

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский Нижегородский государственный университет
им. Н.И. Лобачевского»**

Институт информационных технологий, математики и механики

УТВЕРЖДЕНО

решением президиума Ученого совета ННГУ

протокол № 1 от 16.01.2024 г.

Рабочая программа дисциплины

Технологии сети Интернет

Уровень высшего образования

Бакалавриат

Направление подготовки / специальность

02.03.02 - Фундаментальная информатика и информационные технологии

Направленность образовательной программы

Инженерия программного обеспечения

Форма обучения

очная

г. Нижний Новгород

2024 год начала подготовки

1. Место дисциплины в структуре ОПОП

Дисциплина Б1.В.06 Технологии сети Интернет относится к части, формируемой участниками образовательных отношений образовательной программы.

2. Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения образовательной программы (компетенциями и индикаторами достижения компетенций)

| Формируемые компетенции (код, содержание компетенции) | Планируемые результаты обучения по дисциплине (модулю), в соответствии с индикатором достижения компетенции | | Наименование оценочного средства | |
|---|--|---|------------------------------------|-------------------------------|
| | Индикатор достижения компетенции (код, содержание индикатора) | Результаты обучения по дисциплине | Для текущего контроля успеваемости | Для промежуточной аттестации |
| ПК-3: Способен создавать и исследовать новые математические модели в естественных науках, промышленности и бизнесе, с учетом возможностей современных информационных технологий и программирования и компьютерной техники | ПК-3.1: Знает методы анализа и исследования математических моделей в области фундаментальной информатики и информационных технологий ПК-3.2: Умеет определять ключевые свойства и ограничения системы | ПК-3.1: ЗНАТЬ основы проектирования сайтов и технологии проектирования; основы разработки web-сайтов и Internet программирования; основы программирования сайтов различными программными средствами. ПК-3.2: УМЕТЬ разрабатывать Web-сайты, используя технологии проектирования сайтов и Internet-программирования, и использовать их на практике; создавать информационные ресурсы глобальных сетей. ВЛАДЕТЬ навыками Internet программирования при разработке Web-сайтов. | Проект Тест | Зачёт: Контрольные вопросы |

3. Структура и содержание дисциплины

3.1 Трудоемкость дисциплины

| | |
|--------------------------|--------------|
| | очная |
| Общая трудоемкость, з.е. | 2 |
| Часов по учебному плану | 72 |

| | |
|--|--------------------------|
| в том числе | |
| аудиторные занятия (контактная работа): | |
| - занятия лекционного типа | 32 |
| - занятия семинарского типа (практические занятия / лабораторные работы) | 0 |
| - КСР | 1 |
| самостоятельная работа | 39 |
| Промежуточная аттестация | 0 Зачёт |

3.2. Содержание дисциплины

(структурированное по темам (разделам) с указанием отведенного на них количества академических часов и виды учебных занятий)

| Наименование разделов и тем дисциплины | Всего (часы) | в том числе | | | |
|--|-----------------|--|--|-------------|---|
| | | Контактная работа (работа во взаимодействии с преподавателем), часы из них | | | Самостоятельная работа обучающегося, часы |
| | | Занятия лекционного типа | Занятия семинарского типа (практические занятия/лабораторные работы), часы | Всего | |
| | 0 Ф 0 | 0 Ф 0 | 0 Ф 0 | 0 Ф 0 | 0 Ф 0 |
| Межсетевое взаимодействие в TCP/IP сетях | 9 | 5 | 0 | 5 | 4 |
| Протоколы прикладного уровня | 10 | 5 | 0 | 5 | 5 |
| Принципы работы web-сервера | 11 | 5 | 0 | 5 | 6 |
| Расширение функциональности Web-сервера | 11 | 5 | 0 | 5 | 6 |
| Client-side технологии как часть контента, интерпретируемая клиентским процессом | 10 | 4 | 0 | 4 | 6 |
| Server-side технологии | 10 | 4 | 0 | 4 | 6 |
| СУБД как составная часть Web приложения | 10 | 4 | 0 | 4 | 6 |
| Аттестация | 0 | | | | |
| КСР | 1 | | | 1 | |
| Итого | 72 | 32 | 0 | 33 | 39 |

