

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ВЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Институт математики и информационных систем

Факультет автоматики и вычислительной техники

Кафедра систем автоматизации управления

Отчет по лабораторной работе № 1
по дисциплине
«Веб-программирование»

Выполнил студент гр. ИТб-2301-01-00

_____/Халявин А.К./
(Подпись)

Проверил ст. преподаватель

_____/Земцов М.А./
(Подпись)

Киров 2023

Цель работы: Организовать рабочее пространство для разработки веб-контента, исследовать функциональные возможности инструментов разработчика на стороне браузера и получить навыки веб-программирования и создания приложений JavaScript с помощью Node.js и TypeScript.

Ход работы

В ходе лабораторной работы были пройдены 3 интерактивных курса microsoft “Веб-разработка для начинающих” , “Создание приложений JavaScript с помощью Node.js” , “Создание приложений JavaScript с помощью TypeScript”. Прохождение модулей данных курсов представлены на рисунках 1-24. Результат прохождения тестов представлен на рисунке 25.

1. Программу можно создать, используя:

- ☒ Текстовый редактор
✓ Правильно, текстовый редактор — средство, которое разработчики наиболее часто используют для создания программ.
- ☐ Карандаш и бумага
- ☐ Печатающее устройство

2. Низкоуровневые языки являются популярным вариантом для:

- ☐ веб-сайты;
- ☒ Оборудование
✓ Правильно. Низкоуровневые языки чаще всего используются с аппаратными устройствами.
- ☐ Программное обеспечение для видеоигр

3. Какое из этих средств, скорее всего, будет в окружении веб-разработчика?

- ☐ Оборудование, такое как Raspberry Pi
- ☒ Инструменты разработки браузера
✓ Правильно. Средства разработчика имеют возможности, позволяющие разработчикам собирать и сканировать полезные сведения о приложении.
- ☐ Документация по операционной системе

Рисунок 1 – Прохождение 1 модуля курса “Веб-разработка для начинающих”

1. Какой язык вы, скорее всего, используете для создания веб-сайта?

☐ Машинный код

☒ JavaScript

✓ Правильно. JavaScript используется в основном для веб-приложений и рассматривается как высокоуровневый язык.

☐ Bash

2. Средства, используемые разработчиками для организации рабочего процесса, называются:

☒ Среда разработки

✓ Правильно. Набор средств разработчика формирует полную среду разработки для написания кода.

☐ Набор средств разработки

☐ Арсенал разработки

3. Что делают разработчики для исправления кода, который содержит ошибки?

☐ Применяют выделение синтаксиса

☒ Выполняют отладку кода

✓ Правильно. Лучший способ исправить код, который содержит ошибки, — это отладка!

☐ Применяют форматирование кода

Рисунок 2 – Прохождение 1 модуля курса “Веб-разработка для начинающих”

1. Какая из следующих ссылок является ссылкой, доступной для всех?

☐ `https://www.microsoft.com`

☒ `Microsoft`

✓ Предоставление нескольких слов, чтобы описать, на какой сайт ссылка перенаправит пользователя, является текстом ссылки, доступным для всех.

☐ `Click here`

2. Какое из следующих утверждений является подходящим применением атрибутов ARIA?

☐ Указание средству чтения с экрана, что элемент `div` фактически является элементом `button`.

☐ Предоставление другого текста, который средство чтения с экрана будет озвучивать, если текст ссылки читается как "щелкните здесь".

☒ Предоставление контекста средству чтения с экрана, если элемент управления не существует в семантическом HTML, например является деревом.

✓ Атрибуты ARIA могут предоставлять контекст для средства чтения с экрана при создании пользовательских элементов управления, которые изначально отсутствуют в HTML.

3. Какой элемент HTML следует использовать, чтобы добавить элемент управления, который пользователь выбирает для выполнения действия?

☐ `div`

☐ `span`

☐ `p`

☒ `button`

✓ Используйте `button`, чтобы убедиться, что страница доступна для всех.

Рисунок 3 – Прохождение 2 модуля курса “Веб-разработка для начинающих”

1. В JavaScript можно объявить значения с помощью ключевого слова _____ или _____

☒ `let` или `var`

✓ Правильно. Это ключевые слова, необходимые для создания и объявления переменных.

☐ `get` или `script`

☐ `char` или `log`

2. Константы, так же как и `let` и `var`, используются для объявления переменных, с тем отличием, что...

☐ Константы можно изменять.

☒ Константы необходимо инициализировать.

✓ Правильно. Константы необходимо инициализировать, иначе при запуске кода произойдет ошибка.

☐ Константы можно переназначать.

Рисунок 4 – Прохождение 3 модуля курса “Веб-разработка для начинающих”

1. Что из перечисленного ниже не является типом данных?

☐ Числа

☒ Файлы

✓ Правильно. Хотя вы можете работать с файлами в коде JavaScript, они не считаются типами данных.

☐ Boolean

2. Числа и _____ — это примитивы JavaScript, которые обрабатывают числовые данные.

☒ `bigint`

✓ Правильно. Значения для `bigint` могут использоваться для произвольно больших целых чисел.

☐ `boolean`

☐ `star`

3. Строки могут находиться между одинарными и двойными _____.

☐ скобками

☒ кавычками

✓ Правильно. Для создания строк можно использовать одинарные или двойные кавычки.

☐ обратными косыми чертами

Рисунок 5 – Прохождение 3 модуля курса “Веб-разработка для начинающих”

1. Что такое параметр?

- ☐ Значение, возвращаемое функцией
- ☐ Имя функции
- ☒ Значение, которое функция принимает от вызывающего объекта

✓ Верно! Параметр — это значение, которое функция принимает от вызывающего объекта.

