Автономная некоммерческая организация высшего образования
«Волгоградский институт бизнеса»

Рабочая программа учебной дисциплины

Программирование для дизайна
(Наименование дисциплины)

54.03.01 Дизайн, направленность (профиль) «Цифровой дизайн»
(Направление подготовки / Профиль)

Бакалавр
(Квалификация)

Кафедра разработчик Экономики и управления
Год набора **2026**

Вид учебной деятельности	Трудоемкость (объем) дисциплины	
	Очная форма	Очно-заочная форма
	д	в
Зачетные единицы	**4**	**4**
Общее количество часов	144	144
Аудиторные часы контактной работы обучающегося с преподавателями:	32	24
– Лекционные (Л)	16	12
– Практические (ПЗ)		
– Лабораторные (ЛЗ)	16	12
– Семинарские (СЗ)		
Самостоятельная работа обучающихся (СРО)	76	84
К (Р-Г) Р (П) (+;-)		
Тестирование (+;-)		
ДКР (+;-)		
Зачет (+;-)		
Зачет с оценкой (+;- (Кол-во часов))		
Экзамен (+;- (Кол-во часов))	+(36)	+(36)

Волгоград 2026

Содержание

Раздел 1. Организационно-методический раздел

1.1. Цели освоения дисциплины

Дисциплина **«Программирование для дизайна»** входит в **«Обязательную»** часть дисциплин подготовки обучающихся по направлению подготовки **54.03.01 Дизайн, направленность (профиль) «Цифровой дизайн».**

Целью дисциплины является формирование **компетенций** (*в соответствии с ФГОС ВО и требованиями к результатам освоения основной профессиональной образовательной программы высшего образования (ОПОП ВО)*):

ПК-4. Способен к художественно-технической разработке дизайн-проектов объектов визуальной информации, идентификации и коммуникации

ПК-4.1. Способен использовать специализированное ПО для проектирования.

ПК-4.2. Способен подготовить дизайн-проект проект с учетом современных технологий реализации.

Перечисленные компетенции формируются в процессе достижения **индикаторов компетенций:**

Обобщенная трудовая функция/ трудовая функция	**Код и наименование дескриптора компетенций**	**Код и наименование индикатора достижения компетенций (из ПС)**
ПК-4 Способен к художественно-технической разработке дизайн-проектов объектов визуальной информации, идентификации и коммуникации (ПС 11.013 Графический дизайнер код B/02.6)	ПК-4.1 Способен использовать специализированное ПО для проектирования ПК-4.2 Способен подготовить дизайн-проект проект с учетом современных технологий реализации	**Знание***:* ИД-1 ПК-4.1 Компьютерное программное обеспечение, используемое в дизайне объектов визуальной информации, идентификации и коммуникации B/02.6 ИД-2 ПК-4.2 Технологические процессы производства в области полиграфии, упаковки, кино и телевидения B/02.6 **Умения***:* ИД-3 ПК-4.1 Использовать специальные компьютерные программы для проектирования объектов визуальной информации, идентификации и коммуникации B/02.6 ИД-4 ПК-4.2 Учитывать при проектировании объектов визуальной информации, идентификации и коммуникации свойства используемых материалов и технологии реализации дизайн-проектов B/02.6 **Навыки и (или)опыт деятельности:** ИД-5 ПК-4.1 Разработка дизайн-макета объекта визуальной информации, идентификации и коммуникации B/02.6 ИД-6 ПК-4.2 Подготовка графических материалов для передачи в производство B/02.6

1.2. Место дисциплины в структуре ОПОП ВО направления подготовки 54.03.01 Дизайн, направленность (профиль) «Цифровой дизайн»

№	**Предшествующие дисциплины (дисциплины, изучаемые параллельно)**	**Последующие дисциплины**
1	*2*	*3*
1	Компьютерная графика	Разработка клиент-серверных приложений
2	Графический дизайн	Современные архитектуры нейронных сетей для цифрового дизайна

3		Анимация интерфейсов
4		Моушн-дизайн
5		Программная инженерия

Последовательность формирования компетенций в указанных дисциплинах может быть изменена в зависимости от формы и срока обучения, а также преподавания с использованием дистанционных технологий обучения.

1.3. Нормативная документация

Рабочая программа учебной дисциплины составлена на основе:

- Федерального государственного образовательного стандарта высшего образования по направлению подготовки **54.03.01 Дизайн**;
- Учебного плана направления подготовки **54.03.01 Дизайн, направленность (профиль) «Цифровой дизайн»** 2026 года набора;
- Образца рабочей программы учебной дисциплины (приказ № 113-О от 01.09.2021 г.).