Содержание разделов и тем дисциплины

1. Межсетевое взаимодействие в TCP/IP сетях
2. Протоколы прикладного уровня
3. Принципы работы web-сервера
4. Расширение функциональности Web-сервера
5. Client-side технологии как часть контента, интерпретируемая клиентским процессом
6. Server-side технологии
7. СУБД как составная часть Web приложения

4. Учебно-методическое обеспечение самостоятельной работы обучающихся

Самостоятельная работа обучающихся включает в себя подготовку к контрольным вопросам и заданиям для текущего контроля и промежуточной аттестации по итогам освоения дисциплины приведенным в п. 5.

Для обеспечения самостоятельной работы обучающихся используются:

- электронный курс "Технологии сети Интернет" (<https://e-learning.unn.ru/enrol/index.php?id=6050>).

Иные учебно-методические материалы: 1. Кузнецова Л. Лекции по современным веб-технологиям. – ИНТУИТ. <http://www.intuit.ru/studies/courses/610/466/info>.

2. Брик С., Русак А., Сурин А. Храмцов П. Введение в HTML. – ИНТУИТ. <http://www.intuit.ru/studies/courses/33/33/info>.

3. Гейн А. Web-программирование на PHP 5.2. – ИНТУИТ. <http://www.intuit.ru/studies/courses/985/308/info>.

4. Кан М. Основы программирования на JavaScript. – ИНТУИТ. <http://www.intuit.ru/studies/courses/1093/132/info>.

5. Фонд оценочных средств для текущего контроля успеваемости и промежуточной аттестации по дисциплине (модулю)

5.1 Типовые задания, необходимые для оценки результатов обучения при проведении текущего контроля успеваемости с указанием критериев их оценивания:

5.1.1 Типовые задания (оценочное средство - Проект) для оценки сформированности компетенции ПК-3:

1. Создайте страницу html, содержащую функцию buildTable, которая, принимая двумерный массив элементов добавляет на страницу представление таблицы с двумя столбцами на html.

Таблица должна содержать заголовок, в качестве которого используется первая строка массива, количество строк таблицы должно равняться количеству строк массива - 1. Задание должно быть выполнено с использованием работы с DOM текущей страницы.

1. Создайте страницу html, содержащую функцию buildTable, которая принимает массив объектов с одинаковыми свойствами и строит структуру DOM, представляющую таблицу. У таблицы должна быть строка с заголовками, где имена свойств обернуты в элементы <th>. Каждая строка таблицы формируется из соответствующего объекта, свойства которого обернуты в элементы <td>. Здесь пригодится функция Object.keys, возвращающая массив, содержащий имена свойств объекта.
2. В промежутке с 1928 по 2013 год турецкие законы запрещали использование букв Q, W и X в официальных документах. Это являлось частью общей инициативы подавления курдской культуры – эти буквы используются в языке курдов, но не у турков. Запрограммируйте поле для ввода текста так, чтобы эти буквы нельзя было туда вписать.
3. Создайте страницу HTML, на которой при движении по ней мышью, следом за мышью движется произвольное изображение.
4. Создайте страницу HTML, содержащую четыре кнопки «создать таблицу», «удалить таблицу», «добавить строку», «добавить столбец». При нажатии на кнопку «создать таблицу» на странице отображается таблица с одной строкой и одним столбцом. При нажатии кнопок добавить строку или добавить столбец в таблицу добавляется строка или столбец соответственно. При нажатии кнопки «удалить таблицу» вся таблица удаляется.