2. Какое из следующих имен соответствует рекомендациям по именованию для функций JavaScript?

- ☐ helloworld
- ☒ helloWorld
- ☐ hello_world
- ☐ HelloWorld

✓ Правильно. Имя helloWorld является допустимым и соответствует рекомендациям, которые предлагают использовать "верблюжий" стиль.

3. Какая функция вызывает функцию по истечении указанного времени?

- ☒ setTimeout
- ☐ document.write
- ☐ console.log

✓ Верно! Функция setTimeout вызывает функцию по истечении указанного времени.

Рисунок 6 – Прохождение 4 модуля курса “Веб-разработка для начинающих”

1. Вызывается оператор ===.

- ☐ Равенство
- ☒ Строгое равенство
- ☐ Назначение

✓ Правильно. Этот оператор указывает на строгое равенство.

2. Какой тип результата возвращает сравнение в JavaScript?

- ☒ Логическое
- ☐ NULL
- ☐ Строка

✓ Правильно. Сравнение в JavaScript возвращает значения true и false.

3. ! в JavaScript означает следующее:

- ☒ Логическое НЕ
- ☐ Важно!
- ☐ Равно

✓ Правильно. Логическое НЕ возвращает противоположное значение логического выражения.

Рисунок 7 – Прохождение 5 модуля курса “Веб-разработка для начинающих”

1. Какой результат вернет следующий код: `'1' == 1`?

☒ Да

✓ Правильно. При использовании оператора `==` код вернет `true`, так как выполняется неявная проверка.

☐ false

☐ null

2. Какой результат вернет следующий код: `'1' === 1`?

☐ Да

☒ false

✓ Правильно. Мы сравниваем значения на строгое равенство.

☐ null

3. Выберите оператор, выражающий логику "или".

☐ `a | b`

☒ `a || b`

✓ Правильно. Оператор логического ИЛИ сравнивает два логических выражения.

☐ `a or b`

Рисунок 8 – Прохождение 5 модуля курса “Веб-разработка для начинающих”

1. Чтобы обратиться к конкретному элементу в массиве, следует использовать

☐ Скобки `[]`

☒ индекс;

✓ Правильно. Вы используете индекс для ссылки на определенный элемент в массиве.

☐ Фигурные скобки `{}`

2. Как узнать количество элементов в массиве?

☐ Метод `len(array)`

☐ Размер свойства в массиве

☒ Свойство `length` в массиве

✓ Правильно. Свойство `length` дает вам количество элементов.

3. В JavaScript индексы начинаются с

☒ 0

✓ Правильно. Индексы JavaScript начинаются с индекса 0.

☐ 1

☐ 2

Рисунок 9 – Прохождение 6 модуля курса “Веб-разработка для начинающих”

1. Какую часть цикла `for` вам нужно изменить, чтобы увеличить итерацию на 5?

- ☐ Условие
- ☐ Счетчик
- ☒ Выражение приращения

✓ Правильно. Выражение приращения указывает, насколько необходимо увеличить счетчик.

2. В чем разница между циклом `for` и циклом `while`?

- ☒ Цикл `for` имеет счетчик, условие и выражение приращения, а цикл `while` имеет только условие.
- ☒ Правильно. Циклы `for` и `while` различаются способами их использования.
- ☐ Цикл `while` имеет счетчик и выражение приращения, а цикл `for` имеет только условие.
- ☐ Они совпадают, просто выступают в качестве псевдонимов для друг друга.

3. С учетом кода для `(let i=1; i < 5; i++)`, сколько итераций он выполнит?

- ☐ 5
- ☒ 4
- ☒ Правильно. Код будет повторяться 4 раза, начиная с 1, при условии, что значение `i` меньше 5.
- ☐ 0

Рисунок 10 – Прохождение 6 модуля курса “Веб-разработка для начинающих”

1. Что такое Node.js?

- ☐ Платформа Java
- ☐ Платформа JavaScript
- ☒ Среда выполнения JavaScript

✓ Правильно. Node.js — это среда выполнения для приложений JavaScript.

- ☐ Сервер JavaScript

2. Как Node.js обрабатывает параллельные задачи?

- ☐ Node.js не поддерживает обработку параллельных задач.
- ☐ Node.js использует несколько потоков для параллельных задач.
- ☒ Node.js использует один цикл событий для параллельных задач.
- ☒ Правильно. Node.js использует один поток и цикл событий для обработки параллельных задач ввода-вывода без блокировки.
- ☐ Node.js использует несколько циклов событий для параллельных задач.

3. Для каких из следующих сценариев подходит Node.js?

- ☒ Приложению требуется управляемый событиями сервер без блокировки.
- ☒ Правильно. Node.js идеально подходит для реализации серверных служб API и приложений в реальном времени.
- ☐ Приложению требуется среда выполнения Java.
- ☐ Вам нужно создать мобильное приложение.
- ☐ Приложению требуются вычисления с интенсивным использованием ЦП.

Рисунок 11 – Прохождение 1 модуля курса “Создание приложений JavaScript с помощью Node.js”

1. Как установить платформу тестирования, например Jest?

☐ Выполнить `npm install jest`

☒ Выполните команду `npm install jest --save-dev`.

✓ Правильно. Эта команда устанавливает Jest как зависимость разработки. Платформа тестирования требуется только во время разработки.

☐ Выполните команду `npm install jest --save`.

☐ Выполните команду `npm download jest --save-dev`.

2. Зачем в первую очередь использовать prx?

