

# ОСНОВНАЯ ПРОФЕССИОНАЛЬНАЯ ОБРАЗОВАТЕЛЬНАЯ ПРОГРАММА ПОДГОТОВКИ БАКАЛАВРА

по направлению

**09.03.01 Информатика и вычислительная техника профиль  
"Технологии разработки программного обеспечения"**

**Б. 1.10.3 Модуль "Проектирование и разработка веб-решений".  
Разработка интерфейсов обогащенных веб-приложений**

## Приложение 1

### **Типовые задания для проведения процедур оценивания результатов освоения дисциплины в ходе текущего контроля, шкалы и критерии оценивания**

#### **Содержание**

1. [Типовые задания лабораторных работ по темам](#)
2. [Типовые задания для инвариантной самостоятельной работы по темам](#)
3. [Типовые задания для вариативной самостоятельной работы по темам](#)

#### **1. Типовые задания лабораторных работ по темам**

Задания лабораторных работ направлены на отработку действий, способов, методов решения задач в области веб-проектирования и веб-дизайна, которыми обучающиеся должны владеть.

Задания для лабораторных работ предполагают преимущественно репродуктивный характер действий обучающихся и рассчитаны на выполнение в рамках определенного количества часов лабораторных занятий (Таблица 3). Задания представлены в СДО Moodle в электронном учебном курсе по дисциплине «Разработка интерфейсов обогащенных веб-приложений» и могут иметь одну или несколько из следующих форм:

- краткая формулировка постановки задачи с пояснениями;
- пошаговая инструкция;
- скринкаст.

Далее приводятся ссылки на веб-страницы на сервере Кодактор.ру (например, [https://kodaktor.ru/g/xml\\_intro](https://kodaktor.ru/g/xml_intro)), содержащие более детальные инструкции к заданиям и поясняющие скринкасты.

Выполнение задания предполагает следующие виды деятельности:

- разработку сценария или приложения на языке разметки XML или веб-языке (языке программирования высокого уровня в области веб-ресурсов):
  - JavaScript;
  - ActionScript;
  - и, возможно, другом языке по выбору обучающихся;
- выявление и исправление синтаксических ошибок, выполнение транспилиции, тестирования работы сценария.
- составление отчета о выполненном задании в виде слайдов и/или онлайн-документации, размещаемой в Git-репозитории как части веб-портфолио обучающегося.

**Критерий оценивания.** Лабораторная работа считается выполненной, если XML-документ или веб-сценарий разработан и соответствует заданию, не содержит синтаксических ошибок, а также сопровождается репозиторием (в том числе, возможно,

отчётом в форме слайдов). Часть заданий проверяется (или дополнительно проверяется) с помощью средств автоматизированной проверки, отсылающей запросы к веб-сценариям. Веб-сценарии должны быть размещены на ресурсе, допускающем обращение через Интернет (онлайн-редактор типа Кодактор.ру или codepen.io):

<https://kodaktor.ru/g/z7a>

Secure | <https://kodaktor.ru/g/z7a>

## Автопроверка требования z7a

<h4 id="author" title="GossJS">Иван Иванов</h4> должны помещаться имя и фамилия исполнителя

В случае серверных приложений имя и фамилия исполнителя должны выдаваться ещё и по маршруту GET /author  
Выдача по маршруту /author должна быть с CORS

URL

GossJS?  Author

/author

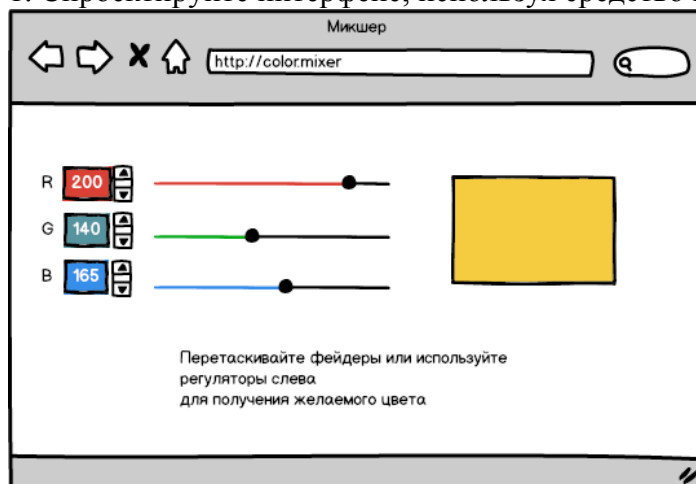
Выполнение заданий самого простого вида в Кодактор.ру: <https://kodaktor.ru/dzjs>

### Тема 1. Интерфейсы HTML5 API и их взаимодействие с языком JavaScript

I. Проектирование и разработка интерфейса микшера цветов - инструмента для подбора цветовых сочетаний для веб-страницы

Цель: создать интерактивный интерфейс с использованием элементов управления HTML5 Shadow DOM таких как `input type="number"` и `input type="range"`

