

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Никишкина Татьяна Юрьевна
Должность: Директор
Дата подписания: 30.01.2026 14:20:28
Уникальный программный ключ:
0a44cc3e5f5bb746f50173cba89718ffd301d635

ФЕДЕРАЛЬНОЕ АГЕНТСТВО МОРСКОГО И РЕЧНОГО ТРАНСПОРТА

**ОМСКИЙ ИНСТИТУТ ВОДНОГО ТРАНСПОРТА (ФИЛИАЛ)
ФЕДЕРАЛЬНОГО ГОСУДАРСТВЕННОГО БЮДЖЕТНОГО
ОБРАЗОВАТЕЛЬНОГО УЧРЕЖДЕНИЯ ВЫСШЕГО ОБРАЗОВАНИЯ
«СИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ВОДНОГО ТРАНСПОРТА»**

УТВЕРЖДАЮ

Заместитель директора по УиНР ВО
ОИВТ (филиал) ФГБОУ ВО «СГУВТ»

Калекин В.В.

« ____ » ____ 20 ____ г.

Шифр дисциплины: 2023.09.03.02.01 . Б1.В.21
(шифр ОПОП) (шифр дисциплины из учебного плана)

Рабочая программа дисциплины (модуля)

Наименование дисциплины (модуля):

Современные технологии разработки

(полное наименование дисциплины (модуля), в строгом соответствии с учебным планом)

Омск

2023

(год набора)

Составлена _____
(должность) кафедрой _____
(сокращенное наименование кафедры)

_____ , _____
(ученая степень) (ученое звание) _____
(И.О. Фамилия)

_____ О.Л. Курнявко « _____ » 20 _____ г.
(подпись) (И.О. Фамилия) число месяц год

Рассмотрена и одобрена на заседании кафедры естественных наук и
информационных технологий

(наименование кафедры полностью)

от « _____ » 20 _____ г., Протокол № _____
число месяц год

Зав. кафедрой _____ ЕНиИТ _____ О.Л. Курнявко
(сокращенное наименование кафедры) (подпись) (И.О.Фамилия)

Согласована:

Руководитель ОПОП _____ К.Т.Н. _____ Т.Л. Иванова
(ученая степень, ученое звание) (подпись) (И.О. Фамилия)

Заведующий библиотекой ОИВТ _____ И.Д. Скоробогатова
(подпись) (И.О. Фамилия)

1 Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения образовательной программы

1.1 Цели дисциплины

Целью данной дисциплины является формирование у студентов фундамента естественно-научных знаний, базирующихся на законах природы, правильных представлений о современной картине мира, а также навыков применения физических законов к прикладным задачам современной техники.

1.2 Перечень формируемых компетенций

В результате освоения дисциплины (модуля) у обучающегося должны сформироваться следующие компетенции, выраженные через результат обучения по дисциплине (модуля), как часть результата освоения образовательной программы (далее – ОП):

Компетенция		Перечень планируемых результатов обучения по дисциплине
Шифр	Содержание	
УК-2	Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений.	УК-2.1. Знать: виды ресурсов и ограничений для решения профессиональных задач; основные методы оценки разных способов решения задач; действующее законодательство и правовые нормы, регулирующие профессиональную деятельность. УК-2.2. Уметь: проводить анализ поставленной цели и формулировать задачи, которые необходимо решить для ее достижения; анализировать альтернативные варианты для достижения намеченных результатов; использовать нормативно-правовую документацию в сфере профессиональной деятельности. УК-2.3. Владеть: методиками разработки цели и задач проекта; методами оценки потребности в ресурсах, продолжительности и стоимости проекта; навыками работы с нормативно-правовой документацией.
ПК-15	Способность выполнять элементы графического дизайна интерфейсов информационных систем и визуализации данных	ПК-3.1. Знать современные технологии, подходы, стандарты разработки дизайна программного обеспечения. ПК-3.2. Уметь понимать процесс разработки дизайна программного обеспечения. ПК-3.3. Иметь навыки создания интерфейсов и визуализации данных.

2 Место дисциплины (модуля) в структуре образовательной программы

Дисциплина (модуль) реализуется в рамках вариативной части
(базовой, вариативной или факультативной)

основной профессиональной образовательной программы.