☒ Зависимость скачивается непосредственно в процесс Node.js и удаляется после выполнения команды. Эта программа очень удобна в случаях, когда команды выполняются редко.

✓ Правильно!

☐ Устанавливать зависимости с помощью prx безопаснее по сравнению с глобальной установкой.

☐ Устанавливать зависимости с помощью prx получается быстрее по сравнению с глобальной установкой.

☐ Использовать sudo перед запуском, как при глобальной установке, необязательно.

3. Как настроить файл package.json так, чтобы получать только исправления ошибок, но не другие обновления?

☐ Найдите запись в `dependencies` или `devDependencies`. Придайте этой записи следующий вид: `"<library>": "*" .`

☐ Найдите запись в `dependencies` или `devDependencies`. Придайте этой записи следующий вид: `"<library>": "1.x.0" .`

☒ Найдите запись в `dependencies` или `devDependencies`. Придайте этой записи следующий вид: `"<library>": "1.0.x" .`

✓ Правильно!

☐ Найдите запись в `dependencies` или `devDependencies`. Придайте этой записи следующий вид: `"<library>": "^1.0.x" .`

4. Какие скрипты рекомендуется настроить в файле package.json в начале проекта?

☐ запуск, тестирование, сборка

☐ запуск, тестирование

☐ запуск

☒ запуск, тестирование, анализ, сборка

✓ Правильно. Настроить эти скрипты будет полезно для любого проекта Node.js.

Рисунок 12 – Прохождение 2 модуля курса “Создание приложений JavaScript с помощью Node.js”

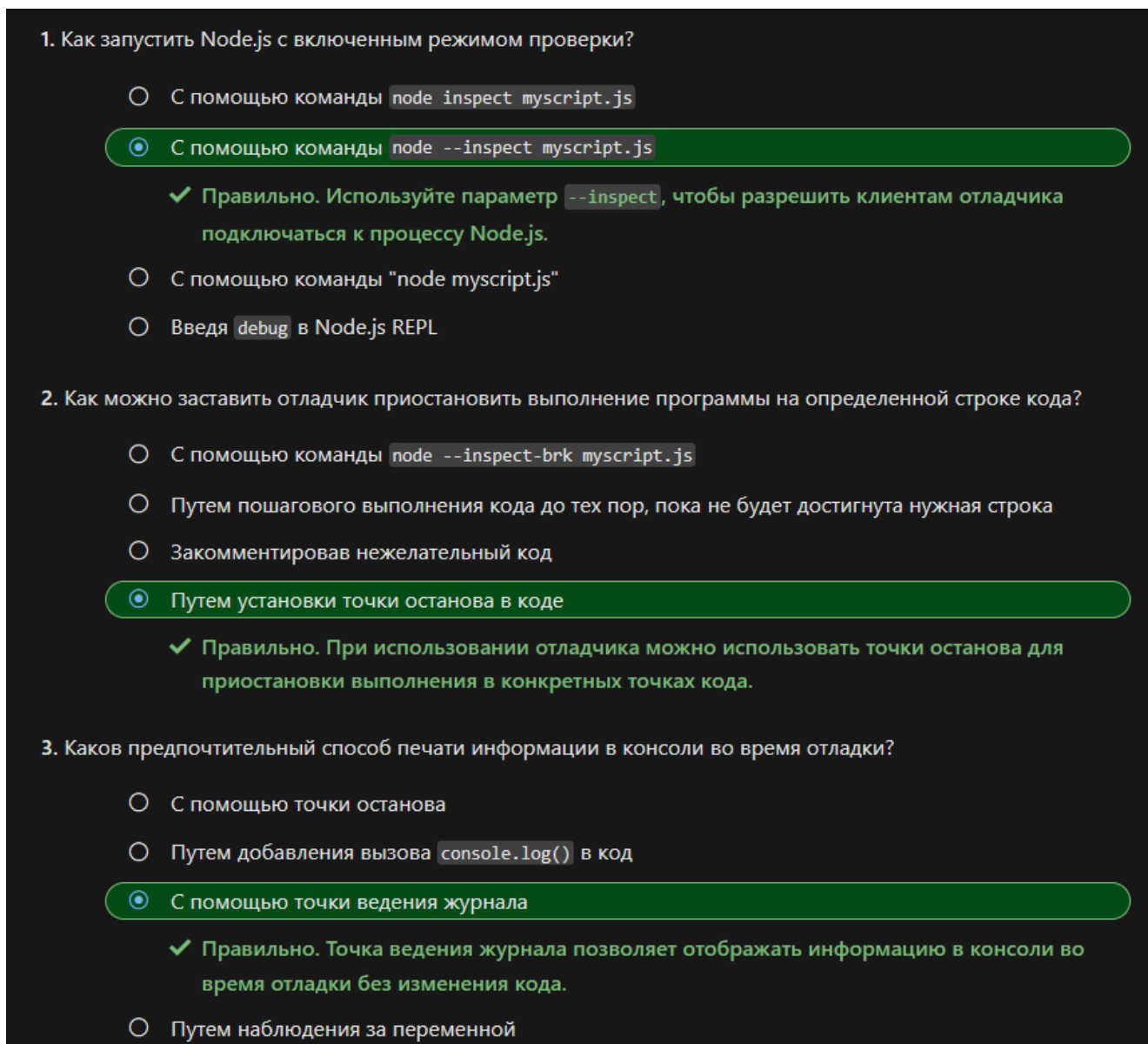


Рисунок 13 – Прохождение 3 модуля курса “Создание приложений JavaScript с помощью Node.js”

