



КГЭУ

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«КАЗАНСКИЙ ГОСУДАРСТВЕННЫЙ ЭНЕРГЕТИЧЕСКИЙ УНИВЕРСИТЕТ»  
(ФГБОУ ВО «КГЭУ»)

УТВЕРЖДАЮ

Директор Института цифровых  
технологий и экономики

\_\_\_\_\_ Э.И. Беляев  
«28» ноября 2023 г.

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ**

**Б1.О.15 Современные языки программирования**

*(Код и наименование дисциплины в соответствии с РУП)*

Направление подготовки

09.03.03 Прикладная информатика

*(Код и наименование направления подготовки)*

Квалификация

Бакалавр

*(Бакалавр / Магистр)*

г. Казань, 2023

Программу разработал(и):

Наименование кафедры	Должность, уч.степень, уч.звание	ФИО разработчика
УРФУ	Доцент, к.т.н., доцент	Обабков И.Н.
УРФУ	Ст. преподаватель	Шадрин Д.Б.
КГЭУ ИТИС	Доцент, к. ф.-м.н.	Андреев А.О.

Согласование	Наименование подразделения	Дата	№ протокола	Подпись
Одобрена	ИТИС	27.11.2023	11	И.о. зав.каф.,к.ф.-м.н., доц. Соловьев С. А.
Согласована	Учебно-методический совет института ИЦТЭ	27.11.2023	3	Директор, к.т.н., доц. Беляев Э.И.
Одобрена	Ученый совет института ИЦТЭ	28.11.2023	3	Директор, к.т.н., доц. Беляев Э.И.

## 1. Цель, задачи и планируемые результаты обучения по дисциплине

Целью освоения дисциплины «Современные языки программирования» является теоретическая и практическая подготовка студентов в области разработки web-страниц. Знания, полученные в результате освоения дисциплины, помогут при разработке интерактивных элементов в такой степени, чтобы студенты могли самостоятельно выбирать средства реализации, находить необходимые программные и технологические решения для практически важных системных и предметно-ориентированных задач, что в итоге позволит освоить такой современный язык программирования, как JavaScript.

Задачами дисциплины являются: – приобретение студентами знаний о работе с HTML/CSS; – ознакомление с современным языком программирования JavaScript; – приобретение практических навыков по работе с браузером (документы, события, интерфейсы)

### Компетенции и индикаторы, формируемые у обучающихся:

Код и наименование компетенции	Код и наименование индикатора
ПК-2 Способен разрабатывать и тестировать программные компоненты решения задач в системах искусственного интеллекта	ПК-2.1 Разрабатывает приложения систем искусственного интеллекта
	ПК-2.2. Проводит тестирование систем искусственного интеллекта
ОПК-7 Способен разрабатывать алгоритмы и программы, пригодные для практического применения;	ОПК-7.1 Способен разрабатывать алгоритмы для решения практических задач
	ОПК-7.2 Способен разрабатывать компьютерные программы, пригодные для практического применения

## 2. Место дисциплины в структуре ОП

Предшествующие дисциплины (модули), практики, НИР, др. - 1. Информационные технологии, 2. Объектно-ориентированное программирование, 3. Проектирование информационных систем.

Последующие дисциплины и практики: 1. Базовая архитектура программного обеспечения, 2. Теория и практика программной инженерии, 3. Виртуализация и облачные технологии

## 3. Структура и содержание дисциплины

### 3.1. Структура дисциплины

Для очной формы обучения

Вид учебной работы	Всего ЗЕ	Всего часов	Семестр
			5
ОБЩАЯ ТРУДОЕМКОСТЬ ДИСЦИПЛИНЫ	3	108	108

КОНТАКТНАЯ РАБОТА *	-	24	24
АУДИТОРНАЯ РАБОТА	1,17	42	42
Лекции	0,5	18	18
Практические (семинарские) занятия		0	0
Лабораторные работы	0,67	24	24
САМОСТОЯТЕЛЬНАЯ РАБОТА ОБУЧАЮЩЕГОСЯ	1,83	66	66
Проработка учебного материала	0,83	30	30
Курсовой проект	-	-	-
Курсовая работа	-	-	-
Подготовка к промежуточной аттестации	1	36	36
Промежуточная аттестация:			3

### 3.2. Содержание дисциплины, структурированное по разделам и видам занятий

Разделы дисциплины	Всего часов	Распределение трудоемкости по видам учебной работы				Формы и вид контроля	Индексы индикаторов формируемых компетенций
		лекции	лаб. раб.	пр. зан.	сам. раб.		
Раздел 1. Основы языка.	12	2	4		6	ТК1	ПК-2.1, ОПК-7.1
Раздел 2. Объекты.	14	4	4		6	ТК2	ПК-2.2, ОПК-7.2
Раздел 3. Прототипы, наследование, классы	18	4	8		6	ТК3	ПК-2.2, ОПК-7.1
Раздел 4. Регулярные выражения как методы строк.	14	4	4		6	ТК4	ПК-2.2, ОПК-7.2
Раздел 5. Браузер: документ, события, интерфейсы	14	4	4		6	ТК5	ПК-2.2, ОПК-7.1
Зачет	36				36	ОМ	ПК-2.1, ПК-2.2, ОПК-7.1, ОПК-7.2
<b>Итого за семестр</b>	<b>108</b>	<b>18</b>	<b>24</b>		<b>66</b>		

### 3.3. Содержание дисциплины

Раздел 1. Основы языка.

Тема 1.1. Переменные, типы данных, взаимодействие с пользователем, преобразование типов, базовые операторы.

Тема 1.2. Условный оператор, логические операторы, циклы, конструкция switch-case.

Тема 1.3. Работа с функциями, синтаксисы Function Declaration и Function Expression, стрелочные функции, рекурсия, замыкание, планирование функций.

## Раздел 2. Объекты.

Тема 2.1. Копирование объектов и ссылки, методы объекта, контекст

Тема 2.2. Конструктор объектов, преобразование объектов в примитивы.

Тема 2.3. Методы примитивов.

Тема 2.4. Массивы, методы массивов, перебираемые объекты.

Тема 2.5. Коллекции Map и Set, методы Object.keys, values, entries, Деструктурирующее присваивание

Тема 2.6. Дата и время, Формат JSON, метод toJSON

Тема 2.7. Флаги и дескрипторы свойств, свойства геттеры и сеттеры.

## Раздел 3. Наследование, классы.

Тема 3.1. Прототипное наследование, встроенные прототипы, методы прототипов, объекты без свойства \_\_proto\_\_

Тема 3.2. Класс: базовый синтаксис, наследование классов, статические свойства и методы, приватные и защищённые методы и свойства, расширение встроенных классов, проверка класса: "instanceof"

## Раздел 4. Регулярные выражения как методы строк.

Тема 4.1. Шаблоны и флаги, Символьные классы, якоря, граница слова, экранирование, наборы и диапазоны, квантификаторы.

Тема 4.2. Жадные и ленивые квантификаторы, скобочные группы, ссылки в шаблоне, альтернатива, опережающие и ретроспективные проверки.

## Раздел 5. Браузер: документ, события, интерфейсы

Тема 5.1. Браузерное окружение, DOM-дерево, навигация по DOM-элементам, Поиск, свойства узлов

Тема 5.2. Атрибуты и свойства, изменение документа, стили и классы.

Тема 5.3. Введение в браузерные события, Генерация пользовательских событий, интерфейсные события.

