

1.) " * + + " , &

Дисциплина «Разработка web – приложений» относится к дисциплинам по выбору учебного плана направления подготовки 09.03.03 Прикладная информатика. Она изучается на 3 курсе в 5 семестре и посвящена разработке web – приложений, содержащих алгоритмы, реализуемые средствами языка Java Script и фреймворка React, и ориентирована на овладение технологией создания web-сайта средствами программирования на стороне клиента и сервера. Для изучения данной дисциплины необходимы компетенции студентов, сформированные при изучении таких дисциплин, как «Основы проектирования сайтов», «Разработка и стандартизация программных средств и информационных технологий», «Языки и методы программирования». Рассматриваются возможные подходы к разработке веб - приложений, предназначенных для построения корпоративных информационных систем современных организаций и предприятий, обеспечивающих возможность удаленного доступа к данным и коллективной работы сотрудников. Дисциплина «Разработка web – приложений» обеспечивает проведение технологической практики студентов на предприятиях.

Изучение курса основано на традиционных методах высшей школы, тесной взаимосвязи со смежными курсами, обобщающими методологию исследований и проектирования корпоративных информационных систем.

2. & + " " " + - + " "

Компетенция	Индикаторы достижения
<p>& . %.Способен создавать прототипы автоматизированных информационных систем (ERP-систем), разрабатывать программный код информационной системы и баз данных информационной системы для управления бизнес-процессами, создавать прикладное программное обеспечение</p>	<p>/ -0 современные языки программирования, их синтаксис, языки программирования и работы с базами данных, теорию баз данных, инструменты и методы тестирования характеристик ИС и прототипирования пользовательского интерфейса, возможности типовой ИС, ее устройство и функционирование, основы современных операционных систем, систем управления базами данных.</p> <p>1 " -: кодировать на языках программирования, тестировать результаты прототипирования, тестировать разрабатываемую ИС (модульное, интеграционное тестирование), обнаруживать и устранять несоответствия и дефекты.</p> <p>" -: навыками по созданию прикладного программного обеспечения, разработке прототипов ИС, разработке кода ИС и баз данных ИС, тестирования ИС, устранения обнаруженных несоответствий и дефектов.</p>

%. 2 " ! "

В дисциплине «Разработка web – приложений» рассматриваются следующие темы.

1. Технологии создания web-сайта. Технологии стороны клиента. Сценарии JavaScript.
2. Основные идеи JavaScript. Структура JavaScript программы.
3. Базовые элементы языка JavaScript. Основные объекты языка. Понятие операторов. Переменные. Базовые типы String, Number, Boolean.
4. Операции в JavaScript. Управляющие структуры и организация циклов. Функции. Объектная модель сценария.
5. Работа со строками JavaScript. Работа с числами - объект Math, функции (методы глобального объекта) parseInt(), parseFloat()).

6. Массивы и объекты JavaScript. Функции и объекты. Управляющие конструкции языка
7. Иерархия объектов браузера. Объект window. Свойства, методы и события объекта window. Создание сценария.
8. Внедрение сценария в HTML. Типовые примеры использования JavaScript-сценариев.
9. Обработка события в JavaScript. События в динамическом HTML. Связывание кода с событиями. Объектная модель браузера и документа.
10. Объект document. Свойства, методы и события объекта document. Объект screen. Свойства, методы и события объекта screen.
11. Технологии создания web-сайта на стороне сервера. Сценарии и обработка событий. Node.js (JavaScript).
12. Фреймворк React, ReactDOM и JSX.
13. Функциональные компоненты React. Свойства компонентов.
14. Родительские и дочерние компоненты React.
15. Компоненты React, основанные на классах.
16. Состояния компонентов React. Модификация состояния компонентов.
17. Обработка событий React.