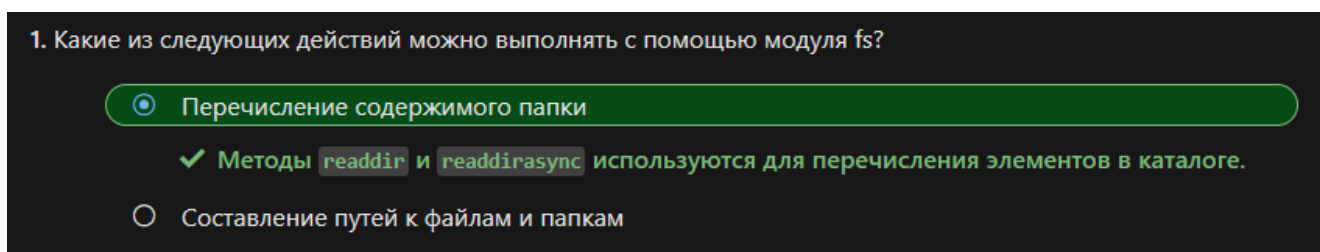


Рисунок 14 – Прохождение 4 модуля курса “Создание приложений JavaScript с помощью Node.js”

1. Какие шаги необходимо выполнить для создания веб-приложения с помощью Express?

- ☐ Создать экземпляр приложения, настроить маршруты, настроить ПО промежуточного слоя, настроить обработчики ошибок, начать ожидание передачи данных от сервера
- ☒ Создать экземпляр приложения, начать ожидание передачи данных от сервера
 - ✓ Правильно. Этих действий достаточно, чтобы запустить приложение. Настоятельно рекомендуется настроить несколько маршрутов.
- ☐ Создать экземпляр приложения, настроить маршруты, начать ожидание передачи данных от сервера
- ☐ Создать экземпляр приложения, настроить маршруты, настроить ПО промежуточного слоя, начать ожидание передачи данных от сервера

2. Каков предпочтительный способ отправки ответа JSON из приложения Express?

- ☒ Вызвать вспомогательный метод `json()` в объекте ответа `res.json({ content: '' })`
 - ✓ Правильно. Существует много способов отправки ответа в формате JSON, но это наиболее распространенный и простой.
 - ☐ Вызовите `res.send({ content: '' })`
 - ☐ Вызовите `res.send(JSON.stringify({ content: '' }))`
- Используйте любой из следующих способов: `res.type('json')`,
- ☐ `res.type('application/json')`, `res.contentType('application/json')`, `res.format({ 'application/json': function() { res.send({}) } })`

3. Как настроить в Express обработку запроса POST с данными JSON?

- ☐ Регистрация маршрута с помощью метода `post`, как в `app.post(<route>, () =>{})`, чтение из объекта `req.body`

Необходимо настроить ПО промежуточного слоя для анализа текста запроса,
- ☐ зарегистрировать маршрут с помощью метода `post`, например `app.post(<route>, () => {})`, а затем выполнить чтение из объекта `req.data`.
- ☒ Вызов `app.use(bodyParser.json())`, регистрация маршрута с помощью метода `post`, как в `app.post(<route>, () =>{})`, чтение из объекта `req.body`
 - ✓ Правильно. Этот вызов настраивает `bodyParser` для интерпретации входящих данных как JSON.
- ☐ Вызов `app.use(bodyParser.urlencoded({ extended: false }))`, регистрация маршрута с помощью метода `post`, как в `app.post(<route>, () =>{})`, чтение из объекта `req.body`

Рисунок 15 – Прохождение 5 модуля курса “Создание приложений JavaScript с помощью Node.js”