### 3.4. Тематический план практических занятий

Данный вид работы не предусмотрен учебным планом.

### 3.5. Тематический план лабораторных работ

Лабораторная работа №1. **Основы языка.** Изучение типов данных, преобразование типов, базовые операторы, логических операторов, циклов, конструкции switch-case. Освоение принципов работы с функциями, синтаксисов Function Declaration и Function Expression, стрелочных функций (4 часа).

Лабораторная работа №2. **Объекты.** Изучение методов объекта, конструкторов объектов, принципов преобразования объектов в примитивы. Применение методов примитивов и массивов. Изучение основ работы с объектами Date, форматом JSON, дескрипторами свойств объекта. (4 часа).

Лабораторная работа №3. **Наследование.** Изучение принципов прототипного наследования, встроенных прототипов, методов прототипов (4 часа).

Лабораторная работа №4. **Классы.** Изучение синтаксиса классов, принципов наследования классов, статических свойства и методов. Реализация приватных и защищённых методов и свойств. (4 часа).

Лабораторная работа №5. *Регулярные выражения как методы строк.*  
Изучение шаблонов и флагов, символьных классы, наборов и диапазонов, квантификаторов, скобочных группы, альтернации, опережающей и ретроспективной проверки (4 часа).

Лабораторная работа №6. *Браузер: документ, события, интерфейсы.*

Изучение концепций браузерного окружения, DOM-дерева, навигации по DOM-элементам. Реализация браузерных событий и генерация пользовательских событий. (4 часа).

### 3.6. Курсовой проект /курсовая работа

Данный вид работы не предусмотрен учебным планом.

## 4. Оценивание результатов обучения

Оценивание результатов обучения по дисциплине осуществляется в рамках текущего контроля и промежуточной аттестации, проводимых по балльно-рейтинговой системе (БРС).

Шкала оценки результатов обучения по дисциплине:

Код компетенции	Код индикатора компетенции	Запланированные результаты обучения по дисциплине	Уровень сформированности индикатора компетенции				
			Высокий	Средний	Ниже среднего	Низкий	
			от 85 до 100	от 70 до 84	от 55 до 69	от 0 до 54	
			Шкала оценивания				
			отлично	хорошо	удовлетворительно	неудовлетворительно	
			зачтено			не зачтено	
ПК-2 Способен разрабатывать и тестировать программные компоненты решения задач в системах искусственного интеллекта	ПК-2.1 Разрабатывает приложения систем искусственного интеллекта	знать:	свободно знает основные принципы, нормы и правила при разработке приложений систем искусственного интеллекта	знает основные стандарты, нормы и правила при разработке приложений систем искусственного интеллекта, делает незначительные ошибки	плохо знает основные стандарты, нормы и правила при разработке приложений систем искусственного интеллекта	не знает основные стандарты, нормы и правила при разработке приложений систем искусственного интеллекта	
		уметь:	Применять основные методы разработки приложений систем искусствен	свободно использует основные стандарты, нормы и правила при разработке приложений	использует основные стандарты, нормы и правила при разработке приложений	неуверенно использует основные стандарты, нормы и правила при разработке приложений	На умеет использовать основные стандарты, нормы и правила при

		ного интеллекта	систем искусственно го интеллекта	й систем искусствен ного интеллекта , допускает ошибки	систем искусствен ного интеллекта	разработке приложени й систем искусствен ного интеллекта
		владеть:				
		навыками разработки приложени й систем искусствен ного интеллекта	свободно владеет основными стандартами, нормами и правилами при разработке приложений систем искусственно го интеллекта	Владеет с ошибками основным и стандарта ми, нормами и правилами при разработке приложени й систем искусствен ного интеллекта	неуверенно владеет основными стандартами , нормами и правилами при разработке приложений систем искусствен ного интеллекта	Не владеет основными стандартам и, нормами и правилами при разработке приложени й систем искусствен ного интеллекта
		знать:				
		Принципы тестирован ия систем искусствен ного интеллекта	свободно знает техническую документаци ю тестирования систем искусственно го интеллекта	хорошо знает техническ ую документа цию тестирован ия систем искусствен ного интеллекта	неуверенно знает техническу ю документац ию тестировани я систем искусствен ного интеллекта	Не знает техническу ю документа цию тестирован ия систем искусствен ного интеллекта
		уметь:				
	ПК-2.2 Проводит тестирова ние систем искусстве ного интелект а	Применять на практике методы тестирован ия систем искусствен ного интеллекта	свободно умеет разрабаты вать техническую документаци ю тестирования систем искусственно го интеллекта	умеет разрабаты вать техническ ую документа цию тестирован ия систем искусствен ного интеллекта , но с ошибками	плохо умеет разрабаты вать техническу ю документац ию тестировани я систем искусствен ного интеллекта	Не умеет разрабаты вать техническу ю документа цию тестирован ия систем искусствен ного интеллекта
		владеть:				
		Навыками тестирован ия систем искусствен ного интеллекта	свободно владеет разработкой технической документации тестирования систем искусственно го интеллекта	хорошо владеет разработко й техническ ой документа ции тестирован ия систем искусствен	плохо владеет разработкой технической документац ии тестировани я систем искусствен ного интеллекта	не владеет разработко й техническо й документа ции тестирован ия систем искусствен ного интеллекта

				ного интеллекта		
ОПК-7 Способен разрабатывать алгоритмы и программы, пригодные для практического применения;	ОПК-7.1. Способен разрабатывать алгоритмы для решения практических задач	знать:				
		основы разработки алгоритмов, необходимых для решения практических задач на Javascript	Отлично знает все основы разработки алгоритмов, необходимых для решения практических задач на Javascript	Хорошо знает большинство основ разработки алгоритмов, необходимых для решения практических задач на Javascript	Удовлетворительно знает некоторые основы разработки алгоритмов, необходимых для решения практических задач на Javascript	Уровень знаний по разработке алгоритмов для решения практических задач на Javascript ниже минимального уровня требований.
		уметь:				
		разрабатывать алгоритмы, необходимые для решения практических задач на Javascript; работать с современными системами программирования	Демонстрирует отличные умения разрабатывать алгоритмы, необходимые для решения практических задач на Javascript; Отлично работает с современным и системами программирования	Демонстрирует хорошие умения разрабатывать алгоритмы, необходимые для решения практических задач на Javascript. Допускает ряд незначительных ошибок. Хорошо работает с современными системами программирования	Демонстрирует удовлетворительные умения разрабатывать алгоритмы, необходимые для решения практических задач на Javascript. Допускает серьезные ошибки. Задания выполняет не в полном объеме. Удовлетворительно работает с современными системами программирования	Допускает грубые ошибки при разработке алгоритмов, необходимых для решения практических задач на Javascript
владеть:						
Навыками разработки алгоритмов для решения практических задач на Javascript для дальнейшего применения в	Демонстрирует отличные навыки разработки алгоритмов для решения практических задач на Javascript для требуемой предметной области с	Демонстрирует базовые навыки разработки алгоритмов для решения практических задач на Javascript для	Демонстрирует минимальные навыки разработки алгоритмов для решения практических задач на Javascript для требуемой	Допускает грубые ошибки на стадиях проектирования алгоритмов для решения практических задач на Javascript		