3 Объем дисциплины (модуля) в зачетных единицах (ЗЕ) с указанием количества академических или астрономических часов, выделенных на контактную работу обучающихся с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающихся

Для очной формы обучения:
(очной, очно-заочной или заочной)

Форма контроля					з.е.		-	Итого акад. часов					Курс 2									
													Сем. 4									
Экзамен	Зачет	Зачет с оц.	КР	Контр.	Экспертное	Факт	Часов в з.е.	Экспертное	По плану	Контакт часы	СР	Контроль	з.е.	Итого	Лек	Лаб	Пр	КСР	КоР	КРП	СР	Контроль
5					5	5	36	180	180	58	86	36	5	180	10	44				4	86	36

Для заочной формы обучения:
(очной, очно-заочной или заочной)

Форма контроля					з.е.		-	Итого акад. часов					Курс 2									
Экзамен	Зачет	Зачет с оц.	КР	Контр.	Экспертное	Факт	Часов в з.е.	Экспертное	По плану	Контакт часы	СР	Контроль	з.е.	Итого	Лек	Лаб	Пр	КСР	КоР	КРП	СР	Контроль
3					5	5	36	180	180	32	130	18	5	180	8	20		2		4	130	18

4 Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

4.1 Разделы и темы дисциплины (модуля) и трудоемкость по видам учебных занятий (в академических часах)

№	Разделы и темы дисциплины (модуля)	Виды учебных занятий, включая СРС											
		Лек			Лаб			Пр			СРС		
		О	ОЗ	З	О	ОЗ	З	О	ОЗ	З	О	ОЗ	З
1.	Система контроля версий Git	2		2	4		4				14		26
2.	React	2		2	10		4				18		26
3.	NestJS	2			10		4				18		26
4.	TypeORM	2		2	10		4				18		26
5.	Типовые паттерны разработки	2		2	10		4				18		26
ИТОГО		10		8	44		20				86		130

Примечания: О – очная форма обучения, ОЗ – очно-заочная форма обучения, З – заочная форма обучения.

4.2 Содержание разделов и тем дисциплины

Тема 1. Понятие фреймворка

Разработка с чистого листа VS фреймворк.

Тема 2. React

Понятие о React как о JavaScript-библиотеке с открытым исходным кодом для разработки пользовательских интерфейсов. React Использование React для разработки одностраничных и мобильных приложений.

Тема 3. NestJS

Понятие о NestJS. Инструмент командной строки. Декораторы. Переменные окружения. DTO и валидация данных. Тестирование приложения. Деплой приложения.

Тема 4. TypeORM

Настройки и подключения. Инструмент командной строки. Сущности и модели. Синхронизация. Сиды и миграции.

Тема 5. Типовые паттерны разработки

Типовые паттерны разработки программного обеспечения как устойчивые и многократно проверенные решения распространённых архитектурных и проектных задач. Роль паттернов в повышении читаемости, сопровождаемости и расширяемости программного кода, а также их значение как общего профессионального языка разработчиков.

Базовая классификация паттернов (порождающие, структурные, поведенческие), принципы их применения и ограничения, а также связь паттернов с объектно-ориентированным подходом и принципами SOLID. Примеры использования типовых паттернов в прикладных задачах разработки и анализу ситуаций, в которых применение паттернов является оправданным.

4.3 Содержание лабораторных работ

Дисциплина не предусматривает проведение лабораторных работ.

4.4 Содержание практических занятий

Раздел (тема) дисциплины	Содержание практического занятия
Система контроля версий Git	Примеры использования
React	Примеры использования
NestJS	Примеры использования
TypeORM	Примеры использования
Типовые паттерны разработки	Примеры использования

4.5 Курсовая работа

Дисциплина не предусматривает курсовую работу.

4.6 Курсовой проект

Дисциплина не предусматривает курсовой проект.

4.7 Самостоятельная работа. Контроль самостоятельной работы

В самостоятельную работу студента входит подготовка к занятиям по дисциплине путём изучения теоретического материала по источникам, указанным в перечне основной и дополнительной литературы, выполнение практических работ, оформление практических заданий и подготовка к их защите.

