

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ  
ФЕДЕРАЦИИ

Ярославский государственный университет им. П.Г. Демидова

Кафедра вычислительных и программных систем

УТВЕРЖДАЮ

Декан факультета ИВТ

 Д.Ю. Чалый

« 18 » мая 2021 г.

**Рабочая программа дисциплины**

«Основы современной промышленной frontend-разработки»

**Направление подготовки**

02.03.02 Фундаментальная информатика и информационные технологии

**Профиль**

«Информатика и компьютерные науки»

**Квалификация выпускника**

Бакалавр

**Форма обучения**

очная

Программа рассмотрена  
на заседании кафедры  
от 23 апреля 2021 г.,  
протокол № 8

Программа одобрена НМК  
факультета ИВТ  
протокол № 7 от  
17 мая 2021 г.

Ярославль  
2021

### 1. Цели освоения дисциплины

Целями дисциплины «Основы современной промышленной frontend-разработки» являются получение обучающимися знаний и умений в области проектирования, тестирования, отладки, внедрения и сопровождения программного обеспечения (ПО).

### 2. Место дисциплины в структуре ОП магистратуры

Дисциплина «Основы современной промышленной frontend-разработки» относится к факультативной части ОП бакалавриата. В курсе рассматриваются вопросы разработки приложений на современных языках программирования с учётом их специфических особенностей.

Содержание курса тесно связано фактически со всеми дисциплинами, которые изучались студентами. Освоению данной программы предшествуют учебные курсы по программированию и современным информационным технологиям.

Дисциплина «Основы современной промышленной frontend-разработки» знакомит магистрантов с современными тенденциями развития информатики и вычислительной техники, компьютерных технологий; основами создания информационных систем и использованием новых информационных технологий разработки программных систем.

### 3. Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения ОП магистратуры

Процесс изучения дисциплины направлен на формирование следующих элементов компетенций в соответствии с ФГОС ВО, ОП ВО и приобретения следующих знаний, умений, навыков и (или) опыта деятельности:

Формируемая компетенция (код и формулировка)	Индикатор достижения компетенции (код и формулировка)	Перечень планируемых результатов обучения
<b>Профессиональные компетенции</b>		
ПК-3 Способен к разработке стратегии тестирования и управлению процессом тестирования	ПК-3.2 Знает и умеет использовать основные методы тестирования программных продуктов	Знать: <ul style="list-style-type: none"><li>● специализированную терминологию;</li><li>● специализированное программное обеспечение для проектирования и разработки информационного контента;</li><li>● технологические стандарты проектирования и разработки информационного контента;</li><li>● принципы построения информационных ресурсов;</li><li>● методы тестирования и отладки программного обеспечения;</li><li>● стандарты составления и оформления технической документации;</li><li>● характеристики качества программного продукта;</li></ul> Уметь <ul style="list-style-type: none"><li>● осуществлять построение информационно-логических моделей</li></ul>

		информационных ресурсов; • разрабатывать программное обеспечение с помощью языков программирования • адаптировать и конфигурировать программное обеспечение для решения поставленных задач; Владеть • навыками разработки и публикации программного обеспечения, • методами верификации и контроля качества продуктов;
--	--	---

#### 4. Объем, структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 1 зач. ед., 36 акад. час.

№ п/п	Темы (разделы) дисциплины, их содержание	Сем естр	Виды учебных занятий, включая самостоятельную работу студентов, и их трудоемкость (в академических часах)						Формы текущего контроля успеваемости	Форма промежуточной аттестации (по семестрам)
			Контактная работа							
			лекц ии	пра кти чес кие	лаб ора тор ные	кон сул ьта ции	атте стац ион ные исп ыта ния	самос тоят ельная работ а		
1.	Технологии, модели и процессы создания ПО	2	1		2			4		
2.	Основы создания ПО	2	1		2			4		
3.	Разработка требований к ПО	2	1		2			4	Работа над проектом	
4.	Управление проектами по созданию и внедрению ПО	2	1		4			4		
5.	Управление персоналом при реализации проектов	2	1		4			6		
6.	Управление качеством созданных программных систем	2	1		4	1		6	Работа над проектом	
7.	Отладка и тестирование ПО	2	1		4			6		

8.	Публикация программного обеспечения	2	1		4	1		6	Защита проекта
	<b>Всего за 2 семестр</b>		<b>8</b>		<b>22</b>	<b>2</b>		<b>40</b>	<b>Зачет</b>
	<b>Всего</b>		<b>8</b>		<b>22</b>	<b>2</b>		<b>40</b>	