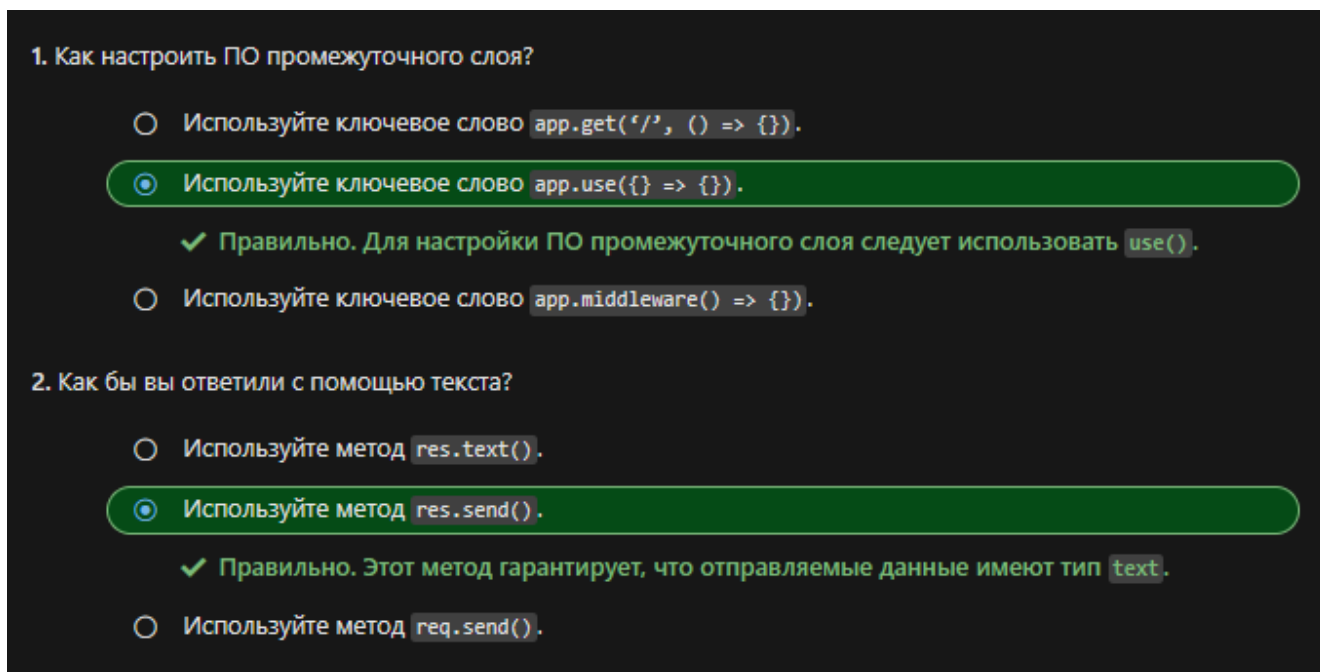


Рисунок 16 – Прохождение 6 модуля курса “Создание приложений JavaScript с помощью Node.js”

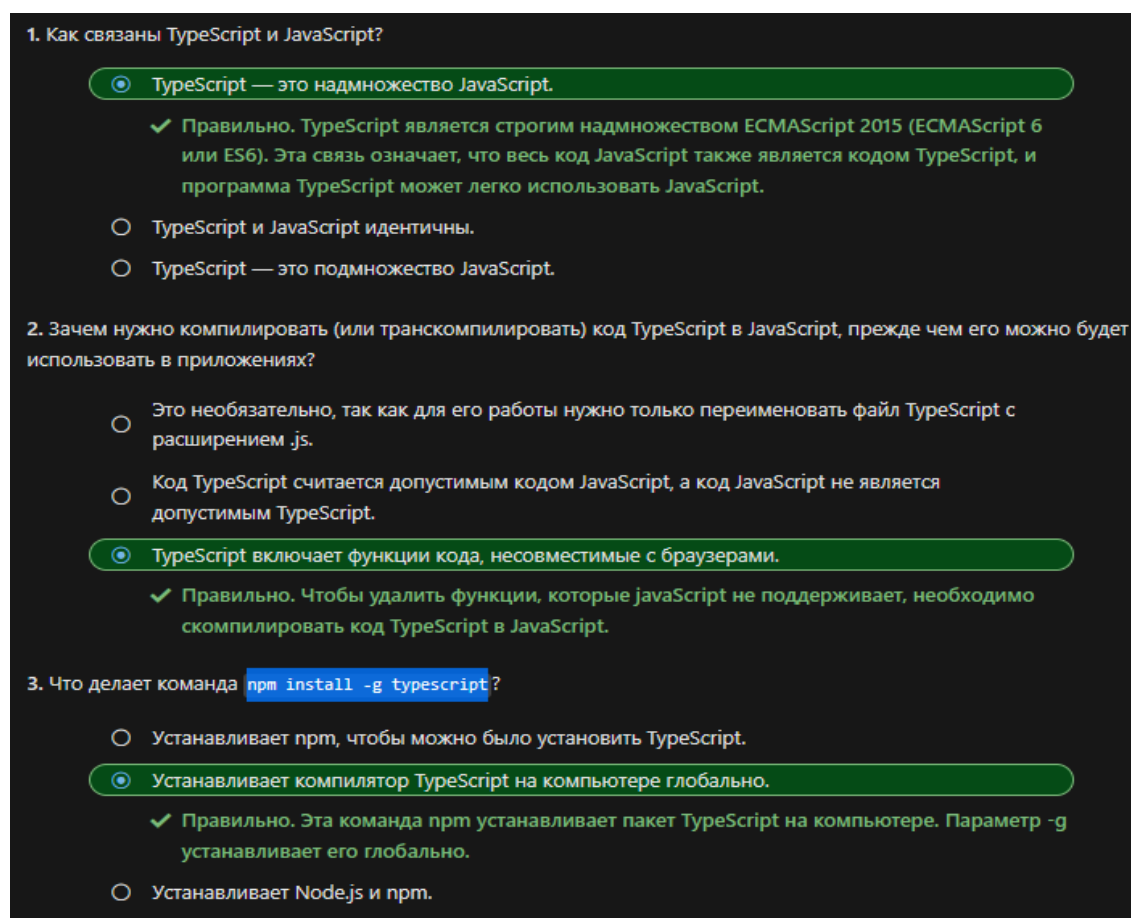


Рисунок 17 – Прохождение 1 модуля курса “Создание приложений JavaScript с помощью TypeScript”

1. К какой категории подтипов `any` относятся типы `boolean`, `number`, `string` и `enum`?

- ☐ Параметры типа.
- ☐ Типы объектов.
- ☒ Прimitives типы.

✓ Верно! Типы `boolean`, `number`, `string` и `enum` — это примеры примитивных подтипов типа `any`.

2. Какие из следующих типов являются примерами типов объектов?

- ☒ `Array`.
- ☐ `void`.
- ☐ Параметры типа.

✓ Верно! `Array` — это пример типа объекта.

3. В чем основное различие между типами `any` и `unknown`?

- ☐ Типу `unknown` можно присвоить любое значение, а у типа `any` есть ряд ограничений.
- ☐ К свойствам типа `unknown` можно обращаться, а к свойствам типа `any` нет.
- ☒ К свойствам типа `any` можно обращаться, а к свойствам типа `unknown` нет.

✓ Верно! При использовании типа можно получить доступ к любым свойствам `any`, а также вызывать или создать их.

4. Как называется функция в TypeScript, сообщающая компилятору: "Я знаю, что делаю"?