1. Спроектируйте интерфейс, используя средство прототипирования:



2. Создайте веб-страницу (борд) на Кодактор.ру или ином онлайн-редакторе
3. Посмотрите, как должен работать интерфейс в динамике, используя борд <https://kodaktor.ru/g/mixer>
4. Задайте минимальные и максимальные значения элементов управления
5. Используя обработчики событий элементов управления, синхронизируйте их между собой так, чтобы изменение значения элемента, отвечающего за конкретный цвет, вызывало реактивное изменение второго, парного ему элемента - и наоборот. Например, перемещение фейдера, задающего синий цвет, должно вызывать

изменение значения соответствующего числового счётчика в левой части интерфейса.

6. Изменение значений элементов управления должно сопровождаться изменением фонового цвета какого-либо прямоугольника на веб-странице либо всей веб-страницы.

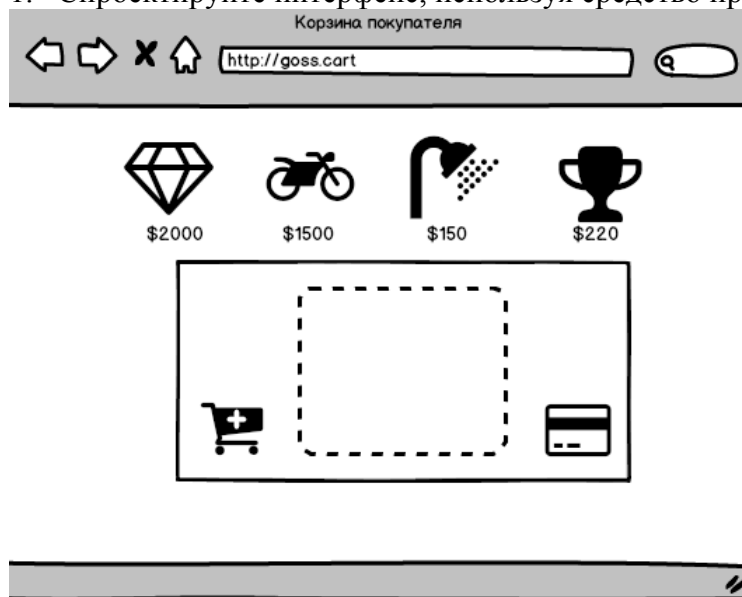
6\*. Добавьте текстовые поля, в которых отображаются доступные для копирования RGB-представления цветов в десятичном и шестнадцатеричном виде

7. Разместите отчёт (лог действий) в репозитории (веб-портфолио).

## II. Проектирование и разработка интерфейса корзины покупателя в интернет-магазине

Цель: создать интерактивный интерфейс с использованием интерфейса перетаскивания, реализованного в HTML5

1. Спроектируйте интерфейс, используя средство прототипирования:



2. В верхней части корзины должна быть представлена галерея доступных товаров с ценниками. Пусть бюджет покупателя ограничен суммой в \$2000.

3. Товары должны добавляться в корзину путём перетаскивания их изображений в область перетаскивания, обозначенную пунктирной рамкой. При этом должен вестись подсчёт суммарной стоимости заказанных товаров

4. При превышении указанного выше лимита перетаскивание не должно завершаться успехом, вместо этого в таком случае должно выдаваться сообщение «Кредит превышен!»

5. Элементы, содержащие товары и «корзина» должны быть реализованы в виде независимых друг от друга компонента.

Используйте интерфейс CustomEvent для связывания компонент между собой путём передачи сообщений и реализации модели событие-слушатель.

6. Используйте атрибуты draggable, ondragstart, ondragover и ondrop (<https://kodaktor.ru/drop4>) для реализации перетаскивания.



7. Разместите отчёт (лог действий) в репозитории (веб-портфолио).

## Тема 2. *Использование интерфейса Canvas для создания графических веб-интерфейсов*

### I. Разработка логотипа для веб-портфолио с помощью интерфейса Canvas

1. Создайте веб-страницу (борд) на Кодактор.ру или ином онлайн-редакторе
2. Вставьте элемент script и определение функции

```
function makeCanvas(x, y) {
  const canvas = document.createElement('canvas'),
    ctx = canvas.getContext('2d');
  canvas.setAttribute('width', x);
  canvas.setAttribute('height', y);
  return { canvas, ctx };
}
```

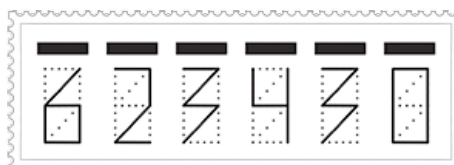
3. Вызовите функцию для создания переменных, содержащих холст и контекст

```
const { canvas, ctx } = makeCanvas(300, 120);
```

