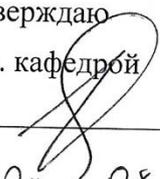


Государственное образовательное учреждение
«Приднестровский государственный университет им. Т.Г. Шевченко»

Физико-технический институт
Физико-математический факультет

Кафедра Высшей и прикладной математики и информатики

Утверждаю
Зав. кафедрой


_____ А.В. Коровай

« 30 » 08 2024г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
по дисциплине

«Web-программирование»

Направление подготовки:
01.04.02 «Прикладная математика и информатика»

Профиль подготовки
«Математические и информационные технологии»

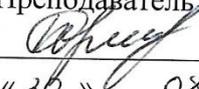
Квалификация
Магистр

Форма обучения
Очная

ГОД НАБОРА 2024

Разработал:

Преподаватель


_____ Юрковская Л.А.

« 30 » 08 2024г.

Тирасполь 2024 г.

**Паспорт фонда оценочных средств по учебной дисциплине
«Технология Web-программирования»**

1. В результате изучения дисциплины «Технология Web-программирования» у обучающихся должны быть сформированы следующие компетенции:

| Категория (группа) компетенций | Код и наименование | Код и наименование индикатора достижения универсальной компетенции |
|---|---|---|
| Общепрофессиональные компетенции выпускников и индикаторы их достижения | | |
| Информационно-коммуникационные технологии для профессиональной деятельности | ОПК-4. Способен комбинировать и адаптировать существующие информационно-коммуникационные технологии для решения задач в области профессиональной деятельности с учетом требований информационной безопасности | ИД ОПК-4.1 Анализирует существующие информационно-коммуникационные технологии для решения задач профессиональной деятельности |
| | | ИД ОПК-4.2 Оценивает существующие информационно-коммуникационные технологии на соответствие основным требованиям информационной безопасности |
| | | ИД ОПК-4.3 Имеет практический опыт применения информационно-коммуникационных технологий, используемых в профессиональной деятельности |
| Обязательные профессиональные компетенции выпускников и индикаторы их достижения | | |
| | ПК-1. Способен проводить научные исследования и получать новые научные и прикладные результаты самостоятельно и в составе научного коллектива | ИД ПК-1.1 Знает классические методы, применяемые в прикладной математике и информатике; необходимые и достаточные условия их реализации |
| | | ИД ПК-1.2 Умеет самостоятельно выбирать эффективные методы решения поставленных задач и разрабатывать новые методы для получения новых научных и прикладных результатов |
| | | ИД ПК-1.3 Владеет наукоемкими технологиями и пакетами прикладных программ для решения прикладных задач |
| | ПК-2. Способен разрабатывать и применять математические методы, системное и прикладное программное обеспечение | ИД ПК-2.1 Знает языки программирования, библиотеки и пакеты программ |

| | | |
|--|--|--|
| | для решения задач научной и технологической деятельности | ИД ПК-2.2 Умеет выбирать оптимальные системы программирования, наиболее подходящие для решения поставленной задачи |
| | | ИД ПК-2.3 Владеет методами моделирования информационных процессов |

2. Программа оценивания контролируемой компетенции

| Текущая аттестация | Контролируемые модули, разделы (темы) дисциплины и их наименование | Код контролируемой компетенции (или ее части) | Наименование оценочного средства |
|---------------------------------|--|---|---|
| 1 | Раздел 1. Библиотеки JS | ОПК-4, ПК-1, ПК-2 | Тест № 1 Тест № 2 Выполнение лабораторных работ |
| Промежуточная аттестация | | Код контролируемой компетенции (или ее части) | Наименование оценочного средства |
| Экзамен | | ОПК-4, ПК-1, ПК-2 | Вопросы к экзамену, комплект задач к экзамену |

ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
«Приднестровский государственный университет им. Т.Г. Шевченко»
Физико-технический институт
Физико-математический факультет
Кафедра Прикладной математики и информатики