		профессиональной деятельности.	использованием современных языков программирования.	требуемой предметной области с использованием современных языков программирования.	предметной области с использованием современных языков программирования.	
ОПК-7.2 Способен разрабатывать компьютерные программы, пригодные для практического применения	знать:					
	основы разработки компьютерных программ на Javascript, пригодных для практического применения	Отлично знает все основы разработки компьютерных программ на Javascript, пригодных для практического применения	Хорошо знает большинство основ разработки компьютерных программ на Javascript, пригодных для практического применения	Удовлетворительно знает некоторые основы разработки компьютерных программ на Javascript, пригодных для практического применения	Уровень знаний по разработке компьютерных программ на Javascript, пригодных для практического применения, ниже минимального уровня требований.	
	уметь:					
	разрабатывать компьютерные программы на Javascript, пригодные для практического применения;	Демонстрирует отличные умения разрабатывать компьютерные программы на Javascript, пригодные для практического применения	Демонстрирует хорошие умения разрабатывать компьютерные программы на Javascript, пригодные для практического применения;	разрабатывать компьютерные программы на Javascript, пригодные для практического применения;	Допускает грубые ошибки при разработке компьютерных программ на Javascript, пригодных для практического применения;	
владеть:						
Навыками разработки компьютерных программ на Javascript, пригодных для практического	Демонстрирует отличные навыки разработки компьютерных программ на Javascript, пригодных для практического применения.	Демонстрирует базовые навыки разработки компьютерных программ на Javascript, пригодных для	Демонстрирует минимальные навыки разработки компьютерных программ на Javascript, пригодных для практическо	Допускает грубые ошибки на стадиях проектирования компьютерных программ на Javascript, пригодных		

		применени я		практическ ого применени я.	го применения.	для практическ ого применени я.
--	--	----------------	--	--------------------------------------	-------------------	---

Оценочные материалы для проведения текущего контроля и промежуточной аттестации приведены в Приложении к рабочей программе дисциплины.

Полный комплект заданий и материалов, необходимых для оценивания результатов обучения по дисциплине, хранится на кафедре разработчика.

## 5. Учебно-методическое и информационное обеспечение дисциплины

### 5.1. Учебно-методическое обеспечение

#### 5.1.1. Основная литература

1. Заяц, А. М. Проектирование и разработка WEB-приложений. Введение в frontend и backend разработку на JavaScript и node.js : учебное пособие для вузов / А. М. Заяц, Н. П. Васильев. — 3-е изд., стер. — Санкт-Петербург : Лань, 2021. — 120 с. — ISBN 978-5-8114-7042-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/154380> (дата обращения: 28.06.2023). — Режим доступа: для авториз. пользователей.

2. Проектирование и разработка WEB-приложений. Введение в frontend и backend разработку на JavaScript и node.js : учебное пособие / А. М. Заяц, Н. П. Васильев. - 3-е изд., стер. - Санкт-Петербург : Лань, 2021. - 119 с. - URL: <https://e.lanbook.com/book/154380>. - ISBN 978-5-8114-7042-6. - Текст : электронный.

3. Клиентские технологии веб-программирования: JavaScript и DOM : учебное пособие / А. В. Диков. - Санкт-Петербург : Лань, 2020. - 122 с. - URL: <https://e.lanbook.com/book/126934>. - ISBN 978-5-8114-4074-0. - Текст : электронный.

#### 5.1.2.Дополнительная литература

1. Старолетов, С. М. Основы тестирования и верификации программного обеспечения / С. М. Старолетов. — 3-е изд., стер. — Санкт-Петербург : Лань, 2023. — 344 с. — ISBN 978-5-507-46773-0. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/319445> (дата обращения: 28.06.2023). — Режим доступа: для авториз. пользователей.

2. Основы информатики и программирования: учебное пособие / Е. А. Роганов. - 2-е изд., испр. - М. : Национальный Открытый Университет "ИНТУИТ", 2016. - 392 с. - URL: <https://e.lanbook.com/book/100298>. - Текст : электронный.

БИБЛИОТЕКА  
КГЭУ

## 5.2. Информационное обеспечение

### 5.2.1. Электронные и интернет-ресурсы

№ п/п	Наименование электронных и интернет-ресурсов	Ссылка
1	Электронно-библиотечная система «Лань»	<a href="https://e.lanbook.com/">https://e.lanbook.com/</a>
2	Электронно-библиотечная система «ibooks.ru»	<a href="https://ibooks.ru/">https://ibooks.ru/</a>
3	Электронно-библиотечная система «book.ru»	<a href="https://www.book.ru/">https://www.book.ru/</a>
4	Портал «Открытое образование»	<a href="https://npoed.ru">https://npoed.ru</a>
5	Российская национальная библиотека	<a href="https://nlr.ru/">https://nlr.ru/</a>

6	КиберЛенинка	<a href="https://cyberleninka.ru">https://cyberleninka.ru</a>
7	Техническая библиотека	<a href="https://techlibrary.ru">https://techlibrary.ru</a>
8	Национальная электронная библиотека (НЭБ)	<a href="https://rusneb.ru/">https://rusneb.ru/</a>

### 5.2.2. Профессиональные базы данных / Информационно-справочные системы

№ п/п	Наименование профессиональных баз данных	Адрес	Режим доступа
1	Официальный интернет-портал правовой информации	<a href="http://pravo.gov.ru">http://pravo.gov.ru</a>	<a href="http://pravo.gov.ru">http://pravo.gov.ru</a>
2	Справочная правовая система «Консультант Плюс»	<a href="http://consultant.ru">http://consultant.ru</a>	<a href="http://consultant.ru">http://consultant.ru</a>
3	Справочно-правовая система по законодательству РФ	<a href="http://garant.ru">http://garant.ru</a>	<a href="http://garant.ru">http://garant.ru</a>

### 5.2.3. Лицензионное и свободно распространяемое программное обеспечение дисциплины

№ п/п	Наименование программного обеспечения	Описание	Реквизиты подтверждающих документов
1	Операционная система Microsoft Windows 10	Пользовательская операционная система	Договор №133/2021 от 12.10.2021, лицензиар - ЗАО «Софт Лайн Трейд», тип (вид) лицензии - неискл. право, срок действия лицензии - бессрочно
2	Microsoft Office 2019	Пакет офисных приложений	Договор №133/2021 от 12.10.2021, лицензиар - ЗАО «Софт Лайн Трейд», тип (вид) лицензии - неискл. право, срок действия лицензии - бессрочно
3	LMS Moodle	ПО для эффективного онлайн- взаимодействия преподавателя и студента	Свободная лицензия, тип (вид) лицензии - неискл. право, срок действия лицензии - бессрочно.
4	Браузер Chrome	Система поиска информации в сети интернет	Свободная лицензия, тип (вид) лицензии - неискл. право, срок действия лицензии - бессрочно.

## 6. Материально-техническое обеспечение дисциплины

№ п/п	Вид учебной работы	Наименование специальных помещений и помещений для СРС	Оснащенность специальных помещений и помещений для СРС
1	Лекционные занятия	Учебная аудитория для проведения занятий лекционного типа	доска аудиторная, акустическая система, проектор, усилитель-микшер для систем громкой связи, экран, микрофон, миникомпьютер, монитор

2	Лабораторные работы	Учебная лаборатория	доска аудиторная, персональный компьютер (25 шт.)
3	Самостоятельная работа обучающегося	Компьютерный класс с выходом в Интернет В-600а	моноблок (30 шт.), система видеонаблюдения (6 видеокамер), проектор, экран
		Читальный зал библиотеки	специализированная мебель, компьютерная техника с возможностью выхода в Интернет и обеспечением доступа в ЭИОС, мультимедийный проектор, экран, программное обеспечение

## 7. Особенности организации образовательной деятельности для лиц с ограниченными возможностями здоровья и инвалидов

Лица с ограниченными возможностями здоровья (ОВЗ) и инвалиды имеют возможность беспрепятственно перемещаться из одного учебно-лабораторного корпуса в другой, подняться на все этажи учебно-лабораторных корпусов, заниматься в учебных и иных помещениях с учетом особенностей психофизического развития и состояния здоровья.

