



**МИНОБРНАУКИ РОССИИ**  
федеральное государственное бюджетное образовательное учреждение  
высшего образования  
**«ИРКУТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**  
ФГБОУ ВО «ИГУ»  
Кафедра Алгебраических и информационных систем

**«УТВЕРЖДАЮ»**  
Директор ИМИТ ИГУ  
  
**М. В. Фалалеев**  
**«25» мая 2022 г.**  


### **Рабочая программа дисциплины**

**Наименование дисциплины (модуля)** Б1.В.22 Веб-технологии

**Направление подготовки** 09.03.03 Прикладная информатика

**Направленность (профиль) подготовки** Проектирование и разработка информационных систем

**Квалификация выпускника** – бакалавр

**Форма обучения** очная

Согласовано с УМК Института математики  
и информационных технологий  
Протокол № 3 от «04» апреля 2022 г.

Председатель \_\_\_\_\_  
  
Антоник В.Г.

Рекомендовано кафедрой Алгебраических и  
информационных систем ИМИТ ИГУ:  
Протокол № 9 От «24» марта 2022 г.

Зав. кафедрой \_\_\_\_\_  
  
Пантелеев В.И.

## СОДЕРЖАНИЕ

1.	Цели и задачи дисциплины	42.	Место дисциплины в структуре опоп во	43.
	Требования к результатам освоения дисциплины	44.	Содержание и структура дисциплины	64.1.
	СОДЕРЖАНИЕ ДИСЦИПЛИНЫ, СТРУКТУРИРОВАННОЕ ПО ТЕМАМ, С УКАЗАНИЕМ ВИДОВ УЧЕБНЫХ ЗАНЯТИЙ И ОТВЕДЕННОГО НА НИХ КОЛИЧЕСТВА АКАДЕМИЧЕСКИХ ЧАСОВ	64.2.	План внеаудиторной самостоятельной работы обучающихся по дисциплине	64.3.
	СОДЕРЖАНИЕ УЧЕБНОГО МАТЕРИАЛА	74.4.	МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ОРГАНИЗАЦИИ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ	94.5.
	ПРИМЕРНАЯ ТЕМАТИКА КУРСОВЫХ РАБОТ (ПРОЕКТОВ)	115.	Учебно-методическое и информационное обеспечение дисциплины (модуля)	126.
	Материально-техническое обеспечение дисциплины	127.	Образовательные технологии	138.
	Оценочные материалы для текущего контроля и промежуточной аттестации	13		

## 1. Цели и задачи дисциплины

### Цель

Формирование у студентов базовых знаний в области создания web-ресурсов, знакомство с базовыми клиентскими технологиями: HTML, CSS, Javascript.

### Задачи:

Обучение студентов технологии создания Интернет-ресурсов (web-сайтов) различных видов.

## 2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП ВО

2.1. Учебная дисциплина (модуль) относится к части программы, формируемой участниками образовательных отношений, и изучается на первом курсе.

2.2. Для изучения данной учебной дисциплины (модуля) необходимы знания, умения и навыки, сформированные: Информатика.

2.3. Перечень последующих учебных дисциплин, для которых необходимы знания, умения и навыки, формируемые данной учебной дисциплиной: Разработка веб-приложений, Веб-программирование на стороне клиента.

## 3. ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Процесс освоения дисциплины направлен на формирование компетенций (элементов следующих компетенций) в соответствии с ФГОС ВО по соответствующему направлению подготовки.

### Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с индикаторами достижения компетенций

Компетенция	Индикаторы компетенций	Результаты обучения
ПК-2 Способность разрабатывать и адаптировать прикладное программное обеспечение.	ИДК ПК2.1 Способен записывать программный код в соответствии с требуемой парадигмой программирования	Знает объектно-ориентированную парадигму программирования Умеет работать с объектами Владеет навыками работы с объектами
	ИДК ПК2.2 Способен выбирать и анализировать способы решения прикладных задач,	Знает гипертекстовый язык разметки HTML, каскадные таблицы стилей CSS, скриптовый язык программирования Javascript Умеет разрабатывать клиентские приложения с использованием современных технологий Владеет навыками верстки веб-страниц, навыками программирования поведения веб-страниц
	ИДК ПК2.3 Способен адаптировать и модифицировать программное обеспечение	Знает гипертекстовый язык разметки HTML, каскадные таблицы стилей CSS,