#### **Содержание разделов дисциплины:**

1. Технологии, модели и процессы создания ПО. Методы создания ПО. Структуры затрат на создание ПО. Основные вопросы, встающие перед специалистами по созданию ПО.
2. Основы создания ПО. Базовые процессы создания ПО. Модели создания ПО. Спецификация ПО. Реализация ПО. Аттестация ПО. Эволюция ПО. CASE-средства автоматизации процессов создания ПО.
3. Разработка требований к ПО. Формирование и анализ требований к ПО. Аттестация требований. Управление требованиями.
4. Управление проектами по созданию и внедрению ПО. Процессы управления. Контрольные отметки. Временные и сетевые диаграммы. Планирование проекта. Управление рисками.
5. Управление персоналом при реализации проектов. Решение задач. Групповая работа. Автоматизированные системы управления проектами.
6. Управление качеством созданных программных систем. Понятие качество программного продукта. Процесс управления качеством. Стандарты для управления качеством. Стандарты на техническую документацию. Показатели качества программного продукта.
7. Отладка и тестирование ПО. Верификация и контроль качества.
8. Публикация программного обеспечения. Разработка и ведение проектной и технической документации.

#### **5. Образовательные технологии, используемые при осуществлении образовательного процесса по дисциплине**

В процессе обучения используются следующие образовательные технологии:

**Лекция-беседа** или «диалог с аудиторией», является наиболее распространенной и сравнительно простой формой активного вовлечения студентов в учебный процесс. Эта лекция предполагает непосредственный контакт преподавателя с аудиторией. Преимущество лекции-беседы состоит в том, что она позволяет привлекать внимание студентов к наиболее важным вопросам темы, определять содержание и темп изложения учебного материала с учетом особенностей студентов.

**Мастер-класс** – это особая форма учебного занятия, когда преподаватель-мастер передает свой опыт путем прямого и комментированного показа последовательности действий, методов, приемов и форм педагогической деятельности. Целью проведения мастер-класса является профессиональное, интеллектуальное и эстетическое воспитание студентов, и прежде всего, развитие в ходе мастер-класса способности студента самостоятельно и нестандартно мыслить.

**Лабораторная работа** – организация учебной работы с реальными материальными и информационными объектами, экспериментальная работа с аналоговыми моделями реальных объектов.

#### **6. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень лицензионного программного обеспечения и информационных справочных систем (при необходимости)**

В процессе осуществления образовательного процесса используются: для разработки документов, презентаций, для работы с электронными таблицами

OfficeStd 2013 RUS OLP NL Acdmc 021-10232

LibreOffice (свободное)  
издательская система LaTeX;  
- Среда разработки программных проектов IntelliJ IDEA  
<http://www.jetbrains.com/idea/> Доступ свободный.  
– OS Linux (свободная)  
– для поиска учебной литературы библиотеки ЯрГУ – Автоматизированная библиотечная информационная система "БУКИ-NEXT" (АБИС "Буки-Next").

## **7. Перечень основной и дополнительной учебной литературы, ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины**

а) основная:

1. Долженко, А. И., Технологии командной разработки программного обеспечения информационных систем [Электронный ресурс] / А. И. Долженко. - 2-е изд., испр., М., Национальный Открытый Университет "ИНТУИТ", 2016, 301с.  
<http://biblioclub.ru/index.php?page=book&id=428801>

2. Парадигмы программирования : анализ и сравнение / Л. В. Городняя; РАН, Ин-т систем информатики. - Новосибирск: Изд-во СО РАН, 2017. - 231 с.

б) дополнительная:

1. Макконнелл, С., Совершенный код. Мастер-класс : пер. с англ / С. Макконнелл, М., Русская редакция, 2013, 868с

в) ресурсы сети «Интернет»

- Среда разработки программных проектов IntelliJ IDEA  
<http://www.jetbrains.com/idea/> Доступ свободный.

## **8. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине**

Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине включает в свой состав специальные помещения:

-учебные аудитории для проведения занятий лекционного типа и практических занятий (семинаров);

-учебные аудитории для проведения лабораторных занятий;

- учебные аудитории для проведения групповых и индивидуальных консультаций,

- учебные аудитории для проведения текущего контроля и промежуточной аттестации;

-помещения для самостоятельной работы;

-помещения для хранения и профилактического обслуживания технических средств обучения.

Специальные помещения укомплектованы средствами обучения, служащими для представления учебной информации большой аудитории.