Для обучения лиц с ОВЗ и инвалидов, имеющих нарушения опорно-двигательного аппарата, обеспечены условия беспрепятственного доступа во все учебные помещения. Информация о специальных условиях, созданных для обучающихся с ОВЗ и инвалидов, размещена на сайте университета [www/kgeu.ru](http://www/kgeu.ru). Имеется возможность оказания технической помощи ассистентом, а также услуг сурдопереводчиков и тифлосурдопереводчиков.

Для адаптации к восприятию лицами с ОВЗ и инвалидами с нарушенным слухом справочного, учебного материала по дисциплине обеспечиваются следующие условия:

- для лучшей ориентации в аудитории, применяются сигналы оповещения о начале и конце занятия (слово «звонок» пишется на доске);

- внимание слабослышащего обучающегося привлекается педагогом жестом (на плечо кладется рука, осуществляется нерезкое похлопывание);

- разговаривая с обучающимся, педагогический работник смотрит на него, говорит ясно, короткими предложениями, обеспечивая возможность чтения по губам.

Компенсация затруднений речевого и интеллектуального развития слабослышащих обучающихся проводится путем:

- использования схем, диаграмм, рисунков, компьютерных презентаций с гиперссылками, комментирующими отдельные компоненты изображения;

- регулярного применения упражнений на графическое выделение существенных признаков предметов и явлений;

- обеспечения возможности для обучающегося получить адресную консультацию по электронной почте по мере необходимости.

Для адаптации к восприятию лицами с ОВЗ и инвалидами с нарушениями зрения справочного, учебного, просветительского материала, предусмотренного образовательной программой по выбранному направлению подготовки,

обеспечиваются следующие условия:

- ведется адаптация официального сайта в сети Интернет с учетом особых потребностей инвалидов по зрению, обеспечивается наличие крупношрифтовой справочной информации о расписании учебных занятий;

- педагогический работник, его собеседник (при необходимости), присутствующие на занятии, представляются обучающимся, при этом каждый раз называется тот, к кому педагогический работник обращается;

- действия, жесты, перемещения педагогического работника коротко и ясно комментируются;

- печатная информация предоставляется крупным шрифтом (от 18 пунктов), тотально озвучивается;

- обеспечивается необходимый уровень освещенности помещений;

- предоставляется возможность использовать компьютеры во время занятий и право записи объяснений на диктофон (по желанию обучающихся).

Форма проведения текущей и промежуточной аттестации для обучающихся с ОВЗ и инвалидов определяется педагогическим работником в соответствии с учебным планом. При необходимости обучающемуся с ОВЗ, инвалиду с учетом их индивидуальных психофизических особенностей дается возможность пройти промежуточную аттестацию устно, письменно на бумаге, письменно на компьютере, в форме тестирования и т.п., либо предоставляется дополнительное время для подготовки ответа.

## **8. Методические рекомендации для преподавателей по организации воспитательной работы с обучающимися.**

Методическое обеспечение процесса воспитания обучающихся выступает одним из определяющих факторов высокого качества образования. Преподаватель вуза, демонстрируя высокий профессионализм, эрудицию, четкую гражданскую позицию, самодисциплину, творческий подход в решении профессиональных задач, в ходе образовательного процесса способствует формированию гармоничной личности.

При реализации дисциплины преподаватель может использовать следующие методы воспитательной работы:

- методы формирования сознания личности (беседа, диспут, внушение, инструктаж, контроль, объяснение, пример, самоконтроль, рассказ, совет, убеждение и др.);

- методы организации деятельности и формирования опыта поведения (задание, общественное мнение, педагогическое требование, поручение, приучение, создание воспитывающих ситуаций, тренинг, упражнение, и др.);

- методы мотивации деятельности и поведения (одобрение, поощрение социальной активности, порицание, создание ситуаций успеха, создание ситуаций для эмоционально-нравственных переживаний, соревнование и др.)

При реализации дисциплины преподаватель должен учитывать следующие направления воспитательной деятельности:

*Гражданское и патриотическое воспитание:*

- формирование у обучающихся целостного мировоззрения, российской идентичности, уважения к своей семье, обществу, государству, принятым в семье и

обществе духовно-нравственным и социокультурным ценностям, к национальному, культурному и историческому наследию, формирование стремления к его сохранению и развитию;

- формирование у обучающихся активной гражданской позиции, основанной на традиционных культурных, духовных и нравственных ценностях российского общества, для повышения способности ответственно реализовывать свои конституционные права и обязанности;

- развитие правовой и политической культуры обучающихся, расширение конструктивного участия в принятии решений, затрагивающих их права и интересы, в том числе в различных формах самоорганизации, самоуправления, общественно-значимой деятельности;

- формирование мотивов, нравственных и смысловых установок личности, позволяющих противостоять экстремизму, ксенофобии, дискриминации по социальным, религиозным, расовым, национальным признакам, межэтнической и межконфессиональной нетерпимости, другим негативным социальным явлениям.

#### *Духовно-нравственное воспитание:*

- воспитание чувства достоинства, чести и честности, совестливости, уважения к родителям, учителям, людям старшего поколения;

- формирование принципов коллективизма и солидарности, духа милосердия и сострадания, привычки заботиться о людях, находящихся в трудной жизненной ситуации;

- формирование солидарности и чувства социальной ответственности по отношению к людям с ограниченными возможностями здоровья, преодоление психологических барьеров по отношению к людям с ограниченными возможностями;

- формирование эмоционально насыщенного и духовно возвышенного отношения к миру, способности и умения передавать другим свой эстетический опыт.

#### *Культурно-просветительское воспитание:*

- формирование эстетической картины мира;

- формирование уважения к культурным ценностям родного города, края, страны;

- повышение познавательной активности обучающихся.

#### *Научно-образовательное воспитание:*

- формирование у обучающихся научного мировоззрения;

- формирование умения получать знания;

- формирование навыков анализа и синтеза информации, в том числе в профессиональной области.

## Вносимые изменения и утверждения на новый учебный год

№ П/П	№ раздела внесения изменений	Дата внесения изменений	Содержание изменений	«Согласовано» Зав. каф. реализующей дисциплину	«Согласовано» председатель УМК института (факультета), в состав которого входит выпускающая
1	2	3	4	5	6
1					
2					
3					



**КГЭУ**

- **МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ  
ФЕДЕРАЦИИ  
Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«КАЗАНСКИЙ ГОСУДАРСТВЕННЫЙ ЭНЕРГЕТИЧЕСКИЙ УНИВЕРСИТЕТ»  
(ФГБОУ ВО «КГЭУ»)**

**ОЦЕНОЧНЫЕ МАТЕРИАЛЫ  
по дисциплине**

**Б1.О.15 Современные языки программирования**

---

*(Наименование дисциплины в соответствии с учебным планом)*

Оценочные материалы по дисциплине «Современные языки программирования», предназначены для оценивания результатов обучения на соответствие индикаторам достижения компетенций.

Оценивание результатов обучения по дисциплине осуществляется в рамках текущего контроля (ТК) и промежуточной аттестации, проводимых по балльно-рейтинговой системе (БРС).