Тест №1

Раздел 1. Библиотеки JS

1. **Что такое Axios и для чего он используется в React?**
 - a) Библиотека для управления состоянием
 - b) Библиотека для работы с HTTP-запросами
 - c) Библиотека для анимаций
 - d) Библиотека для работы с формами
2. **Как установить Axios в проекте React?**
 - a) `npm install axios`
 - b) `npm install react-axios`
 - c) `npm install axios-react`
 - d) `npm install http-axios`
3. **Напишите пример отправки GET-запроса с использованием Axios.**
 - a) `axios.get('/api/data')`
 - b) `axios.fetch('/api/data')`
 - c) `axios.request('/api/data', method='GET')`
 - d) `axios.retrieve('/api/data')`
4. **Как обрабатывать ошибки при использовании Axios?**
 - a) Используя блок `try-catch`
 - b) Используя метод `.catch()`
 - c) Используя метод `.error()`
 - d) Используя метод `.then()`
5. **Что такое интерцептор в Axios?**
 - a) Функция для обработки ошибок
 - b) Функция для обработки запросов и ответов
 - c) Функция для создания запросов
 - d) Функция для отмены запросов
6. **Как создать интерцептор для запросов в Axios?**
 - a) `axios.interceptors.request.use(config => { /* ... / }, error => { / ... */ })`
 - b) `axios.requestInterceptor(config => { /* ... / }, error => { / ... */ })`
 - c) `axios.createInterceptor(config => { /* ... / }, error => { / ... */ })`
 - d) `axios.request.use(config => { /* ... / }, error => { / ... */ })`
7. **Как создать интерцептор для ответов в Axios?**
 - a) `axios.interceptors.response.use(response => { /* ... / }, error => { / ... */ })`
 - b) `axios.responseInterceptor(response => { /* ... / }, error => { / ... */ })`
 - c) `axios.createInterceptor(response => { /* ... / }, error => { / ... */ })`
 - d) `axios.response.use(response => { /* ... / }, error => { / ... */ })`
8. **Напишите пример использования интерцептора в Axios.**

- a) `axios.interceptors.request.use(config => { console.log('Request:', config); return config; })`
 - b) `axios.interceptors.config.use(config => { console.log('Config:', config); return config; })`
 - c) `axios.interceptors.data.use(data => { console.log('Data:', data); return data; })`
 - d) `axios.interceptors.url.use(url => { console.log('URL:', url); return url; })`
9. **Какие преимущества использования интерцепторов в Axios?**
- a) Позволяют легко создавать запросы
 - b) Позволяют автоматически обрабатывать запросы и ответы
 - c) Улучшают производительность
 - d) Упрощают интеграцию с React
10. **Что такое мемоизация и как она помогает оптимизировать производительность кода?**
- a) Кэширование результатов вычислений
 - b) Уменьшение размера кода
 - c) Улучшение интерфейса пользователя
 - d) Увеличение времени загрузки
11. **Как использовать React.memo для оптимизации компонентов?**
- a) Оборачивать компоненты в `React.memo()`
 - b) Использовать `React.memo()` для запроса данных
 - c) Использовать `React.memo()` для рендеринга компонентов
 - d) Использовать `React.memo()` для изменения состояния
12. **Что такое useMemo и когда его следует использовать?**
- a) Хук для мемоизации значений
 - b) Хук для управления состоянием
 - c) Хук для выполнения побочных эффектов
 - d) Хук для работы с формами
13. **Как библиотека Lodash может помочь в оптимизации кода?**
- a) Предоставляет функции для работы с массивами и объектами
 - b) Предоставляет функции для работы с HTTP-запросами
 - c) Предоставляет функции для создания анимаций
 - d) Предоставляет функции для работы с формами
14. **Напишите пример использования функции debounce из библиотеки Lodash.**
- a) `_.debounce(func, wait)`
 - b) `_.throttle(func, wait)`
 - c) `_.delay(func, wait)`
 - d) `_.defer(func, wait)`
15. **Какие преимущества использования ленивой загрузки ресурсов?**
- a) Улучшение производительности страницы
 - b) Уменьшение размера кода
 - c) Уменьшение числа HTTP-запросов
 - d) Увеличение времени загрузки
16. **Какой инструмент можно использовать для минификации файлов JavaScript?**
- a) UglifyJS
 - b) Babel

- c) Webpack
 - d) ESLint
- 17. Какие преимущества использования CDN для статических ресурсов?**
- a) Снижение задержек и улучшение производительности загрузки
 - b) Уменьшение размера кода
 - c) Уменьшение числа HTTP-запросов
 - d) Увеличение времени загрузки
- 18. Как настроить кэширование на стороне клиента?**
- a) Настройка заголовков кэширования в HTTP-ответах
 - b) Использование локального хранилища
 - c) Использование сессий
 - d) Использование файлов cookies
- 19. Что такое D3.js и для чего он используется?**
- a) Библиотека для создания графиков и визуализаций данных
 - b) Библиотека для управления состоянием
 - c) Библиотека для работы с HTTP-запросами
 - d) Библиотека для работы с формами
- 20. Как интегрировать D3.js в компоненты React?**
- a) Использовать D3.js внутри хуков useEffect и useRef
 - b) Использовать D3.js внутри хуков useState и useEffect
 - c) Использовать D3.js внутри хуков useMemo и useCallback
 - d) Использовать D3.js внутри хуков useReducer и useContext

ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
«Приднестровский государственный университет им. Т.Г. Шевченко»
Физико-технический институт
Физико-математический факультет
Кафедра Прикладной математики и информатики

Тест №2

Раздел 1. Библиотеки JS

1. **Как получить данные с сервера с помощью Axios?**
 - a) `axios.get('/api/data')`
 - b) `axios.post('/api/data')`
 - c) `axios.fetch('/api/data')`
 - d) `axios.retrieve('/api/data')`
2. **Напишите пример использования Axios для отправки POST-запроса.**
 - a) `axios.post('/api/data', { data: 'value' })`
 - b) `axios.get('/api/data', { data: 'value' })`
 - c) `axios.put('/api/data', { data: 'value' })`
 - d) `axios.send('/api/data', { data: 'value' })`
3. **Как обновить состояние компонента на основе данных, полученных с сервера?**
 - a) `setState(response.data)`
 - b) `useState(response.data)`
 - c) `updateState(response.data)`
 - d) `setData(response.data)`
4. **Как визуализировать данные с использованием D3.js?**
 - a) `d3.select().append()`
 - b) `d3.visualize().render()`
 - c) `d3.create().draw()`
 - d) `d3.plot().display()`
5. **Напишите пример создания простого столбчатого графика с использованием D3.js.**
 - a) `svg.selectAll('rect').data(data).enter().append('rect')`
 - b) `svg.selectAll('circle').data(data).enter().append('circle')`
 - c) `svg.selectAll('line').data(data).enter().append('line')`
 - d) `svg.selectAll('path').data(data).enter().append('path')`
6. **Как создать линейный график с использованием D3.js?**
 - a) `svg.selectAll('path').data(data).enter().append('path')`
 - b) `svg.selectAll('rect').data(data).enter().append('rect')`
 - c) `svg.selectAll('line').data(data).enter().append('line')`
 - d) `svg.selectAll('circle').data(data).enter().append('circle')`
7. **Какие преимущества использования библиотек для визуализации данных?**
 - a) Уменьшение объема данных
 - b) Обеспечение наглядности данных
 - c) Уменьшение количества запросов
 - d) Увеличение времени загрузки
8. **Как создать дашборд с использованием React и D3.js?**

- a) Использовать компоненты React для построения интерфейса и D3.js для визуализации данных
 - b) Использовать только компоненты React для визуализации данных
 - c) Использовать только D3.js для построения интерфейса
 - d) Использовать jQuery и React для визуализации данных
9. **Как оптимизировать производительность приложения с помощью мемоизации?**
- a) Кэшировать результаты вычислений, чтобы избежать повторных вычислений
 - b) Увеличить количество HTTP-запросов
 - c) Уменьшить размер изображений
 - d) Использовать более сложные алгоритмы
10. **Как использовать useCallback для оптимизации функций обратного вызова?**
- a) Оборачивать функции в useCallback()
 - b) Оборачивать функции в useMemo()
 - c) Оборачивать функции в React.memo()
 - d) Оборачивать функции в useEffect()
11. **Напишите пример использования функции throttle из библиотеки Lodash.**
- a) _.throttle(func, wait)
 - b) _.debounce(func, wait)
 - c) _.delay(func, wait)
 - d) _.defer(func, wait)
12. **Какие преимущества кэширования данных в приложениях?**
- a) Улучшение производительности за счет уменьшения количества запросов
 - b) Уменьшение объема данных
 - c) Уменьшение размера кода
 - d) Увеличение времени загрузки
13. **Как использовать Chrome DevTools для анализа производительности?**
- a) Открыть вкладку "Performance" и записать производительность
 - b) Открыть вкладку "Console" и записать производительность
 - c) Открыть вкладку "Network" и записать производительность
 - d) Открыть вкладку "Elements" и записать производительность
14. **Какие методы кэширования можно использовать в веб-приложениях?**
- a) HTTP-кэширование, кэширование в локальном хранилище, кэширование в сервис-воркерах
 - b) Кэширование в памяти, кэширование на сервере, кэширование в кэше браузера
 - c) Кэширование в базах данных, кэширование в сессиях, кэширование в файлах cookies
 - d) Кэширование в облаке, кэширование на диске, кэширование в процессах
15. **Какие инструменты можно использовать для анализа и улучшения производительности веб-приложений?**
- a) Chrome DevTools, Lighthouse, WebPageTest
 - b) Photoshop, Illustrator, After Effects
 - c) MySQL, PostgreSQL, MongoDB
 - d) Git, GitHub, Bitbucket

ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
«Приднестровский государственный университет им. Т.Г. Шевченко»
Физико-технический институт
Физико-математический факультет
Кафедра Прикладной математики и информатики

Вопросы к экзамену
по дисциплине «Web-программирование»

1. Что такое библиотека JavaScript?
2. Какие преимущества использования библиотек JavaScript?
3. Чем отличается библиотека от фреймворка?
4. Назовите несколько популярных библиотек JavaScript.
5. Каковы основные задачи Lodash?
6. Что такое React?
7. Какие концепции лежат в основе React?
8. Почему React считается декларативным?
9. Что такое JSX?
10. Как в React реализуется однонаправленный поток данных?
11. Что такое состояние (state) в React?
12. Чем props отличаются от состояния?
13. Как обновить состояние в функциональном компоненте React?
14. Что такое Context API и для чего он используется?
15. Назовите методы управления глобальным состоянием в React.
16. Для чего используется jQuery?
17. Как выбрать элемент с определённым классом в jQuery?
18. Какие методы jQuery используются для добавления и удаления классов?
19. Как в jQuery выполнить анимацию элемента?
20. Какие методы доступны для обработки событий в jQuery?
21. Что такое AJAX?
22. Чем GET-запрос отличается от POST-запроса?
23. Как выполнить AJAX-запрос с использованием jQuery?
24. Какие параметры можно передать методу AJAX в jQuery?
25. Как обрабатывать ошибки AJAX-запросов в jQuery?
26. Как в Lodash клонировать объект?
27. Какие методы Lodash используются для работы с массивами?
28. Как с помощью Lodash исключить повторяющиеся элементы в массиве?
29. Что делает функция `_.merge`?
30. Для чего используется `_.debounce` в Lodash?
31. Что такое D3.js и для чего он используется?
32. Как с помощью D3 выбрать элемент SVG?
33. Какие основные концепции лежат в основе работы с D3?

34. Чем D3.js отличается от других библиотек визуализации?
35. Как подключить D3.js к проекту?
36. Как добавить ось к графику в D3?
37. Как настроить масштабирование в D3.js??
38. Какие методы используются для добавления переходов и анимации в D3.js??
39. Как управлять данными, отображаемыми на графике, с помощью D3.js??
40. Как настроить интерактивное поведение (например, всплывающие подсказки) в D3.js??

ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
«Приднестровский государственный университет им. Т.Г. Шевченко»
Физико-технический институт
Физико-математический факультет
Кафедра Прикладной математики и информатики

Комплект задач к экзамену
по дисциплине «Web-программирование»

1. Реализуйте функцию, которая с использованием Lodash создаёт глубокую копию объекта.
2. Напишите код, который использует jQuery для выбора всех элементов с классом `active` и добавляет к ним класс `highlighted`.
3. Создайте простой функциональный компонент React, который отображает кнопку и счётчик кликов.
4. Напишите компонент, который получает список элементов через props и рендерит их в виде списка ``.
5. Используя хук `useState`, создайте компонент React, который меняет текст по нажатию кнопки.
6. Реализуйте компонент React, который использует Context API для передачи данных между вложенными компонентами.
7. Создайте анимацию с помощью jQuery, которая увеличивает ширину элемента с идентификатором `box` до `200px` за 2 секунды.
8. Напишите скрипт jQuery, который обрабатывает событие нажатия кнопки и меняет текст параграфа с классом `info`.
9. Напишите код, который отправляет GET-запрос с использованием jQuery и выводит полученные данные в консоль.
10. Создайте скрипт, который отправляет AJAX-запрос для добавления нового элемента на сервер (POST-запрос) и выводит сообщение об успехе или ошибке.
11. Используя Lodash, напишите функцию для удаления всех повторяющихся значений из массива.
12. Создайте объект и с использованием Lodash напишите код для сортировки его свойств по значениям.
13. Напишите код, который добавляет SVG-прямоугольник на страницу с использованием D3.js..
14. С помощью D3.js создайте ось с метками от 0 до 100 с интервалом 10.
15. Реализуйте интерактивный график, где при наведении курсора на точку отображается всплывающая подсказка с её значением.
16. Используя D3.js, создайте анимацию, которая изменяет цвет столбцов диаграммы при клике на них.
17. Напишите пример компонента React, который визуализирует данные с помощью D3.js..
18. Создайте проект, который использует React и jQuery для рендеринга списка элементов и обработки событий.

19. Реализуйте проект, который получает данные с API с использованием AJAX и отображает их в интерактивной таблице с помощью React.
20. Создайте веб-приложение, которое показывает график с использованием D3.js и динамически обновляет его данные каждые 5 секунд.