Для проведения занятий лекционного типа предлагаются наборы демонстрационного оборудования и учебно-наглядных пособий, хранящиеся на электронных носителях и обеспечивающие тематические иллюстрации, соответствующие рабочим программам дисциплин.

Помещения для лабораторных занятий и самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду организации.

Число посадочных мест в лекционной аудитории больше либо равно списочному составу потока, а в аудитории для практических занятий (семинаров) – списочному составу группы обучающихся.

**Автор(ы) :**

Старший преподаватель кафедры ВПС А.М.Васильев

**Приложение №1 к рабочей программе дисциплины  
«Основы современной промышленной frontend-  
разработки»**

**Фонд оценочных средств  
для проведения текущей и промежуточной аттестации студентов  
по дисциплине**

**1. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций**

**1.1. Контрольные задания и иные материалы, используемые в процессе текущей аттестации**

**Темы проектов**

1. Разработка проекта «Случайный выбор»
2. Разработка проекта «Словарь»
3. Разработка проекта «Двигающаяся кнопка»
4. Разработка проекта «Альбом»
5. Разработка проекта «Таблица умножения»
6. Разработка проекта «Светофор»
7. Разработка проекта «Ханойские башни»
8. Разработка проекта «Электронный альбом»
9. Разработка проекта «Вычисление процентов»
10. Разработка проекта «Калькулятор»
11. Разработка проекта «Строковый калькулятор»
12. Разработка проекта «Обработка одномерных и двумерных массивов»
13. Разработка проекта «Текущее время и текущая дата»
14. Разработка проекта «Электронные часы»
15. Разработка проекта «Блокнот–органайзер»
16. Разработка проекта «Справочник»
17. Разработка описания и анализ информационной системы
18. Разработка требований к информационной системе
19. Методология функционального моделирования
20. Методология объектно-ориентированного моделирования
21. Методология управление проектами
22. Анализ осуществимости внедрения системы

**Требования к выполнению заданий 1 – 16**

**Общие требования**

1. Сдаваемая программа должна быть концептуально и текстуально понятной, функционировать правильно, обладать хорошо документированным исходным текстом.
2. Система классов программы должна быть осмысленной и отвечать парадигме объектно-ориентированного программирования.
3. Программа должна быть спроектирована с соответствии архитектурой «модель-вид-контроллер» или одной из её вариаций с обязательным отделением бизнес-логики приложения (уровень модели) от логики представления (пользовательского интерфейса).
4. Настоятельно рекомендуется разрабатывать модульные тесты для всех классов приложения, для которых это представляется рациональным (как минимум, для классов уровня бизнес-логики).

5. Главный класс приложения должен инкапсулировать лишь наиболее общую логику программы. В качестве главного класса графического приложения может выступать класс главной формы, если он осуществляет лишь инициализацию и отображение этой формы. Рекомендуется использовать шаблоны проектирования во всех случаях, когда это представляется уместным.
6. Не допускается выполнять такие оптимизации программы, которые ухудшают качество её исходного текста (затрудняют восприятие кода человеком, усложняют модификацию, рефакторинг, отладку и т. д.).
7. Программа должна корректно обрабатывать все ошибочные ситуации. Сообщения об ошибках должны быть максимально информативными.

#### Требования к отдельным деталям реализации программы

##### 1. Обязательно выполнение следующих соглашений по именам:

Имена должны быть осмысленными словами (или словосочетаниями в значении соответствующей части речи) английского или (категорически не рекомендуется!) русского языка (транслитом). Для локальных переменных допускается использование сокращённых имён.

Имена классов – существительные или словосочетания в значении существительных: в нижнем регистре, первые буквы слов – в верхнем регистре, разделители слов не используются.

Имена полей и локальных переменных – существительные в нижнем регистре, первые буквы слов начиная со второго – в верхнем регистре, разделители слов не используются.

Имена методов – глаголы в нижнем регистре (либо словосочетания, отражающие действия), первые буквы слов начиная со второго – в верхнем регистре, разделители слов не используются. Для имён методов, возвращающих значения величин допускается использование существительных, а не глаголов, если только это не приведёт к неоднозначности.

Имена методов, возвращающих результат типа `boolean`, начинаются на `is` (`isOk`); выполняющих чтение/изменение значений полей класса – на `get` и `set` соответственно (`getFileAttr`, `setFileAttr`); выполняющих преобразование к другому типу данных – на `to` (`toString`);

Имена констант – существительные или словосочетания в верхнем регистре, слова разделены подчёркиваниями.