Контроль самостоятельной работы осуществляется в ходе практических и лабораторных занятий.

4.8 Интерактивные формы обучения

В процессе освоения данной учебной дисциплины интерактивные формы используются при изучении следующих тем:

Раздел (тема) дисциплины	Количество часов		
	О	ОЗ	З
Система контроля версий Git	10		4
React	10		4
NestJS	10		4
TypeORM	10		4
Типовые паттерны разработки	10		4
Итого	50		20

5 Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине (модулю)

5.1 Описание показателей и критериев оценивания компетенций, описание шкал оценивания

Показатели оценивания	Описание показателя
Оценка «отлично» (зачтено)	Обучаемый демонстрирует способность к полной самостоятельности (допускаются консультации с преподавателем по сопутствующим вопросам) в выборе способа решения неизвестных или нестандартных заданий с использованием знаний, умений и навыков, полученных в ходе проведения освоения дисциплины, следует считать компетенцию сформированной на высоком уровне. Присутствие сформированной компетенции на высоком уровне означает способность студента к ее дальнейшему саморазвитию и высокой адаптивности практического применения к изменяющимся условиям профессиональной задачи.

Показатели оценивания	Описание показателя
Оценка «хорошо» (зачтено)	Обучаемый демонстрирует самостоятельное применение знаний, умений и навыков при решении заданий, аналогичных тем, которые предоставлял преподаватель при потенциальном формировании компетенции, подтверждает наличие сформированной компетенции, причем на более высоком уровне. Наличие сформированной компетенции на повышенном уровне самостоятельности со стороны обучаемого при ее практической демонстрации в ходе решения аналогичных заданий следует оценивать как положительное и устойчиво закрепленное в практическом навыке.
Оценка «удовлетворительно» (зачтено)	Обучаемый демонстрирует самостоятельность в применении знаний, умений и навыков к решению учебных заданий в полном соответствии с образцом, данным преподавателем, по заданиям, решение которых было показано преподавателем, следует считать, что компетенция сформирована, но ее уровень недостаточно высок. Поскольку выявлено наличие сформированной компетенции, ее следует оценивать положительно, но на низком уровне.
Оценка «неудовлетворительно» (не зачтено)	Обучаемый не способен самостоятельно продемонстрировать наличие знаний при решении заданий, которые были представлены преподавателем вместе с образцом их решения, отсутствие самостоятельности в применении умения к использованию методов проведения пред- дипломной практики и неспособность самостоятельно проявить навык повторения решения поставленной задачи по стандартному образцу свидетельствуют об отсутствии сформированной компетенции.

5.2 Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений и (или) навыков, характеризующих этапы формирования компетенций в процессе освоения образовательной программы

Задача: разверните шаблон React-проекта, используя инструмент командной строки.

Задача: создайте компонент React.

Задача: реализуйте юнит-тест для указанной функциональности.

Задача: разверните шаблон NestJS-проекта, используя инструмент командной строки.

Задача: подключите БД MySQL, используя TypeORM.

5.3 Перечень вопросов для итогового контроля по дисциплине

Система контроля версий Git

1. Назначение систем контроля версий. Основные преимущества использования Git.
2. Архитектура Git: рабочий каталог, индекс (staging area), репозиторий.
3. Основные команды Git и их назначение (`init`, `clone`, `status`, `add`, `commit`, `log`).
4. Ветвление в Git: понятие ветки, создание и переключение веток.
5. Слияние веток: `merge` и `rebase`, отличия и области применения.
6. Разрешение конфликтов при слиянии.
7. Работа с удалёнными репозиториями: `remote`, `fetch`, `pull`, `push`.
8. Назначение `.gitignore`.
9. Коммиты как единицы истории проекта: требования к сообщениям коммитов.
10. Типовой Git-workflow в командной разработке.

React

11. Назначение React и его место во frontend-разработке.
12. Компонентная модель React. Функциональные и классовые компоненты.
13. JSX: назначение и особенности синтаксиса.
14. Свойства (props) и состояние (state): различия и области применения.
15. Жизненный цикл компонента в React.
16. Хуки React: назначение и примеры (`useState`, `useEffect`).
17. Управление событиями в React-приложении.
18. Условный рендеринг и списки элементов.
19. Однонаправленный поток данных в React.
20. Архитектура frontend-приложения на React.