3. 4" "

№ п/п	Разделы и темы	Всего часов	Формы занятий			
			лекции	практические занятия	лабораторные занятия	самостоятельная работа
1	Технологии создания web-сайта. Технологии стороны клиента. Сценарии JavaScript.	6	–	–	4	2
2	Основные идеи JavaScript. Структура JavaScript программы.	6	–	–	4	2
3	Базовые элементы языка JavaScript. Основные объекты языка. Понятие операторов. Переменные. Базовые типы String, Number, Boolean.	6	–	–	4	2
4	Операции в JavaScript. Управляющие структуры и организация циклов. Функции. Объектная модель сценария.	6	–	–	4	2
5	Работа со строками JavaScript. Работа с числами - объект Math, функции (методы глобального объекта) parseInt(), parseFloat().	6	–	–	4	2
6	Массивы и объекты JavaScript. Функции и объекты. Управляющие конструкции языка	6	–	–	4	2
7	Иерархия объектов браузера. Объект window. Свойства, методы и события объекта window. Создание сценария.	6	–	–	4	2
8	Внедрение сценария в HTML. Типовые примеры использования JavaScript-сценариев.	6	–	–	4	2
9	Обработка события в JavaScript. События в динамическом HTML. Связывание кода с событиями. Объектная модель браузера и документа.	6	–	–	4	2
10	Объект document. Свойства, методы и	6	–	–	4	2

	события объекта document. Объект screen. Свойства, методы и события объекта screen.					
11	Технологии создания web-сайта на стороне сервера. Сценарии и обработка событий. Node.js (JavaScript).	6	–	–	4	2
12	Фреймворк React, ReactDOM и JSX.	7	–	–	4	3
13	Функциональные компоненты React. Свойства компонентов.	7	–	–	4	3
14	Родительские и дочерние компоненты React.	7	–	–	4	3
15	Компоненты React, основанные на классах.	7	–	–	4	3
16	Состояния компонентов React. Модификация состояния компонентов.	7	–	–	4	3
17	Обработка событий React.	7	–	–	4	3
	(4 , 5 ,	106		7	86	30

9. * " - # " " -

: "

ЛБ 1-2 «JS. Работа с переменными»

ЛБ 3-4 «JS. Работа со строками»

ЛБ 5-6 «JS. Обращение к символам строки»

ЛБ 7-8 «Работа с DOM в JavaScript»

ЛБ 9-10 «Математические калькуляторы на JavaScript»

ЛБ 11-12 «Анализатор текста на JavaScript»

ЛБ 13-14 «Игра угадай число на JavaScript»

ЛБ 15-16 «Линейный календарь на JavaScript»

ЛБ 17-18 «Поиск исторических события на JavaScript»

ЛБ 19-20 «Календарь на JavaScript»

ЛБ 21-22 «Установка фреймворка React»

ЛБ 23-24 «Компонентный подход в React»

ЛБ 25-26 «Язык JSX в React»

ЛБ 27-28 «React. Применение условий в JSX»

ЛБ 29-30 «React. Теги в массивах и циклах JSX»

ЛБ 31-32 «React. Навешивание событий в JSX»

ЛБ 33-34 «React. Вывод массива объектов в виде HTML таблицы»

Задания для лабораторных работ, размещены в системе дистанционного обучения СмолГУ (www.moodle.smolgu.ru). На занятиях для каждой работы задание предоставляется студентам в электронном виде.

2 " -

Текущая самостоятельная работа студента направлена на углубление и закрепление знаний студентов, развитие практических умений. Она заключается в работе с лекционными материалами, поиске и обзоре литературы и электронных источников, информации по заданным темам курса, опережающей самостоятельной работе, в изучении тем, вынесенных на самостоятельную проработку, подготовке к лабораторным занятиям.

Самостоятельная внеаудиторная работа студентов включает:

- проработку лекционного материала, составление конспекта лекций по темам, вынесенным на самостоятельное изучение;
- выполнение домашних заданий;
- подготовку к защите лабораторных работ.

1. Обработка форм средствами JavaScript.
2. Структура формы.
3. Верификация формы.
4. Ввод цвета на Web-форму.
5. Применение стилей к цвету на Web-форме.
6. Ввод – вывод даты на Web-формах.
7. Улучшение web-форм.

Консультирование студентов осуществляется в индивидуальном порядке на занятиях и во внеурочное время. Выполнение самостоятельной работы оценивается по электронным материалам, подготовленным студентами. Результаты деятельности накапливаются в индивидуальных портфолио студентов.

