кубенский, А. А. Функциональное программирование: учебник и практикум для вузов / А. А. Кубенский. — Москва: Издательство Юрайт, 2022. — 348 с. — (Высшее образование). — ISBN 978-5-9916-9242-7. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/490015>

5.3. Ресурсы информационно-коммуникационной сети «Интернет»

Адрес Интернет ресурса	Название Интернет ресурса	Режим доступа
http://intuit.ru/	Интернет-университет информационных технологий	Свободный
http://vkit.ru/	Сайт журнала «Вестник компьютерных и информационных технологий»	Свободный
http://ru.wikipedia.org/	Свободная общедоступная мультязычная универсальная интернет-энциклопедия	Свободный

6. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ

Дисциплина «**Пакеты и комплексы прикладных программ**» изучается в течение одного семестра. При планировании и организации времени, необходимого на изучение обучающимся дисциплины, необходимо придерживаться следующих рекомендаций.

В период между сессиями студенты должны вести конспект лекций, изучать теоретический материал в соответствии с программой курса, выполнять предложенные преподавателем задания для самостоятельной работы, готовиться к сдаче зачета и экзамена, прорабатывая необходимый материал согласно перечню вопросов для подготовки к зачету и экзамену и списку рекомендованной литературы.

Выполнение лабораторных работ относится к числу обязательных видов работ. Перед выполнением работы необходимо внимательно ознакомиться с теоретическим материалом, представленным в методических указаниях к соответствующей лабораторной работе. При необходимости можно воспользоваться рекомендуемой литературой. В ходе выполнения работы необходимо руководствоваться порядком выполнения лабораторной работы и указаниями преподавателя, при этом должны соблюдаться правила техники безопасности. Результатом выполнения работы является отчет, который должен быть аккуратно оформлен и выполнен в соответствии с требованиями, приведенными в методических указаниях.

В указанное преподавателем время обучающиеся защищают отчеты. Защита проводится в виде собеседования по контрольным вопросам, приведенным в методических

указаниях. Кроме того, преподаватель может задавать дополнительные вопросы, касающиеся результатов эксперимента, выводов по результатам опытов и т.п. К промежуточной аттестации допускаются обучающиеся, выполнившие все лабораторные работы и защитившие отчеты по ним. При наличии задолженности по лабораторным работам, по согласованию с преподавателем, возможна замена работы по выполнению отчета на реферат по теме соответствующего лабораторного занятия с последующей его защитой.

В течение семестра и во время сессии основным видом подготовки являются самостоятельные занятия. Они включают в себя изучение вопросов, вынесенных на самостоятельное изучение, оформление отчетов по лабораторным работам, а так же подготовку к промежуточной аттестации

Систематическая работа в соответствии с программой дисциплины – условие успешного освоения материала.

Методические рекомендации по обучению лиц с ограниченными возможностями здоровья и инвалидов

Освоение дисциплины обучающимися с ограниченными возможностями здоровья может быть организовано как совместно с другими обучающимися, так и в отдельных группах. Предполагаются специальные условия для получения образования обучающимися с ограниченными возможностями здоровья.

Профессорско-педагогический состав знакомится с психолого-физиологическими особенностями обучающихся инвалидов и лиц с ограниченными возможностями здоровья, индивидуальными программами реабилитации инвалидов (при наличии). При необходимости осуществляется дополнительная поддержка преподавания тьюторами, психологами, социальными работниками, прошедшими подготовку ассистентами.

В соответствии с методическими рекомендациями Минобрнауки РФ (утв. 8 апреля 2014 г. N АК-44/05вн) в курсе предполагается использовать социально-активные и рефлексивные методы обучения, технологии социокультурной реабилитации с целью оказания помощи в установлении полноценных межличностных отношений с другими студентами, создании комфортного психологического климата в студенческой группе. Подбор и разработка учебных материалов производятся с учетом предоставления материала в различных формах: аудиальной, визуальной, с использованием специальных технических средств и информационных систем.

Медиа материалы также следует использовать и адаптировать с учетом индивидуальных особенностей обучения лиц с ОВЗ.

Освоение дисциплины лицами с ОВЗ осуществляется с использованием средств обучения общего и специального назначения (персонального и коллективного использования). Материально-техническое обеспечение предусматривает приспособление аудиторий к нуждам лиц с ОВЗ.

Форма проведения аттестации для студентов-инвалидов устанавливается с учетом индивидуальных психофизических особенностей. Для студентов с ОВЗ предусматривается доступная форма предоставления заданий оценочных средств, а именно:

- в печатной или электронной форме (для лиц с нарушениями опорно-двигательного аппарата);
- в печатной форме или электронной форме с увеличенным шрифтом и контрастностью (для лиц с нарушениями слуха, речи, зрения);
- методом чтения ассистентом задания вслух (для лиц с нарушениями зрения).

Студентам с инвалидностью увеличивается время на подготовку ответов на контрольные вопросы. Для таких студентов предусматривается доступная форма предоставления ответов на задания, а именно:

- письменно на бумаге или набором ответов на компьютере (для лиц с нарушениями слуха, речи);
- выбором ответа из возможных вариантов с использованием услуг ассистента (для лиц с нарушениями опорно-двигательного аппарата);

- устно (для лиц с нарушениями зрения, опорно-двигательного аппарата).

При необходимости для обучающихся с инвалидностью процедура оценивания результатов обучения может проводиться в несколько этапов.

7. ПЕРЕЧЕНЬ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ И ИНФОРМАЦИОННЫХ СПРАВОЧНЫХ СИСТЕМ

При проведении занятий по дисциплине используются следующие программные продукты:

1. Windows (для академических организациях, лицензия Microsoft Imagine (ранее MSDN AA, Dream Spark);
2. Open Office (свободное ПО)
3. NetBeans 8.2 IDE (бесплатное ПО, <https://netbeans.org/downloads/>)
4. Java SE Development Kit (свободное ПО <http://www.oracle.com/technetwork/java/javase/downloads/2133151>)
5. Доступ к электронным изданиям ЭБС ЮРАЙТ (www.biblio-online.ru).

8. НЕОБХОДИМАЯ МАТЕРИАЛЬНО-ТЕХНИЧЕСКАЯ БАЗА

Оборудование лекционных аудиторий 504, 509, 604, 609: офисная мебель, экран – 1 шт.; проектор – 1 шт.; ПК – 1шт.

Оборудование аудиторий для лабораторных занятий ауд. 609: 10 ПК с доступом в Интернет, офисная мебель.

Оборудование аудиторий для самостоятельной работы: читальный зал НТБ: 5 ПК с доступом в Интернет; ауд. 609: 10 ПК с доступом в Интернет

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ
ОБРАЗОВАТЕЛЬНАЯ АВТОНОМНАЯ НЕКОММЕРЧЕСКАЯ
ОРГАНИЗАЦИЯ ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОЛЖСКИЙ УНИВЕРСИТЕТ имени В.Н. ТАТИЩЕВА» (институт)**

Фонд оценочных средств

«Пакеты и комплексы прикладных программ»

для направления подготовки

09.03.02 «Информационные системы и технологии»

Квалификация (степень) выпускника – бакалавриат

1 Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы

Оценочные средства разработаны для оценки профессиональных компетенций: ОПК-1, ОПК-7.

1. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Планируемые результаты освоения основной профессиональной образовательной программы (ОПОП) – компетенции обучающихся определяются требованиями стандарта по направлению подготовки (специальности) и формируются в соответствии с матрицей компетенций ОПОП (Таблица 2)

Планируемые результаты обучения по дисциплине – знания, умения, навыки и (или) опыт деятельности, характеризующие этапы формирования компетенций и обеспечивающие достижение планируемых результатов освоения образовательной программы, формируются в соответствии с картами компетенций ОПОП.