Раздел 2. Тематический план

Очная форма обучения (полный срок)

№	Тема дисциплины	Трудоемкость				
		Всего	Аудиторные занятия		СРО	Код индикатора и дескриптора достижения компетенций
			Л	ПЗ (ЛЗ, СЗ)		
1	*2*	*3*	*4*	*5*	*6*	*7*
1	Роль программирования в дизайне и настройка рабочей среды	12	2	2	8	ИД-1 ПК-4.1 ИД-2 ПК-4.2
2	Основы JavaScript и работа с объектной моделью документа	12	2	2	8	ИД-3 ПК-4.1 ИД-4 ПК-4.2
3	Автоматизация монотонных задач: файловые операции и парсинг данных	12	2	2	8	ИД-5 ПК-4.1 ИД-6 ПК-4.2
4	Интеграция внешних данных: REST API, асинхронные запросы и обработка ответов	12	2	2	8	ИД-3 ПК-4.1 ИД-6 ПК-4.2
5	Логика интерактивных прототипов: машины состояний и управление интерфейсом	14	2	2	10	ИД-3 ПК-4.1 ИД-4 ПК-4.2
6	Творческое программирование и генеративная визуализация (p5.js)	14	2	2	10	ИД-5 ПК-4.1 ИД-6 ПК-4.2
7	Архитектура расширений для графических редакторов	14	2	2	10	ИД-3 ПК-4.1 ИД-6 ПК-4.2
8	Контроль качества, отладка и передача кода смежным специалистам	18	2	2	14	ИД-5 ПК-4.1 ИД-6 ПК-4.2
Вид промежуточной аттестации (Экзамен)		**36**				
Итого		**144**	**16**	**16**	**76**	

Очно-заочная форма обучения (полный срок)

№	Тема дисциплины	Трудоемкость				
		Всего	Аудиторные занятия		СРО	Код индикатора и дескриптора достижения компетенций
			Л	ПЗ (ЛЗ, СЗ)		
1	*2*	*3*	*4*	*5*	*6*	*7*
1	Роль программирования в дизайне и настройка рабочей среды	12	1	1	10	ИД-1 ПК-4.1 ИД-2 ПК-4.2
2	Основы JavaScript и работа с объектной моделью документа	12	1	1	10	ИД-3 ПК-4.1 ИД-4 ПК-4.2
3	Автоматизация монотонных задач: файловые операции и парсинг данных	12	1	1	10	ИД-5 ПК-4.1 ИД-6 ПК-4.2
4	Интеграция внешних данных: REST API, асинхронные запросы и обработка ответов	12	1	1	10	ИД-3 ПК-4.1 ИД-6 ПК-4.2
5	Логика интерактивных прототипов: машины состояний и управление интерфейсом	14	2	2	10	ИД-3 ПК-4.1 ИД-4 ПК-4.2

6	Творческое программирование и генеративная визуализация (p5.js)	14	2	2	10	ИД-5 ПК-4.1 ИД-6 ПК-4.2
7	Архитектура расширений для графических редакторов	14	2	2	10	ИД-3 ПК-4.1 ИД-6 ПК-4.2
8	Контроль качества, отладка и передача кода смежным специалистам	18	2	2	14	ИД-5 ПК-4.1 ИД-6 ПК-4.2
Вид промежуточной аттестации (Экзамен)		**36**				
Итого		**144**	**12**	**12**	**84**	

Раздел 3. Содержание дисциплины

3.1. Содержание дисциплины

Тема 1. Роль программирования в дизайне и настройка рабочей среды

Границы применения сценариев в дизайн-процессе: оценка целесообразности автоматизации, выбор между JavaScript и Python в зависимости от контекста задачи. Настройка локального окружения: установка интерпретаторов, создание виртуальных окружений, конфигурация редактора кода VS Code, подключение встроенного отладчика. Основы работы в командной строке: навигация по каталогам, установка пакетов через менеджеры npm и pip, запуск сценариев. Безопасность разработки: аудит сторонних зависимостей, управление правами доступа. Инициализация репозитория Git, создание структуры проекта и файла с инструкциями по запуску.

Тема 2. Основы JavaScript и работа с объектной моделью документа

Базовый синтаксис: переменные, типы данных, функции, стрелочные функции и обратные вызовы. Объектная модель документа (DOM): иерархия HTML-элементов, методы поиска узлов, создание и вставка элементов в дерево страницы. Обработка пользовательских действий: привязка обработчиков к событиям нажатия, ввода текста и завершения загрузки страницы, принцип делегирования событий. Динамическое управление стилями: переключение CSS-классов вместо прямого изменения свойств, использование CSS-переменных. Оптимизация производительности: предотвращение лишних перерисовок интерфейса, пакетное обновление элементов. Практика добавления интерактивности к статичной вёрстке без применения сторонних фреймворков.

Тема 3. Автоматизация монотонных задач: файловые операции и парсинг данных

Взаимодействие с файловой системой: использование стандартных модулей Node.js и Python. Пакетная обработка материалов: массовое переименование, конвертация графических форматов, фильтрация по метаданным. Чтение и разбор структурированных данных: работа с форматами JSON и CSV, сопоставление полей, автоматическая генерация отчётов. Обработка ошибок: конструкция try/catch, ведение журналов работы, обеспечение устойчивости сценария к некорректным входным данным. Практические кейсы: автогенерация спецификаций, контрольных списков, экспорт ресурсов с контролем версий. Написание сценария с отображением прогресса и сравнением времени ручной и автоматической обработки.