	в соответствии с требованиями	скриптовый язык программирования Javascript  Умеет разрабатывать клиентские приложения с использованием современных технологий Владеет навыками верстки веб-страниц, навыками программирования поведения веб-страниц
--	-------------------------------	---

#### 4. СОДЕРЖАНИЕ И СТРУКТУРА ДИСЦИПЛИНЫ

Объем дисциплины составляет 5 зачетных единиц, 180 часов, в том числе 44 часа на контроль, практическая подготовка 180.  
 Форма промежуточной аттестации: 2 семестр – экзамен.

##### 4.1. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ, СТРУКТУРИРОВАННОЕ ПО ТЕМАМ, С УКАЗАНИЕМ ВИДОВ УЧЕБНЫХ ЗАНЯТИЙ И ОТВЕДЕННОГО НА НИХ КОЛИЧЕСТВА АКАДЕМИЧЕСКИХ ЧАСОВ

№ п/п	Раздел дисциплины/темы	Се мес тр	Виды учебной работы, включая самостоятельную работу обучающихся и трудоемкость (в часах)				Самостоятельная работа + контроль		Формы текущего контроля успеваемости
			Контактная работа преподавателя с обучающимися			Контроль обучения			
			Лекции	Семинарские (практические занятия)					
1	Тема 1. Гипертекстовый язык разметки HTML	2		16	3	10	10	лаб.	
2	Тема 2. Каскадные таблицы стилей CSS	2		22	3	16	17	лаб.	
3	Тема 3. Скриптовый язык программирования Javascript	2		34	4	28	17	лаб.	
<b>Итого часов</b>			0	72	10	98			

##### 4.2. План внеаудиторной самостоятельной работы обучающихся по дисциплине

Семестр	Название раздела, темы	Самостоятельная работа обучающихся			Оценочное средство	Учебно-методическое обеспечение самостоятельной работы
		Вид самостоятельной работы	Сроки выполнения	Затраты времени (час.)		
2	Тема 1. Гипертекстовый язык разметки HTML	Выполнение практической работы	15.03	20	Проверка домашней работы	Материалы курса на платформе ИОС DOMIC
2	Тема 2. Каскадные таблицы стилей CSS	Выполнение практической работы	26.04	33	Проверка домашней работы	Материалы курса на платформе ИОС DOMIC
2	Тема 3. Скриптовый язык программирования Javascript	Выполнение практической работы	7.06	45	Проверка домашней работы	Материалы курса на платформе ИОС DOMIC
Общая трудоемкость самостоятельной работы по дисциплине (час)				98		
Из них объем самостоятельной работы с использованием электронного обучения и дистанционных образовательных технологий (час)				98		

### 4.3. СОДЕРЖАНИЕ УЧЕБНОГО МАТЕРИАЛА

Тема 1. Гипертекстовый язык разметки HTML

Браузеры, версии языка, перспективы развития, инструментарий. Структура HTML-документа: списки, ссылки, изображения, карты изображений, таблицы, формы, семантические теги, встроенные фреймы. Валидация документа.

Тема 2. Каскадные таблицы стилей CSS

Преимущества таблиц стилей, способы добавления стилей на страницу, типы носителей, базовый синтаксис CSS. Селекторы тегов, классы, идентификаторы, контекстные селекторы, соседние селекторы, дочерние селекторы, селекторы атрибутов, универсальный селектор, группировка селекторов. Псевдоклассы, определяющие состояние элементов, псевдоклассы, имеющие отношение к дереву документа, псевдоэлементы. Наследование, изменение свойств наследуемого элемента, особенности каскадирования. Позиционирование, фиксированная и «резиновая» верстки, блочная верстка. Верстка с использованием модуля Flexbox. Фреймворк Bootstrap, верстка на сетках Bootstrap. Возможности CSS3. Плагин Emmet.