4. Добавьте холст в дерево DOM

```
document.body.appendChild(canvas);
```

5. Используя методы контекста, изобразите свои инициалы, используя в качестве образца написание в стиле почтового индекса:



6. Используя кривую Безье, нарисуйте под инициалами дугу другого цвета:

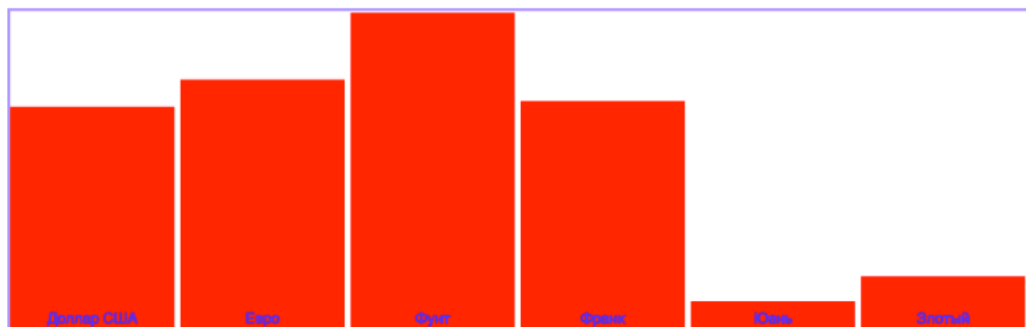


7. Используя аффинные преобразования (<https://kodaktor.ru/affine>), осуществите поворот этого изображения на 90 градусов против часовой стрелки.
8. Создайте форк этого борда (поставив на него ссылку в первом и обратно), в котором этот логотип используется как фоновый узор.
9. Разместите отчёт (лог действий) в репозитории (веб-портфолио).

### **Тема 3. Использование интерфейса Canvas или графической библиотеки GD2 языка PHP для визуализации инфографики**

Цель: создать столбчатую диаграмму для визуализации <https://kodaktor.ru/j/rates>

1. Возьмите за основу борд [https://kodaktor.ru/canvas\\_1](https://kodaktor.ru/canvas_1)
2. Создайте кнопку «Рисовать» и добавьте к ней единожды срабатывающий асинхронный слушатель события, который должен извлекать имена и цены валют из документа ([https://kodaktor.ru/js01\\_26bc1](https://kodaktor.ru/js01_26bc1))
3. Примите за 100% самую высокую цену и пусть по соглашению соответствующий столбец диаграммы занимает всю высоту холста
4. Получите количество валют и пусть ширина каждого столбца будет равна ширине холста, разделённой на это количество
5. Расположите столбцы растущими с нижней кромки холста
6. Расположите названия валют у дна каждого столбца по центру столбца



7. Обеспечьте выполнение требования z7a
8. Разместите отчёт (лог действий) в репозитории (веб-портфолио).

Проверочный код: <https://kodaktor.ru/30>

### **Тема 4. Разработка реактивных веб-интерфейсов на основе языка JSX и React-компонентов.**

I. Развёртывание простого приложения на основе React с компонентным интерфейсом в браузере с использованием транспилятора babel, получающего данные из JSON-ресурса

Пример работы React-приложения в браузере: [https://kodaktor.ru/min\\_react\\_live](https://kodaktor.ru/min_react_live)

1. Создайте веб-страницу (борд) на Кодактор.ру или ином онлайн-редакторе, в которой есть только пустой элемент `<div></div>` в разделе body
2. Подключите react, react-dom и babel
3. Пишите код в элементе `<script type="text/babel"></script>`
4. Главным компонентом должен быть список **VoteButtonList** кнопок для голосования, в каждую из которых передаётся название фреймворка и количество актуальных голосов.

Подключите компонент в точку монтирования

```
ReactDOM.render(  
  <VoteButtonList url="https://kodaktor.ru/j/react5b_6cbf2"/>,  
  document.querySelector('div')  
);
```

передав ему в параметре url адрес документа

[https://kodaktor.ru/j/react5b\\_6cbf2](https://kodaktor.ru/j/react5b_6cbf2)



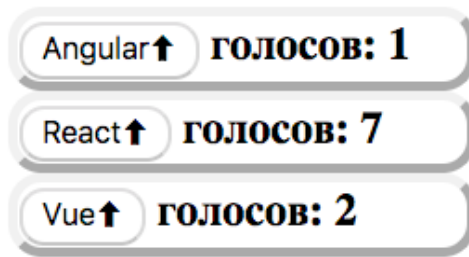
описывающего результаты голосования разработчиков при выборе наиболее перспективного фреймворка (библиотеки) для JavaScript. Необходимо извлечь массив данных в событии жизненного цикла компонента `ComponentDidMount` и сохранить в состоянии главного компонента

```
componentDidMount() {  
  fetch(this.props.url)  
    .then(x => x.json())  
    .then(frameworks => this.setState({ frameworks }));  
}
```

или в асинхронной форме

```
async componentDidMount() {  
  const frameworks = await fetch(this.props.url).then(x => x.json());  
  this.setState({ frameworks });  
}
```

5. Напишите код компонента `VoteButton`, в состоянии которого хранится количество голосов, отданных за тот фреймворк, которому этот компонент соответствует. Каждый такой компонент должен содержать кнопку, увеличивающую хранимое внутри его состояния количество голосов. Вот пример списка из трёх компонентов `VoteButton`



6. Разместите адрес борда (а также его скриншот и лог действий) в репозитории (веб-портфолио) и в специально созданном форуме в соответствующем курсе в СДО Moodle.