Тема 4. Интеграция внешних данных: REST API, асинхронные запросы и обработка ответов

Архитектура клиент-серверного обмена: точки входа, типы операций, коды статусов, заголовки запросов. Асинхронное программирование: обещания (Promises), синтаксис async/await, обработка параллельных запросов. Встроенный метод fetch: отправка запросов, преобразование ответов, различие сетевых ошибок и проблем на стороне сервера. Методы авторизации: ключи доступа, маркеры-носители, безопасное хранение учётных данных в переменных окружения. Эмуляция ответов удалённого сервера: настройка локальной заглушки json-server, имитация задержек и сбоев. Сборка прототипа с предсказуемой реакцией интерфейса на успешные данные и сетевые неполадки.

Тема 5. Логика интерактивных прототипов: машины состояний и управление интерфейсом

Концепция машины состояний для проектирования сценариев: переходы между режимами ожидания, загрузки, успеха и ошибки. Организация хранения данных без фреймворков: использование объектов и логических флагов, синхронизация состояний с CSS-классами. Валидация вводимых данных: применение регулярных выражений, вывод индивидуальных сообщений, блокировка отправки формы. Динамическое формирование списков: генерация HTML-элементов из массивов данных, обновление без полной перерисовки. Оптимизация обработки событий: ограничение частоты вызовов, ленивая инициализация тяжёлых компонентов. Проектирование компонента с полной картой переходов и тестирование всех веток логики.

Тема 6. Творческое программирование и генеративная визуализация (p5.js)

Введение в кодовое искусство: режим изоляции области видимости (instance mode), работа с холстом (canvas), цикл отрисовки, система координат и трансформаций. Визуализация метрик: привязка числовых значений к размеру, цвету, прозрачности и позиции объектов. Обработка ввода от устройств: отслеживание положения указателя и нажатий клавиш, связывание параметров скетча с CSS-переменными проекта. Оптимизация анимации: управление частотой кадров, очистка холста, синхронизация с частотой обновления экрана, создание резервного режима для слабых устройств. Интеграция библиотеки p5.js в веб-страницу: предотвращение конфликтов с DOM, экспорт результатов. Создание интерактивного фона или визуализатора, встроенного в HTML-прототип.

Тема 7. Архитектура расширений для графических редакторов

Модель безопасности и изолированная среда выполнения (sandbox): ограничения доступа к дереву слоёв, взаимодействие с пользовательским окном. Структура модуля расширения: файл конфигурации manifest.json, основной сценарий, разделение логики и интерфейса. Базовые операции с документом: создание геометрических фигур, переименование слоёв, экспорт материалов, управление автоматическим выравниванием. Ограничения производительности: пакетная обработка операций (batching), лимиты асинхронных вызовов, обработка файлов с большим количеством элементов. Публикация и отладка: локальный режим разработки, ведение журналов, контроль прав доступа. Разработка базовой версии расширения с простым интерфейсом и инструкцией по установке.

Тема 8. Контроль качества, отладка и передача кода смежным специалистам

Автоматическая проверка стиля: настройка линтеров ESLint и форматтеров Prettier, интеграция в процесс сохранения файлов. Методы пошаговой отладки: установка точек останова, анализ сетевых запросов и потребления ресурсов, поиск утечек памяти в панелях разработчика. Документирование проекта: описание назначения функций, составление руководства по запуску, фиксация ограничений и зависимостей. Управление версиями: работа с ветками, создание меток релизов, ведение журнала изменений. Культура передачи материалов: подготовка чек-листа, оформление примеров использования, совместная поддержка кода. Рефакторинг студенческих работ, перекрёстная проверка и подготовка к защите итогового артефакта.

3.2. Содержание практического блока дисциплины

Очная форма обучения (полный срок)

№	Тема практического (семинарского, лабораторного) занятия
1	***2***
ПЗ 1	Роль программирования в дизайне и настройка рабочей среды
ПЗ 2	Основы JavaScript и работа с объектной моделью документа
ПЗ 3	Автоматизация монотонных задач: файловые операции и парсинг данных
ПЗ 4	Интеграция внешних данных: REST API, асинхронные запросы и обработка ответов
ПЗ 5	Логика интерактивных прототипов: машины состояний и управление интерфейсом
ПЗ 6	Творческое программирование и генеративная визуализация (p5.js)
ПЗ 7	Архитектура расширений для графических редакторов
ПЗ 8	Контроль качества, отладка и передача кода смежным специалистам

Очно-заочная форма обучения (полный срок)