### 1. Технологическая карта

#### Семестр 5

Наименование раздела	Формы и вид контроля	Рейтинговые показатели						Промежуточная аттестация
		I текущий контроль	II текущий контроль	III текущий контроль	IV текущий контроль	V текущий контроль	Итого	
<b>Раздел 1. «Основы языка»</b>	<b>ТК1</b>	<b>11</b>					<b>0-55</b>	<b>0-45</b>
Защита лабораторной работы №1		11						
<b>Раздел 2. «Объекты»</b>	<b>ТК2</b>		<b>11</b>				<b>0-55</b>	<b>0-45</b>
Защита лабораторной работы №2			11					

<b>Раздел 3. « Прототипы, наследование, классы»</b>	<b>TK3</b>			11			<b>0-55</b>	<b>0-45</b>
Защита лабораторной работы №3				5				
Защита лабораторной работы №4				6				
<b>Раздел 4. « Регулярные выражения как методы строк.»</b>	<b>TK4</b>				11		<b>0-55</b>	<b>0-45</b>
Защита лабораторной работы №5					11			
<b>Раздел 5. « Браузер: документ, события, интерфейсы »</b>	<b>TK5</b>					<b>11</b>	<b>0-55</b>	<b>0-45</b>

Защита лабораторной работы №6						11		
<b>Промежуточная аттестация (зачет, экзамен, КП, КР)</b>	<b>ОМ</b>							<b>0-45</b>
В письменной форме по билетам								<b>0-45</b>

**2. Оценочные материалы текущего контроля и промежуточной аттестации**  
Шкала оценки результатов обучения по дисциплине:

Код компетенции	Код индикатора компетенции	Запланированные результаты обучения по дисциплине	Уровень сформированности индикатора компетенции			
			Высокий	Средний	Ниже среднего	Низкий
			от 85 до 100	от 70 до 84	от 55 до 69	от 0 до 54
			Шкала оценивания			
			отлично	хорошо	удовлетворительно	неудовлетворительно
			зачтено			не зачтено
ПК-2 Способен разрабатывать и тестировать программные компоненты решения задач в системах искусственного интеллекта	ПК-2.1 Разрабатывает приложения систем искусственного интеллекта	знать:  Принципы разработки приложений систем искусственного интеллекта	свободно знает основные принципы, нормы и правила при разработке приложений систем искусственного интеллекта	знает основные стандарты, нормы и правила при разработке приложений систем искусственного интеллекта, делает незначительные	плохо знает основные стандарты, нормы и правила при разработке приложений систем искусственного интеллекта	не знает основные стандарты, нормы и правила при разработке приложений систем искусственного интеллекта

			ьные ошибки		
		уметь:			
	Применять основные методы разработки приложений систем искусственного интеллекта	свободно использует основные стандарты, нормы и правила при разработке приложений систем искусственного интеллекта	использует основные стандарты, нормы и правила при разработке приложений систем искусственного интеллекта, допускает ошибки	неуверенно использует основные стандарты, нормы и правила при разработке приложений систем искусственного интеллекта	На умеет использовать основные стандарты, нормы и правила при разработке приложений систем искусственного интеллекта
		владеть:			
	навыками разработки приложений систем искусственного интеллекта	свободно владеет основными стандартами, нормами и правилами при разработке приложений систем искусственного интеллекта	Владеет с ошибками основным стандарта ми, нормами и правилами при разработке приложений систем искусственного интеллекта	неуверенно владеет основными стандартами, нормами и правилами при разработке приложений систем искусственного интеллекта	Не владеет основными стандартами и нормами и правилами при разработке приложений систем искусственного интеллекта
ПК-2.2 Проводит тестирование систем искусственного интеллекта		знать:			
	Принципы тестирования систем искусственного интеллекта	свободно знает техническую документацию тестирования систем искусственного интеллекта	хорошо знает техническую документацию тестирования систем искусственного интеллекта	неуверенно знает техническую документацию тестирования систем искусственного интеллекта	Не знает техническую документацию тестирования систем искусственного интеллекта
		уметь:			
	Применять на практике методы тестирования систем искусственного интеллекта	свободно умеет разрабатывать техническую документацию тестирования систем искусственного интеллекта	умеет разрабатывать техническую документацию тестирования систем искусственного интеллекта, но с ошибками	плохо умеет разрабатывать техническую документацию тестирования систем искусственного интеллекта	Не умеет разрабатывать техническую документацию тестирования систем искусственного интеллекта
		владеть:			

		Навыками тестирования систем искусственного интеллекта	свободно владеет разработкой технической документации тестирования систем искусственного интеллекта	хорошо владеет разработкой технической документации тестирования систем искусственного интеллекта	плохо владеет разработкой технической документации тестирования систем искусственного интеллекта	не владеет разработкой технической документации тестирования систем искусственного интеллекта
ОПК-7 Способен разрабатывать алгоритмы и программы, пригодные для практического применения;	ОПК-7.1. Способен разрабатывать алгоритмы для решения практических задач	знать:				
		основы разработки алгоритмов, необходимых для решения практических задач на Javascript	Отлично знает все основы разработки алгоритмов, необходимых для решения практических задач на Javascript	Хорошо знает большинство основ разработки алгоритмов, необходимых для решения практических задач на Javascript	Удовлетворительно знает некоторые основы разработки алгоритмов, необходимых для решения практических задач на Javascript	Уровень знаний по разработке алгоритмов для решения практических задач на Javascript ниже минимального уровня требований.
		уметь:				
		разрабатывать алгоритмы, необходимые для решения практических задач на Javascript;  работать с современными системами программирования	Демонстрирует отличные умения разрабатывать алгоритмы, необходимые для решения практических задач на Javascript;  Отлично работает с современным и системами программирования	Демонстрирует хорошие умения разрабатывать алгоритмы, необходимые для решения практических задач на Javascript. Допускает ряд незначительных ошибок.  Хорошо работает с современными системами программирования	Демонстрирует удовлетворительные умения разрабатывать алгоритмы, необходимые для решения практических задач на Javascript. Допускает серьезные ошибки. Задания выполняет не в полном объеме.  Удовлетворительно работает с современными системами программирования	Допускает грубые ошибки при разработке алгоритмов, необходимых для решения практических задач на Javascript
		владеть:				
Навыками разработки	Демонстрирует отличные	Демонстрирует	Демонстрирует	Допускает грубые		

		алгоритмов для решения практических задач на Javascript для дальнейшего применения в профессиональной деятельности.	навыки разработки алгоритмов для решения практических задач на Javascript для требуемой предметной области с использованием современных языков программирования.	базовые навыки разработки алгоритмов для решения практических задач на Javascript для требуемой предметной области с использованием современных языков программирования.	минимальные навыки разработки алгоритмов для решения практических задач на Javascript для требуемой предметной области с использованием современных языков программирования.	ошибки на стадиях проектирования алгоритмов для решения практических задач на Javascript
	ОПК-7.2 Способен разрабатывать компьютерные программы, пригодные для практического применения	знать:				
		основы разработки компьютерных программ на Javascript, пригодных для практического применения	Отлично знает все основы разработки компьютерных программ на Javascript, пригодных для практического применения	Хорошо знает большинство основ разработки компьютерных программ на Javascript, пригодных для практического применения	Удовлетворительно знает некоторые основы разработки компьютерных программ на Javascript, пригодных для практического применения	Уровень знаний по разработке компьютерных программ на Javascript, пригодных для практического применения, ниже минимального уровня требований
		уметь:				
		разрабатывать компьютерные программы на Javascript, пригодные для практического применения;	Демонстрирует отличные умения разрабатывать компьютерные программы на Javascript, пригодные для практического применения	Демонстрирует хорошие умения разрабатывать компьютерные программы на Javascript, пригодные для практического применения;	разрабатывать компьютерные программы на Javascript, пригодные для практического применения;	Допускает грубые ошибки при разработке компьютерных программ на Javascript, пригодных для практического применения;
		владеть:				