2. Все элементы программы должны иметь минимально возможную область видимости.

3. Запрещается использование статических полей и методов при наличии возможности достичь результата другими средствами.

4. Запрещается использование внутренних анонимных классов, за исключением классов-обработчиков событий. Последние должны выполнять лишь переброс вызова другому методу класса и не должны содержать других операторов.

5. Запрещается использование методов, выполняющих две или более самостоятельные операции. Каждый такой метод подлежит разбиению на более мелкие.

6. Не рекомендуется использование методов, занимающих более 15 строк. Использование методов, занимающих более 50 строк, запрещается, кроме исключительных случаев (исключительность должна быть обоснована!).

7. Запрещается собственная реализация средств, имеющих аналоги в стандартной библиотеке. Во всех случаях, когда возможно использование библиотечных классов, они должны быть использованы.

8. Запрещается посимвольная обработка строковых данных в стиле языка C. Вместо этого необходимо использовать средства стандартной библиотеки.



9. Метод, являющийся точкой входа в программу, не должен выбрасывать исключений.

#### Требования по оформлению исходного текста программы

- Исходный текст каждого класса программы должен быть размещён в отдельном файле (кроме вложенных классов).
- Программы должны быть выровнены в соответствии с одним из двух допустимых стилей: стиль Кёрнигана и Ричи или стиль Олмана. Недопустимо смешение различных стилей выравнивания в рамках одного проекта.
- Величина отступа всюду должна быть одинаковой (рекомендуется 4 символа).
- Все операторы линейной части программы должны иметь один и тот же отступ. Отступ увеличивается для объявлений вложенных классов, полей и методов классов, тел методов, субоператоров (в т. ч. для всех операторов блока, образующего субоператор).
- При использовании множественного ветвления допускается и рекомендуется размещать конструкции `else if` строго друг под другом без дополнительных отступов.
- Фигурная скобка, открывающая класс, метод или блок размещается либо на той же строке, что и описание конструкции, которую она открывают (стиль Кёрнигана и Ричи), либо на отдельной строке без дополнительного отступа (стиль Олмана).
- Закрывающая фигурная скобка размещается на отдельной строке с отступом влево относительно той конструкции которую она закрывает.
- Точка с запятой, завершающая пустой оператор, должна размещаться на отдельной строке.
- Не допускается использование строк, выходящих за пределы экрана. Не вмещающиеся части строки переносятся на следующие строки с отступом относительно предыдущей строки. Рекомендуется использовать отступ вдвое больше определённого для программы. Перенос строк, содержащих объявление метода, допускается осуществлять без отступа.
- Не рекомендуется размещение нескольких операторов в одной строке, кроме случаев, когда это не ухудшает удобочитаемости программы.
- Рекомендуется использование пробелов для отбивок отдельных лексем в программе. Способ расстановки пробелов в этом случае должен соответствовать полиграфическим правилам.
- Многострочный документационный комментарий выравнивается следующим образом: первая и последняя строки содержат только символы `/**` и `*/` (или `/**/`) соответственно. Все промежуточные строки начинаются со звёздочки, за которой следует текст комментария.

#### Требования по документированию программы

Для сдаваемой программы создаётся описание в формате HTML всех пакетов, классов, полей и методов.

Обязательными являются комментарии для всех классов, полей и методов.

Комментарий в начале каждого файла должен содержать: имя проекта и его описание (краткое или полное), полное имя класса, содержащегося в файле, описание этого класса, Ф. И. О. автора, название учебной группы.



Если смысл хотя бы одного из принимаемых или возвращаемых методом значений или выбрасываемых исключений не является интуитивно понятным, то соответствующий комментарий должен содержать описание всех указанных элементов.

В случае использования меток в операторах break и continue обязательно наличие комментария в строке, содержащей один из указанных операторов.

Комментарии не должны содержать орфографических и пунктуационных ошибок.

### **Требования к выполнению заданий 17 – 22**

Задание выполняется письменно в виде реферата.

Требования к выполнению задания:

1. Наличие авторской позиции, самостоятельность суждений.