Тема 3. Скриптовый язык программирования Javascript

Добавление скриптов на страницу, отладка, структура кода. Переменные и типы данных. Понятие объекта в Javascript. Объекты window, document, location, navigator, history, их свойства и методы. Объекты Array, Date, Math, Number, String, их свойства и методы. Объектная модель документа DOM. Библиотека jQuery. Событийно-ориентированное программирование на Javascript, технология AJAX, обработка файлов типа JSON и XML. ООП в Javascript. Объект WebStorage. Объект Promise и асинхронное программирование.

#### 4.3.1. Перечень семинарских, практических занятий и лабораторных работ

№ п/п	№ раздела и темы	Наименование семинаров, практических и лабораторных работ	Трудоемкость (час.)		Оценочные средства	Формируемые компетенции (индикаторы)*
			Всего часов	Из них практическая подготовка		
1	2	3	4	5	6	7
1	1	Тема 1. Гипертекстовый язык разметки HTML	16	16	Проверка лабораторной работы	ПК-2
2	2	Тема 2. Каскадные таблицы стилей CSS	22	22	Проверка лабораторной работы	ПК-2
3	3	Тема 3. Скриптовый язык программирования Javascript	34	34	Проверка лабораторной работы	ПК-2
		<b>Всего</b>	<b>72</b>	<b>72</b>		

#### 4.3.2. Перечень тем (вопросов), выносимых на самостоятельное изучение студентами в рамках самостоятельной работы (СР)

Тема	Задание	Формируемые компетенции
Тема 1. Гипертекстовый язык разметки HTML	Решить задачи для самостоятельного выполнения	ПК-2
Тема 2. Каскадные таблицы стилей CSS	Решить задачи для самостоятельного выполнения	ПК-2



Тема 3. Скриптовый язык программирования Javascript	Решить задачи для самостоятельного выполнения	ПК-2
---	---	------

#### 4.4. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ОРГАНИЗАЦИИ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

Во время изучения дисциплины студент посещает лекции, практические занятия, выполняет лабораторные задания, готовится к тестам, зачетам и экзаменам. Для каждого вида деятельности необходимо правильно организовать самостоятельную работу.

Лабораторное занятие. Лабораторные занятия по решению задач существенно дополняют лекции. В процессе анализа и решения задач студенты расширяют и углубляют знания, полученные из лекционного курса и учебников, приобретают умение применять общие закономерности к конкретным случаям.

Необходимо, чтобы студенты готовили теоретический материал, т.к. именно невыполнение этого требования приводит к неудаче при решении задач.

Несмотря на различие в видах задач, их решение можно проводить по следующему общему плану (некоторые пункты плана могут выпадать в некоторых конкретных случаях): а) прочесть внимательно условие задачи; б) посмотреть, все ли термины в условиях задачи известны и понятны (если что-то неясно, следует обратиться к учебнику, просмотреть решения предыдущих задач, посоветоваться с преподавателем); в) произвести анализ задачи, (нужно чётко понимать, в чем будет заключаться решение задачи); г) решить задачу; д) протестировать полученное решение на данных из примеров к задаче, а также на дополнительных данных.

Если задача не решена или «не решается», то необходимо еще раз вернуться к пунктам а) и б). Сколько раз нужно возвращаться к этим пунктам? Практика показывает, что не более десяти раз. Если и после этого задача «не решается», то можно попытаться найти решение этой или похожей задачи в различных источниках.

Домашнее задание. При выполнении домашнего задания необходимо просмотреть текст лекции, разобраться с новыми определениями, посмотреть задания, которые были выполнены на лабораторной работе и применить полученные знания для выполнения домашней работы.

Тест. В первую очередь постарайтесь узнать чего ждать от теста, какие примерно там будут задания. Если вам доступны образцы теста (как, например, при сдаче ЕГЭ), необходимо этим воспользоваться и ежедневно тренироваться.

Не оставляйте все на самый последний момент. Если будете постоянно готовиться к тесту, вы наверняка улучшите свои знания. Для этого составьте план на каждый день, чтобы правильно распределять свое время.

Делайте небольшие перерывы во время учебы. В промежутках можно дать себе небольшую физическую нагрузку. Мозг лучше всего работает, когда умственный труд сменяется физическим. Прогуляйтесь, побегайте, поиграйте в баскетбол, попинайте мяч – помимо стимуляции умственной деятельности, это снимет стресс.