Таблица 1

Планируемые результаты обучения по дисциплине

Код и наименование компетенции	Код и наименование индикатора достижения компетенции
ОПК-1. Способен применять естественнонаучные и общинженерные знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности;	ОПК-2.1. Осуществляет анализ современных информационных технологий и выбирает программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности. ОПК-2.2. Применяет современные информационных технологий и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности.
ОПК-7. Способен осуществлять выбор платформ и инструментальных программно-аппаратных средств для реализации информационных систем	ОПК-7.1. Анализирует платформы и инструментальные программно-аппаратные средства для реализации информационных систем ОПК-7.2. Использует основные платформы, технологии и инструментальные программно-аппаратные средства для реализации информационных систем ОПК-7.3. Применяет современные технологии для реализации информационных систем

2 Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания

Результаты обучения по дисциплине «Пакеты и комплексы прикладных программ» направления подготовки 09.03.02 «Информационные системы и технологии» определяются показателями и критериями оценивания сформированности компетенций на этапах их формирования представлены в табл. 2.

Таблица 2

Матрица соответствия оценочных средств запланированным результатам обучения

Компетенции	Оценочные средства	
	Текущий контроль	Промежуточный контроль
и		

	Оценочное средство 1 (лабораторное задание)	Оценочное средство 2	Экзамен
ОПК-1	ОПК-1.1. ОПК -1.2.		ОПК-1.1. ОПК -1.2.
ОПК-7	ОПК-7.1. ОПК -7.2. ОПК -7.3.		ОПК-7.1. ОПК -7.2. ОПК -7.3.

Показатели и критерии оценивания сформированности компетенций (промежуточного контроля)

На этапе промежуточной аттестации используется система оценки успеваемости обучающихся, которая позволяет преподавателю оценить уровень освоения материала обучающимися. Критерии оценивания сформированности планируемых результатов обучения (дескрипторов) представлены в карте компетенции ОПОП.

Форма оценки знаний: оценка - 5 «отлично»; 4 «хорошо»; 3 «удовлетворительно»; 2 «неудовлетворительно». Лабораторные работы, практические занятия, практика оцениваются: «зачет», «незачет». Возможно использование балльно-рейтинговой оценки.

Шкала оценивания:

«Зачет» – выставляется, если сформированность заявленных дескрипторов компетенций на 51% и более оценивается не ниже «удовлетворительно» при условии отсутствия критерия «неудовлетворительно». Выставляется, когда обучающийся показывает хорошие знания изученного учебного материала; самостоятельно, логично и последовательно излагает и интерпретирует материалы учебного курса; полностью раскрывает смысл предлагаемого вопроса; владеет основными терминами и понятиями изученного курса; показывает умение переложить теоретические знания на предполагаемый практический опыт.

«Отлично» – выставляется, если сформированность заявленных дескрипторов компетенций 85% более (в соответствии с картами компетенций ОПОП) оценивается критериями «хорошо» и «отлично», при условии отсутствия оценки «неудовлетворительно»: студент показал прочные знания основных положений фактического материала, умение самостоятельно решать конкретные практические задачи повышенной сложности, свободно использовать справочную литературу, делать обоснованные выводы из результатов анализа конкретных ситуаций;

«Хорошо» – выставляется, если сформированность заявленных дескрипторов компетенций на 61% и более (в соответствии с картами компетенций ОПОП) оценивается критериями «хорошо» и «отлично», при условии отсутствия оценки «неудовлетворительно», допускается оценка «удовлетворительно»: обучающийся показал прочные знания основных положений фактического материала, умение самостоятельно решать конкретные практические задачи, предусмотренные рабочей программой, ориентироваться в рекомендованной справочной литературе, умеет правильно оценить полученные результаты анализа конкретных ситуаций;

«Удовлетворительно» – выставляется, если сформированность заявленных дескрипторов компетенций 51% и более (в соответствии с картами компетенций ОПОП) оценивается критериями «удовлетворительно», «хорошо» и «отлично»: обучающийся показал знание основных положений фактического материала, умение получить с помощью преподавателя правильное решение конкретной практической задачи из числа предусмотренных рабочей программой, знакомство с рекомендованной справочной литературой;

«Неудовлетворительно» «Незачет» – выставляется, если сформированность заявленных дескрипторов компетенций менее чем 51% (в соответствии с картами компетенций ОПОП): при ответе обучающегося выявились существенные пробелы в знаниях основных положений фактического материала, неумение с помощью преподавателя получить правильное решение конкретной практической задачи из числа предусмотренных рабочей

программой учебной дисциплины.

Ответы и решения обучающихся оцениваются по следующим общим критериям: распознавание проблем; определение значимой информации; анализ проблем; аргументированность; использование стратегий; творческий подход; выводы; общая грамотность.

Соответствие критериев оценивания сформированности планируемых результатов обучения (дескрипторов) системам оценок представлено в табл.

Таблица 4

Интегральная оценка

Критерии	Традиционная оценка	Балльно-рейтинговая оценка
5	5	86 - 100
4	4	61-85
3	3	51-60
2 и 1	2, Незачет	0-50
5, 4, 3	Зачет	51-100

Обучающиеся обязаны сдавать все задания в сроки, установленные преподавателем. Оценка «Удовлетворительно» по дисциплине, может выставляться и при неполной сформированности компетенций в ходе освоения отдельной учебной дисциплины, если их формирование предполагается продолжить на более поздних этапах обучения, в ходе изучения других учебных дисциплин.

Показатели и критерии оценки достижений студентом запланированных результатов освоения дисциплины в ходе текущего контроля и промежуточной аттестации

Оценка, уровень	Критерии
«отлично», повышенный уровень	Студент показал прочные знания основных положений фактического материала, умение самостоятельно решать конкретные практические задачи повышенной сложности, свободно использовать справочную литературу, делать обоснованные выводы из результатов анализа конкретных ситуаций
«хорошо», пороговый уровень	Студент показал прочные знания основных положений фактического материала, умение самостоятельно решать конкретные практические задачи, предусмотренные рабочей программой, ориентироваться в рекомендованной справочной литературе, умеет правильно оценить полученные результаты анализа конкретных ситуаций
«удовлетворительно», пороговый уровень	Студент показал знание основных положений фактического материала, умение получить с помощью преподавателя правильное решение конкретной практической задачи из числа предусмотренных рабочей программой, знакомство с рекомендованной справочной литературой
«неудовлетворительно», уровень не сформирован	При ответе студента выявились существенные пробелы в знаниях основных положений фактического материала, неумение с помощью преподавателя получить правильное решение конкретной практической задачи из числа предусмотренных рабочей программой учебной дисциплины

3 Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы

3.1 Перечень вопросов для промежуточной аттестации (экзамен)

1. Цикл разработки программных средств с использованием Java.
2. Минимальный комплект для разработки программ на Java.
3. Средства разработки и системные требования.
4. Основные технические возможности NetBeans.
5. Переменные, константы, область видимости Java.
6. Примитивные типы данных.
7. Ссылочные типы данных.
8. Логические операторы.
9. Арифметические операторы.
10. Преобразования типов. Оболочечные классы.
11. Класс Math.
12. Сложные типы данных. Массивы.
13. Сложные типы данных. Строки.
14. Управляющие конструкции. Оператор If, Switch.
15. Операторы цикла.
16. Операторы перехода.
17. Основные принципы ООП. Инкапсуляция.
17. Основные принципы ООП. Наследование.
19. Основные принципы ООП. Полиморфизм.
20. Понятие объекта и работа с ним.
21. Типы отношений между классами. Агрегация, ассоциация.
22. Достоинства и недостатки ООП.
24. Класс. Описание полей класса. New.
25. Описание методов класса. Модификаторы доступа.
26. Конструкторы.
27. Ссылка This. Перегрузка методов.
27. Наследование. Super. Замещение методов (override).
28. Динамическое назначение методов.
29. Ввод вывод в Java.
30. Элементы графического интерфейса.