- ☐ Сужение литерала.
- ☒ Утверждение типа.
- ☐ Условие типа.

✓ Верно! Утверждение типа сообщает TypeScript, что вы выполнили все необходимые специальные проверки перед вызовом инструкции.

5. Что такое кортеж?

- ☐ Массив с бесконечным числом элементов одного типа.
- ☐ Массив с определенным числом элементов одного типа.
- ☒ Массив с определенным числом элементов одного или нескольких типов.

✓ Верно! Кортеж — это массив с определенным числом элементов, каждый из которых имеет определенный тип. При присвоении значений все элементы должны располагаться в точно заданном порядке.

Рисунок 18 – Прохождение 2 модуля курса “Создание приложений JavaScript с помощью TypeScript”

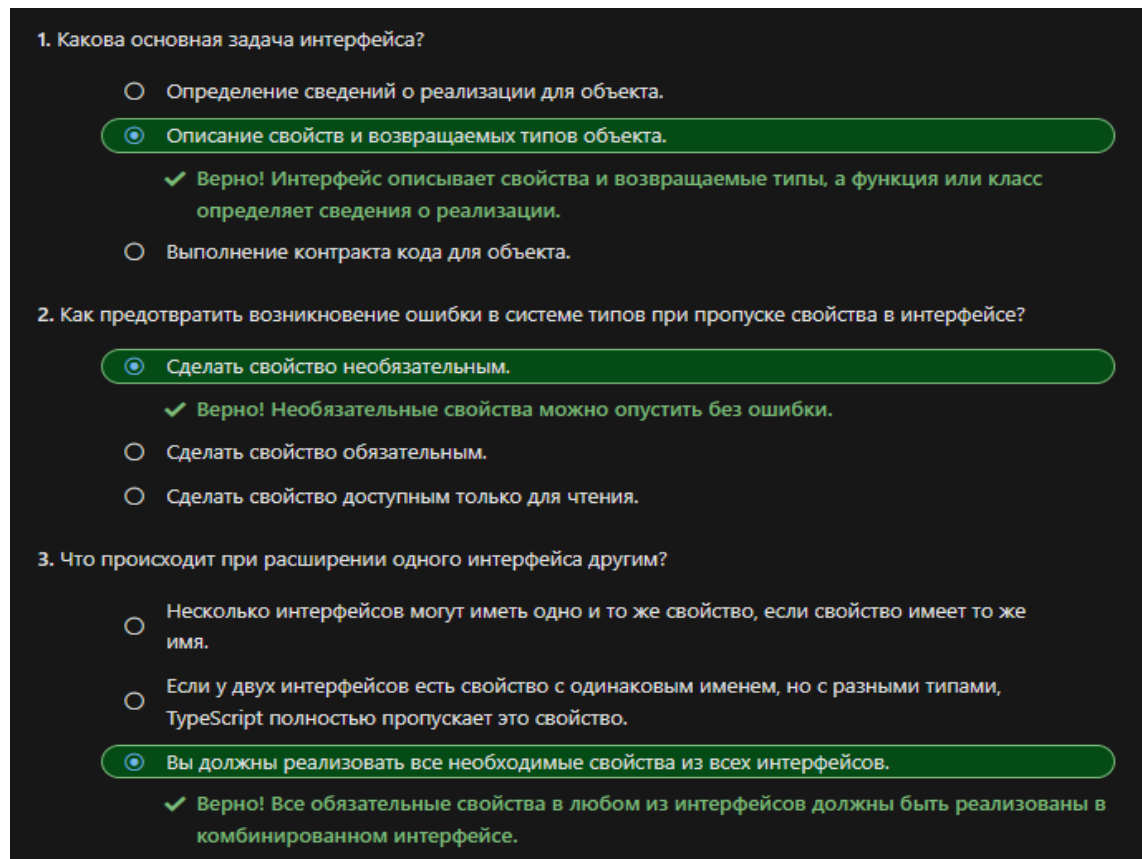


Рисунок 19 – Прохождение 3 модуля курса “Создание приложений JavaScript с помощью TypeScript”