№	Тема практического (семинарского, лабораторного) занятия
1	***2***
ПЗ 1	Роль программирования в дизайне и настройка рабочей среды. Основы JavaScript и работа с объектной моделью документа
ПЗ 2	Автоматизация монотонных задач: файловые операции и парсинг данных. Интеграция внешних данных: REST API, асинхронные запросы и обработка ответов
ПЗ 3	Логика интерактивных прототипов: машины состояний и управление интерфейсом
ПЗ 4	Творческое программирование и генеративная визуализация (p5.js)
ПЗ 5	Архитектура расширений для графических редакторов
ПЗ 6	Контроль качества, отладка и передача кода смежным специалистам

3.3. Образовательные технологии

Очная форма обучения (полный срок)

№	Тема занятия	Вид учебного занятия	Форма / Методы интерактивного обучения	% учебного времени
1	*2*	*3*	*4*	*5*
1.	Роль программирования в дизайне и настройка рабочей среды	ПЗ	Кейс-метод	100
2.	Основы JavaScript и работа с объектной моделью документа	ПЗ	Кейс-метод	100
3.	Автоматизация монотонных задач: файловые операции и парсинг данных	ПЗ	Кейс-метод	100
4.	Интеграция внешних данных: REST API, асинхронные запросы и обработка ответов	ПЗ	Кейс-метод	100
5.	Логика интерактивных прототипов: машины состояний и управление интерфейсом	ПЗ	Кейс-метод	80
6.	Творческое программирование и генеративная визуализация (p5.js)	ПЗ	Кейс-метод	100
7.	Архитектура расширений для	ПЗ	Кейс-метод	100

	графических редакторов			
8.	Контроль качества, отладка и передача кода смежным специалистам	ПЗ	Кейс-метод	80
Итого %				**30%**

Очно-заочная форма обучения (полный срок)

№	Тема занятия	Вид учебного занятия	Форма / Методы интерактивного обучения	% учебного времени
1	*2*	*3*	*4*	*5*
1.	Роль программирования в дизайне и настройка рабочей среды. Основы JavaScript и работа с объектной моделью документа	ПЗ	Кейс-метод	100
2.	Автоматизация монотонных задач: файловые операции и парсинг данных. Интеграция внешних данных: REST API, асинхронные запросы и обработка ответов	ПЗ	Кейс-метод	100
3.	Логика интерактивных прототипов: машины состояний и управление интерфейсом	ПЗ	Кейс-метод	100
4.	Творческое программирование и генеративная визуализация (p5.js)	ПЗ	Кейс-метод	100
5.	Архитектура расширений для графических редакторов	ПЗ	Кейс-метод	80
6.	Контроль качества, отладка и передача кода смежным специалистам	ПЗ	Кейс-метод	100
Итого %				**30%**

Раздел 4. Организация самостоятельной работы обучающихся

4.1. Организация самостоятельной работы обучающихся

№	Тема дисциплины	№ вопросов	№ рекомендуемой литературы
1	*2*	*3*	*4*
1	Роль программирования в дизайне и настройка рабочей среды	1-3	1-11
2	Основы JavaScript и работа с объектной моделью документа	4-6	1-11
3	Автоматизация монотонных задач: файловые операции и парсинг данных	7-9	1-11
4	Интеграция внешних данных: REST API, асинхронные запросы и обработка ответов	10-12	1-11
5	Логика интерактивных прототипов: машины состояний и управление интерфейсом	13-15	1-11
6	Творческое программирование и генеративная визуализация (p5.js)	16-18	1-11
7	Архитектура расширений для графических редакторов	19-21	1-11
8	Контроль качества, отладка и передача кода смежным специалистам	22-24	1-11

Перечень вопросов, выносимых на самостоятельную работу обучающихся

1. Каковы критерии выбора скриптового языка для автоматизации дизайн-задач?
2. В чём различие между интерпретируемым и компилируемым выполнением кода?
3. Какие принципы безопасности необходимо соблюдать при подключении внешних зависимостей?
4. Что такое объектная модель документа и как она представляет структуру веб-страницы?
5. В чём разница между всплытием и погружением событий в браузерной среде?
6. Почему прямое изменение inline-стилей считается антипаттерном при работе с интерфейсом?
7. Какие преимущества предоставляет пакетная обработка файлов по сравнению с ручной?
8. Как обеспечивается устойчивость сценария при обработке некорректных входных данных?
9. В чём различие между синхронным и асинхронным чтением файлов?
10. Что определяет выбор метода запроса при взаимодействии с сервером?
11. Как различаются ошибки сетевого уровня и ошибки на стороне сервера?
12. Зачем необходимо эмулировать ответы сервера при разработке прототипов?
13. Какие состояния выделяются в машине состояний пользовательского интерфейса?
14. Как обеспечивается предсказуемость переходов между состояниями компонента?
15. В чём назначение ограничения частоты обработки событий ввода?
16. Как работает цикл отрисовки в библиотеке для творческого программирования?
17. Какие способы привязки данных к визуальным параметрам существуют?
18. Как оптимизируется анимация при ограниченной производительности устройства?
19. Зачем расширениям дизайн-инструментов требуется изолированная среда выполнения?
20. Какие операции с документом доступны через прикладной интерфейс редактора?
21. Как влияют ограничения производительности на архитектуру плагина?
22. Какие задачи решает автоматическая проверка стиля кода?
23. В чём преимущество пошаговой отладки перед выводом диагностических сообщений?
24. Какие элементы должны входить в документацию передаваемого сценария?