Отдых и контроль над волнением — одни из главных составляющих успеха при подготовке к тесту. Часто ошибки совершаются только из-за стресса, который мешает сконцентрироваться и собраться. Чтобы быть отдохнувшим и расслабленным, соблюдайте составленный режим и старайтесь высыпаться.

Экзамен. На экзамене оцениваются: 1) понимание и степень усвоения теории; 2) методическая подготовка; 3) знание фактического материала; 4) знакомство с обязательной литературой; 5) умение приложить теорию к практике, решать практические задачи и т. д.; 6) логика,

структура и стиль ответа, умение защищать выдвигаемые положения. Но значение экзаменов не ограничивается проверкой знаний. Являясь естественным завершением работы студента, они способствуют обобщению и закреплению знаний и умений, приведению их в строгую систему, а также устранению возникших в процессе занятий пробелов.

Студенты готовятся к экзаменам по-разному. Одни из них прорабатывают лишь некоторые вопросы, выбранные наугад, другие стремятся запомнить весь материал подряд, не вникая глубоко в его суть. Работа при этом концентрируется на одном стремлении – сдать экзамен. Недостатки такой системы очевидны. Очевидно также, что подготовка не должна ограничиваться чтением лекционных записей. Первоначальные необработанные конспекты студента содержат факты, определения, выводы, сделанные преподавателем, но в них, как правило, слабо просматривается связующая идея курса, так как студент, записывая каждую лекцию в отдельности, редко способен сразу и достаточно точно уловить общую направляющую мысль. Поэтому конспект требует дополнительной обработки на основе использования учебников и рекомендованной литературы.

Существенные недостатки имеет и такой способ подготовки к экзаменам, как беглый просмотр всего материала. Он эффективен только на некоторых этапах планирования и закрепляющего повторения. Более надежный и целесообразный путь – это тщательная систематизация материала при вдумчивом повторении, установлении внутрисубъектных связей, увязке различных тем и разделов, закреплении путем решения задач.

Перед экзаменом назначается консультация. Цель ее – дать ответы на вопросы, возникшие в ходе самостоятельной подготовки. Хотелось бы обратить особое внимание на важность предэкзаменационных консультаций. Здесь студент имеет полную возможность получить ответ на все неясные ему вопросы. А для этого он должен проработать до консультации весь курс. Кроме того, преподаватель будет отвечать на вопросы других студентов, что будет для вас повторением и закреплением знаний. И еще очень важное обстоятельство: лектор на консультации, как правило, обращает внимание на те разделы, по которым на предыдущих экзаменах ответы были неудовлетворительными, а также фиксирует внимание на наиболее трудных разделах курса. Некоторые студенты не приходят на консультации либо потому, что считают, что у них нет вопросов к лектору, либо полагают, что у них и так мало времени и лучше самому почитать материал по конспекту или в учебнике. Это глубокое заблуждение. Никакая другая работа не сможет принести столь значительного эффекта накануне экзамена, как консультация преподавателя.

Подготовку к экзамену следует начинать с первого дня изучения дисциплины. Как правило, на лекциях подчеркиваются наиболее важные и трудные вопросы или разделы курса, требующие внимательного изучения и обдумывания. Нужно эти вопросы выделить и обязательно постараться разобраться в них, не дожидаясь экзамена, проработать их, готовясь к лабораторным занятиям, попробовать самостоятельно решить несколько типовых задач. И если, несмотря на это, часть материала осталась неувоенной, ни в коем случае нельзя успокаиваться, надеясь на то, что это не попадет на экзамене. Факты говорят об обратном: если те или другие вопросы курса не вошли в экзаменационный билет, преподаватель может их задать (и часто задает) в виде дополнительных вопросов. Точно такое же отношение должно быть выработано к вопросам и задачам, перечисленным в экзаменационной программе, выдаваемой студентам еще до экзамена. Обычно эти же вопросы и аналогичные задачи содержатся в экзаменационных билетах. Не следует оставлять без внимания ни одного раздела курса; если не удалось в чем-то разобраться самому, нужно обратиться к товарищам; если и это не помогло выяснить какой-либо вопрос до конца, нужно обязательно задать этот вопрос преподавателю на предэкзаменационной консультации. Чрезвычайно важно приучить себя к умению самостоятельно мыслить, учиться думать, понимать суть дела. Очень полезно после проработки каждого раздела восстановить в памяти

содержание изученного материала, кратко записав это на листе бумаги. Если этого не сделать, то большая часть материала останется не понятой, а лишь формально заученной, и при первом же вопросе экзаменатора студент убедится в том, насколько поверхностно он усвоил материал.