2. Степень раскрытия сущности проблемы

- соответствие плана теме реферата;
- соответствие содержания теме и плану реферата;
- владение терминологией и понятийным аппаратом проблемы;
- полнота и глубина раскрытия основных понятий проблемы;
- обоснованность способов и методов работы с материалом;
- умение работать с литературой, систематизировать и структурировать материал;
- умение обобщать, сопоставлять различные точки зрения по рассматриваемому

вопросу, аргументировать основные положения и выводы

3. Уровень выбора источников

- обоснованность,
- полнота использования,
- современность,
- научность;

4. Соблюдение требований к оформлению, грамотность:

- правильное оформление ссылок на используемую литературу;
- соблюдение требований к объему реферата (от 10 до 15 страниц);
- культура оформления: выделение абзацев
- грамотность и культура изложения;
- отсутствие орфографических и синтаксических ошибок,
- отсутствие стилистических погрешностей;
- отсутствие опечаток, сокращений слов, кроме общепринятых;
- научно-литературный стиль.

Работа, в которой дословно переписаны текст учебника, пособия или аналогичная работа, защищенная ранее другим студентом, не оценивается, а тема заменяется на новую.

Критерии оценивания выполнения проектов

<b>Оценка</b>	<b>Критерии</b>
Отлично Уровень формирования компетенций: высокий	ПК-4: Выполнены все описанные выше требования к заданию. Знает технологические стандарты проектирования и разработки информационного контента. Выбирает оптимальные средства и инструменты разработки. Умеет работать с документацией. Анализирует поставленную задачу, использует при этом несколько источников информации: учебники, статьи, лекционные материалы.
Хорошо Уровень формирования компетенций:	ПК-4: Выполнены почти все описанные выше требования к заданию, нарушено не более одного – трех пунктов из каждого набора требований. Знает технологические стандарты проектирования и разработки информационного контента.

продвинутый	Выбирает оптимальные средства и инструменты разработки для большинства используемых алгоритмов. Умеет работать с документацией. Использует при решении задачи в основном лекционный материал, но может пользоваться другими источниками информации, возможно не полностью понимая качество получаемого решения.
Удовлетворительно Уровень формирования компетенций: пороговый	ПК-4: Выполнены описанные выше требования к заданию, нарушено не более половины пунктов из каждого набора требований. Знает технологические стандарты проектирования и разработки информационного контента. Выбирает подходящие средства и инструменты разработки, возможно не достаточно эффективные. Работает с документацией с затруднениями. Использует при решении задачи в только лекционный материал. Другие источники информации воспринимает с трудом или не пытается анализировать.
Неудовлетворительно	ПК-4: Описанные выше требования к заданию не выполнены по большинству пунктов из каждого набора требований. Не знает технологические стандарты проектирования и разработки информационного контента. Не может выбрать средства и инструменты разработки. Не умеет работать с документацией. Не умеет использовать лекционный материал, а также другие источники информации для решения поставленной задачи.

#### Список вопросов к зачету:

1. Технологии, модели и процессы создания ПО.
2. Методы создания ПО.
3. Структуры затрат на создание ПО.
4. Базовые процессы создания ПО.
5. Модели создания ПО.
6. Спецификация ПО.
7. CASE-средства автоматизации процессов создания ПО.
8. Разработка требований к ПО.
9. Формирование и анализ требований к ПО.
10. Аттестация требований. Управление требованиями.
11. Управление проектами по созданию и внедрению ПО.
12. Процессы управления. Контрольные отметки. Временные и сетевые диаграммы.
13. Планирование проекта.
14. Управление рисками.
15. Управление персоналом при реализации проектов. Решение задач.
16. Групповая работа.
17. Автоматизированные системы управления проектами.
18. Управление качеством созданных программных систем.
19. Понятие качество программного продукта.
20. Процесс управления качеством.
21. Стандарты для управления качеством.
22. Стандарты на техническую документацию.
23. Показатели качества программного продукта.
24. Отладка и тестирование ПО.
25. Верификация и контроль качества.
26. Публикация программного обеспечения.
27. Разработка и ведение проектной и технической документации.

### **Методические указания по выставлению зачета**

Зачет выставляется по результатам выполнения проекта на оценку не ниже удовлетворительно. Проект может быть сдан в течение семестра последовательно в процессе освоения материала или на зачете. В случае необходимости преподаватель в ходе сдачи проекта может провести беседу по вопросам к зачету, связанным в первую очередь с тематикой проекта.

## **2. Перечень компетенций, этапы их формирования, описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкалы оценивания**

### **2.1. Шкала оценивания сформированности компетенций и ее описание**

Оценивание уровня сформированности компетенций в процессе освоения дисциплины осуществляется по следующей трехуровневой шкале:

*Пороговый уровень* - предполагает отражение тех ожидаемых результатов, которые определяют минимальный набор знаний и (или) умений и (или) навыков, полученных студентом в результате освоения дисциплины. Пороговый уровень является обязательным уровнем для студента к моменту завершения им освоения данной дисциплины.