Проверочный код: [https://kodaktor.ru/min\\_react\\_live2](https://kodaktor.ru/min_react_live2)

## II. Создание интерфейса MaterialUI на React-платформе

Цель: создать клиентский интерфейс с компонентом Date Picker (<http://www.material-ui.com/#/components/date-picker>) для реализации вычисления разницы между текущей датой и выбранной с помощью компонента

1. Создайте проект `mkdir $(date +%Y%m%d_%H%M%S) && cd $_ && yarn init -y`
2. Установите линтер и его файл настройки:  
`bash <(curl -s https://kodaktor.ru/g/eslint_exec)`
3. Установите минимально необходимые зависимости, включая девелоперские:  
`yarn add react react-dom moment react-tap-event-plugin material-ui`  
`yarn add --dev webpack webpack-cli webpack-dev-server babel-core babel-loader babel-preset-env babel-preset-react`
3. Создайте файл настройки транспилятора:  
`echo ' {"presets":["env"]}' > .babelrc`
4. Создайте файл настройки webpack  
`curl 'https://kodaktor.ru/j/min_react_webpack4' -o 'webpack.config.babel.js'`

```

1  const WDS_PORT = 1234;
2
3  export default {
4    mode: 'development',
5    devtool: 'sourcemap',
6    resolve: { extensions: ['.js', '.jsx'] },
7    module: {
8      rules: [{
9        test: /\.jsx$/,
10       loader: 'babel-loader',
11       query: {
12         babelrc: false,
13         presets: [
14           'react', ['env', {
15             modules: false,
16           }],
17         ],
18       },
19     }],
20   },
21   devServer: {
22     port: WDS_PORT,
23     host: '0.0.0.0',
24   },
25 };

```

5. Добавьте запускающий скрипт в файл package.json

```

"scripts": {
  "start": "webpack-dev-server"
},

```

6. В папке ./src создайте файл index.jsx

и поместите в него JavaScript-сценарий, созданный в результате выполнения предыдущей лабораторной работы

7. В корне приложения создайте файл index.html, в котором содержатся строк  
 <div></div><script src="./main.js"></script>

8. Проверьте работу приложения, запустив его

**yarn start**

и перейдя по адресу localhost:1234 в браузере

9. Для работы с элементами интерфейса MaterialUI нужно импортировать их

```

import MuiThemeProvider from 'material-ui/styles/MuiThemeProvider';
import DatePicker from 'material-ui/DatePicker';
import injectTapEventPlugin from 'react-tap-event-plugin';

```

и

вставить метку, при щелчке по которой элемент будет появляться на веб-странице:

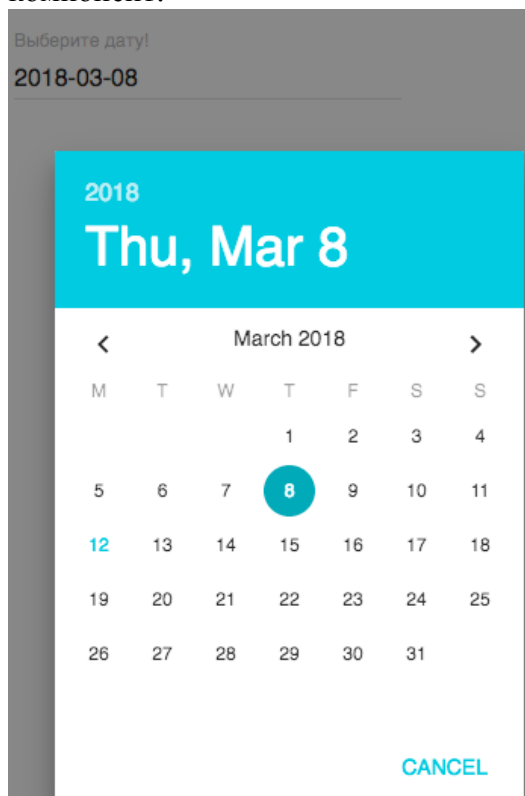


```

<MuiThemeProvider>
  <DatePicker
    onChange={(n = null, date) => alert(date)}
    floatingLabelText="Выберите дату!"
    autoOk={true}
  />
</MuiThemeProvider>

```

При щелчке по надписи «Выберите дату» должен отображаться искомый компонент:



10. Добавьте код, который с помощью `moment` получает текущую дату и вычисляет разницу в годах, месяцах и днях между ней и той, которая выбрана с помощью установленного компонента.