		Навыками разработки компьютерных программ на Javascript, пригодных для практического применения	Демонстрирует отличные навыки разработки компьютерных программ на Javascript, пригодных для практического применения.	Демонстрирует базовые навыки разработки компьютерных программ на Javascript, пригодных для практического применения.	Демонстрирует минимальные навыки разработки компьютерных программ на Javascript, пригодных для практического применения.	Допускает грубые ошибки на стадиях проектирования компьютерных программ на Javascript, пригодных для практического применения.
--	--	---	---	--	--	--

Оценка **«отлично»** выставляется за безошибочное выполнение лабораторных работ и их защиту в семестре; глубокое понимание современных языков программирования, полные и содержательные ответы на вопросы билета (теоретическое задание);

Оценка **«хорошо»** выставляется за выполнение лабораторных работ и их защиту в семестре с незначительными ошибками; хорошее понимание современных языков программирования, достаточно полные и содержательные ответы на вопросы билета (теоретическое задание);

Оценка **«удовлетворительно»** выставляется за выполнение лабораторных работ и их защиту в семестре с ошибками; среднее понимание современных языков программирования, недостаточно полные и содержательные ответы на вопросы билета (теоретическое задание);

Оценка **«неудовлетворительно»** выставляется за слабое и неполное выполнение лабораторных работ и их защиту в семестре; слабое понимание современных языков программирования, неполные и несодержательные ответы на вопросы билета (теоретическое задание);

### 3. Перечень оценочных средств

Краткая характеристика оценочных средств, используемых при текущем контроле успеваемости и промежуточной аттестации обучающегося по дисциплине:

Наименование оценочного средства	Краткая характеристика оценочного средства	Описание оценочного средства
Отчет по лабораторной работе (ОЛР)	Выполнение лабораторной работы, обработка результатов моделирования. Оформление отчета, защита результатов лабораторной работы по отчету	Перечень заданий и вопросов для защиты лабораторной работы, перечень требований к отчету

**4. Перечень контрольных заданий или иные материалы, необходимые для оценки знаний, умений и навыков, характеризующих этапы формирования компетенций в процессе освоения дисциплины**

## *Пример задания*

### **Для текущего контроля ТК1:**

Проверяемые компетенции: ПК-2.1 Разрабатывает приложения систем искусственного интеллекта, ОПК-7.1 Способен разрабатывать алгоритмы для решения практических задач

### **Задачи, относящиеся к лабораторной работе ТК1**

1. Создать переменную и присвоить ей значение 8. Вывести значение этой переменной на экран с помощью функции alert.
2. Создать переменные  $a = 8$  и  $b = 4$ . Вывести на экран их сумму, разность, произведение и частное (результат деления).
3. Создать переменные  $c = 11$  и  $d = 4$ . Просуммировать их, результат присвоить переменной result. Вывести на экран значение переменной result.
4. Создать переменные  $a = 17$  и  $b = 10$ . Отнять от  $a$  переменную  $b$ , результат присвоить переменной  $c$ . Затем создать переменную  $d$ , присвоить ей значение 7. Сложить переменные  $c$  и  $d$ , а результат записать в переменную result. Вывести на экран значение переменной result.
5. Создать переменную str и присвоить ей значение 'Hello World!'. Вывести значение этой переменной на экран.
6. Создать переменные  $str1 = \text{'Hello'}$  и  $str2 = \text{'World!'}$ . С помощью этих переменных и операции сложения вывести на экран фразу 'Hello World!'.
7. Создать переменную name и присвоить ей свое имя. Вывести на экран фразу 'Привет, %Имя%!'
8. Создать переменную age и присвоить ей свой возраст. Вывести на экран 'Мне %Возраст% лет!'.
9. Создать страницу, которая отобразит сообщение «Hello World».
10. Решить предыдущую задачу, вызвав alert из внешнего файла alert.js, лежащего в той же папке, что и исходная страница.

### **План Лабораторной работы № 1**

Цель – изучение типов данных, преобразование типов, базовые операторы, логических операторов, циклов, конструкции switch-case. Освоение принципов работы с функциями, синтаксисов Function Declaration и Function Expression, стрелочных функций.

Итог – освоение обучающимися навыков преобразования типов, использования логических операторов, циклов, конструкции switch-case функций, создаваемых с помощью синтаксисов Function Declaration и Function Expression, стрелочных функций.

### **Для текущего контроля ТК2:**

Проверяемые компетенции: ПК-2.2 Проводит тестирование систем искусственного интеллекта, ОПК-7.2 Способен разрабатывать компьютерные программы, пригодные

для практического применения

### **Задачи, относящиеся к лабораторной работе ТК2.**

1. Создать объект, в котором ключами являются числа от 1 до 7, а значениями – соответствующие этим числам названия дней недели. Вывести текущий день недели, обратившись к нужному ключу объекта.
2. Создать функцию `checkProp(obj)`, возвращающую `false`, если у объекта нет свойств, и `true` – если свойства есть. Протестировать функцию на пустом и непустом объектах.
3. Создать функцию, которая принимает в качестве параметра объект, суммирует зарплаты сотрудников и возвращает сумму зарплат.
4. Создать функцию, которая принимает в качестве параметра объект, умножает все его числовые свойства на 10, не изменяя при этом нечисловые, и возвращает измененный объект.
5. Создать функцию `max(a, b)`, которая возвращает большее из чисел `a` и `b`. В случае их равенства функция должна возвращать строку «Эти числа равны».
6. Создать функцию, которая принимает 2 параметра `x` и `n` и с помощью цикла возвращает `x` в степени `n`. Параметр  $n \geq 1$ .
7. Создать функцию, которая принимает в качестве параметра число от 1 до 7 и возвращает день недели на русском языке.
8. Создать функцию `celsiusToFahrenheit`, которая принимает температуру в градусах по Цельсию и переводит ее в градусы по Фаренгейту.
9. Создать функцию, принимающую параметром число и возвращающую факториал этого числа.
10. Создать функцию, проверяющую число на простоту. Если число простое, функция должна вернуть `true`, иначе `false`.

### **План Лабораторной работы № 2**

Цель – изучение методов объекта, конструкторов объектов, принципов преобразования объектов в примитивы. Применение методов примитивов и массивов. Изучение основ работы с объектами `Date`, форматом `JSON`, дескрипторами свойств объекта.

Итог – освоение обучающимися навыков работы с методами объектов, примитивов, массивов, функциями-конструкторами, преобразованиями объектов в примитивы, объектами `Date`, форматом `JSON`, дескрипторами свойств объекта.

### **Для текущего контроля ТК3:**

Проверяемые компетенции: ПК-2.2 Проводит тестирование систем искусственного интеллекта, ОПК-7.1 Способен разрабатывать алгоритмы для решения практических задач

### **Задачи, относящиеся к лабораторной работе ТК3.**

1. Создать функцию-конструктор, принимающую в качестве параметра название самолета и возвращающую объект с 2 свойствами: 1) `name`, значение которого берется из параметра; 2) `flies`, значение которого по

умолчанию для всех объектов задано как false.