3.3 Оценочное средство 1 (лабораторное задания)

ЛАБОРАТОРНАЯ РАБОТА №1. СОЗДАНИЕ ПРОГРАММ НА ЯЗЫКЕ JAVA

ЛАБОРАТОРНАЯ РАБОТА №2. СОЗДАНИЕ СОБСТВЕННЫХ КЛАССОВ

ЛАБОРАТОРНАЯ РАБОТА №3. ГРАФИКА В JAVA. ПАКЕТ JAVA.AWT

ЛАБОРАТОРНАЯ РАБОТА №4. ОБРАБОТКА СОБЫТИЙ В JAVA

ЛАБОРАТОРНАЯ РАБОТА №5. ПАКЕТ java.io. СЕРИАЛИЗАЦИЯ ОБЪЕКТОВ

ЛАБОРАТОРНАЯ РАБОТА №6. ПАКЕТ javax.swing. ГРАФИЧЕСКИЙ ИНТЕРФЕЙС

**Критерии конкретного оценочного средства (согласно ПОЛОЖЕНИЮ
о промежуточной аттестации обучающихся ВУиТ
по программам высшего образования – программам бакалавриата и программам
специалитета)**

По итогам тестирования оценка знаний обучающегося производится в соответствии со

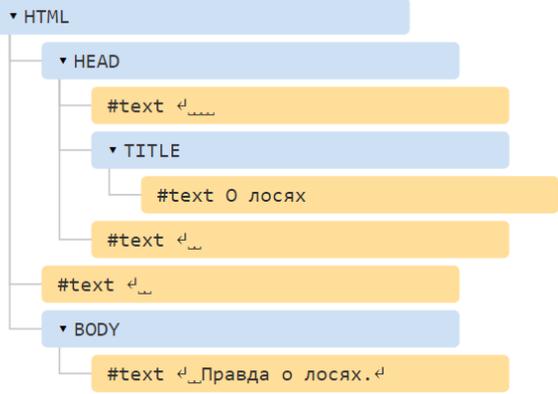
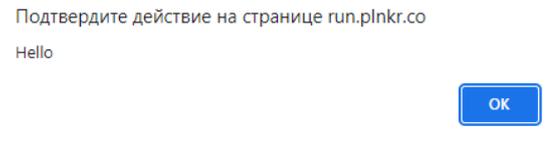
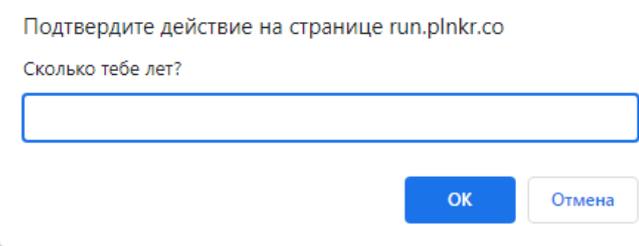
следующими критериями:

- правильных ответов 0-39% – «неудовлетворительно»/«не зачтено»;
- правильных ответов 40-59% – «удовлетворительно»/«зачтено»;
- правильных ответов 60-79% – «хорошо»/«зачтено»;
- правильных ответов 80-100% – «отлично»/«зачтено».

Промежуточный контроль

Код и наименование компетенции	Код и наименование индикатора достижения компетенции
ОПК-1. Способен применять естественнонаучные и общеинженерные знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности;	ОПК-2.1. Осуществляет анализ современных информационных технологий и выбирает программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности. ОПК-2.2. Применяет современные информационных технологий и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности.

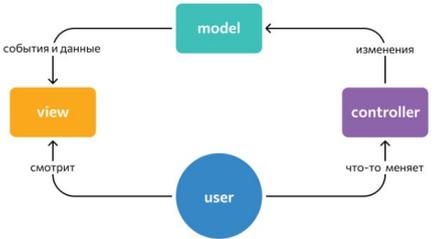
Номер задания	Содержание вопроса	Правильный ответ
1.	<p>Выберите правильный вариант ответа. Представлен код, как часть Web страницы написанный на языке _____.</p> <pre><script> "use strict"; function showMessage() { alert('Всем привет!'); } showMessage(); showMessage(); </script></pre> <p>A) PHP B) JavaScript C) DOM</p>	В
2.	<p>Выберите правильный вариант ответа. На рисунке представлена архитектура _____ представление HTML-документа в виде дерева тегов.</p>	С

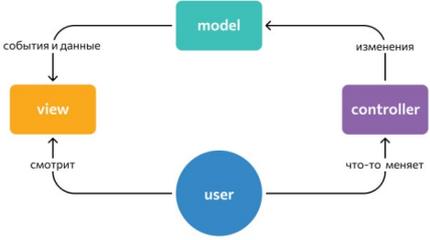
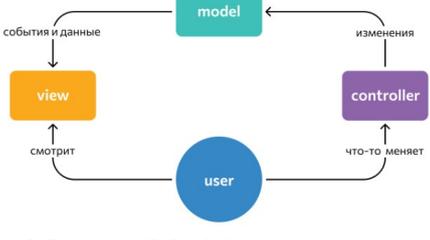
Номер задания	Содержание вопроса	Правильный ответ
	 <p>A) PHP B) JavaScript C) DOM</p>	
3.	<p>Выберите правильный вариант ответа. При создании Web приложений поставлена задача сформировать диалоговое окно средствами JavaScript.</p>  <p>Выберите функцию с помощью которой можно выполнить задачу средствами JavaScript.</p> <p>A) alert B) prompt C) confirm</p>	A
4.	<p>Выберите правильный вариант ответа. При создании Web приложений поставлена задача сформировать диалоговое окно средствами JavaScript.</p>  <p>Выберите функцию с помощью которой можно выполнить задачу средствами JavaScript.</p> <p>A) alert B) prompt C) confirm</p>	B
5.	<p>Выберите правильный вариант ответа. При создании Web приложений поставлена задача сформировать диалоговое окно средствами JavaScript.</p>	C

Номер задания	Содержание вопроса	Правильный ответ
	<p>Подтвердите действие на странице gun.plnkr.co</p> <p>Ты здесь главный?</p> <div style="text-align: center;"> <input type="button" value="OK"/> <input type="button" value="Отмена"/> </div> <p>Выберете функцию с помощью которого можно выполнить задачу средствами JavaScript.</p> <p>A) alert B) prompt C) confirm</p>	
6.	<p>Выберите правильный вариант ответа. Рассмотрите перечисление паттерны и выберите тип этих паттернов</p> <ul style="list-style-type: none"> • Singleton (Одиночка) - ограничивает создание одного экземпляра класса, обеспечивает доступ к его единственному объекту. • Factory (Фабрика) - используется, когда у нас есть суперкласс с несколькими подклассами и на основе ввода, нам нужно вернуть один из подкласса. • Abstract Factory (Абстрактная фабрика) - используем супер фабрику для создания фабрики, затем используем созданную фабрику для создания объектов. • Builder (Строитель) - используется для создания сложного объекта с использованием простых объектов. Постепенно он создает большой объект от малого и простого объекта. • Prototype (Прототип) - помогает создать дублированный объект с лучшей производительностью, вместо нового создается возвращаемый клон существующего объекта. <p>A) структурные B) порождающие C) поведенческие</p>	C
7.	<p>Выберите правильный вариант ответа. Рассмотрите перечисление паттерны и выберите тип этих паттернов</p> <ul style="list-style-type: none"> • Template Method (Шаблонный метод) - определяющий основу алгоритма и позволяющий наследникам переопределять некоторые шаги алгоритма, не изменяя его структуру в целом. • Mediator (Посредник) - предоставляет класс посредника, который обрабатывает все коммуникации между различными классами. • Chain of Responsibility (Цепочка обязанностей) - позволяет избежать жесткой 	C