1 " " " " " " " " " -

1. Электронный вариант лабораторных работ по курсу.
2. Интернет-Университет Информационных Технологий – <http://www.intuit.ru>
3. Каталог образовательных Internet-ресурсов – <http://window.edu.ru>.
4. библиотека разработчика Microsoft – <http://msdn.microsoft.com>.
5. Веб-программирование – <http://codeacademy.ru>.
6. Веб-программирование с азов – <http://appleinsider.ru/developer/web-programmirovaniye-s-azov-vvedenie.html>.

8. . " " * " + - * * " ; + <
8.1. , " " " * " " * " +="# " "

4" " " " *

1. Технологии создания web-сайта. Технологии стороны клиента. Сценарии JavaScript.
2. Основные идеи JavaScript. Структура JavaScript программы.
3. Базовые элементы языка JavaScript. Основные объекты языка. Понятие операторов. Переменные. Базовые типы String, Number, Boolean.
4. Операции в JavaScript. Управляющие структуры и организация циклов. Функции. Объектная модель сценария.
5. Работа со строками JavaScript. Работа с числами - объект Math, функции (методы глобального объекта) parseInt(), parseFloat().
6. Массивы и объекты JavaScript. Функции и объекты. Управляющие конструкции языка.
7. Иерархия объектов браузера. Объект window. Свойства, методы и события объекта window. Создание сценария.
8. Внедрение сценария в HTML. Типовые примеры использования JavaScript-сценариев.
9. Обработка события в JavaScript. События в динамическом HTML. Связывание кода с событиями. Объектная модель браузера и документа.
10. Объект document. Свойства, методы и события объекта document. Объект screen. Свойства, методы и события объекта screen.
11. Технологии создания web-сайта на стороне сервера. Сценарии и обработка событий. Node.js (JavaScript).
12. Фреймворк React, ReactDOM и JSX.
13. Функциональные компоненты React. Свойства компонентов.
14. Родительские и дочерние компоненты React.
15. Компоненты React, основанные на классах.
16. Состояния компонентов React. Модификация состояния компонентов.
17. Обработка событий React.

1. Нормы оценивания ответов на теоретические вопросы

№ п/п	Теоретический вопрос	Количество баллов (*)
1	Дан краткий ответ на поставленный вопрос	1 балл
2	Дан развернутый ответ на вопрос с анализом результатов	2 балла

(*) Возможна градация в 0,25 балла.

2. Шкала оценивания. Оценка «зачтено» за ответы на теоретические вопросы выставляется, если набрано не менее 3 баллов при ответе на три вопроса, в противном случае выставляется «не зачтено».

: 1 2 ?@A. " " " В > #
/ 0

- Создайте переменную `z` и присвойте ей значение 3. Выведите значение этой переменной на экран с помощью метода `alert`.
- Создайте переменные `CD10` и `BD2`. Выведите на экран их сумму, разность, произведение и частное (результат деления).
- Создайте переменные `ED19` и `FD2`. Просуммируйте их, а результат присвойте переменной `GHJK`. Выведите на экран значение переменной `GHJK`.
- Создайте переменные `CD10`, `BD2` и `ED9`. Выведите на экран их + +.
- Создайте переменные `CD1` и `BD10`. Отнимите от `CD1` переменную `BD10` и результат присвойте переменной `E`. Затем создайте переменную `F`, присвойте ей значение `L`. Сложите переменные `E` и `F`, а результат запишите в переменную `GHJK`. Выведите на экран значение переменной `GHJK`.

: % 3 ?@A. В
/ 0

- Создайте переменную `str1` и присвойте ей значение 'Привет, Мир!'. Выведите значение этой переменной на экран.
- Создайте переменные `str1='Привет, Мир!'` и `str2='Мир!'`. С помощью этих переменных и операции сложения строк выведите на экран фразу 'Привет, Мир!'.
• Создайте переменную `name` и присвойте ей ваше имя. Выведите на экран фразу 'Привет, %Имя%!'.
• Создайте переменную `age` и присвойте ей ваш возраст. Выведите на экран 'Мне %Возраст % лет!'.
• Спросите имя пользователя с помощью метода `prompt`. Выведите с помощью `alert` сообщение 'Ваше имя %имя%'.
• Спросите у пользователя число. Выведите с помощью `alert` квадрат этого числа.