#### **4.5. ПРИМЕРНАЯ ТЕМАТИКА КУРСОВЫХ РАБОТ (ПРОЕКТОВ)**

Не предусмотрено.

## 5. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

### а) перечень литературы

#### Основная

1. Диков, А. В. Клиентские технологии веб-программирования: JavaScript и DOM : учебное пособие / А. В. Диков. — Санкт-Петербург : Лань, 2020. — 124 с. — ISBN 978-5-8114-4074-0. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/126934>. — Режим доступа: для авториз. пользователей.

+

#### Дополнительная

1. Побединский, Е. В. Проектирование веб-сайтов с использованием технологий PHP, HTML, CSS и WordPress : учебное пособие / Е. В. Побединский, В. В. Побединский. — Екатеринбург : УГЛТУ, 2018. — 115 с. — ISBN 978-5-94984-651-3. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/142518>. — Режим доступа: для авториз. пользователей. +
2. Курзыбова Я. В. Средства создания динамических Web-сайтов: учеб. пособие. / Я.В. Курзыбова – Иркутск: Изд-во Иркут. гос. ун-та. – 2011. – ISBN: 978-5-9624-0558-2. – 122 с. – (ЭБС «БиблиоТех»). +
3. Web-программирование JavaScript / Т. В. Зудилова. - Москва : СПбНИУ ИТМО (Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики), 2012. - Режим доступа: ЭБС "Издательства Лань". +
4. Джереми К. HTML5 для веб-дизайнеров / К. Джереми. – Москва : Манн, Иванов и Фербер, 2013. – 105 с. – Режим доступа: ЭБС "РУКОНТ". – ISBN 978-5-91657-596-5. +
5. Сидерхолм Д. CSS3 для веб-дизайнеров / Д. . Сидерхолм. – Москва : Манн, Иванов и Фербер, 2013. – 137 с. – Режим доступа: ЭБС "РУКОНТ". – ISBN 978-5-91657-595-8. +

### б) периодические издания

### в) список авторских методических разработок:

Материалы курса, опубликованные в ИОС «DOMIC».

### г) базы данных, информационно-справочные и поисковые системы \_\_\_\_\_

1. HTML Standard URL: <https://html.spec.whatwg.org/multipage/>
2. 2. Описания всех спецификаций CSS URL: <https://www.w3.org/Style/CSS/specs.ru.html>
3. 3. Javascript | MDN URL: <https://developer.mozilla.org/ru/docs/Web/JavaScript>

## 6. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

### 6.1. УЧЕБНО-ЛАБОРАТОРНОЕ ОБОРУДОВАНИЕ:

Для проведения лабораторных занятий необходима компьютерная аудитория с презентационным оборудованием, необходима аудитория на 25-30 рабочих мест (в зависимости от численности учебной группы), оборудованная доской, презентационной техникой.

### 6.2. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ:

1. Интегрированная среда разработки NetBeans IDE 12 (распространяется бесплатно, LGPLv2.1, GPLv2 with Classpatch exception).
2. Браузер Google Chrome.
3. Браузер Mozilla Firefox.

### 6.3. ТЕХНИЧЕСКИЕ И ЭЛЕКТРОННЫЕ СРЕДСТВА:

ИОС EDUCA, DOMIC, презентационное оборудование, персональный компьютер с возможностью демонстрации презентаций в формате pdf.

## 7. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

При реализации данного курса используются следующие образовательные технологии: технологии традиционного обучения, технологии проблемного обучения, технологии контекстного обучения, интерактивные технологии, технологии дистанционного обучения.

## 8. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

### 8.1. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ВХОДНОГО КОНТРОЛЯ

Не предусмотрены.