11. Разместите приложение в Интернете, а его адрес (а также скриншот и лог действий) в репозитории (веб-портфолио) и в специально созданном форуме в соответствующем курсе в СДО Moodle.

Контрольный код к шагам до 10:

[https://github.com/GossJS/react\\_starters1/tree/voters\\_material](https://github.com/GossJS/react_starters1/tree/voters_material)

## 2. Типовые задания для инвариантной самостоятельной работы по темам

Задания для самостоятельной инвариантной работы обобщают и расширяют задания лабораторных работ, преимущественно предполагают поисковую деятельность в аспекте модификация и совершенствования алгоритмов и моделей, предложенных в лабораторных работах и лекционном материале по теме. Эти задания нацелены на формирование видов деятельности, которые обучающиеся должны уметь осуществлять. Задания представлены в СДО Moodle в электронном учебном курсе по дисциплине «Разработка интерфейсов обогащенных веб-приложений».

Выполнение задания предполагает следующие виды деятельности:

- поиск информации в Интернете по изучаемым технологиям и инструментам, анализ справочных материалов и документации,
- развёртывание и настройку окружения (программной среды, IDE, папки проекта, репозитория);
- проектирование формата данных веб-приложения или формального языка (языка разметки), основанного на XML;
- разработку сценария или приложения на веб-языке (языке программирования высокого уровня в области веб-ресурсов):
  - JavaScript;
  - PHP;
  - и, возможно, другом языке по выбору обучающихся;
- выявление и исправление синтаксических ошибок, выполнение транспилиции, тестирования работы сценария.
- составление отчета о выполненном задании в виде слайдов и/или онлайн-документации, размещаемой в Git-репозитории как части веб-портфолио обучающегося.

**Критерии оценивания.** Задание считается выполненным, если сценарий или отчёт соответствует заданию, сценарий успешно проходит процедуру модульного тестирования и сопровождается репозиторием.

№ темы	Содержание самостоятельной работы обучающихся
1	<ol style="list-style-type: none"><li>1. Создание интерфейса проекта «Микшер цветов веб-страницы»</li><li>2. Разработка проекта «Интерфейс корзины покупателя»</li><li>3. Разработка приложения с реализацией перетаскивания и анимации.</li><li>4. Формирование отчета по выполнению задания (тема 1) и размещение его в портфолио.</li></ol>
2	<ol style="list-style-type: none"><li>1. Разработка приложения на основе интерфейса Canvas</li></ol>

	<p>2. Разработка приложения с использованием фреймворка Fabric.</p> <p>3. Разработка приложения на основе интерфейса fetch и интерфейса потоков.</p> <p>4. Формирование отчета по выполнению задания (тема 2) и размещение его в портфолио.</p>
3	<p>1. Разработка приложения для визуализации инфографики</p> <p>2. Разработка приложения для реализации CAPTCHA</p> <p>3. Разработка приложения с подключением к базе данных с помощью объектно-ориентированного интерфейса.</p> <p>4. Формирование отчета по выполнению задания (тема 3) и размещение его в портфолио.</p>
4	<p>1. Разработка интерфейса приложения с использованием реактивных компонентов на языке JSX, а также мультимедийных компонентов.</p> <p>2. Разработка приложения на основе интерфейса MaterialUI</p> <p>3. Сборка проекта с помощью бандлера и с использованием транспилятора.</p> <p>4. Развёртывание приложения на деплой-платформе.</p> <p>5. Формирование отчета по выполнению задания (тема 4) и размещение его в портфолио.</p>

### 3. Типовые задания для вариативной самостоятельной работы по темам

Задания для вариативной самостоятельной работы позволяют углубить и детализировать важные аспекты содержания дисциплины, представляющие интерес для каждого конкретного обучающегося, включая углубление и детализацию теоретической подготовки и формирование исследовательской деятельности, для реализации чего предназначены задания на подготовку выступлений, докладов и аналитики.

Для выполнения этих заданий, например, проектирования дизайна главной страницы мобильного приложения, требуется выполнение следующих действий:

- анализ программных средств и инструментов, их документации;
- прототипирование (моделирование) веб-ресурса;
- анализ и использование фреймворков, логирование

При подготовке выступления следует руководствоваться следующей дорожной картой презентации:

- обзор по теме (сравнение, таблица, ... - слайды в google drive или инструменте вещания слайдов);
- демонстрация в live-режиме (slides.com, например <http://slides.com/elizabethanatskaya-1/deck-2#/12> и др.);
- выводы;
- примеры заданий для аудитории на овладение материалом (возможно, опрос, одним словом, интерактив);
- поддержка в репозитории (ссылки на слайды / ресурсы / ...).