: 9 8 ?@A. , = " " * В
/ 0

- Создайте переменную `JKG` и присвойте ей значение 'abcde'. Обращаясь к отдельным символам этой строки выведите на экран символ 'a', символ 'c', символ 'e'.
- Создайте переменную `MIN` и присвойте ей значение '12345'. Найдите произведение (умножение) цифр этого числа.
- Напишите скрипт, который считает количество секунд в " , в + > , в " " .
- Создайте три переменные - , + , " + . С их помощью выведите текущее время в формате 0 0 + 0 " + 0 .
- Создайте переменную, присвойте ей число. Возведите это число в квадрат. Выведите его на экран.

: L 6 ? PQR * @CSCAEGTUKB
/ 0

- Выведите сообщение по нажатию на кнопку.
- Повторите поведение кнопки по нажатию на нее (она меняет текст в инпуте).
- Повторите поведение кнопки по нажатию на нее (она осуществляет обмен содержимым между двумя инпутами, можете понажимать на нее несколько раз или вручную сменить содержимое инпутов).
- Повторите поведение кнопки по нажатию на нее (поменяется ее текст).
- Повторите поведение кнопки по нажатию на нее (она меняет цвет текста в инпуте).
- Напишите сценарий на языке Javascript, позволяющий для изображения на web-странице менять ширину и высоту изображения, создавать рамку вокруг изображения, менять ее толщину и цвет, задавать альтернативный текст.
- Напишите сценарий на JavaScript, который реализует обмен рисунков на web-странице. Пусть на web-странице расположено четыре изображения, пронумерованных от 1 до 4. В текстовых полях указываются номера рисунков, которые необходимо поменять местами. Требуется, чтобы после нажатия на кнопку "Поменять местами" изображения переместились на нужные места.

: \$ 10 ?) " " " - + @CSCAEGTUK»

Задание 1

Напишите скрипт, который будет находить корни квадратного уравнения. Для этого сделайте 3 инпута, в которые будут вводиться коэффициенты уравнения.

Задание 2

Даны 3 инпута. В них вводятся числа. Проверьте, что эти числа являются тройкой Пифагора: квадрат самого большого числа должен быть равен сумме квадратов двух остальных.

Задание 3

Дан инпут и кнопка. В инпут вводится число. По нажатию на кнопку выведите список делителей этого числа.

Задание 4

Даны 2 инпута и кнопка. В инпуты вводятся числа. По нажатию на кнопку выведите список общих делителей этих двух чисел.

Задание 5

Даны 2 инпута и кнопка. В инпуты вводятся числа. По нажатию на кнопку выведите наибольший общий делитель этих двух чисел.

Задание 6

Даны 2 инпута и кнопка. В инпуты вводятся числа. По нажатию на кнопку выведите наименьшее число, которое делится и на одно, и на второе из введенных чисел.

: 11 12 ? V " @CSCAEGTUKB

Задание 1

Дан `textarea`. Пусть в него вводится текст. Сделайте так, чтобы по потере фокуса под `textarea` вывелось сообщение о том, сколько в этом тексте слов.

Задание 2

Модифицируйте предыдущую задачу так, чтобы также вывелось сообщение о том, сколько в тексте символов.

Задание 3

Модифицируйте предыдущую задачу так, чтобы также вывелось сообщение о том, сколько в тексте символов за вычетом пробелов.

Задание 4

Модифицируйте предыдущую задачу так, чтобы также вывелось сообщение о процентном содержании каждого символа в тексте.

Задание 5

В предыдущих задачах мы сделали так, что для нашего текста выводятся 4 параметра. Сделайте 4 чекбокса, которые будут регулировать, какие именно параметры показывать.

: 1% 13 ? (+ # @CSCAEGTUKB

Задание 1. Сейчас мы с вами реализуем игру *угадай число*. В этой игре компьютер загадывает число от 1 до 100. В инпут на экране игрок вводит число от 1 до 100, пытаясь угадать, что же загадал компьютер.

Если игрок ввел число, меньше загаданного, компьютер должен написать '
' , а если введено больше загаданного, то, соответственно, компьютер должен написать '
' . Реализуйте описанную игру.