5. Создайте страницу, содержащую тестовое поле с возможностью ввести несколько строк текста и кнопку. В текстовое поле можно ввести код на языке JavaScript, а при нажатии на кнопку этот код должен выполняться.
6. Найдите в интернете изображения с игральными картами. Выберите одну любую масть и создайте страницу HTML со следующим функционалом. При старте страницы отображается колода карт, лежащих друг на друге рубашками вверх (все карты одной масти). При нажатии мышью на рубашку верхней карты карта «открывается» и откладывается вправо, уже рубашкой вниз. Нажав мышью 9 раз, мы откроем всю колоду. Начальный порядок карт – случайный.
7. Создайте страницу, содержащую текстовое поле. Создайте скрипт, который при вводе текста в текстовое поле *под текстовым полем* отображает список вариантов автозавершения. У вас есть заданный в тексте программы массив возможных вариантов, и показывать нужно те из них, которые начинаются с вводимого текста. Когда пользователь щёлкает по предложенному варианту, он меняет содержимое поля на него.
8. Создайте страницу, содержащую элемент canvas. С этим элементом должны быть связаны функция «рисования», работающая следующим образом. Если нажать клавишу мыши на элементе, затем продвинуть мышь в сторону и отпустить, должна быть нарисована линия с началом в первой точке (точке нажатия клавиши мыши) и окончанием во второй точке (точке, где клавиша мыши была отпущена).
9. Создайте страницу, содержащую 4 поля редактирования: одно поле типа текст, одно поле для ввода чисел, одно поле для выбора даты и одно поле для выбора цвета. При нажатии кнопки «добавить», значения, введенные пользователем в эти поля, должны быть добавлены в *первую* строку таблицы, расположенной на этой же странице. Таким образом, каждый раз добавляя записи в таблицу, мы как бы «сдвигаем» строки таблицы вниз раз за разом.
10. Создайте страницу следующей структуры: на ней должны быть три «чекбокса» помеченные метками «фрукты», «овощи», «ягоды». В программе должны быть определены три массива, содержащие по меньшей мере по пять элементов для названий фруктов, овощей и ягод. Также на странице должен быть выпадающий список, содержимое которого должно зависеть от выбранных элементов. При выборе фруктов в нем должны быть названия фруктов, при выборе овощей – названия овощей – названия и фруктов и овощей и т.д.
11. Создайте страницу, содержащую функцию, реализованную на JavaScript со следующим функционалом. Функция принимает массив строк в качестве аргумента, выделяет на странице прямоугольные области одинакового размера (размер областей одинаков и задается как константа) и размещает элементы массива в этих областях. Т.е. 0-й элемент массива в первой области, 1-й во второй и т.д. Указанные элементы должны полностью по ширине укладываться в ширину окна браузера, не приводя к появлению горизонтального скроллера.
12. Создайте страницу, содержащую форму, включающую 4 текстовых поля. Под формой расположите два переключателя с подписями «горизонтально» и «вертикально». При выборе переключателя «горизонтально» все элементы формы должны выстроиться в одну линию, горизонтально. При этом возможно появление горизонтального скроллера в браузере. При выборе переключателя «вертикально», все элементы формы должны выстроиться друг под другом, с выравниванием по левому краю относительно формы.
13. Создайте страницу, которая при загрузке вызывает функцию, написанную на JavaScript. Функция принимает массив строк, каждая строка массива может содержать значения “text”, “number”, “date” или “color”. Функция должна сформировать на странице форму, включающую элементы управления, заданные элементами массива. Т.е. для элемента text в форму должно быть включено текстовое поле, для элемента number – поле для ввода чисел и т.д.
14. Создайте страницу, содержащую два поля ввода и две кнопки. Первое поле ввода служит для ввода номера элемента списка, *после* которого нужно осуществлять вставку. Второе поле ввода служит для ввода текста, который нужно вставлять. С помощью первой кнопки текст можно вставить на этом же уровне, что и элемент, после которого мы его вставляем. С помощью второй

кнопки можно вставить текст как вложенный список (его уровень будет на один уровень ниже). Таким образом, можно выбрать элемент списка, ввести текст, который будет вставлен после этого элемента и выбрать, будет ли вставленный текст вставлен на этом же уровне, или же уровнем ниже. Использовать нумерованный список (элемент ``).