NestJS

21. Назначение NestJS и его основные архитектурные принципы.
22. Модульная структура NestJS-приложения.
23. Контроллеры и маршрутизация запросов.
24. Провайдеры и сервисы: назначение и роль `dependency injection`.
25. `Dependency Injection` в NestJS: принципы и преимущества.
26. `Middleware`, `Guards`, `Pipes` и `Interceptors`: назначение и различия.
27. Обработка ошибок и исключений в NestJS.
28. Конфигурация приложения и работа с переменными окружения.
29. `REST-API` в NestJS: принципы проектирования.
30. Взаимодействие NestJS с frontend-приложением (например, React).

TypeORM

31. Назначение ORM и преимущества использования ORM-подхода.
32. Основные концепции TypeORM: сущность (Entity), репозиторий, менеджер.
33. Описание сущностей и их отображение на таблицы базы данных.
34. Типы связей между сущностями: `OneToOne`, `OneToMany`, `ManyToMany`.
35. Работа с репозиториями: основные CRUD-операции.

36. Запросы к базе данных в TypeORM.
37. Миграции базы данных: назначение и жизненный цикл.
38. Валидация данных и ограничения на уровне сущностей.
39. Интеграция TypeORM с NestJS.
40. Типовые ошибки при работе с ORM и способы их предотвращения.

Интеграция технологий

41. Общая архитектура full-stack-приложения (React + NestJS + TypeORM).
42. Взаимодействие frontend и backend через REST-API.
43. Управление состоянием данных между клиентом и сервером.
44. Роль Git в разработке и сопровождении full-stack-проекта.
45. Типовой жизненный цикл запроса пользователя от UI до базы данных.

5.4 Методические рекомендации к проведению итогового контроля по дисциплине

На итоговый контроль выносятся учебный материал, рассмотренный на лекционных и практических занятиях, а также темы, которые были предложены для самостоятельного изучения.

Экзамен по дисциплине может проводиться в одном из следующих форматов: экзамен в устной форме (классический), защита проектной работы. Формат экзамена определяется преподавателем и доводится до сведения обучающихся заблаговременно.

Возможен комбинированный вариант, когда часть студентов защищает проект, а другая часть – сдает классический экзамен.

Экзамен в устной форме (классический)

Экзамен проводится в устной форме с использованием билетов, содержащих теоретическую и практическую часть. Студент получает экзаменационный билет случайным образом, осуществляет подготовку и затем отвечает преподавателю. Во время подготовки к ответу на экзамене использование справочной литературы не допускается.

Экзаменационный билет включает:

- два теоретических вопроса;
- одну практическую задачу.

Время на подготовку составляет **40 минут**. Ответ предполагает устное изложение теоретического материала, пояснение используемых понятий и решений, а также разбор практической задачи. Преподаватель вправе задавать уточняющие и дополнительные вопросы в рамках содержания дисциплины.

Экзамен в форме защиты/демонстрации проекта

Экзамен может быть проведён в форме защиты/демонстрации учебного проекта. Для выполнения проекта обучающиеся объединяются в проектные группы. Все проектные группы получают **единое техническое задание**, в рамках которого осуществляется разработка программного продукта.

Внутри каждой проектной группы формируются функциональные подгруппы:

- команда frontend-разработки;
- команда backend-разработки;
- команда менеджмента (координация, планирование, документация).

Экзамен включает:

- демонстрацию работоспособности итогового продукта;
- представление архитектурных и технологических решений;
- ответы студентов на вопросы преподавателя по реализованному проекту.

При оценивании учитываются:

- соответствие результата поставленной задаче;
- корректность архитектурных решений;
- качество реализации;
- понимание студентами своей роли в проекте и используемых технологий.