4.2. Перечень учебно-методического обеспечения самостоятельной работы обучающихся

Самостоятельная работа обучающихся обеспечивается следующими учебно-методическими материалами:

1. Указаниями в рабочей программе по дисциплине (п.4.1.)
2. Лекционные материалы в составе учебно-методического комплекса по дисциплине
3. Заданиями и методическими рекомендациями по организации самостоятельной работы обучающихся в составе учебно-методического комплекса по дисциплине.
4. Глоссарием по дисциплине в составе учебно-методического комплекса по дисциплине.

Раздел 5. Фонд оценочных средств для проведения текущего контроля и промежуточной аттестации обучающихся

Фонд оценочных средств по дисциплине представляет собой совокупность контролирующих материалов, предназначенных для измерения уровня достижения обучающимися установленных результатов образовательной программы. ФОС по дисциплине используется при проведении оперативного контроля и промежуточной аттестации обучающихся. Требования к структуре и содержанию ФОС дисциплины регламентируются Положением о фонде оценочных материалов по программам высшего образования – программам бакалавриата, магистратуры.

5.1. Паспорт фонда оценочных средств
Очная форма обучения (полный срок)

№	Контролируемые разделы (темы) дисциплины	Оценочные средства			
		Л	ПЗ (ЛЗ, СЗ)	СРО	Код индикатора и дескриптора достижения компетенций
1	*2*	*3*	*4*	*5*	*6*
1	Роль программирования в дизайне и настройка рабочей среды	ЛС	КМ	ПРВ	ИД-1 ПК-4.1 ИД-2 ПК-4.2
2	Основы JavaScript и работа с объектной моделью документа	ЛС	КМ	ПРВ	ИД-3 ПК-4.1 ИД-4 ПК-4.2
3	Автоматизация монотонных задач: файловые операции и парсинг данных	ЛС	КМ	ПРВ	ИД-5 ПК-4.1 ИД-6 ПК-4.2
4	Интеграция внешних данных: REST API, асинхронные запросы и обработка ответов	ЛС	КМ	ПРВ	ИД-3 ПК-4.1 ИД-6 ПК-4.2
5	Логика интерактивных прототипов: машины состояний и управление интерфейсом	ЛС	КМ	ПРВ	ИД-3 ПК-4.1 ИД-4 ПК-4.2
6	Творческое программирование и генеративная визуализация (p5.js)	ЛС	КМ	ПРВ	ИД-5 ПК-4.1 ИД-6 ПК-4.2
7	Архитектура расширений для графических редакторов	ЛС	УО	ПРВ	ИД-3 ПК-4.1 ИД-6 ПК-4.2
8	Контроль качества, отладка и передача кода смежным специалистам	ЛС, УО	УО	ПРВ	ИД-5 ПК-4.1 ИД-6 ПК-4.2

Очно-заочная форма обучения (полный срок)

№	Контролируемые разделы (темы) дисциплины	Оценочные средства			
		Л	ПЗ (ЛЗ, СЗ)	СРО	Код индикатора и дескриптора достижения компетенций
1	*2*	*3*	*4*	*5*	*6*
1	Роль программирования в дизайне и настройка рабочей среды. Основы JavaScript и работа с объектной моделью документа	ЛС	КМ	ПРВ	ИД-1 ПК-4.1 ИД-2 ПК-4.2
2	Автоматизация монотонных задач: файловые операции и парсинг данных. Интеграция внешних данных: REST API, асинхронные запросы и обработка ответов	ЛС	КМ	ПРВ	ИД-3 ПК-4.1 ИД-4 ПК-4.2
3	Логика интерактивных прототипов:	ЛС	КМ	ПРВ	ИД-5 ПК-4.1

	машины состояний и управление интерфейсом				ИД-6 ПК-4.2
4	Творческое программирование и генеративная визуализация (p5.js)	ЛС	УО	ПРВ	ИД-3 ПК-4.1 ИД-6 ПК-4.2
5	Архитектура расширений для графических редакторов	ЛС	КМ	ПРВ	ИД-3 ПК-4.1 ИД-4 ПК-4.2
6	Контроль качества, отладка и передача кода смежным специалистам	ЛС, УО	УО, ЗЗ	ПРВ, ЗЗ	ИД-5 ПК-4.1 ИД-6 ПК-4.2