Номер задания	Содержание вопроса	Правильный ответ
	<p>зависимости отправителя запроса от его получателя, при этом запрос может быть обработан несколькими объектами.</p> <ul style="list-style-type: none"> • Observer (Наблюдатель) - позволяет одним объектам следить и реагировать на события, происходящие в других объектах. <p>А) структурные В) порождающие С) поведенческие</p>	
8.	<p>Выберите правильный вариант ответа. Рассмотрите перечисление паттерны и выберите тип этих паттернов</p> <ul style="list-style-type: none"> • Strategy (Стратегия) - алгоритм стратегии может быть изменен во время выполнения программы. • Command (Команда) - интерфейс команды объявляет метод для выполнения определенного действия. • State (Состояние) - объект может изменять свое поведение в зависимости от его состояния. • Visitor (Посетитель) - используется для упрощения операций над группировками связанных объектов. • Interpreter (Интерпретатор) - определяет грамматику простого языка для проблемной области. • Iterator (Итератор) - последовательно осуществляет доступ к элементам объекта коллекции, не зная его основного представления. • Memento (Хранитель) - используется для хранения состояния объекта, позже это состояние можно восстановить. <p>А) структурные В) порождающие С) поведенческие</p>	С
9.	<p>Выберите правильный вариант ответа. Рассмотрите перечисление паттерны и выберите тип этих паттернов</p> <ul style="list-style-type: none"> • Adapter (Адаптер) - это конвертер между двумя несовместимыми объектами. Используя паттерн адаптера, мы можем объединить два несовместимых интерфейса. • Composite (Компоновщик) - использует один класс для представления древовидной структуры. • Proxy (Заместитель) - представляет функциональность другого класса. • Flyweight (Легковес) - вместо создания 	А

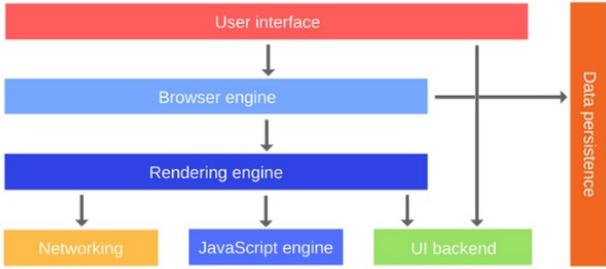
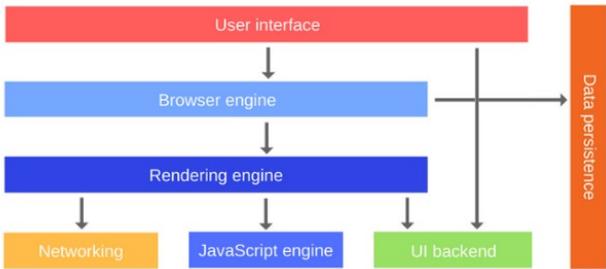
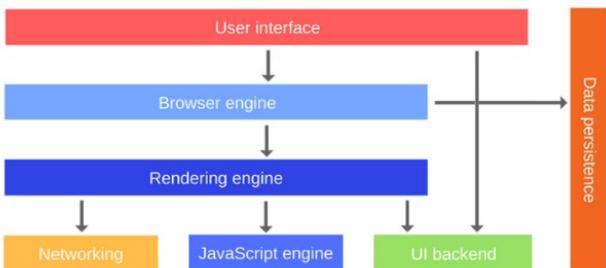
Номер задания	Содержание вопроса	Правильный ответ
	<p>большого количества похожих объектов, объекты используются повторно.</p> <ul style="list-style-type: none"> • Facade (Фасад) - беспечивает простой интерфейс для клиента, и клиент использует интерфейс для взаимодействия с системой. • Bridge (Мост) - делает конкретные классы независимыми от классов реализации интерфейса. • Decorator (Декоратор) - добавляет новые функциональные возможности существующего объекта без привязки его структуры. <p>A) структурные B) порождающие C) поведенческие</p>	
10.	<p>Выберите правильный вариант ответа. При создание Web приложений поставлена задача определить какое свойств объектно ориентированного программирования реализовано средствами JavaScript.</p> <pre>class Logger { constructor() { this.stream = null; } log(message) { if (this.stream) { this.stream.write(message); } } } const logger = new Logger(); logger.stream = process.stdout; logger.log('Here we are');</pre> <p>A) агрегация B) ассоциация C) композиция</p>	B
11.	<p>Выберите правильный вариант ответа. При создание Web приложений поставлена задача определить какое свойств объектно ориентированного программирования реализовано средствами JavaScript.</p> <pre>class Logger { constructor(stream) { this.stream = stream; } log(message) { if (this.stream) { this.stream.write(message); } } }</pre>	A

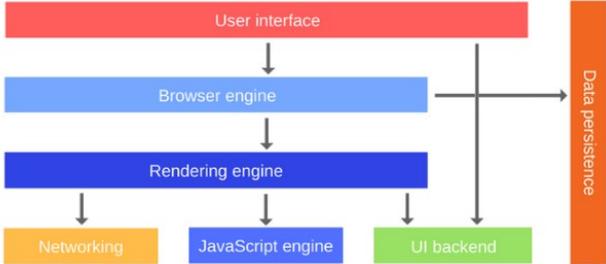
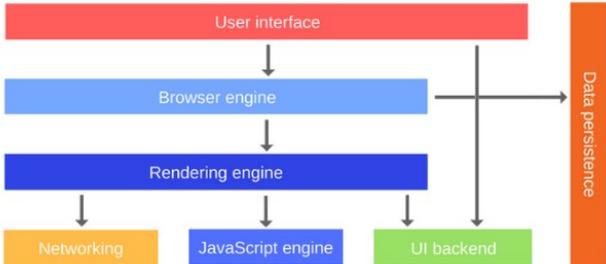
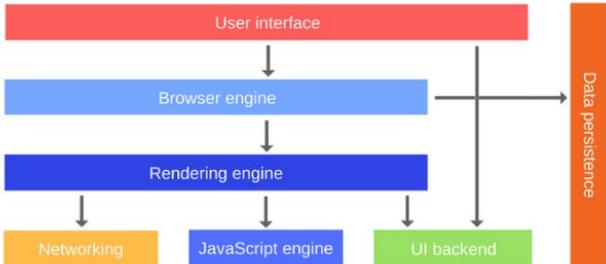
Номер задания	Содержание вопроса	Правильный ответ
	<pre> } const logger = new Logger(process.stdout); logger.log('Here we are'); </pre> <p>A) агрегация B) ассоциация C) композиция</p>	
12.	<p>Выберите правильный вариант ответа. При создание Web приложений поставлена задача определить какое свойств объектно ориентированного программирования реализовано средствами JavaScript.</p> <pre> const fs = require('fs') class Logger { constructor(name) { this.stream = fs.createWriteStream(name); } log(message) { this.stream.write(message); } } const logger = new Logger('file.log'); logger.log('Here we are'); </pre> <p>A) агрегация B) ассоциация C) композиция</p>	C
13.	<p>Выберите правильный вариант ответа. В архитектуре Web приложения MVC (Model, View, Controller) часть _____ отвечает за задачи компонента — визуализацию информации, которую он получает от модели; отображает данные на уровне пользовательского интерфейса. Например, в виде таблицы или списка.</p>  <pre> graph TD user((user)) -- "СМОТРИТ" --> view[view] view -- "события и данные" --> model[model] model -- "изменения" --> controller[controller] controller -- "что-то меняет" --> user </pre> <p>A) Модель (Model) B) Представление (View) C) Контроллер (Controller)</p>	B
14.	<p>Выберите правильный вариант ответа. В архитектуре Web приложения MVC (Model, View, Controller) часть _____ отвечает за основную логику приложения; за данные, методы работы с ними и структуру программы;</p>	A