15. Создайте страницу, содержащую пустую форму (не содержащую элементов), поле редактирования (для задания значения метки) и четыре кнопки. Каждая кнопка добавляет в форму один элемент. Первая кнопка – тестовое поле, вторая кнопка – кнопку, третья – метку (label), причем метка должна быть привязана к последнему добавленному до нее элементу управления, четвертая – поле для ввода даты. В случае добавления метки, текстовое значение для нее берется из поля редактирования.
16. Создайте страницу, содержащую таблицу (элемент `<table>`). При нажатии на значение заголовка столбца таблицы, строки таблицы должны сортироваться по этому столбцу. При нажатии на любую строку таблицы (кроме строки заголовка) должен выводиться вопрос «удалить строку таблицы?» и при нажатии кнопки Ok, строка таблицы – удаляться.
17. Создайте страницу, содержащую элемент canvas. С этим элементом должна быть связана функция «рисования» квадратов, работающая следующим образом. Если нажать клавишу мыши на элементе, затем продвинуть мышь в сторону и отпустить, должен быть нарисован квадрат с центром в первой точке (точке нажатия клавиши мыши) и углом во второй точке (точке, где клавиша мыши была отпущена). Предполагается, что первая и вторая точки не находятся на одной горизонтали или вертикали.
18. Создайте страницу, содержащую произвольный ненумерованный список (``). Список может включать в себя вложенные списки. Уровень вложенности и количество элементов в списке – произвольные. Реализуйте функцию на JavaScript, которая принимая в качестве аргумента корневой элемент списка (``) сортирует в лексикографическом порядке все элементы списка, включая и вложенные списки. Результатом работы функции должно быть изменение первоначального списка. Функция должна запускаться при щелчке мышью на любом элементе списка.
19. Напишите функцию на JavaScript, работающую следующим образом. Она принимает в качестве аргумента элемент, являющийся `<div>`, этот элемент определен на текущей странице и содержит произвольное изображение. При приближении курсора мыши к этому элементу, элемент пытается «убежать» от мыши, перескакивая в произвольном направлении.
20. Напишите функцию на JavaScript, принимающую элемент, являющийся таблицей (`<table>`) и массив, содержащий номера столбцов таблицы. Для каждого из столбцов, содержащихся в массиве:
 - a. Если столбец таблицы с этим номером содержит числовые данные – подсчитать сумму всех элементов этого столбца
 - b. Если столбец таблицы содержит хотя бы одно значение, не являющееся числом – подсчитать количество НЕПУСТЫХ значений в этом столбце

В конец таблицы добавить строку, в которой для каждого из столбцов, переданных в функцию вывести либо сумму по столбцу (для числовых данных), либо количество не пустых элементов.

21. Напишите функцию на JavaScript, принимающую в качестве аргумента элемент html и выводящую на страницу «дерево» DOM, для которого переданный в качестве аргумента элемент является корневым. Дерево должно выводиться в виде ненумерованного списка (``) с вложениями. В качестве текста, выводимого в элементе списка использовать имя элемента (т.е. head, body, p, div и др) и id элемента в том случае, если он указан. Вызов функции на странице осуществлять при нажатии кнопки, элемент, начиная с которого вывести дерево, указывать через его id, которое вводить в текстовом поле.

Критерии оценивания (оценочное средство - Проект)

| Оценка | Критерии оценивания |
|------------|---|
| зачтено | Выполнены все или большая часть этапов решения задачи или задача решена с незначительными недочетами. Программа и результаты работы представлены преподавателю в срок. |
| не зачтено | Выполнены не все лабораторные работы или выполнены не в полном объеме (представлено не полное описание этапов выполнения заданий, программа работает некорректно, результаты работы не представлены преподавателю). |

5.1.2 Типовые задания (оценочное средство - Тест) для оценки сформированности компетенции ПК-3:

1. Каким тегом объявляется web-страница?