*Продвинутый уровень* - предполагает способность студента использовать знания, умения, навыки и (или) опыт деятельности, полученные при освоении дисциплины, для решения профессиональных задач. Продвинутый уровень превосходит пороговый уровень по нескольким существенным признакам.

*Высокий уровень* - предполагает способность студента использовать потенциал интегрированных знаний, умений, навыков и (или) опыта деятельности, полученных при освоении дисциплины, для творческого решения профессиональных задач и самостоятельного поиска новых подходов в их решении путем комбинирования и использования известных способов решения применительно к конкретным условиям. Высокий уровень превосходит пороговый уровень по всем существенным признакам.

**2.2. Перечень компетенций, этапы их формирования, описание показателей и критериев оценивания компетенций на различных этапах их формирования**

Код компетенции	Форма контроля	Этапы формирования (№ темы (раздела))	Показатели оценивания	Шкала и критерии оценивания компетенций на различных этапах их формирования		
				Пороговый уровень	Продвинутый уровень	Высокий уровень
Профессиональные компетенции						
ПК-3	Проект Зачет.	1-8	Знать: <ul style="list-style-type: none"><li>• специализированную терминологию;</li><li>• специализированное программное обеспечение для проектирования и разработки информационного контента;</li><li>• технологически е стандарты проектирования и разработки информационного контента;</li><li>• принципы построения информационных ресурсов;</li><li>• методы тестирования и отладки программного обеспечения;</li><li>• стандарты составления и оформления технической</li></ul>	обучающийся должен выполнять: 1) Сбор информации для определения потребностей клиентов осуществлен в соответствии с техникой проведения интервьюирования 2) Отладка и тестирование программного обеспечения выполнена в соответствии с техническим заданием проекта 3) Проектная и техническая документация оформлена в соответствии со стандартами 4) Измерение и контроль качества продукта выполнен в соответствии с	обучающийся должен выполнять: 1) Сбор информации для определения потребностей клиентов осуществлен в соответствии с техникой проведения интервьюирования 2) Анализ информации для определения потребностей клиентов осуществлен в соответствии с требованиями к оформлению технического задания 3) Отладка и тестирование программного обеспечения выполнена в соответствии с техническим заданием проекта 4) Проектная и техническая документация оформлена в соответствии со стандартами 5) Измерение и контроль качества продукта выполнен в соответствии с	обучающийся должен выполнять: 1) Сбор информации для определения потребностей клиентов осуществлен в соответствии с техникой проведения интервьюирования 2) Анализ информации для определения потребностей клиентов осуществлен в соответствии с требованиями к оформлению технического задания 3) Отладка и тестирование программного обеспечения выполнена в соответствии с техническим заданием проекта 4) Адаптация программного обеспечения проведена в соответствии с техническим заданием 5) Проектная и техническая документация оформлена в соответствии со стандартами 6) Измерение и контроль качества продукта выполнен в соответствии с техническим

		<p>документации;</p> <ul style="list-style-type: none"> <li>• характеристики качества программного продукта;</li> </ul> <p>Уметь</p> <ul style="list-style-type: none"> <li>• осуществлять построение информационно-логических моделей информационных ресурсов;</li> <li>• разрабатывать программное обеспечение с помощью языков программирования</li> <li>• адаптировать и конфигурировать программное обеспечение для решения поставленных задач;</li> </ul> <p>Владеть</p> <ul style="list-style-type: none"> <li>• навыками разработки и публикации программного обеспечения,</li> <li>• методами верификации и контроля качества продуктов;</li> </ul>	<p>техническим заданием</p> <p>5) Решение стандартных профессиональных задач в соответствии с поставленной задачей</p>	<p>техническим заданием</p> <p>6) Выбор метода и способа решения профессиональных задач с согласно заданной ситуации.</p> <p>7) Решение стандартных и нестандартных профессиональных задач в соответствии с поставленной задачей</p> <p>8) Демонстрация собственной деятельности в условиях коллективной и командной работы в соответствии с заданной ситуацией</p>	<p>заданием</p> <p>7) Выбор метода и способа решения профессиональных задач с согласно заданной ситуации.</p> <p>Оценка эффективности и качества выполнения согласно заданной ситуации</p> <p>8) Решение стандартных и нестандартных профессиональных задач в соответствии с поставленной задачей</p> <p>9) Информация, подобранная из разных источников в соответствии с заданной ситуацией</p> <p>10) Моделирование профессиональной деятельности с помощью прикладных программ в соответствии с заданной ситуацией</p> <p>11) Демонстрация собственной деятельности в условиях коллективной и командной работы в соответствии с заданной ситуацией</p> <p>12) Демонстрация, самоанализ и коррекция результатов собственной рабо</p>
--	--	--	--	---	--