Добавить в прототип функции-конструктора 2 метода: 1) обозначает взлет и меняет значение `flies` на true; 2) обозначает посадку и меняет значение `flies` на false.

2. Создать функцию-конструктор, принимающую 2 параметра: 1) `carModel` – название автомобиля; 2) `kmPerLiter` – количество километров, которое можно проехать на 1 литре топлива. Еще 2 свойства – `fuelTank` (топливный бак) и `odometer` (одометр) – должны задаваться как 0 по умолчанию.

Добавить в прототип функции-конструктора следующие методы:

I) `refuel(liters)` – добавляет `liters` к свойству `fuelTank`, т.е. автомобиль заправляется `liters` литрами топлива.

II) `drive(distance)` – увеличивает свойство `odometer` на `distance`, уменьшает свойство `fuelTank`. Если `distance` оказывается больше или равно максимальному количеству км, которое может проехать автомобиль, `drive` должен вернуть строку «Закончилось топливо после преодоления \*\*\* км. На одометре \*\*\* км. Осталось \*\*\* литров топлива». Если `distance` оказывается меньше, чем максимальное количество км, которое может проехать автомобиль, `drive` должен вернуть строку «На одометре \*\*\* км, в топливном баке \*\*\* литров топлива».

3. Создать собственную реализацию метода `Object.create(prototype, [properties])`, принимающего 2 параметра: `prototype` – прототип, который будет наследоваться создаваемым объектом; `properties` – необязательный параметр, который будет передаваться в `Object.defineProperties`. Если `prototype` не является объектом или null, вернуть строку 'Ошибка'.

### План Лабораторной работы № 3

Цель – изучение принципов прототипного наследования, встроенных прототипов, методов прототипов

Итог – освоение обучающимися навыков работы с прототипным наследованием, встроенными прототипами, методами прототипов.

### Задачи, относящиеся к лабораторной работе ТКЗ.

Создать класс `Human`, принимающий следующие параметры:

имя `firstName` (по умолчанию задается как "John"), фамилию `lastName` (по умолчанию задается как "Smith"), возраст `age` (по умолчанию задается как 0), пол `gender` (по умолчанию задается как "undefined").

Созданный класс должен иметь метод `fullName`, не принимающий аргументы и возвращающий полное имя (имя + фамилия), а также статический метод `greetAliens(race)`, принимающий в качестве параметра название расы инопланетян и возвращающий строку "Welcome to the Earth \*raceName\*"

Дан класс `Animal`:

```
class Animal {
  constructor(name, age, legs, species, status) {
    this.name = name;
    this.age = age;
    this.legs = legs;
```

```

this.species = species;
this.status = status;
}
introduce() {
return `Hello, my name is ${this.name} and I am ${this.age} years
old.`;
}
}

```

Создать следующие классы, наследующие от Animal:

Класс акул Shark, конструктор которого принимает 3 аргумента в следующем порядке: name, age, status. Значение свойства legs должно быть равно 0 для всех акул, а значение свойства species должно быть равно "shark".

Класс кошек Cat, конструктор которого также принимает 3 аргумента: name, age, status. Метод introduce для класса Cat должен быть таким же, как в Animal, но дополнительно иметь после "...old." 2 пробела и слова "Meow meow!". Т.е. метод introduce() для Cat должен возвращать строку "Hello, my name is \*name\* and I am \*age\* years old. Meow meow!".

Класс собак Dog, конструктор которого принимает 4 аргумента в следующем порядке: name, age, status, master. Значением master должно быть имя хозяина собаки. Значение свойства legs должно быть равно 4 для всех собак, а значение свойства species должно быть равно "dog". Для собак должен быть свой аналог метода introduce, который называется greetMaster и возвращает строку "Hello \*master\*".

Реализовать класс Vector, с помощью которого создаются объекты-векторы, имеющие 3 координаты (x, y, z) и следующие методы:

plus, реализующий сложение двух векторов;

minus, реализующий вычитание двух векторов;

multiply, реализующий скалярное умножение двух векторов;

vec\_length, реализующий определение длины вектора.

#### **План Лабораторной работы № 4**

Цель – изучение синтаксиса классов, принципов наследования классов, статических свойства и методов. Реализация приватных и защищённых методов и свойств

Итог – освоение обучающимися навыков работы с классами, наследованием классов, использования статических свойств и методов, реализации приватных и защищённых методов и свойств.

#### **Для текущего контроля ТК4:**

Проверяемые компетенции: ПК-2.2 Проводит тестирование систем искусственного интеллекта, ОПК-7.2 Способен разрабатывать компьютерные программы, пригодные для практического применения

#### **Задачи, относящиеся к лабораторной работе ТК4.**

1. Дана строка "x1x x2x x3x x4x x5x xbх xvх".

Написать регулярное выражение, ищущее в этой строке подстроки, по краям которых стоят буквы "x", а между буквами "x" – одна цифра.

2. Дана строка "x1x x22x x333x x4444x x55555x xx хсх хtrх".

Написать регулярное выражение, ищущее в этой строке подстроки, в которых по краям стоят буквы "x", а между буквами "x" – любое количество цифр (включая ноль цифр – "xx").

3. Дана строка "aba aea asa aza аха".

Написать регулярное выражение, которое в этой строке найдет только следующие подстроки: "aba", "aea", "аха".

4. Дана строка "aba aea asa aza аха a.a a+a a\*a".

Написать регулярное выражение, которое в этой строке найдет только следующие подстроки: "aba", "a.a", "a+a", "a\*a".

5. Дана строка "aaa aBa asa aDa afa aZa aga".

Написать регулярное выражение, ищущее в этой строке подстроки, по краям которых стоят буквы "a", а между буквами "a" – буква из набора от a до f и от A до Z.

6. Написать регулярное выражение, которое с помощью метода test проверяет, является ли строка URL-адресом страницы, оканчивающимся на .com или .ru, протокол может быть как http, так и https.

Регулярное выражение должно корректно работать для URL-адресов следующего вида:

1) <https://www.yandex.ru> 2) <http://www.yandex.ru> 3) <https://www.google.com>

7. Сформировать строку из 5 целых случайных чисел от 1 до 9, разделенных пробелами. С помощью методов регулярного выражения заменить эти 5 случайных чисел на их квадраты.

Пример работы:

'5 7 1 2 3' → '25 49 1 4 9'

8. Дана строка вида ' a + b = x + y = i + j = ', в которой a, b, x, y, i, j – произвольные числа. С помощью методов регулярного выражения вставить в строку после знака равенства результаты сложений.

Пример работы:

'1 + 2 = 3 + 4 = 5 + 6 = ' // получили '1 + 2 = 3 3 + 4 = 7 5 + 6 = 11'

9. С помощью метода регулярного выражения test определить, является ли строка временем в формате '11.59 am', '00.01 pm'.

10. С помощью метода регулярного выражения test определить, лежит ли число в диапазоне [1500, 1800].

## План Лабораторной работы № 5

Цель – изучение шаблонов и флагов, символьных классы, наборов и диапазонов, квантификаторов, скобочных группы, альтернативы, опережающей и ретроспективной проверки

Итог – освоение обучающимися навыков работы с шаблонами регулярных выражений и использования флагов, принципов работы с символьными классами, наборами и диапазонами, квантификаторами, скобочными группами, альтернативой, опережающей и ретроспективной проверками.