- 1) (+)<html></html>
- 2) <head></head>
- 3) <title></title>
- 4) <body></body>

2. Каким тегом объявляется заголовок web-страницы?

- 1) <html></html>
- 2) (+)<head></head>
- 3) <title></title>
- 4) <body></body>

3. Вкакой тег заключается основное содержание web-страницы?

- 1) <html></html>
- 2) <head></head>
- 3) <title></title>
- 4) (+)<body></body>

4. Вкакой тег заключается название web-страницы?

- 1) <html></html>
- 2) <head></head>
- 3) (+)<title></title>
- 4) <body></body>

5. Какой код для пустой web-страницы правильный?

- 1) <html><head><title></head><body></body></html>
- 2)(+)<html><head><title></title></head><body></body></html>
- 3) <html><head><title><body></body></html>
- 4) <html><head><title></title></head><body></body>

Критерии оценивания (оценочное средство - Тест)

| Оценка | Критерии оценивания |
|------------|--|
| зачтено | как минимум 80% правильных ответов в тесте |
| не зачтено | менее 80% правильных ответов в тесте |

5.2. Описание шкал оценивания результатов обучения по дисциплине при промежуточной аттестации

Шкала оценивания сформированности компетенций

| Уровень сформированности компетенций (индикатор достижения компетенций) | плохо | неудовлетворительно | удовлетворительно | хорошо | очень хорошо | отлично | превосходно |
|---|---|--|--|---|---|--|--|
| | не зачтено | | | зачтено | | | |
| <u>Знания</u> | Отсутствие знаний теоретического материала. Невозможность оценить полноту знаний вследствие отказа обучающегося от ответа | Уровень знаний ниже минимальных требований. Имели место грубые ошибки | Минимально допустимый уровень знаний. Допущено много негрубых ошибок | Уровень знаний в объеме, соответствующем программе подготовки. Допущено несколько негрубых ошибок | Уровень знаний в объеме, соответствующем программе подготовки. Допущено несколько несущественных ошибок | Уровень знаний в объеме, соответствующем программе подготовки. Ошибок нет. | Уровень знаний в объеме, превышающем программу подготовки. |
| <u>Умения</u> | Отсутствие минимальных умений. Невозможность оценить наличие умений вследствие отказа обучающегося от ответа | При решении стандартных задач не продемонстрированы основные умения. Имели место грубые ошибки | Продemonстрированы основные умения. Решены типовые задачи с негрубыми ошибками. Выполнены все задания, но не в полном объеме | Продemonстрированы все основные умения. Решены все основные задачи с негрубыми ошибками. Выполнены все задания в полном объеме, но некоторые с недочетами | Продemonстрированы все основные умения. Решены все основные задачи. Выполнены все задания в полном объеме, но некоторые с недочетами. | Продemonстрированы все основные умения. Решены все основные задачи с отдельным и несущественными недочетами, выполнены все задания в полном объеме | Продemonстрированы все основные умения. Решены все основные задачи. Выполнены все задания, в полном объеме без недочетов |
| <u>Навыки</u> | Отсутствие базовых навыков. Невозможность оценить наличие навыков вследствие отказа обучающегося от | При решении стандартных задач не продемонстрированы базовые навыки. Имели место грубые ошибки | Имеется минимальный набор навыков для решения стандартных задач с некоторым | Продemonстрированы базовые навыки при решении стандартных задач с некоторым и | Продemonстрированы базовые навыки при решении стандартных задач без ошибок и недочетов | Продemonстрированы навыки при решении нестандартных задач без ошибок и | Продemonстрирован творческий подход к решению нестандартных задач |

| | | | | | | | |
|--|--------|--|-----------------|------------|--|-----------|--|
| | ответа | | и недочетами | недочетами | | недочетов | |
|--|--------|--|-----------------|------------|--|-----------|--|