### **3. Методические рекомендации преподавателю по процедуре оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций**

Целью процедуры оценивания является определение степени овладения студентом ожидаемыми результатами обучения (знаниями, умениями, навыками и (или) опытом деятельности).

Процедура оценивания степени овладения студентом ожидаемыми результатами обучения осуществляется с помощью методических материалов, представленных в разделе «Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций»

#### **3.1 Критерии оценивания степени овладения знаниями, умениями, навыками и (или) опытом деятельности, определяющие уровни сформированности компетенций**

Пороговый уровень (общие характеристики):

- владение основным объемом знаний по программе дисциплины;
- знание основной терминологии данной области знаний, стилистически грамотное, логически правильное изложение ответа на вопросы без существенных ошибок;
- владение инструментарием дисциплины, умение его использовать в решении стандартных (типовых) задач;
- способность самостоятельно применять типовые решения в рамках рабочей программы дисциплины;
- усвоение основной литературы, рекомендованной рабочей программой дисциплины;
- знание базовых теорий, концепций и направлений по изучаемой дисциплине;
- самостоятельная работа на практических и лабораторных занятиях, периодическое участие в групповых обсуждениях, достаточный уровень культуры исполнения заданий.

Продвинутый уровень (общие характеристики):

- достаточно полные и систематизированные знания в объёме программы дисциплины;
- использование основной терминологии данной области знаний, стилистически грамотное, логически правильное изложение ответа на вопросы, умение делать выводы;
- владение инструментарием дисциплины, умение его использовать в решении учебных и профессиональных задач;
- способность самостоятельно решать сложные задачи (проблемы) в рамках рабочей программы дисциплины;
- усвоение основной и дополнительной литературы, рекомендованной рабочей программой дисциплины;
- умение ориентироваться в базовых теориях, концепциях и направлениях по изучаемой дисциплине и давать им сравнительную оценку;
- самостоятельная работа на практических и лабораторных занятиях, участие в групповых обсуждениях, высокий уровень культуры исполнения заданий.

Высокий уровень (общие характеристики):

- систематизированные, глубокие и полные знания по всем разделам дисциплины;
- точное использование терминологии данной области знаний, стилистически грамотное, логически правильное изложение ответа на вопросы, умение делать обоснованные выводы;



- безупречное владение инструментарием дисциплины, умение его использовать в постановке и решении научных и профессиональных задач;
- способность самостоятельно и творчески решать сложные задачи (проблемы) в рамках рабочей программы дисциплины;
- полное и глубокое усвоение основной и дополнительной литературы, рекомендованной рабочей программой дисциплины;
- умение ориентироваться в основных теориях, концепциях и направлениях по изучаемой дисциплине и давать им критическую оценку;
- активная самостоятельная работа на практических и лабораторных занятиях, творческое участие в групповых обсуждениях, высокий уровень культуры исполнения заданий.

### **3.2 Описание процедуры выставления оценки**

В зависимости от уровня сформированности каждой компетенции по окончании освоения дисциплины студенту выставляется оценка «зачтено», «незачтено».

Показатели и критерии, используемые при выставлении оценки подробно описаны в разделе «Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций».

Высокий уровень формирования компетенций соответствует оценке «отлично» за проект.

Продвинутый уровень формирования компетенций соответствует оценке «хорошо» за проект.

Пороговый уровень формирования компетенций соответствует оценке «удовлетворительно» за проект.

Оценка «отлично» выставляется студенту, у которого каждая компетенция (полностью или частично формируемая данной дисциплиной) сформирована на высоком уровне.

Оценка «хорошо» выставляется студенту, у которого каждая компетенция (полностью или частично формируемая данной дисциплиной) сформирована не ниже, чем на продвинутом уровне.

Оценка «удовлетворительно» выставляется студенту, у которого каждая компетенция (полностью или частично формируемая данной дисциплиной) сформирована не ниже, чем на пороговом уровне.

Оценка «неудовлетворительно» выставляется студенту, у которого хотя бы одна компетенция (полностью или частично формируемая данной дисциплиной) сформирована ниже, чем на пороговом уровне.