Задание 2. Давайте теперь реализуем игру *угадай ячейку*. В этой игре будет дана таблица 10 на 10. Компьютер случайным образом запоминает 10 ячеек из этой таблицы. Игроку нужно кликать на клетки пока он не найдет все загаданные компьютером клетки.

: 19 18 ? : "# # " - @CSCAEGTB

Задание 1

Выведите в виде списка все числа текущего месяца. Средствами CSS поставьте пункты списка в ряд.

Задание 2

Сделайте так, чтобы текущий день в календаре был выделен каким-нибудь цветом.

Задание 3

Сделайте так, чтобы над списком было написано название текущего месяца по-русски и номер года.

Задание 4

Сделайте так, чтобы над календарем появились ссылки вперед и назад, позволяющие менять месяц. Месяц и год, выводимые над календарем, должны соответствовать отображаемому месяцу.

: 1L 16?& " > @CSCAEGTUK»
/ "0

Пусть мы хотим написать сайт, показывающий таблицу исторических событий за определенный год. Пусть год будет вводится в инпут. По нажатию Enter давайте выведем таблицу событий, случившихся в этот год. Пусть в качестве информации о событии мы хотим видеть дату, название и описание события.

: 1\$ 20 ? . " - @CSCAEGTUKB
/ "0

В данном разделе мы с вами реализуем календарь. Для простоты просто сделаем так, чтобы календарь выводился за текущий месяц. Но с прицелом на то, что его можно будет доработать так, чтобы можно было менять месяц и год. Вот образец того, что у нас должно получиться:

пн	вт	ср	чт	пт	сб	вс
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

: 21 22? 1 * ' "# * WeCEKB

Задание 1

Установите фреймворк React.

Для установки фреймворка вам понадобятся NodeJS и npm на вашем компьютере. Для создания проекта в командной строке выполните следующие команды:

```
npm create-react-app my-app
```

```
cd my-app
```

```
npm start
```

После этого у вас появится папка my-app, содержащая в себе фреймворк. Найдите в этой папке папку src - это будет ваша рабочая папка, в которой вы будете вести разработку вашего проекта. В папке src найдите файл App.js. Этот файл будет вашим основным рабочим файлом. В нем вы будете писать код, наблюдая его результаты в окне браузера.

Откройте этот файл в редакторе и удалите из него все лишнее, приведя его вот к такому виду: `import React from 'react';`

```
function App() {  
  return <div>  
    text  
  </div>;  
}
```

```
export default App;
```

После того, как ваш проект установлен, для следующего запуска (например, после перезагрузки компьютера) вам достаточно будет перейти через терминал в папку my-app и выполнить следующую команду:

```
npm start
```

Установите в ваш браузер специальный плагин react-devtools, помогающий вести разработку на React.

Задание 2

Используя конструкцию ReactDOM.render, выведите в браузере произвольный текст.

```
: 2% 23? . " # > * WeCEKB  
/  
"
```

Организуите возврат кода из функционального компонента.

В папке !"#%&'()*+,-./:;<=>?@A-Z[a-z]0-9 в файле <code>(ex)</code> создайте и расположите макет сайта. Вы можете размещать в нем любой HTML код - и вы увидите результат этого кода в браузере. Кроме того, в макете сайта есть специальный див <code>с</code>, равным <code>г++</code>, в который <code>монтируется</code> основной компонент. Мод монтированием понимается то, что в этот див будет выводиться результат работы нашего компонента.

В див с результатом будет выведено то, что возвращает через <code>return</code> функция компонента. В следующем примере это будет див с текстом:

```
function App() {  
  return <div>  
    text  
  </div>;  
}
```

```
: 29 28? X @AY*WeCEKB
```

Задание 1

Поменяйте текст внутри дива. Посмотрите на изменения, произошедшие в браузере.

Задание 2

Добавьте в див несколько абзацев.

Задание 3

Добавьте в див несколько инпутов, разделенных тегами `br`.

Задание 4

Сделайте внутри дива список `ul`, содержащий в себе 10тегов `li`.

Задание 5

Сделайте внутри дива таблицу с тремя рядами и тремя колонками.