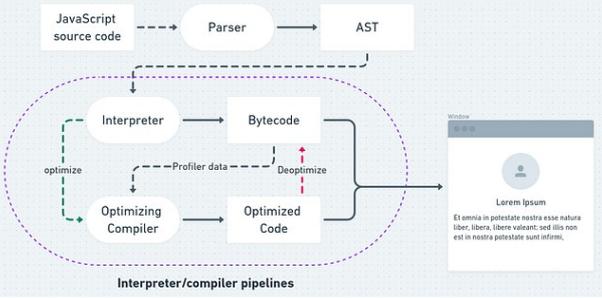
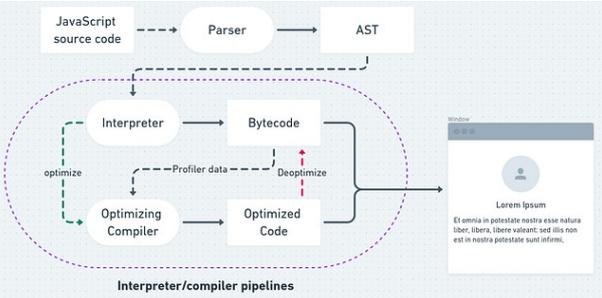
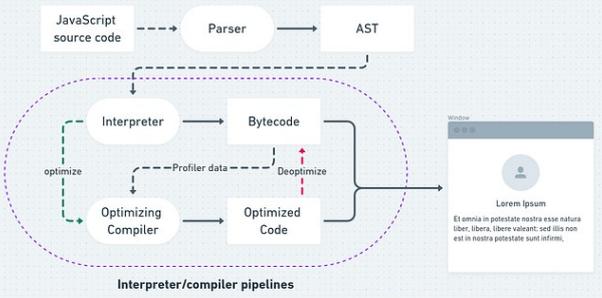
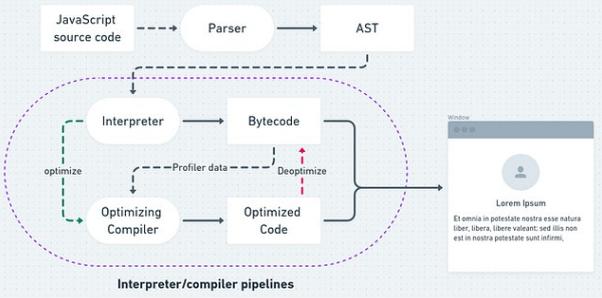
Номер задания	Содержание вопроса	Правильный ответ
	<p>реагирует на команды из контроллера и выдает информацию и/или изменяет свое состояние; передает данные в представление.</p>  <p>A) Модель (Model) B) Представление (View) C) Контроллер (Controller)</p>	
15.	<p>Выберите правильный вариант ответа. В архитектуре Web приложения MVC (Model, View, Controller) часть _____ обеспечивает взаимодействие с системой: обрабатывает действия пользователя, проверяет полученную информацию и передает ее модели; определяет, как приложение будет реагировать на действия пользователя; может отвечать за фильтрацию данных и авторизацию.</p>  <p>A) Модель (Model) B) Представление (View) C) Контроллер (Controller)</p>	С
16.	<p>Дайте описание понятию AJAX (Asynchronous JavaScript and XML).</p>	<p>AJAX — это комплекс методов для создания веб-сайтов и веб-приложений с динамически загружаемым контентом без создания новой записи данных.-</p>
17.	<p>Дайте описание части скрипта JavaScript Web приложения <code>if (year == 2015) alert('Вы правы!');</code></p>	<p>Инструкция if вычисляет условие в скобках и, если результат true, то выполняет блок кода, в данном случае alert('Вы правы!')-</p>
18.	<p>В каких случаях в скриптах JavaScript Web приложений в условном операторе используется <u>блок «else»</u></p>	<p>Инструкция if может содержать необязательный блок «else» («иначе»). Он выполняется, когда условие ложно.-</p>
19.	<p>Как может использоваться оператор ? в скриптах JavaScript Web приложений</p>	<p>Например «условный» оператор «вопросительный знак» позволяет вернуть значение, которое зависит от более чем одного условия.-</p>

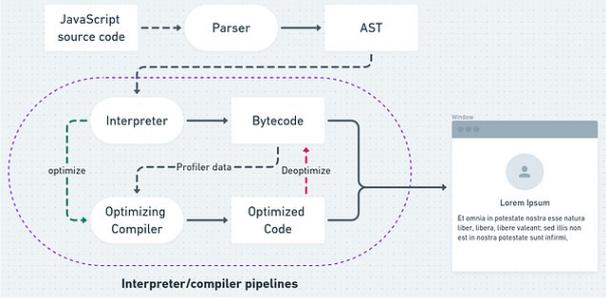
Номер задания	Содержание вопроса	Правильный ответ
20.	<p>Определите количество диалоговых окон по части скрипта JavaScript как части Web приложения.</p> <pre>let i = 0; while (i < 3) { alert(i); i++; }</pre>	<p>Для пользователя отобразится три диалоговых окна alert со значениями 0,1,2</p>
21.	<p>Определите количество диалоговых окон по части скрипта JavaScript как части Web приложения.</p> <pre>let i = 0; do { alert(i); i++; } while (i < 3);</pre>	<p>Для пользователя отобразится три диалоговых окна alert со значениями 0,1,2.</p>
22.	<p>Определите количество диалоговых окон по части скрипта JavaScript как части Web приложения.</p> <pre>for (let i = 0; i < 3; i++) { // выведет 0, затем 1, затем 2 alert(i); }</pre>	<p>Для пользователя отобразится три диалоговых окна alert со значениями 0,1,2.</p>
23.	<p>Поставлена задача: определить действие директивы break в данной части скрипта JavaScript как части Web приложения.</p> <pre>let sum = 0; while (true) { let value = +prompt("Введите число", ''); }; if (!value) break; // (*) sum += value; } alert('Сумма: ' + sum);</pre>	<p>Директива break в строке (*) полностью прекращает выполнение цикла и передаёт управление на строку за его телом, то есть на alert.</p>
24.	<p>Дана часть скрипта JavaScript как части Web приложения. Какое последнее значение выведет этот код?</p> <pre>let i = 3; while (i) { alert(i--);</pre>	<p>Последнее значение диалогового окна будет 1.</p>

Номер задания	Содержание вопроса	Правильный ответ
	}	
25.	<p>Дана часть скрипта JavaScript как части Web приложения. Определите вид переменной объявленной в функции</p> <pre>function showMessage() { let message = "Привет, я JavaScript!"; alert(message); } showMessage();</pre>	В функции объявлена локальная переменная
26.	Поставлена задача: организовать взаимодействие с посетителями Web приложения. Приведите варианты организации взаимодействия через скрипты JavaScript.	<p>В JavaScript возможно организовать взаимодействие с посетителями Web приложения через:</p> <ol style="list-style-type: none"> 1. Модальные окна 2. Формы
27.	Охарактеризуйте системные методы alert(), confirm() и prompt() как блокирующие.	Методы alert(), confirm() и prompt() являются блокирующими, т. е. они не возвращают управление документом, пока пользователь не отреагирует на запрос диалогового окна.
28.	Дайте описание понятию «замыкание» в функциях JavaScript при проектировании Web приложений.	замыкание – это область действия, которая создается при объявлении функции и позволяет ей получать доступ к внешними (свободным) по отношению к ней переменными.
29.	Для каких целей используют «замыкание» в функциях JavaScript при проектировании Web приложений.	«Замыкание» в функциях JavaScript используется например при сохранении переменных при полном выполнении функции.
30.	Поставлена задача: перечислить инструменты концепции MyBatis JavaScript при доступе Web приложения к БД.	<p>Различные инструменты MyBatis JavaScript, например доступа приложения к БД:</p> <ul style="list-style-type: none"> - ORM-фреймворки — преобразующие информацию, хранящуюся в базе данных, в объекты с помощью объектно-ориентированных языков программирования; - СУБД — системы управления базами данных, представляющие собой специализированное программное обеспечение, работа которого зависит от типа БД.
31.	Кратко опишите как MyBatis JavaScript при доступе Web приложения к БД реализует Динамический SQL.	Динамический SQL — функция, которая позволяет точно структурировать сложный SQL-оператор, тем самым избежав ошибок в его синтаксисе, которые часто возникают при ручном написании.