Шкала оценивания при промежуточной аттестации

| Оценка | | Уровень подготовки |
|------------|---------------------|--|
| зачтено | превосходно | Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «превосходно», продемонстрированы знания, умения, владения по соответствующим компетенциям на уровне выше предусмотренного программой |
| | отлично | Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «отлично». |
| | очень хорошо | Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «очень хорошо» |
| | хорошо | Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «хорошо». |
| | удовлетворительно | Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «удовлетворительно», при этом хотя бы одна компетенция сформирована на уровне «удовлетворительно» |
| не зачтено | неудовлетворительно | Хотя бы одна компетенция сформирована на уровне «неудовлетворительно». |
| | плохо | Хотя бы одна компетенция сформирована на уровне «плохо» |

5.3 Типовые контрольные задания или иные материалы, необходимые для оценки результатов обучения на промежуточной аттестации с указанием критериев их оценивания:

5.3.1 Типовые задания (оценочное средство - Контрольные вопросы) для оценки сформированности компетенции ПК-3

1. Глобальные компьютерные сети: основные понятия, принципы функционирования. Протокол HTTP
2. Язык гипертекстовой разметки страниц HTML: общая структура документа, абзацы, цвета, ссылки.
3. Язык гипертекстовой разметки страниц HTML: списки, графика (графические форматы, графический объект как ссылка), таблицы, фреймы, формы.
4. Общие подходы к дизайну сайта. Разработка макета страницы.
5. Язык гипертекстовой разметки страниц HTML: формы.
6. Преимущества и ограничения программ, работающих на стороне клиента. Язык JavaScript: основы синтаксиса.
7. Исполнение кода на стороне сервера. исторический подходы (CGI, JSP)

8. Исполнение кода JavaScript на стороне сервера. node.js: инфраструктура.
9. Обзор некоторых пакетов node.js
10. Использование фреймворка express
11. Архитектурный шаблон REST. SPA-приложения. AJAX.
12. Работа с базами данных. Принципы доступа к данным из приложения.
13. Сессии и их поддержка (самостоятельная и с помощью готовых библиотек)
14. Взаимодействие со стороны сервера: Longpooling, SSE, Websocket`s

Критерии оценивания (оценочное средство - Контрольные вопросы)

| Оценка | Критерии оценивания |
|------------|--|
| зачтено | Студент ответил на большую часть вопросов возможно с незначительными недочетами. |
| не зачтено | При ответе студент допускает грубые ошибки в основном материале и решении стандартных задач. |

6. Учебно-методическое и информационное обеспечение дисциплины (модуля)

Основная литература:

1. Лекции по современным веб-технологиям / Кузнецова Л.В. - Москва : ИНТУИТ, 2016., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=662939&idb=0>.
2. Введение в HTML5. - Москва : ИНТУИТ, 2016., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=662805&idb=0>.

Дополнительная литература:

1. Основы программирования на JavaScript / Кан М. - Москва : ИНТУИТ, 2016., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=663437&idb=0>.

Программное обеспечение и Интернет-ресурсы (в соответствии с содержанием дисциплины):

- Операционные системы семейства Microsoft Windows, лицензия по подписке Microsoft Imagine.
- Среда разработки семейства Microsoft Visual Studio, лицензия по подписке Microsoft Imagine.
- Компилятор семейства Intel C++ Compiler из пакета Intel Parallel Studio по бесплатной лицензии для преподавателей вузов.

7. Материально-техническое обеспечение дисциплины (модуля)

Учебные аудитории для проведения учебных занятий, предусмотренных образовательной программой, оснащены мультимедийным оборудованием (проектор, экран), техническими

средствами обучения, компьютерами.

Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к сети "Интернет" и обеспечены доступом в электронную информационно-образовательную среду.

Программа составлена в соответствии с требованиями ОС ННГУ по направлению подготовки/специальности 02.03.02 - Фундаментальная информатика и информационные технологии.

Автор(ы): Свистунов Алексей Николаевич.

Заведующий кафедрой: Мееров Иосиф Борисович, кандидат технических наук.

Программа одобрена на заседании методической комиссии от 13.12.2023, протокол № 3.