**Критерии оценивания.** Задание считается выполненным, если сценарий или отчёт соответствует заданию, сценарий успешно проходит процедуру модульного тестирования и сопровождается репозиторием, выступление логично и последовательно охватывает заявленную тему.

№ темы	Содержание самостоятельной работы обучающихся
1	1. Подготовка сообщения и мастер-класса по интерфейсу Web Audio 2. Подготовка сообщения и мастер-класса по интерфейсу Drag-and-Drop
2	1. Подготовка выступления и мастер-класса по сравнению характеристик Canvas и SVG 2. Подготовка презентации и мастер-класса по интерфейсу WebGL
3	1. Подготовка презентации и мастер-класса по сравнению характеристик GD2 и Canvas 2. Подготовка материалов для вебинара по разработке Captcha

4	<ol style="list-style-type: none"> <li>1. Разработка интерфейса веб-ресурса на базе библиотеки React и подготовка сообщения и мастер-класса по этой теме.</li> <li>2. Разработка интерфейса веб-ресурса на базе фреймворка Angular и подготовка сообщения и мастер-класса по этой теме.</li> </ol>
---	--

# ОСНОВНАЯ ПРОФЕССИОНАЛЬНАЯ ОБРАЗОВАТЕЛЬНАЯ ПРОГРАММА ПОДГОТОВКИ БАКАЛАВРА

по направлению

**09.03.01 Информатика и вычислительная техника**

**профиль "Технологии разработки программного обеспечения"**

**Б. 1.10 Модуль "Проектирование и разработка веб-решений".**

**Приложение 3**

**Типовые задания для проведения процедур оценивания результатов освоения модуля  
в ходе промежуточного контроля**

## Содержание

- [1. Процедура промежуточной аттестации](#)
- [2. Типовые вопросы к экзамену по модулю](#)

### 1. Процедура промежуточной аттестации

Промежуточная аттестация по модулю направлена на оценивание теоретических знаний основных понятий дисциплин, входящих в модуль, базовых алгоритмов и методов веб-программирования и практической составляющей – готовности разрабатывать компоненты аппаратно-программных комплексов, алгоритмы, структуры данных и веб-приложения для решения различных классов задач. Аттестация осуществляется в виде экзамена в форме набора заданий, которые обучающиеся выполняют на платформе Moodle, сопровождаемого защитой веб-портфолио.

### Перечень результатов обучения по модулю

Обучающийся должен демонстрировать владение следующими компетенциями: ПК-2 – способностью разрабатывать компоненты аппаратно-программных комплексов и баз данных, используя современные инструментальные средства и технологии программирования.

### 2. Типовые вопросы к экзамену по модулю

Это задания открытого или фасетного типа. Они формулируются однозначно, так чтобы ответом могли быть:

- ключевое слово языка, в том числе пропущенное слово,
- число или строка, получающиеся в результате работы алгоритма,
- номер строки в сценарии, содержащей подлежащую исправлению ошибку и т.п.

Билет содержит два вопроса, в зависимости от типа вопроса ответ должен содержать:

- описание соответствующего алгоритма или структуры данных;
- описание процедуры валидации / тестирования / развёртывания веб-приложения.

Типовые задания представлены в следующем перечне.

N	Формулировка	Ответ
1.	Какой тег языка HTML5 позволяет внедрять в веб-страницы видеоконтент?	video

2.	Какой тег языка XSLT позволяет применить подготовленный шаблон к подходящим элементам XML-разметки?	<xsl:apply-templates/>
3.	Какой тип поля input в языке HTML5 позволяет вызвать стандартный диалог выбора цвета?	color
4.	Какое свойство каскадных стилей отвечает за скругление границ объектов?	border-radius
5.	В инструкции инициализации литерала массива квадратными скобками определите и выпишите символ, завершающий литерал массива.  let r = ['first', 'second', 'third']; // массив r	]
6.	Назовите имя переменной-счётчика в инструкции цикла for (let c=0;c<d;c++)	c
7.	Назовите встроенный объект языка JavaScript, который инкапсулирует математическую функциональность (его методы sqrt() или floor())	Math
8.	Какое значение должно быть возвращено при вычислении данного выражения: (x=>y=>x<<y)(5)(3)	40
9.	Выберите примитивные типы из списка:  Выберите один или несколько ответов:  Function Array <b>Number</b> <b>Symbol</b> <b>Boolean</b>	Number ; Symbol ; Boolean
10.	Назовите метод класса Function, позволяющий вызвать функцию, переопределив её this-контекст и передав её массив аргументов.	apply
11.	Назовите бинарный оператор метапрограммирования JavaScript, позволяющий проверить, что левый операнд является экземпляром второго операнда-как-класса	instanceof
12.	Какое значение в JavaScript не равно само себе?	NaN