Задание 6

Сделайте внутри дива три абзаца с различными CSS классами.

```
: 2L 26 ?WeCEK. & " " " + * # * @AYB
```

Задание 1

Пусть в константе `isAdult` содержится `true`, если пользователю уже есть 18 лет, и `false`, если нет:

```
function App() {  
  const isAdult = true;  
}
```

Сделайте так, чтобы в зависимости от значения `isAdult` на экране показался или один абзац с текстом, или другой.

Задание 2

Пусть в константе `isAdmin` содержится `true`, если пользователь админ, и `false`, если нет:

```
function App() {  
  const isAdmin = true;  
}
```

Сделайте так, чтобы, если `isAdmin` имеет значение `true`, на экране показался див с абзацами. В противном случае ничего показывать не нужно.

```
: 2$ %0 ?WeCEK. 4" * * > > @AYB
```

Задание 1

Дан массив:

```
function App() {  
  const arr = [  
    <li>1</li>,  
    <li>2</li>,  
    <li>3</li>,  
    <li>4</li>,  
    <li>5</li>,  
  ];  
}
```

С помощью этого массива получите результатом рендеринга следующий код:

```
<ul>  
  <li>1</li>  
  <li>2</li>  
  <li>3</li>  
  <li>4</li>  
  <li>5</li>  
</ul>
```

Задание 2

Сформируйте следующий код с помощью цикла `, + r`:

```
<ul>  
  <li>1</li>  
  <li>2</li>  
  <li>3</li>  
  <li>4</li>  
  <li>5</li>  
</ul>
```

```
: %1 %2 ?WeCEK. Z *"[ * " # * @AYB
```

Задание 1

Дан следующий код:

```
functionApp() {  
  function show1() {  
    alert(1);  
  }  
  
  function show2() {  
    alert(2);  
  }  
  
  return <div>  
    <button>act1</button>  
    <button>act2</button>  
  </div>;  
}
```

Сделайте так, чтобы по клику на первую кнопку срабатывала первая функция, а по клику на вторую кнопку - вторая функция.

Задание 2

Даны три кнопки:

```
functionApp() {  
  return <div>  
    <button>act1</button>  
    <button>act2</button>  
    <button>act3</button>  
  </div>;  
}
```

Сделайте так, чтобы по клику на первую кнопку алертом выводилось число 1, по клику на вторую кнопку - число 2, а по клику на третью - число 3.

```
: %% %3 ?WeCEK. * * \" * * * \" ] ^ R _ B  
/ \"0
```

В компоненте App дан следующий массив:

```
const users = [  
  {id: 1, name: 'user1', surn: 'surn1', age: 30},  
  {id: 2, name: 'user2', surn: 'surn2', age: 31},  
  {id: 3, name: 'user3', surn: 'surn3', age: 32},  
];
```

Выведите элементы этого массива в виде таблицы `table` так, чтобы каждое поле объекта попало в свой `td`. Сделайте заголовки колонок вашей таблицы.

```
. " " * * " >
```

1. Нормы оценивания каждой лабораторной работы:

№п/п	Структурная часть работы	Количество баллов (*)
1	Ответ на теоретические вопросы по теме лабораторной работы	1 балл
2	Демонстрация выполнения конкретного задания, предложенного для самостоятельного решения к лабораторной работе	2 балла

(*) с возможностью градации до 0,25 балла.

2. Шкала оценивания. Оценка «зачтено» за лабораторную работу выставляется, если набрано не менее 2 баллов, в противном случае за работу выставляется «не зачтено».

8.2. , " " " * " " * " !+ # "

/ " - ; " <

1. Напишите сценарий на языке Javascript, позволяющий для изображения на web-странице менять ширину и высоту изображения, создавать рамку вокруг изображения, менять ее толщину и цвет, задавать альтернативный текст.
2. Средствами React выведите элементы массива в виде таблицы table так, чтобы каждое поле объекта попало в свой тег td.

. " " * " # - #

1. Нормы оценивания работы

№ п/п	Структурная часть контрольной работы	Количество баллов (*)
1	Правильно реализован каждый метод решения	1 балл
2	Анализ результатов	2 балла

(*) Возможна градация в 0,25 балла.