Номер задания	Содержание вопроса	Правильный ответ
32.	Кратко опишите понятие Открытый исходный код.	Открытый исходный код — платформа распространяется по бесплатной лицензии, ее можно дорабатывать под конкретные задачи, что делает ее более гибкой в сравнении с конкурентами.
33.	Опишите место Node.js (Node) в системе проектирования Web приложений на JavaScript.	Node или Node.js — программная платформа, основанная на движке V8 (компилирующем JavaScript в машинный код), превращающая JavaScript из узкоспециализированного языка в язык общего назначения.
34.	<p>На рисунке показаны основные компоненты браузера и их влияние на процесс обработки веб-страниц. Кратко опишите компонент Пользовательский интерфейс (User Interface).</p> 	Пользовательский интерфейс (User Interface). Этот компонент браузера включает в себя адресную строку, кнопки «Вперёд» и «Назад», команды для работы с закладками, и так далее. В целом, это всё то, что выводит на экран браузер — за исключением той области его окна, где находится отображаемая им веб-страница.
35.	<p>На рисунке показаны основные компоненты браузера и их влияние на процесс обработки веб-страниц. Кратко опишите компонент Движок браузера (Browser Engine).</p> 	Движок браузера (Browser Engine). Он занимается поддержкой взаимодействия между пользовательским интерфейсом и движком рендеринга.
36.	<p>На рисунке показаны основные компоненты браузера и их влияние на процесс обработки веб-страниц. Кратко опишите компонент Сетевая подсистема (Networking).</p> 	Сетевая подсистема (Networking). Эта подсистема ответственна за сетевое взаимодействие браузера с внешним миром, в частности, например, её средствами выполняются XMLHttpRequest-запросы. Она поддерживает платформенно-независимый интерфейс, за которым скрываются конкретные реализации различных сетевых механизмов, специфичные для различных платформ.
37.	На рисунке показаны основные компоненты браузера и их влияние на процесс обработки веб-страниц. Кратко опишите компонент	JavaScript-движок (JavaScript Engine). Именно здесь осуществляется выполнение JS-кода.

Номер задания	Содержание вопроса	Правильный ответ
	<p>JavaScript-движок (JavaScript Engine).</p> 	
38.	<p>На рисунке показаны основные компоненты браузера и их влияние на процесс обработки веб-страниц. Кратко опишите компонент Подсистема поддержки пользовательского интерфейса (UI Backend).</p> 	<p>Подсистема поддержки пользовательского интерфейса (UI Backend). Эта подсистема отвечает за вывод базовых компонентов интерфейса, таких, как окна и элементы управления, вроде чекбоксов. Здесь браузеру предоставляется универсальный интерфейс, не зависящий от платформы, на которой он работает, а в основе этой подсистемы лежат возможности формирования элементов пользовательского интерфейса, предоставляемые конкретной операционной системой.</p>
39.	<p>На рисунке показаны основные компоненты браузера и их влияние на процесс обработки веб-страниц. Кратко опишите компонент Подсистема постоянного хранения данных (Data Persistence).</p> 	<p>Подсистема постоянного хранения данных (Data Persistence). Если приложению нужны возможности локального хранения данных, оно может пользоваться различными механизмами, предоставляемыми этой подсистемой.</p>
40.	<p>Кратко опишите этапы Процесс рендеринга JavaScript Web приложений.</p>	<p>Основные этапы процесса рендеринга:</p> <ol style="list-style-type: none"> 1. Обработка HTML для создания дерева DOM. 2. Создание дерева рендеринга. 3. Расчёт параметров расположения элементов дерева рендеринга на экране, формирование макета страницы. 4. Визуализация (отрисовка) дерева рендеринга.
41.	<p>На рисунке отображена структура движка JavaScript Web приложений (на примере V8). Дайте описание работы компонента Парсер.</p>	<p>Парсер выполняет действия лексического анализатора. Это первое, что происходит с файлом JavaScript при попадании в движок. Код разбивается на части, называемые токенами, для выявления их</p>

Номер задания	Содержание вопроса	Правильный ответ
	 <p>The diagram illustrates the JavaScript engine pipeline. It starts with 'JavaScript source code' which goes to a 'Parser' to create an 'AST'. The 'AST' then goes to an 'Interpreter' to produce 'Bytecode'. The 'Bytecode' is then processed by an 'Optimizing Compiler' to create 'Optimized Code'. The 'Optimized Code' is then executed by the 'Interpreter' to produce the final output. A feedback loop labeled 'Profiler data' goes from the 'Interpreter' back to the 'Optimizing Compiler', which then feeds back into the 'Interpreter' via an 'optimize' step. A 'Deoptimize' step also feeds back from the 'Optimized Code' to the 'Bytecode'. The final output is shown as a browser window with 'Lorem Ipsum' text.</p>	<p>назначения, после чего мы узнаём, что код пытается сделать.</p>
42.	<p>На рисунке отображена структура движка JavaScript Web приложений (на примере V8). Дайте описание работы компонента Абстрактное синтаксическое дерево (AST).</p> 	<p>На основе токенов создаётся Абстрактное синтаксическое дерево (AST). Синтаксическое дерево — это древовидное представление синтаксической структуры кода JavaScript, и мы можем использовать этот инструмент для анализа преобразования кода AST.</p>
43.	<p>На рисунке отображена структура движка JavaScript Web приложений (на примере V8). Дайте описание работы компонента Интерпретатор.</p> 	<p>Интерпретатор читает файлы JavaScript построчно и преобразовывает их на ходу (во время работы программы, не прерывая её выполнение). На основе сгенерированного кода AST интерпретатор начинает быстро создавать байт-код. Никаких оптимизаций здесь не выполняется, так что байт-код этот неоптимизированный.</p>
44.	<p>На рисунке отображена структура движка JavaScript Web приложений (на примере V8). Дайте описание работы компонента Profiler & Compiler.</p> 	<p>Профайлер отвечает за проверку кода. Он вызывает специальное средство контроля, которое отслеживает код и наблюдает за ходом его выполнения, обращая внимание на то, как можно оптимизировать код. Выдаёт, например, информацию о том, сколько раз код запускался, какие типы используются и т.д.</p>

Номер задания	Содержание вопроса	Правильный ответ
45.	<p>На рисунке отображена структура движка JavaScript Web приложений (на примере V8). Дайте описание работы компонента Оптимизирующий компилятор.</p> 	<p>Задача оптимизирующего компилятора — определить, что делает программа, подлежащая оптимизации, и создать из неё оптимизированную программу, выполняющую всё то же самое, только быстрее</p>
46.	<p>Приведите пример концепции позволяющую упрощать чтение и поддержание кода JavaScript Web приложений.</p>	<p>Вертикальное форматирование кода играет важную роль и упрощает чтение и поддержание кода.</p>
47.	<p>Кратко опишите Стек вызовов (Call Stack) движка JavaScript Web приложений.</p>	<p>Стек вызовов (Call Stack) — то место, куда в процессе выполнения кода попадают так называемые стековые кадры.</p>
48.	<p>Приведите описание понятию Промисы кода JavaScript Web приложений.</p>	<p>Промисы это способ работы с асинхронным кодом. В коде Промис это объект, который представляет собой асинхронный таск (возвращаемый тип), который должен завершиться.</p>
49.	<p>Кратко опишите цель концепции встроенного кэширования JavaScript Web приложений.</p>	<p>Цель встроенного кэширования в том, чтобы ускорить привязку метода времени выполнения. Происходит это за счёт запоминания результатов поиска предыдущего метода непосредственно в месте вызова.</p>
50.	<p>Кратко опишите причину существования скрытых классов JavaScript Web приложений.</p>	<p>Основная причина существования скрытых классов — концепция встроенных кэшей.</p>