6 Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины (модуля)

6.1 Основная учебная литература

1. Б. Карривей. Интернет, краткая история и влияние на общество. – LAMBERT Academic Publishin, 2015 г. – 368 с.
2. Хоган Б. HTML5 и CSS3. Веб-разработка по стандартам нового поколения. 2 изд. – СПб: Питер, 2014 – 320 с.
3. М. Мориссон. Изучение JavaScript. – СПб: Питер, 2012 – 590 с.
4. М.С. Миковски, Дж.К. Пауэлл. Разработка одностраничных веб-приложений. – М: ДМК Пресс, 2014 г. – 512 с.

6.2 Дополнительная учебная литература

1. А. Фримен. ASP.NET MVC 5 с примерами на C# 5.0. – Apress, 2014г. – 736 с.

6.3 Периодические издания

1. Вестник Омского университета [Текст]: ежекварт. науч. журнал/ Мин-во образ. и науки РФ, Омский гос. ун-т. – Омск, 2013-2016
2. Известия ВУЗ. Физика [Текст]: ежемес. науч. журнал/ Мин-во образ. и науки РФ, Томский гос. ун-т. - Томск: Сиб. физико-технич. ин-т, 2012-2018
3. Известия ТулГУ. Естественные науки [Текст]: ежемес. науч. журнал/ Мин-во образ. и науки РФ, Тульский гос. ун-т. - Тула, 2008-2015

7 Перечень ресурсов информационно-телекоммуникационной сети "Интернет" (далее - сеть "Интернет"), необходимых для освоения дисциплины (модуля)

1. Электронная библиотечная система «Лань» [Электронный ресурс]. – Режим доступа: <https://e.lanbook.com/>
2. Научная электронная библиотека [Электронный ресурс]. – Режим доступа: <https://elibrary.ru/>
3. Математический портал «EqWorld» [Электронный ресурс]. – Режим доступа: <http://eqworld.ipmnet.ru/ru/info/mathwebs.htm>
4. Электронный каталог НГАВТ [Электронный ресурс]. – Режим доступа: <http://libcat.nsawt.ru/>
5. Статья «Wiki-учебник по веб-технологиям: HTML» и далее по ссылкам. <http://www.webmasterwiki.ru/HTML>
6. Статья «Wiki-учебник по веб-технологиям: CSS» и далее по ссылкам. <http://www.webmasterwiki.ru/CSS>
7. Статья «Wiki-учебник по веб-технологиям: JavaScript» и далее по ссылкам. <http://www.webmasterwiki.ru/JavaScript>

8 Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения и информационных справочных систем (при необходимости)

1. Среда разработки «Node.js».
2. Фреймворк «NestJS».
3. Редактор «VS Code»

9 Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)

№ п/п	Наименование вида образования, уровня образования, профессии, специальности, направления подготовки (для профессионального образования), подвида дополнительного образования	Наименование оборудованных учебных кабинетов, объектов для проведения практических занятий, объектов физической культуры и спорта с перечнем основного оборудования	Адрес (местоположение) учебных кабинетов, объектов для проведения практических занятий, объектов физической культуры и спорта (с указанием номера помещения в соответствии с документами бюро технической инвентаризации)
15.14	Современные технологии разработки	1. PyCharm (Python IDE) 2. Visual Studio Code (редактор кода) 3. Visual Studio Community Edition a. Разработка на Node.js b. Разработка на Python c. Разработка классических приложений на C++ d. Разработка приложений WinUI 4. MAMP 5. Git (система контроля версий) 6. SourceTree 7. Java SE 8. Tomcat 9. Node.js 10. Eclipse IDE for Enterprise Java and Web Developers 11. Eclipse IDE for C and C++ Developers 12. Postman Учебная аудитория для проведения групповых и индивидуальных консультаций: ПК – 14 шт., ученическое место - не менее 21 шт., стол преподавательский с ПК - не менее 1 шт., доска учебная - не менее 1 шт., экран проекционный - не менее 1 шт., стул - не менее 43 шт., рамки портретные - не менее 6 шт., плакаты - не менее 5 шт.	644099, г.Омск, ул. Ивана Алексеева, д. 2 (№№ 52, 53, 54, 55, 56 на поэтажном плане (1П), 4 этаж) 644099, г.Омск, ул. Ивана Алексеева, д. 2 (№№ 31, 32 на поэтажном плане (1П), 4 этаж)