Условные обозначения оценочных средств (Столбцы 3, 4, 5):
ЗЗ – Защита выполненных заданий (творческих, расчетных и т.д.), представление презентаций;
Т – Тестирование компьютерное;
УО – Устный (фронтальный, индивидуальный, комбинированный) опрос;
КР – Контрольная работа (аудиторные или домашние, индивидуальные, парные или групповые контрольные, самостоятельные работы, диктанты и т.д.);
К – Коллоквиум;
ПРВ – Проверка рефератов, отчетов, рецензий, аннотаций, конспектов, графического материала, эссе, переводов, решений заданий, выполненных заданий в электронном виде и т.д.;
ДИ – Деловая игра;
РИ – Ролевая игра;
КМ – Кейс-метод;
КС – Круглый стол;
КСМ – Компьютерная симуляция;
МШ – Метод мозгового штурма;
ЛС – Лекция-ситуация;
ЛК – Лекция-конференция;
ЛВ – Лекция-визуализация;
ПЛ – Проблемная лекция;
Д – Дискуссия, полемика, диспут, дебаты;
П – Портфолио;
ПВУ – Просмотр видеоуроков;
МП – Метод проектов.

5.2. Оценочные средства текущего контроля

Перечень практических (семинарских) заданий

Тема 1. Роль программирования в дизайне и настройка рабочей среды
Цель практики и её задачи: Освоить базовую настройку локального окружения, работу в командной строке и систему контроля версий. Научиться запускать простые сценарии, управлять зависимостями и документировать проект.
Входные данные:
• Шаблон файла brief.json с данными учебного проекта
• Инструкция по установке Node.js / Python и настройке редактора
• Пустой Git-репозиторий курса
Инструменты: VS Code, Terminal/CLI, Node.js или Python 3.x, Git
Пошаговое задание:

1. Инициализировать Git-репозиторий, настроить файл .gitignore под проект
2. Скопировать brief.json, проверить валидность структуры данных
3. Написать скрипт summarize.js или summarize.py, читающий JSON и выводящий структурированную сводку в консоль
4. Добавить обработку ошибок: отсутствие файла, повреждённый JSON, неверные типы данных
5. Закоммитить изменения, создать ветку develop, написать README.md с инструкцией по установке зависимостей и запуску

Итоговый результат:
.zip архив или ссылка на репозиторий с рабочим скриптом, скриншоты вывода в консоли, файл README.md / Фамилия_ПР1_СкриптСводка
.txt черновик заметок по выбору менеджера пакетов и настройке переменных окружения / Фамилия_ПР1_Заметки_Среда
Критерии оценки (чек-лист):
• [ ] Скрипт корректно читает JSON и обрабатывает типовые ошибки ввода
• [ ] Репозиторий содержит .gitignore, структура каталогов логична
• [ ] README.md включает команды запуска, описание зависимостей и требования к версии интерпретатора
• [ ] Код содержит комментарии, поясняющие основные этапы выполнения

Тема 2. Основы JavaScript и работа с объектной моделью документа
Цель практики и её задачи: Научиться динамически управлять структурой HTML-документа через DOM API, обрабатывать пользовательские события и управлять стилями без inline-переопределений.
Входные данные:
• Статичная HTML/CSS вёрстка (материал параллельной дисциплины)
• Файл-заготовка script.js с базовой структурой
• Браузер с включёнными инструментами разработчика
Инструменты: VS Code, Live Server, DevTools браузера
Пошаговое задание:

1. Подключить скрипт к странице, проверить доступ к DOM через console.log
2. Реализовать переключение вкладок или модальных окон через classList и делегирование событий
3. Добавить валидацию формы (email, телефон) с выводом индивидуальных сообщений об ошибках
4. Заменить прямое изменение style на переключение CSS-классов и работу с CSS Custom Properties

5. Протестировать производительность через DevTools (вкладка Performance, отслеживание reflow/repaint)

Итоговый результат:
.html + .css + .js файлы с работающим интерактивом / Фамилия_ПР2_Интерактив
.png скриншот вкладки Console с логами событий и подтверждением отсутствия предупреждений / Фамилия_ПР2_ЛогиСобытий
Критерии оценки (чек-лист):
• [ ] Отсутствуют inline-стили, управление оформлением вынесено в CSS-классы
• [ ] Обработчики событий привязаны через делегирование или корректно удаляются
• [ ] Валидация блокирует отправку формы при некорректном вводе
• [ ] Код не вызывает лишних перерисовок интерфейса

Тема 3. Автоматизация монотонных задач: файловые операции и парсинг данных
Цель практики и её задачи: Научиться автоматизировать пакетную обработку файлов, парсить структурированные данные и строить устойчивые сценарии с логированием.
Входные данные:
• Папка с 20+ изображениями/файлами разного формата и имён
• Шаблон отчёта report_template.csv
• Тестовые данные с некорректными расширениями и дубликатами
Инструменты: Node.js (модули fs, path, sharp) или Python (os, shutil, Pillow), Terminal
Пошаговое задание:

1. Написать скрипт для переименования файлов по шаблону project_001.webp
2. Реализовать конвертацию в заданный формат и удаление дубликатов по хэш-сумме
3. Добавить прогресс-лог в консоль и генерацию итогового CSV-отчёта с метаданными
4. Обработать исключения: неподдерживаемый формат, отсутствие прав записи, повреждённые файлы
5. Зафиксировать время ручной и автоматической обработки, рассчитать экономию ресурсов

Итоговый результат:
Скрипт batch-process.js/.py, файл журнала log.txt,
отчёт report.csv / Фамилия_ПР3_ПакетнаяОбработка
.md документ с расчётом времени, сравнением подходов и описанием ограничений / Фамилия_ПР3_ОтчётВремя
Критерии оценки (чек-лист):
• [ ] Скрипт корректно обрабатывает папку с неструктурированными данными
• [ ] Дубликаты определяются и удаляются без потери оригиналов
• [ ] Логирование включает этапы, ошибки и итоговую статистику
• [ ] Документ содержит измеримые показатели эффективности

Тема 4. Интеграция внешних данных: REST API, асинхронные запросы и обработка ответов
Цель практики и её задачи: Освоить безопасную работу с сетевыми запросами, асинхронный поток выполнения и эмуляцию серверных ответов для прототипирования.
Входные данные:
• Прототип из ПР2 (статичная вёрстка)
• URL открытого API (погода, курсы валют, открытые данные РФ)
• Локальная заглушка json-server с тестовыми данными
Инструменты: fetch API, async/await, json-server, DevTools Network
Пошаговое задание:

1. Настроить fetch-запрос с async/await, вынести конфигурацию в отдельный модуль

2. Реализовать ветвление обработки: успешный ответ, ошибка сервера (4xx/5xx), сетевая неполадка
3. Отобразить данные в DOM через динамический рендеринг карточек/списков
4. Визуализировать состояния интерфейса: загрузка, успех, ошибка, кнопка повтора
5. Замокать ответы через json-server, протестировать сценарии с задержкой и обрывом соединения

Итоговый результат:
Файл api-integration.js, работающая страница с динамическими данными
/ Фамилия_ПР4_APIПрототип
.md схема обработки ошибок и таблица статус-кодов с действиями интерфейса
/ Фамилия_ПР4_КартаОшибок
Критерии оценки (чек-лист):
• [ ] Запрос не блокирует основной поток выполнения, используется async/await
• [ ] Интерфейс предсказуемо реагирует на все типы сетевых ответов
• [ ] Ключи доступа не хранятся в открытом виде в клиентском коде
• [ ] Реализован режим работы с локальными моками

Тема 5. Логика интерактивных прототипов: машины состояний и управление интерфейсом
Цель практики и её задачи: Спроектировать сложную логику компонента без фреймворков, используя паттерн конечного автомата, и обеспечить предсказуемость переходов.
Входные данные:
• Прототип из ПР2/4 с формой или загрузчиком
• Описание компонента и требования к пользовательскому сценарию
• Библиотека Mermaid.js для визуализации
Инструменты: Vanilla JS, Mermaid.js, DevTools
Пошаговое задание:

1. Определить состояния компонента: idle, loading, success, error, disabled
2. Написать JS-объект-машину с таблицей переходов, условиями и валидацией
3. Привязать состояния к CSS-классам и DOM-элементам через единый метод обновления
4. Реализовать ограничение частоты действий (debounce/throttle) и блокировку повторных отправок
5. Построить диаграмму переходов в Mermaid.js, протестировать все ветки логики

Итоговый результат:
state-machine.js, диаграмма diagram.md, работающий компонент
/ Фамилия_ПР5_МашинаСостояний
.txt лог тестирования с описанием покрытых сценариев и найденных граничных случаев
/ Фамилия_ПР5_ТестЛог
Критерии оценки (чек-лист):
• [ ] Все состояния описаны явно, переходы не допускают «зависаний»
• [ ] Логика вынесена из обработчиков событий в отдельный объект-машину
• [ ] Диаграмма в Mermaid.js соответствует реализации в коде
• [ ] Реализована защита от дублирующих действий пользователя

Тема 6. Творческое программирование и генеративная визуализация (p5.js)
Цель практики и её задачи: Расширить выразительность интерфейса через программируемую графику, освоить цикл отрисовки, оптимизацию анимации и интеграцию холста в веб-документ.
Входные данные:
• Верстка из ПР2 с контейнером для графики
• Документация p5.js (режим instance)

• Набор метрик или массив данных для визуализации
Инструменты: Библиотека p5.js, DevTools Performance, VS Code
Пошаговое задание:

1. Подключить p5.js в режиме instance, инициализировать canvas внутри заданного контейнера
2. Создать скетч, реагирующий на положение указателя или передаваемые данные
3. Связать параметры визуализации (цвет, размер, прозрачность) с CSS-переменными проекта
4. Оптимизировать frameRate, добавить очистку холста и fallback для устройств с низкой производительностью
5. Встроить скетч в страницу, проверить отсутствие конфликтов с DOM и утечек памяти