13.	Какой тип структуры возвращает метод <code>document.querySelectorAll</code> ?	<code>NodeList</code>
14.	Назовите статический класс (объект) современного JavaScript, позволяющий переопределять в рамках метапрограммирования операции по отношению к объектам, используя <code>Proxy</code> в качестве обёртки вокруг каждого конкретного объекта.	<code>Reflect</code>
15.	Пусть массив <code>m</code> есть <code>[3,2]</code> ; Сколько различных элементов будет в массиве <code>[1,2, ...m, 1]</code> ?	3
16.	Какое значение возвращает функция в JavaScript, если в определении функции не указано явно, что она возвращает?	<code>undefined</code>
17.	Назовите класс DOM/ JavaScript для создания собственных пользовательских событий (сообщений, обрабатываемых слушателями событий типа 'click')	<code>CustomEvent</code>
18.	Что является прототипом объекта, созданного инициализатором литерала объекта?	<code>Object.prototype</code>
19.	Что в языке JavaScript является результатом извлечения квадратного корня из отрицательного числа?	<code>NaN</code>
20.	Сколько раз в консоль будет выведена единица? <code>for(let i=0;i&lt;3;++i){for(let i=0;i&lt;4;++i){console.log(1)}}</code>	12
21.	Какова сумма элементов массива, получаемого в результате вычисления выражения <code>[1,2,3].map(x=&gt;x*x)</code>	14
22.	Назовите унарный оператор JavaScript, который, будучи поставлен перед выражением (в том числе выражением вызова функции), превращает его значение в <code>undefined</code> .	<code>void</code>
23.	Какое значение будет вычислено при исполнении этого выражения: <code>((a=2,b=10)=&gt;Math.pow(a,b))()</code>	1024
24.	Назовите имя синхронной функции подключения внешних файлов в Node.js, включая зависимости и файлы JSON.	<code>require</code>



25.	Назовите стандартное имя файла (с расширением) дескрипторов пакетов CommonJS, в котором задаются зависимости для современного веб-приложения на JavaScript	package.json
26.	Какое ключевое слово ES2015 позволяет обозначить экспортируемую сущность как доступную для импорта по умолчанию?	default
27.	Какая папка содержит в Node.js-проекте зависимости, загруженные с помощью менеджера пакетов?	node_modules
28.	Какой метод протокола HTTP позволяет передавать данные от клиента серверу в открытом виде (в том числе в адресной строке), с использованием вопросительного знака для отделения запроса от основной части URL?	GET
29.	Назовите метод объекта response (ответ сервера, res) в Node.js, который позволяет одновременно вывести текст клиенту и завершить соединение.	end
30.	Как называется объект, являющийся свойством объекта Request, который позволяет обратиться к параметрам строки запроса в URL? (т.е. того, что начинается с вопросительного знака) Впишите одно английское слово.	query
31.	Раскодируйте последовательность русских букв в нижеследующем адресе: <a href="https://ru.wikipedia.org/wiki/%D0%A1%D0%B0%D0%B9%D1%82">https://ru.wikipedia.org/wiki/%D0%A1%D0%B0%D0%B9%D1%82</a>	Сайт
32.	На какой знак необходимо заменить вопросительный знак в нижеследующем заголовке HTTP, чтобы он был корректным заголовком, предшествующим отсылке данных в формате JSON? Content-Type: application?json; charset=utf-8	/
33.	Назовите аббревиатуру из 3 латинских букв, обозначающую подход к разработке приложений, заключающийся в строгой необходимости разрабатывать тесты перед написанием собственно кода.	TDD
34.	Назовите метод объекта console (одно слово без скобок), позволяющий сверить два значения, например, на равенство, и выводящий сообщение об ошибке в случае невыполнения этого «утверждения»	Assert

35.	В терминологии разработчиков программного обеспечения [_____] тестирование – это то, которое осуществляется на уровне очень мелких структурных единиц, тестирование отдельных компонентов для проверки того, работают ли они должным образом, в то время как интеграционное тестирование тестирует взаимодействие многих компонентов или даже всей системы	Модульное Юнит unit
-----	--	---------------------------