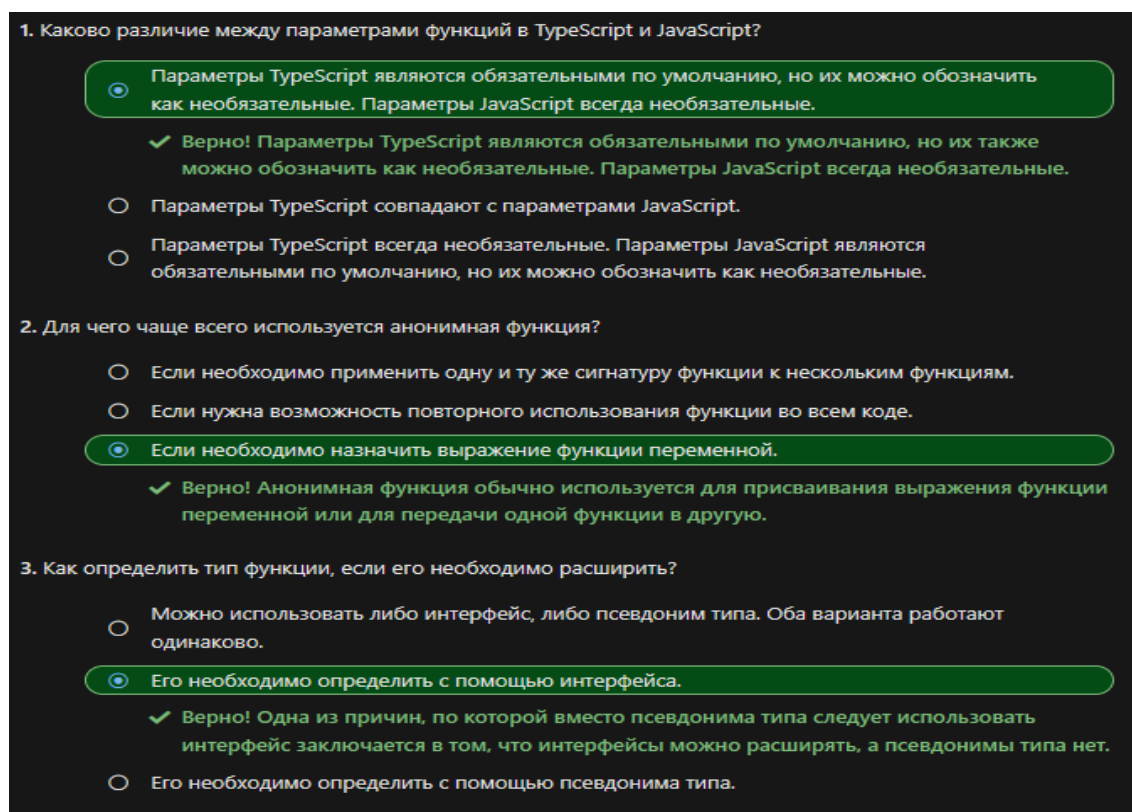


Рисунок 20 – Прохождение 4 модуля курса “Создание приложений JavaScript с помощью TypeScript”

1. Сколько функций `constructor` можно включить в определение класса?

☒ Один.

✓ Верно! В определении класса можно объявить одну функцию `constructor`. Если она не объявлена, для вас будет предоставлен автоматический конструктор.

☐ Любое число.

☐ Отсутствует.

2. Что произойдет, если опустить метод доступа `get` для свойства класса?

☐ Вы не сможете задать его значение из кода.

☐ Ничего. TypeScript автоматически предоставляет функциональные возможности метода доступа `get`, если он пропущен.

☒ Вы не сможете вернуть его значение из кода.

✓ Верно! Если метод доступа `get` опущен, свойство становится недоступным.

3. Какой модификатор доступа следует использовать, если вы не хотите, чтобы метод был доступен из кода за пределами класса, но при этом был доступен в других классах, производных от данного класса?

☐ `readonly`

☒ `protected`

✓ Верно! Модификатор `protected` делает метод невидимым за пределами класса, но при этом позволяет использовать его в производных классах.

☐ `private`

4. Какова цель ключевого слова `super` при расширении класса?

☐ Оно инициализирует свойства в подклассе.

☐ Он переопределяет свойства базового класса.

☒ Оно обеспечивает выполнение функции `constructor` базового класса при его запуске.

✓ Верно! Перед инициализацией свойств в подклассе ключевое слово `super` выполняет функцию `constructor` базового класса. Оно должно располагаться перед всеми экземплярами `this`.

Рисунок 21 – Прохождение 5 модуля курса “Создание приложений JavaScript с помощью TypeScript”

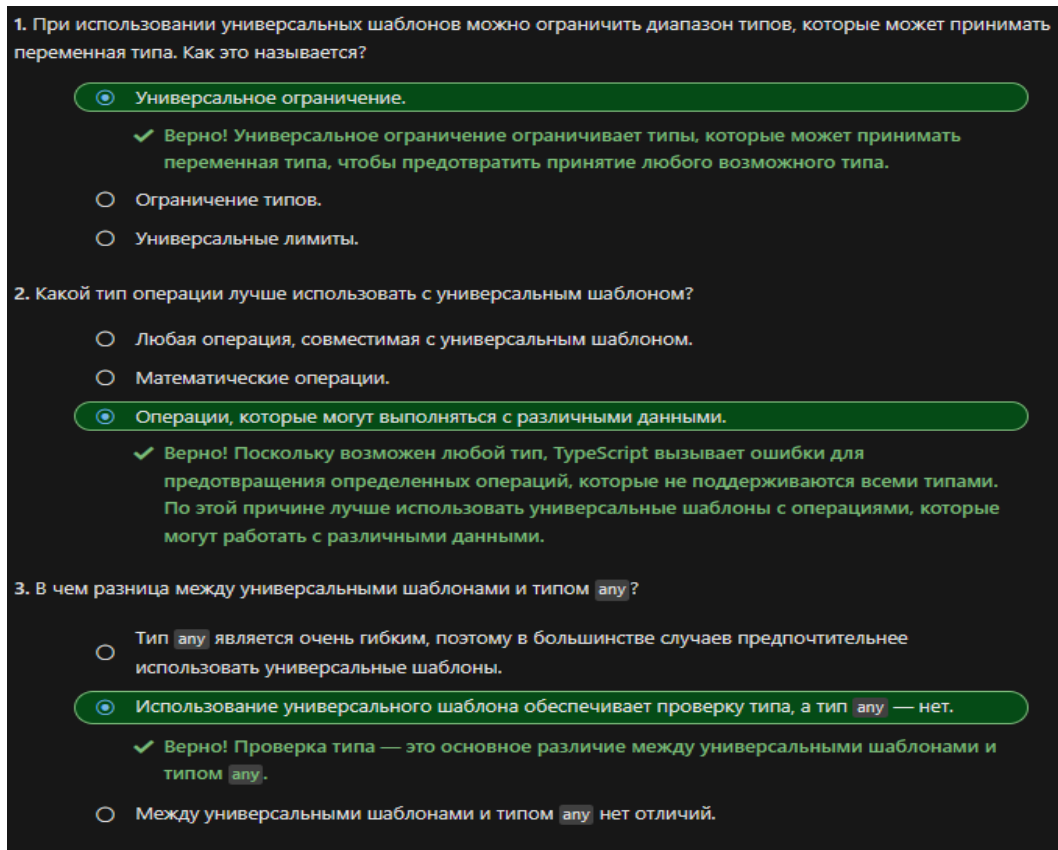


Рисунок 22 – Прохождение 6 модуля курса “Создание приложений JavaScript с помощью TypeScript”