### 8.2. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ТЕКУЩЕГО КОНТРОЛЯ

Разноуровневые задания к лабораторному практикуму  
Примеры оценочных средств текущего контроля

#### 1. Лабораторная работа по теме «Верстка на Flexbox»

С помощью CSS-модуля **FlexBox** создайте **отзывчивую верстку** (% + медиа-запросы) согласно вашему варианту из предыдущей лабораторной работы «**Блочная верстка**», на основе заготовки кода из предыдущей работы.

Для страницы установите в значения параметров метатега **viewport**:

- ширину области отображения, совпадающую с шириной устройства, на котором просматривается страница,
- начальный коэффициент масштабирования 1,
- максимальный размер масштабирования страницы 130%.

Разработайте стили для устройств следующих типов:

- С минимальной шириной экрана **768px** согласно изображению в варианте. Размеры блоков выберите самостоятельно с соблюдением пропорций на рисунке.
- С максимальной шириной экрана **768px** для мобильного устройства. Протестируйте данный вариант верстки с помощью встроенных средств для разработчика в браузере (откройте панель разработчика и нажмите сочетание клавиш **Ctrl** + **Shift** + **M**), выберите устройство с подходящим разрешением экрана.

Размеры основных элементов задайте с помощью свойства **flex-basis**. Внутри жёлтого блока разместите элементы с использованием свойства **position**.

#### 2. Лабораторная работа по теме «Библиотека jQuery»

Создайте проект и поместите в папку **public\_html** [страницу](#) (скачать файлы в [архиве](#)).

Скачайте с официального сайта **jQuery** последнюю стабильную версию библиотеки, выберите сжатый (compressed) вариант, поместите файл библиотеки в папку проекта и подключите его к вашей HTML-странице.

С помощью библиотеки **jQuery** (без использования DOM), выполните следующие задания:

- в папке проекта создайте файл **rect.js** (ссылка на него уже прописана в **rect.html**), в этом файле должны находиться все функции и их вызовы.
- при загрузке страницы нужно создать заданное количество прямоугольников (элементов **div** с классом **"rectangle"**) и поместить их в элемент **div** с идентификатором **"rectanglearea"**. Прямоугольники должны располагаться в родительском блоке случайным образом, но не выходить за его границы (блоки могут перекрывать друг друга).
- в поле для ввода цвета можно вводить только значения цвета в шестнадцатеричном формате, например **3366FF**, **AC4394**, **fc0** (событие **keydown** в поле для ввода цвета). Также предусмотрена возможность удалить неверно введенные символы с помощью клавиш **Del** и **Backspace**. При вводе символов в указанное поле, нужно проверять, что они соответствуют алфавиту шестнадцатеричной системы счисления **0-9**, **a-f**. Для проверки можно использовать регулярное выражение **ch.match(/^[\a-f0-9]\$/i)**, где **ch** – символ, который пытается ввести пользователь. Если символ не соответствует регулярному выражению, то он не должен появиться в поле для ввода (для этого нужно отменить стандартное поведение браузера).
- при нажатии кнопки "Цвет" нужно прочитать цвет, записанный в поле для ввода цвета, и закрасить все прямоугольники в указанный цвет.
- при наведении курсора на один из прямоугольников, он должен изменить цвет фона на случайный цвет.
- при нажатии пользователем сочетания клавиш **ctrl** + **вверх**, все прямоугольники должны закраситься случайным образом (событие **keydown** для всего документа, код кнопки **вверх** = 38).
- (дополнительное задание) скачайте с официального сайта **jQuery UI** последнюю стабильную версию библиотеки, выберите сжатый (compressed) вариант, поместите файл библиотеки в папку проекта и подключите его к вашей HTML-странице (библиотеку **jQuery UI** нужно подключить после подключения библиотеки **jQuery**). С помощью **jQuery UI** сделайте возможным перетаскивание прямоугольников, но только внутри родительской области (используйте свойство **Draggable**).
- выполните задание по [варианту](#).
- при нажатии пользователем сочетания клавиш **shift** + **0**, прямоугольники выстраиваются последовательно внутри области-контейнера, заполняя всю область. Те прямоугольники, которые не поместились в область, удаляются.