#### Шкала критериев оценивания экзамена по модулю

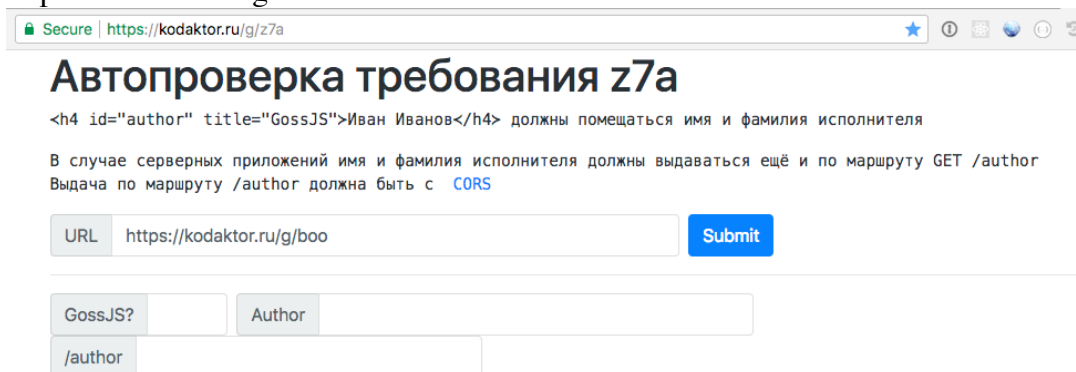
Шкала	Критерии
отлично	<b>обучающийся должен:</b> продемонстрировать сформированность требуемой компетенции – знания изучаемых аспектов языков веб-программирования и веб-интерфейсов в полном объёме: дать корректные ответы на 85% тестовых заданий и более, распознавать операторы и структуры данных веб-языков, исправлять ошибки в языковых конструкциях, быть способным корректно сформулировать определения, воспроизвести по запросу информацию о функциональности всех изучаемых средств разработки и проектирования веб-приложений, должно быть выполнено 85% и более заданий лабораторных работ, нацеленных на разработку компонентов программно-аппаратных комплексов, заданий для самостоятельной работы, представленных в виде корректно функционирующего веб-портфолио.
хорошо	<b>обучающийся должен:</b> продемонстрировать знания изучаемых аспектов языков веб-программирования в достаточном объёме: дать корректные ответы не менее чем на 70% тестовых заданий, распознавать основные компоненты веб-языков, воспроизвести по запросу информацию о функциональности основных средств разработки и проектирования веб-приложений, должно быть выполнено не менее 70% заданий лабораторных работ, заданий для самостоятельной работы, представленных в виде функционирующего веб-портфолио.
удовлетворительно	<b>обучающийся должен:</b> продемонстрировать способность в целом ориентироваться в структуре изучаемого материала: дать корректные ответы не менее чем на 55% тестовых заданий; уметь объяснить назначение веб-языков и работу стандартных компонентов веб-приложений; показать общее владение понятийным аппаратом дисциплины. У обучающегося должны присутствовать представления об основных управляющих конструкциях, операторах и структурах данных, должно быть выполнено не менее 60% заданий лабораторных работ, заданий для самостоятельной работы, отражённых в веб-портфолио.

неудовлетворительно	<b>обучающийся:</b> не знает значительной части программного материала (ответил менее чем на 55% тестовых заданий); не владеет понятийным аппаратом дисциплины; не способен продемонстрировать знание ни одной синтаксической конструкции веб-языков и не способен оценить корректность работы веб-сценария; выполнено менее 60% заданий лабораторных работ, заданий для самостоятельной работы, которые не отражены в веб-портфолио.
---------------------	---

### Типовая процедура защиты веб-портфолио

**Критерий оценивания.** Задание в составе веб-портфолио выполненной, если XML-документ или веб-сценарий разработан и соответствует заданию, не содержит синтаксических ошибок, а также сопровождается репозиторием (в том числе, возможно, отчётом в форме слайдов). Часть заданий проверяется (или дополнительно проверяется) с помощью средств автоматизированной проверки, отсылающей запросы к веб-сценариям. Веб-сценарии должны быть размещены на ресурсе, допускающем обращение через Интернет (онлайн-редактор типа Кодактор.ру или codepen.io):

<https://kodaktor.ru/g/z7a>



Типовые вопросы к экзамену по модулю представлены в следующем перечне.

1. Основные стандарты в области веб-технологий. Языки XML, HTML5, SVG.
2. Элементы клиентской шаблонизации
3. Основы синтаксиса JavaScript. Переменные, операторы, литералы и выражения.
4. Функции JavaScript
5. Управление зависимостями
6. Экосистема JavaScript / Node.JS
7. Слушатели событий, коллбэки (обратные вызовы). Использование AJAX для создания запросов.
8. Асинхронное программирование и неблокирующий код. Промисы.
9. Строки и массивы. Мутабельность.
10. Объекты и формат JSON.
11. Перечислимость ключей. Собственные свойства объекта.
12. Глобальный объект и объект window. Взаимодействие с хостом и с DOM.
13. Управление API с помощью JavaScript. Создание элементов audio и video.

14. Пользовательские (кастомные) события DOM и их обработка по модели «подписчик-издатель».
15. Перетаскивание элементов с помощью интерфейса Drag - drop.
16. Понятие холста и контекста.
17. Программная вставка холста в дерево DOM.
18. Основные характеристики и конструкции языка PHP. Сравнение с JavaScript. Приём данных в виде GET и POST-запросов
19. Функциональные и классные компоненты. Язык JSX. Жизненный цикл проекта на React и жизненный цикл React-компонента.
20. Одностраничное приложение (SPA)