Код и наименование компетенции	Код и наименование индикатора достижения компетенции
ОПК-7. Способен осуществлять выбор платформ и инструментальных программно-аппаратных средств для реализации информационных систем	ОПК-7.1. Анализирует платформы и инструментальные программно-аппаратные средства для реализации информационных систем ОПК-7.2. Использует основные платформы, технологии и инструментальные программно-аппаратные средства для реализации информационных систем ОПК-7.3. Применяет современные технологии для реализации информационных систем

Номер задания	Содержание вопроса	Правильный ответ
1.	Выберите правильный вариант ответа.	A

Номер задания	Содержание вопроса	Правильный ответ
	<p>Вид представленной функции при JavaScript</p> <pre>(function ()){})();</pre> <p>A) Самовызывающая B) Конструктор</p>	
2.	<p>Выберите правильный вариант ответа.</p> <p>Вид представленной функции при JavaScript</p> <p>1.</p> <pre>function Constructor(){}</pre> <pre>var newObj = new Constructor;</pre> <p>A) Самовызывающая B) Конструктор</p>	В
3.	<p>Выберите правильный вариант ответа.</p> <p>Вид представленной функции при JavaScript</p> <pre>function foo(){}</pre> <p>A) Выражения B) Декларативные C) Созданные конструктором function</p>	В
4.	<p>Выберите правильный вариант ответа.</p> <p>Вид представленной функции при JavaScript</p> <pre>var foo = new Function();</pre> <p>A) Выражения B) Декларативные C) Созданные конструктором function</p>	С
5.	<p>Выберите правильный вариант ответа.</p> <p>Вид представленной функции при JavaScript</p> <pre>var foo = function(){};</pre> <p>A) Выражения B) Декларативные C) Созданные конструктором function</p>	А
6.	<p>Выберите правильный вариант ответа.</p>	С

Номер задания	Содержание вопроса	Правильный ответ
	<p>При проектировании Web приложения на JavaScript использовался шаблон. Определите по предложенному коду и рисунку с диалоговым окном определите отображение браузера</p> <pre data-bbox="284 443 944 1153"> <!doctype html> <body> <script src="https://cdnjs.cloudflare.com/ajax/libs/knockout/3.3.0/knockout-min.js"></script> Поменяйте имя: <input data-bind="value: name, valueUpdate: 'input'"> <hr> <h1>Здравствуй, !</h1> <script> var user = { name: ko.observable("СТРАНА") }; ko.applyBindings(user, document.body); </script> </body> </pre>  <p>А) Здравствуй, СТРАНА! Б) Здравствуй, Тольятти! С) Здравствуй, город!</p>	
7.	<p>Выберите правильный вариант ответа.</p> <p>Выбрать блок обработки ошибка на JavaScript при проектировании Web приложения</p> <p>A) try..catch B) do..while C) if else</p>	А
8.	<p>Выберите правильный вариант ответа.</p> <p>Поставлена задача восстановления состояния программы (то есть, некоторых её объектов) после запуска. С помощью какой технологии это можно реализовать?</p> <p>А) Пакетизация В) Сериализация C) Синхронизация</p>	В
9.	<p>Выберите правильный вариант ответа.</p> <p>Выберите формат для представления значений</p>	А

Номер задания	Содержание вопроса	Правильный ответ
	<p>и объектов при сериализация, задокументированный в стандарте RFC 4627, при работе JavaScript.</p> <p>A) JSON (JavaScript Object Notation) B) UDP (User Datagram Protocol)</p>	
10.	<p>Выберите правильный вариант ответа.</p> <p>При создании формы необходимо средствами JavaScript реализовать представленный ниже объект</p> <p></p> <p>A) Мужской <code><INPUT TYPE="checkbox" NAME="ch"></code></p> <p>B) <code><INPUT TYPE="RADIO" NAME="Sex" VALUE="Man" CHECKED></code> Мужской</p> <p>C) <code><INPUT TYPE="button" NAME="bt" VALUE="Мужской"></code></p>	В
11.	<p>Выберите правильный вариант ответа.</p> <p>При создании формы необходимо средствами JavaScript реализовать представленный ниже объект</p> <p></p> <p>A) Мужской <code><INPUT TYPE="checkbox" NAME="ch"></code></p> <p>B) <code><INPUT TYPE="RADIO" NAME="Sex" VALUE="Man" CHECKED></code> Щелкни здесь!</p> <p>C) <code><INPUT TYPE="button" NAME="bt" VALUE="Щелкни здесь!" onClick="btnClick();"></code></p>	С
12.	<p>Выберите правильный вариант ответа.</p> <p>При работе с окном пользователя необходимо отслеживать координаты положения мыши. Выберите свойства события мыши для данной задачи.</p> <p>A) clientX/clientY B) pageX/pageY</p>	А

Номер задания	Содержание вопроса	Правильный ответ
	С) e.metaKey / e.ctrlKey	
13.	Выберите правильный вариант ответа. Ключевое слово свойства-аксессуара (accessor properties) для записи/присваивания значения объекта в JavaScript. А) геттеры (getters) ключевое слово get В) сеттеры (setters) ключевое слово set	В
14.	Выберите правильный вариант ответа. Ключевое слово свойства-аксессуара (accessor properties) для чтения/получения значение объекта в JavaScript. А) геттеры (getters) ключевое слово get В) сеттеры (setters) ключевое слово set	А
15.	Неконфигурируемое свойство в JavaScript.не может быть удалено, его атрибуты не могут быть изменены. А) writable: true В) prop1:descriptor1 С) configurable:false	С
16.	Задача: Для реализации информационных систем необходимо организовать Web интерфейс. Перечислите возможные программные платформ JavaScript инструменты для сборки и автоматизации.	Для фронтенда Web систем для JavaScript для сборки и автоматизации можно использовать: Webpack Grunt Gulp Browserify Brunch.io Yeoman Visual Studio Code Brackets и т.д.
17.	Дайте описание используемым при проектировании Web приложений паттернам (design pattern).	В сфере разработки программного обеспечения паттерн проектирования — это повторяемая архитектурная конструкция, представляющая собой решение проблемы проектирования в рамках некоторого часто возникающего контекста..
18.	Задача: Для реализации информационных систем необходимо организовать Web интерфейс. Перечислите возможные программные платформ JavaScript инструменты для документирования кода.	Для фронтенда Web систем для JavaScript для документирования кода можно использовать: Swagger JSDoc jGrouseDoc (jGD) YUIDoc Docco и т.д.
19.	Задача: Для реализации информационных систем необходимо организовать Web интерфейс. Перечислите возможные программные платформ JavaScript	Для фронтенда Web систем для JavaScript тестирования кода можно использовать: Jasmine Mocha