## Для текущего контроля ТК5:

Проверяемые компетенции: ПК-2.2 Проводит тестирование систем искусственного интеллекта, ОПК-7.1 Способен разрабатывать алгоритмы для решения практических задач

## Задачи, относящиеся к лабораторной работе ТК5.

1. Даны ссылки (`<a href="ссылка">... </a>`). Добавить этим ссылкам следующее событие: при наведении курсора мыши на ссылку в ее свойство `innerHTML` добавляется значение ее атрибута `href` в круглых скобках.
2. Модифицировать с помощью метода `removeEventListener` решение предыдущей задачи так, чтобы после первого наведения курсора на ссылку от нее отвязывалось событие, которое добавляет в свойство `innerHTML` значение атрибута `href`.
3. Написать код, согласно которому при нажатии на каждый из инпутов с помощью `alert` выводится значение атрибута `value`, но только в случае первого нажатия на инпут. Последующие нажатия на инпут не должны приводить к вызову `alert`.
4. Даны абзацы (`<p>...</p>`) с числами. Написать код, согласно которому по клику на абзаце в нем должен появиться квадрат числа.
5. Даны 2 `input` с произвольным текстом (тексты в `input` не должны совпадать). Создать кнопку, при нажатии на которую `input` обменяются содержимым.
6. Дан `input` с произвольным текстом. Создать 2 кнопки «Скрыть» и «Вернуть», которые выполняют соответствующие команды, т.е. кнопка «Скрыть» должна скрывать инпут, а кнопка «Вернуть» должна возвращать инпут.
7. Даны 2 инпута и `span` (в примере ниже `span` введен как `<span>?</span>`). Реализовать программу, которая выполняет сложение чисел, вводимых пользователем в `input`.
8. Создать кнопку, при нажатии на которую содержимое абзацев поменяется на их порядковые номера в коде.
9. На главной странице в шапке сайта разместить ссылки: главная/новости/регистрация. На странице «новости» разместить информацию новостного характера. На странице «регистрация» разместить форму регистрации, использовать текстовые поля, радио, флажки, меню `select`, `textarea`, кнопки «Зарегистрироваться» и «Отменить».
10. На кнопке «Зарегистрироваться» написать обработчик события на языке JavaScript, отображающий введенные данные в форму на новой странице. На главной странице осуществить отображение даты и времени используя методы работы с датой и временем `javascript`

## План Лабораторной работы № 6

Цель – изучение концепций браузерного окружения, DOM-дерева, навигации по DOM-элементам. Реализация браузерных событий и генерация пользовательских событий.

Итог – освоение обучающимися концепций браузерного окружения, DOM-дерева, навыков реализации браузерных событий и генерации пользовательских событий.

## Вопросы к зачету

1. Строковое преобразование, численное преобразование, логическое преобразование.
2. Операторы ИЛИ (`||`), И (`&&`), НЕ (`!`).
3. Циклы **while** и **for**.
4. Конструкция **switch**.
5. Глобальные и локальные переменные. Особенности использования ключевого слова **var**.
6. Пользовательские функции: синтаксис Function Declaration и синтаксис Function Expression. Отличия Function Declaration от Function Expression.
7. Стрелочные функции. Многострочные стрелочные функции.
8. Объекты. Литералы и свойства. Квадратные скобки.
9. Объекты. Вычисляемые свойства. Свойство из переменной. Ограничения на имена свойств.
10. Проверка существования свойства, оператор **in**. Цикл **for...in**.
11. Копирование объектов по ссылке. Сравнение объектов. Объекты-константы.
12. Клонирование объектов. Метод **Object.assign**.
13. Конструкторы, создание объектов через **new**. Строковое, численное и логическое преобразования объектов.
14. Создание методов в конструкторе. Методы у примитивов. Примитив как объект.
15. Способы записи числа. Шестнадцатеричные, двоичные и восьмеричные числа. Метод **toString(base)**. Функции округления.
16. Методы строк. Сравнение строк.
17. Объявление массивов. Методы **pop/push, shift/unshift** и их эффективность.
18. Перебор элементов массива. Свойство **length**. Синтаксис **new Array()**. Реализация метода **toString** у массивов.
19. Методы **splice, slice, concat**.
20. Методы **forEach, indexOf, lastIndexOf, includes**.
21. Методы **find, findIndex, filter, map**.
22. Методы **sort, reverse, split, join**.
23. Итерируемые объекты и псевдомассивы. Метод **Array.from**.
24. Коллекция **Map**. Перебор **Map**. **Map** из объекта и объект из **Map**.
25. Коллекция **Set**. Перебор **Set**.
26. Деструктуризация массива. Остаточные параметры. Значения по умолчанию.
27. Деструктуризация объекта. Остаток объекта.
28. Объект **Date**. Создание объекта даты. Получение компонентов даты. Установка компонентов даты.
29. Автоисправление даты. Преобразование даты к числу, разность дат. Метод **Date.parse**.
30. Регулярные выражения. Флаги. Методы **match, replace, test**.
31. Символьные классы. Обратные символьные классы. Флаг **"u"**. Юникодные свойства.
32. Якоря. Многострочный режим якорей.
33. Экранирование символов. Особенности экранирования при использовании синтаксиса **new RegExp**.
34. Наборы. Диапазоны. Символьные классы в диапазонах. Исключающие диапазоны.
35. Квантификаторы **+, \*, ?, {n}**.

36. Жадный и ленивый виды поиска.
37. Скобочные группы.
38. Метод **matchAll**.
39. Именованные группы. Скобочные группы при замене. Исключение скобочной группы из запоминания.
40. Альтернатива. Отличия альтернативы от наборов. Альтернатива на примере шаблона для времени в формате 00:00–23:59.
41. Опережающая проверка. Негативная опережающая проверка. Ретроспективная проверка. Опережающая и ретроспективная проверки и скобочные группы.
42. Остаточные параметры функции. Переменная **arguments**.
43. Оператор расширения.
44. Лексическое окружение. Лексическое окружение в Function Declaration. Внутреннее и внешнее лексическое окружение.
45. Вложенные функции.
46. Рекурсия, контекст выполнения, стек.
47. Функции как объект. Свойства **name** и **length**. Пользовательские свойства.
48. Named Function Expression.
49. Формат JSON. Методы **JSON.stringify** и **JSON.parse**.
50. Флаги и дескрипторы свойств объекта.
51. Свойства-геттеры и свойства-сеттеры.
52. Свойство **[[Prototype]]**.
53. Свойство **prototype** в конструкторах объектов.
54. Встроенные прототипы.
55. Методы **Object.create**, **Object.getPrototypeOf**, **Object.setPrototypeOf**.
56. Синтаксис классов.
57. Class Expression. Геттеры, сеттеры, вычисляемые свойства в классах. Свойства классов.
58. Наследование классов.
59. Статические свойства и методы, наследование статических свойств и методов.
60. Приватные и защищенные методы и свойства.
61. DOM (Document Object Model). Пример DOM. Автоисправление.
62. Навигация по DOM-элементам. `documentElement` и `body`. `childNodes`, `firstChild`, `lastChild`.
63. DOM-коллекции. Навигация только по элементам.
64. Методы поиска элементов по `id`, тегу и классу. Живые коллекции.
65. Свойства `innerHTML`, `outerHTML`, `hidden`, `value`, `href`.
66. Браузерные события. Обработчики событий. Назначение обработчика. Доступ к элементу через `this`.