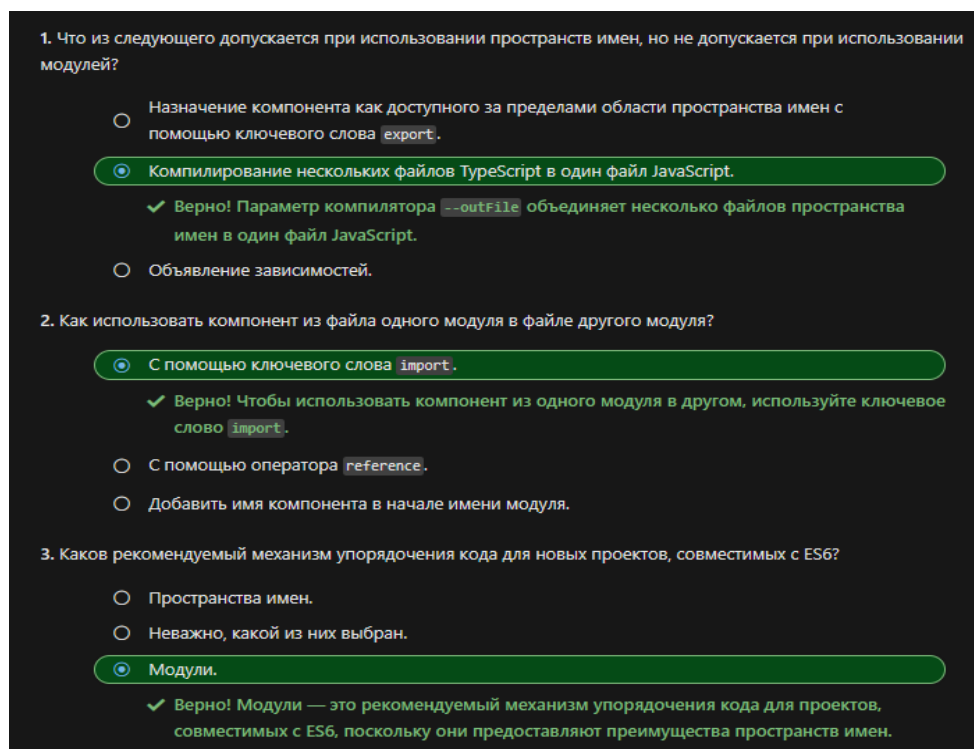


Рисунок 23 – Прохождение 7 модуля курса “Создание приложений JavaScript с помощью TypeScript”

1. Что происходит при добавлении кода в пространство имен?

- ☒ **Объявления удаляются из глобального пространства имен.**
 - ✓ Верно! Объявления удаляются из глобального пространства имен, предоставляя контекст для имен и помогая сократить конфликты именования.
- ☐ Объявления удаляются из глобального пространства имен, только если пространство имен находится в отдельном файле.
- ☐ Объявления приводят к загрязнению глобальной области.

2. Как сделать компонент доступным за пределами области пространства имен?

- ☐ Добавьте ключевое слово `import` в файл, который будет использовать компонент.
- ☒ **Добавьте ключевое слово `export` в объявление.**
 - ✓ Верно! Добавление ключевого слова `export` делает компонент доступным для использования за пределами пространства имен.
- ☐ Добавьте инструкцию `reference` в файл, который будет использовать компонент.

3. Какой параметр указывает компилятору TypeScript выводить несколько файлов, которые используют и содержат пространства имен, в один файл JavaScript?

- ☐ `--multifile`
- ☒ **`--outfile`**
 - ✓ Верно! Параметр `--outfile` указывает компилятору выводить один файл JavaScript.
- ☐ Никаких параметров не требуется. Компилятор TypeScript выполняет эту операцию по умолчанию.

Рисунок 24 – Прохождение 8 модуля курса “Создание приложений JavaScript с помощью TypeScript”

The screenshot shows a user profile for "Халявин Арсений Константинович" (stud143703@vyatsu.ru). The profile includes statistics: 20 значки (badges), 3 награды (awards), 0 баллов репутации (reputation points), 0 принятых ответов (accepted answers), 0 подписок (subscriptions), and 0 подписчиков (followers). A progress bar indicates "УРОВЕНЬ 6" (Level 6) with a score of 31,400/32,299.

Under the "Достижения" (Achievements) section, there are three displayed achievements:

- НАГРАДА** (Award): Создание приложений JavaScript с помощью TypeScript. Дата завершения: 13.03.2023.
- НАГРАДА** (Award): Создание приложений JavaScript с помощью Node.js. Дата завершения: 13.03.2023.
- НАГРАДА** (Award): Веб-разработка для начинающих. Дата завершения: 07.03.2023.

Рисунок 25 – Пройденные интерактивные курсы microsoft

Вывод: в ходе лабораторной работы были получены знания по данным курсам: “Веб-разработка для начинающих” , “Создание приложений JavaScript с помощью Node.js” , “Создание приложений JavaScript с помощью TypeScript”.