2. Шкала оценивания работы:

п/п	Оценка	Количество баллов
1	Отлично	4,75-5
2	Хорошо	3,75-4,5
3	Удовлетворительно	3-3,5
4	Неудовлетворительно	менее 3

. " # + " "

Зачет выставляется по результатам работы студента в течение семестра согласно Положению о текущем контроле успеваемости и промежуточной аттестации обучающихся в федеральном государственном бюджетном образовательном учреждении высшего профессионального образования «Смоленский государственный университет» (утверждено приказом ректора № 01-113 от 26.09.2019 г.; внесены дополнения приказом ректора № 01-48 от 30.04.2020).

Для получения зачета студент должен:

- уметь отвечать на теоретические вопросы, рассмотренные на лекциях;
- уметь решать задачи, предложенные на лабораторных занятиях;
- уметь решать задачи, предложенные на зачетной контрольной работе.

L. &" " " - * # " - # + " # " +

L.1. , * " +

1. Грекул В. И. Проектирование информационных систем: учебник и практикум для вузов / В. И. Грекул, Н. Л. Коровкина, Г. А. Левочкина. – Москва: Издательство Юрайт, 2022. – 385 с. – (Высшее образование). – ISBN 978-5-9916-8764-5. – URL: <https://urait.ru/bcode/450997>
2. Зыков С. В. Программирование. Объектно-ориентированный подход: учебник и практикум для вузов / С. В. Зыков. – Москва: Издательство Юрайт, 2022. – 155 с. – (Высшее образование). – ISBN 978-5-534-00850-0. – URL : <https://urait.ru/bcode/451488>
3. Лаврищева Е. М. Программная инженерия и технологии программирования сложных систем: учебник для вузов / Е. М. Лаврищева. – 2-е изд., испр. и доп. – Москва: Издательство Юрайт, 2022. – 432 с. – (Высшее образование). – ISBN 978-5-534-07604-2. – URL: <https://urait.ru/bcode/452137>

4. Лаврищева Е. М. Программная инженерия. Парадигмы, технологии и CASE-средства: учебник для вузов / Е. М. Лаврищева. – 2-е изд., испр. – Москва: Издательство Юрайт, 2022. – 280 с. – (Высшее образование). – ISBN 978-5-534-01056-5. – URL: <https://urait.ru/bcode/452156>
5. Соколова В. В. Разработка мобильных приложений: учебник и практикум для вузов / В. В. Соколова. – Москва: Издательство Юрайт, 2022. – 175 с.
6. Черткова Е. А. Программная инженерия. Визуальное моделирование программных систем: учебник для вузов/ Е. А. Черткова. – 2-е изд., испр. и доп. – Москва: Издательство Юрайт, 2022. – 147 с. – (Профессиональное образование). – ISBN 978-5-534-09823-5. – Текст: электронный // Образовательная платформа Юрайт [сайт]. – URL: <https://urait.ru/bcode/493226>.

L.2. " - " +

1. Зудилова Т.В. Web-программирование JavaScript [Электронный ресурс] / Т.В. Зудилова, М.Л. Буркова. – Электрон. текстовые данные. – СПб.: Университет ИТМО, 2020. – 68 с. – 2227-8397. – Режим доступа: <http://www.iprbookshop.ru/65749.html>
2. Брик С. и др. Введение в JavaScript [Электронный ресурс] / Брик С. и др. – Электрон. текстовые данные. – НОУ «Интуит», URL: <http://www.intuit.ru/studies/courses/35/35/info>.

L.%.&" " " - " + * ' " " + # " ?(" " В

1. Система дистанционного обучения СмолГУ (moodle.smolgu.ru).
2. Интернет-Университет Информационных Технологий (www.intuit.ru).
3. Веб-программирование (codeacademy.ru).
4. Веб-программирование с азов (appleinsider.ru/developer/veb-programmirovanie-s-azov-vvedenie.html).
5. Лаборатория Web-программистов (dklab.ru).
6. Сайтостроениеотадоя (internet-technologies.ru/articles/category_36.html).

6.) " - "> " " " " " "

Для занятий необходимы:

1. проектор;
2. интерактивная доска;
3. персональные компьютеры.

Для самостоятельной работы подготовлены аудитории № 224, 226, 230, 234 с выходом в Интернет, оснащенные компьютерами IBMPCс процессорами IntelCore 7 и оперативной памятью не менее 16 Гб.

\$.& " " " " "

1. Node.js