### 8.3. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПРОМЕЖУТОЧНОГО КОНТРОЛЯ

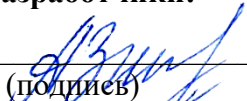
Список вопросов для промежуточной аттестации:

1. Структура HTML страницы
2. Элементы и атрибуты языка HTML

3. Понятие строчных и блочных элементов HTML
4. Семантические теги HTML
5. Веб-стандарты
6. Кодирование символов в HTML
7. Синтаксис CSS, группировка стилей
8. CSS-свойство для текста
9. CSS-свойства задания фона
10. Наследование стилей. Конфликты стилей
11. Способы подключения стилей
12. Аппаратно-зависимые стили
13. Понятие псевдоэлементов и псевдоклассов
14. Селекторы по идентификатору и по классу
15. Контекстные и соседние селекторы
16. Конфликты стилей и алгоритм вычисления специфичности селектора
17. Блочная модель CSS
18. Плавающие элементы
19. Виды позиционирования элементов
20. Понятие вендерного префикса и его использование
21. Функции трансформации в CSS
22. Модель FlexBox и ее свойства
23. Фреймворк Bootstrap
24. Язык программирования Javascript и его особенности
25. Синтаксис Javascript
26. Типы Javascript
27. Особенности сравнения значений в Javascript
28. Понятие событийно-ориентированного программирования на Javascript
29. Методы получения доступа к элементам (по идентификатору, по классу, по имени элемента, по имени элемента, по селектору)
30. Отладка кода на Javascript
31. Ненавязчивый JavaScript
32. Понятие анонимной функции и анонимного объекта
33. Объектная модель документа (DOM = Document Object Model)
34. Свойства DOM-объекта
35. Настройка стилей с помощью DOM
36. Глобальные объекты DOM
37. События таймера
38. Функции как объекты первого класса
39. Библиотека jQuery
40. Обращение к узлам в jQuery
41. Сравнение идентификации в DOM и jQuery
42. Объект jQuery
43. Идентификация в контексте в DOM и jQuery
44. Методы обхода дерева в DOM
45. Использование метода each() в jQuery
46. Доступ к стилям в jQuery
47. Создание новых узлов в DOM и jQuery
48. Изменение дерева в DOM и jQuery
49. Сигнатура функции jQuery \$
50. События JavaScript
51. Установка обработчиков событий с помощью jQuery
52. Объект event в jQuery
53. Событие мыши и объект события мыши

54. События страницы и окна
55. События клавиатуры и объект события клавиатуры
56. События форм
57. Отмена действия браузера по умолчанию
58. Высплывающие события
59. Остановка всплывающего объекта события
60. Отмена выполнения следующих обработчиков текущего события
61. Синхронная модель веб-приложения
62. Объект XMLHttpRequest и его ограничения безопасности
63. Асинхронная модель веб-приложения
64. Схема выполнения типичного AJAX-запроса
65. Методы jQuery \$.ajax(), \$.get(), \$.post() и их опции
66. Реализация обратной связи AJAX с пользователем
67. Понятие формата XML
68. Получение и обработка XML с помощью AJAX
69. Взаимодействие с узлами XML DOM
70. Свойства и методы XML DOM
71. Понятие формата JSON
72. Методы браузера для работы с JSON
73. Способы хранения данных на клиенте (в браузере)
74. HTML5 Web Storage
75. Объекты localStorage и sessionStorage и работа с ними
76. Событие onStorage и объект storageEvent

**Разработчики:**

  
\_\_\_\_\_  
(подпись)                      \_\_\_\_\_                      \_\_\_\_\_                      Зинченко А.С. \_\_\_\_\_  
(занимаемая должность)                      (Ф.И.О.)

Программа составлена в соответствии с требованиями ФГОС ВО по направлению подготовки 09.03.03 «Прикладная информатика» (уровень бакалавриата), утвержденный приказом Министерства образования и науки Российской Федерации от «19» сентября 2017 г. № 922, зарегистрированный в Минюсте России «12» октября 2017 г. № 48531 с изменениями и дополнениями от 26.11.2020 г., 8.02.2021 г.

Программа рассмотрена на заседании кафедры Алгебраических и информационных систем ИМИТ ИГУ «24» марта 2022 г.

Протокол № 9 Зав. кафедрой \_\_\_\_\_ Пантелеев В.И.

*Настоящая программа, не может быть воспроизведена ни в какой форме без предварительного письменного разрешения кафедры-разработчика программы.*