Номер задания	Содержание вопроса	Правильный ответ
	инструменты тестирования.	PhantomJS Protractor и т.д.
20.	Задача: Для реализации информационных систем необходимо организовать Web интерфейс. Перечислите возможные программные платформ JavaScript инструменты отладки.	Для фронтенда Web систем для JavaScript отладки кода можно использовать: Chrome Dev Tools JavaScript Debugger ng-inspector Augury и т.д
21.	Задача: Для реализации информационных систем необходимо организовать Web интерфейс. Перечислите возможные программные платформ JavaScript инструменты безопасности.	Для фронтенда Web систем для JavaScript безопасности кода можно использовать: Snyk Node Security Project RetireJS Gemnasium OSSIndex и т.д
22.	Задача: Для реализации информационных систем необходимо организовать Web интерфейс. Перечислите возможные программные платформы JavaScript инструменты аналитики и оптимизации кода.	Для фронтенда Web систем для JavaScript инструменты аналитики и оптимизации кода можно использовать: JSLint JSHint ESLint Flow и т.д
23.	Задача: Для реализации информационных систем необходимо организовать Web интерфейс. Перечислите характеристики выбора редактора для JavaScript.	При выборе редактора для JavaScript обращают внимание на следующие характеристики: 1. Работа с кодом 2. Расширяемость 3. Актуальность и популярность 4. Скорость работы 5. Внешний вид 6. Цена
24.	Перечислите достоинства паттернов JavaScript.	Достоинства использования паттернов Выверенное решение Легко используются повторно Формируют структуру
25.	Дайте описание части программного кода на JavaScript. <code>rabbit.__proto__ = animal</code>	При установке свойства <code>rabbit.__proto__ = animal</code> говорят, что объект <code>animal</code> будет «прототипом» <code>rabbit</code> .
26.	Перечислите основные свойства объектно ориентированного программирования JavaScript.	Свойства объектно ориентированного программирования JavaScript: 1. Инкапсуляция 2. Наследование 3. Абстракция 4. Полиморфизм
27.	Дайте описание понятию класс объектно ориентированного программирования	Классом в объектно-ориентированной разработке называют

Номер задания	Содержание вопроса	Правильный ответ
	JavaScript.	шаблон/программный код, предназначенный для создания объектов и методов.
28.	Описание способа создания нового объекта JavaScript.	Создать функцию-конструктор и сделать экземпляр объекта можно с помощью функции инициализатора объекта и оператора new.
29.	Дайте описание понятию Колбэк JavaScript.	Колбэк-функция (или обратный вызов) - это функция, переданная в другую функцию в качестве аргумента, которая затем вызывается по завершению какого-либо действия.
30.	Приведите примеры операционных систем для которых возможно написание Web приложения на платформе Node.js	Node.js позволяет написать программу для разных ОС: Linux , OS X и Windows, может использоваться для создания API.
31.	Приведите примеры компонентов для которых возможно написание Web приложения на платформе Node.js	Платформу используют для написания fronted-разработки , backend-разработки и другие.
32.	Дайте краткое описание понятию Сериализация.	Сериализация — это преобразование объекта или дерева объектов в какой-либо формат с тем, чтобы потом эти объекты можно было восстановить из этого формата. Используется, например, для сохранения состояния программы (то есть, некоторых её объектов) между запусками. Или для передачи данных между различными экземплярами программы (или различными программами), например, по сети.
33.	Перечислите методы Сериализации в JavaScript	JavaScript предоставляет методы: 1.JSON.stringify для преобразования объектов в JSON. 2.JSON.parse для преобразования JSON обратно в объект.
34.	Поставлена задача определить по программному коду результат нажатия на кнопку <pre><input id="elem" type="button" value="Нажми меня!"> <script> elem.onclick = function() { alert('Спасибо'); }; </script></pre>	После нажатия на кнопку появится диалоговое окно (alert) с надписью Спасибо .
35.	Опишите понятие Обработчик события.	Событию можно назначить обработчик, то есть функцию, которая сработает, как только событие произошло. Мы можем назначить обработчиком не только

Номер задания	Содержание вопроса	Правильный ответ
		функцию, но и объект.
36.	<p>Поставлена задача определить по программному коду результат нажатия на кнопку</p> <pre data-bbox="284 383 941 510"><body> <input type="button" onclick="this.hidden=true" value="Нажми"> </body></pre>	В соответствии с обработчиком события "this.hidden=true" кнопка после нажатия исчезнет с экрана.
37.	Является ли CGI языком программирования Web приложений.	CGI – это не язык программирования и не какой-то специализированный софт. CGI – это стандарт по которому передается информация между внешней программой и ее серверной частью. По CGI-интерфейсом может скрываться любая серверная программа, написанная на любом серверном языке программирования.
38.	Опишите понятие Сценарий в Web приложениях.	Сценарии (scripts) являются еще одним расширением языка HTML. Сценарии представляют собой небольшие программы, текст которых включается в голову HTML-файла с помощью специальных тегов.
39.	Почему пользователь должен иметь возможность (при настройке операционной системы на своем компьютере) отказаться от сценариев Web страниц?	Сценарии JavaScript загружаются с интернет-сайта на компьютер пользователя и выполняются на нем. Этот факт порождает потенциальную угрозу для компьютера пользователя!
40.	Как связаны файлы «cookie» Web страниц и конфиденциальность данных пользователя.	Файлы cookie – это небольшие фрагменты текста, передаваемые в браузер с сайта, который открывает пользователь. С их помощью сайт запоминает информацию о посещениях пользователя. Защиту конфиденциальности данных пользователя берет на себя браузер-система.
41.	Дайте описание понятию Фреймворк (с англ. framework — «каркас, структура») в Web системе.	Фреймворк— заготовка, готовая модель в IT для быстрой разработки, на основе которой можно дописать собственный код. Он задает структуру, определяет правила и предоставляет необходимый набор инструментов для создания проекта. В основном фреймворки используются в Web-разработке JavaScript.
42.	Перечислите достоинства использования фреймворков JavaScript.	Особенности Web -фреймворков JavaScript: Универсальность Эффективность и простота Надежность

Номер задания	Содержание вопроса	Правильный ответ
		Безопасность Адаптивность
43.	При реализации Web системы на основе модулей требуется динамическая подгрузка модулей. Приведите примеры библиотеки для динамической подгрузки модулей.	Библиотеки для динамической подгрузки модулей: AMD – одна из самых старых модульных систем, изначально реализована библиотекой require.js . CommonJS – модульная система, созданная для сервера Node.js. UMD – ещё одна модульная система, предлагается как универсальная, совместима с AMD и CommonJS.
44.	Дайте описание понятию модуль в JavaScript для Web систем.	Модуль – это просто файл. Один скрипт – это один модуль. Модули не работают локально, только через HTTP(s). Модули работают как в браузере, так и в серверном JavaScript.
45.	Для чего используется Инструкция export в JavaScript для Web систем?	Инструкция export используется для экспорта функций, объектов или примитивов из файла (или модуля).
46.	Для чего используется Инструкция import в JavaScript для Web систем?	Инструкция import используется для импорта ссылок на значения, экспортированные из внешнего модуля.
47.	В JavaScript при проектировании Web систем необходимо расширить один класс другим классом. Каким приемом объектно ориентированного программирования это реализуемо?	Наследование классов – это способ расширения одного класса другим классом.
48.	В JavaScript при проектировании Web систем необходимо что бы данные объекта не были напрямую доступны. Каким приемом объектно ориентированного программирования это реализуемо?	Инкапсуляция включает в себя идею о том, что данные объекта не должны быть напрямую доступны. Нужно вызывать методы вместо прямого доступа к данным. Инкапсуляция позволяет нам скрывать/показывать свойства функций.
49.	В JavaScript при проектировании Web систем необходимо скрыть детали реализации и показать пользователям только функциональность. Каким приемом объектно ориентированного программирования это реализуемо?	Абстракция - это способ создания простой модели, которая содержит только важные свойства с точки зрения контекста приложения, из более сложной модели. Иными словами - это способ скрыть детали реализации и показать пользователям только функциональность.
50.	В JavaScript при проектировании Web систем необходимо проектировать объекты таким образом, чтобы они могли совместно использовать или переопределять любое поведение с конкретными предоставленными объектами. Каким приемом объектно ориентированного программирования это реализуемо?	Полиморфизмом помогает проектировать объекты таким образом, чтобы они могли совместно использовать или переопределять любое поведение с конкретными предоставленными объектами.