Оценка «зачет» выставляется студенту, у которого каждая компетенция (полностью или частично формируемая данной дисциплиной) сформирована не ниже, чем на пороговом уровне.

Оценка «незачтено» выставляется студенту, у которого хотя бы одна компетенция (полностью или частично формируемая данной дисциплиной) сформирована ниже, чем на пороговом уровне.

## **Приложение №2 к рабочей программе дисциплины «Основы современной промышленной frontend-разработки»**

### **Методические указания для студентов по освоению дисциплины**

Занятия проводятся в интерактивной форме с использованием мультимедиа-технологий. Занятия предполагают наличие дискуссий по поводу тех или иных вопросов разработки программных приложений осуществляемых в результате соответствующего предложения преподавателя.

Практическое применение полученных знаний отрабатывается и во время лабораторных занятий, ориентированных помимо закрепления лекционного материала на разбор различных модельных ситуаций, характерных для разработки современных программных продуктов. Для успешного освоения дисциплины очень важно решение достаточно большого количества задач, как в аудитории, так и самостоятельно в качестве домашних заданий. В основном такими задачами являются лабораторные работы различного объема. Примеры решения задач разбираются на занятиях. Для решения всех задач необходимо знать и понимать лекционный материал. Поэтому в процессе изучения дисциплины рекомендуется регулярное повторение пройденного лекционного материала. Материал, законспектированный на лекциях, необходимо дома еще раз прорабатывать и при необходимости дополнять информацией, полученной на консультациях, практических занятиях или из учебной литературы.

Текущий контроль успеваемости осуществляется в форме опросов по основным понятиям и концепциям курса, осуществляемый в ходе лабораторных занятий. Для самостоятельной работы студентам предлагается разрабатывать и реализовывать алгоритмы создания компьютерных систем с применением разобранных во время лекций и лабораторных занятий подходов и методик. Окончательная аттестация осуществляется в форме зачета, основную часть которого составляют результаты лабораторных работ, а также собеседование по тематике курса.

### **Учебно-методическое обеспечение самостоятельной работы студентов по дисциплине**

Для самостоятельной работы особенно рекомендуется использовать учебную литературу, указанную в разделе № 7 данной рабочей программы.

Также для подбора учебной литературы рекомендуется использовать широкий спектр интернет-ресурсов:

1. Электронно-библиотечная система «Университетская библиотека online» ([www.biblioclub.ru](http://www.biblioclub.ru)) - электронная библиотека, обеспечивающая доступ к наиболее востребованным материалам-первоисточникам, учебной, научной и художественной литературе ведущих издательств (\*регистрация в электронной библиотеке – только в сети университета. После регистрации работа с системой возможна с любой точки доступа в Internet.).

2. Для самостоятельного подбора литературы в библиотеке ЯрГУ рекомендуется использовать:

1. Личный кабинет ([http://lib.uniyar.ac.ru/opac/bk\\_login.php](http://lib.uniyar.ac.ru/opac/bk_login.php)) дает возможность получения on-line доступа к списку выданной в автоматизированном режиме литературы, просмотра и копирования электронных версий изданий сотрудников университета (учеб. и метод. пособия, тексты лекций и т.д.) Для работы в «Личном кабинете» необходимо зайти на сайт Научной библиотеки ЯрГУ с любой точки, имеющей доступ в Internet, в пункт меню «Электронный каталог»; пройти процедуру авторизации, выбрав вкладку «Авторизация», и заполнить представленные поля информации.

2. Электронная библиотека учебных материалов ЯрГУ ([http://www.lib.uniyar.ac.ru/opac/bk\\_cat\\_find.php](http://www.lib.uniyar.ac.ru/opac/bk_cat_find.php)) содержит более 2500 полных текстов учебных и учебно-методических материалов по основным изучаемым

дисциплинам, изданных в университете. Доступ в сети университета, либо по логину/пароллю.

### 3. Электронная картотека «Книгообеспеченность»

([http://www.lib.uniyar.ac.ru/opac/bk\\_bookreq\\_find.php](http://www.lib.uniyar.ac.ru/opac/bk_bookreq_find.php)) раскрывает учебный фонд научной библиотеки ЯрГУ, предоставляет оперативную информацию о состоянии книгообеспеченности дисциплин основной и дополнительной литературой, а также цикла дисциплин и специальностей. Электронная картотека «Книгообеспеченность» доступна в сети университета и через Личный кабинет.