Итоговый результат:
sketch.js, интегрированная HTML-страница, отчёт о производительности / Фамилия_ПР6_GenArt .png скриншоты DevTools Performance с графиками нагрузки до и после оптимизации / Фамилия_ПР6_PerfОтчёт
Критерии оценки (чек-лист):
• [ ] Скетч изолирован от глобальной области видимости, не конфликтует с другими скриптами
• [ ] Частота кадров стабильна, реализована очистка ресурсов
• [ ] Параметры визуализации синхронизированы с дизайн-системой проекта
• [ ] Предусмотрен корректный деградация для слабых устройств

Тема 7. Архитектура расширений для графических редакторов
Цель практики и её задачи: Понять модель безопасности дизайн-инструментов, освоить структуру плагина и реализовать базовую операцию с документом через прикладной интерфейс.
Входные данные:
• Шаблон плагина (manifest.json, code.js, ui.html)
• Тестовый макет в Penpot или Figma
• Документация API выбранной платформы
Инструменты: Penpot API или Figma Plugin API, Dev-режим редактора, текстовый редактор
Пошаговое задание:

1. Настроить локальный дев-режим, проверить корректность загрузки manifest.json
2. Реализовать функцию: авто-переименование слоёв по шаблону или генерация палитры цветов
3. Добавить простое UI-окно (iframe) для ввода параметров, связать его с code.js через postMessage
4. Обработать ограничения песочницы: асинхронность, лимиты операций, права доступа к файлу
5. Протестировать на макете, зафиксировать ограничения, логи ошибок и время выполнения

Итоговый результат:
Архив плагина, файл code.js, инструкция по установке / Фамилия_ПР7_Плагин .md документ с перечнем ограничений API, логом выполнения и рекомендациями по масштабированию / Фамилия_ПР7_Ограничения
Критерии оценки (чек-лист):
• [ ] Плагин успешно загружается в редакторе и не вызывает ошибок в консоли
• [ ] Операции с деревом слоёв выполняются асинхронно, не блокируя интерфейс
• [ ] UI-окно корректно передаёт данные в основной сценарий
• [ ] Задокументированы лимиты и меры предосторожности при работе с API

Тема 8. Контроль качества, отладка и передача кода смежным специалистам
Цель практики и её задачи: Подготовить код к передаче, провести автоматическую проверку стиля, отладку и документирование, освоить культуру handoff и перекрёстную проверку.
Входные данные:
• Все скрипты и прототипы ПР1–7, собранные в одном репозитории
• Конфигурационные файлы ESLint, Prettier, правила JSDoc
• Шаблон чек-листа передачи материалов
Инструменты: ESLint, Prettier, DevTools (Sources, Performance, Memory), Git
Пошаговое задание:

1. Запустить линтер и форматтер, устранить все предупреждения и ошибки стиля
2. Добавить JSDoc-комментарии к основным функциям, обновить README.md с инструкциями по запуску
3. Установить точки останова, протестировать граничные сценарии, исправить утечки памяти
4. Подготовить 3-минутное демо: показать архитектуру, ключевые функции, обработку ошибок
5. Провести перекрёстную проверку (peer-review), зафиксировать правки и обновить changelog.md

Итоговый результат:
Чистый Git-репозиторий с документацией, скринкаст демо, отчёт peer-review / Фамилия_ПР8_FinalRepo
.md чек-лист handoff с подтверждением прохождения всех пунктов / Фамилия_ПР8_Checklist
Критерии оценки (чек-лист):
• [ ] Код проходит автоматическую проверку без критических предупреждений
• [ ] Документация покрывает установку, запуск, ограничения и примеры использования
• [ ] Демо демонстрирует работоспособность всех ключевых модулей
• [ ] Учтены и применены замечания по результатам peer-review

5.4. Перечень вопросов промежуточной аттестации по дисциплине

Вопросы к экзамену:

1. Критерии выбора языка сценариев для автоматизации дизайн-задач.
2. Различия между интерпретируемым и компилируемым выполнением кода.
3. Принципы безопасного управления зависимостями и аудит внешних библиотек.
4. Назначение системы контроля версий Git в рабочем процессе.
5. Роль командной строки в настройке окружения и запуске сценариев.
6. Структура и назначение объектной модели документа (DOM) в браузере.
7. Принцип делегирования событий и его влияние на производительность.
8. Различия между всплытием и погружением событий в иерархии элементов.
9. Почему прямое изменение inline-стилей считается антипаттерном?
10. Понятие reflow и repaint, методы их минимизации при динамическом обновлении страницы.
11. Архитектурные различия между синхронным и асинхронным выполнением файловых операций.
12. Принципы обеспечения устойчивости сценария при обработке некорректных данных.
13. Методы вычисления хэш-сумм для выявления дубликатов ассетов.
14. Структура и назначение форматов JSON и CSV в обмене данными.
15. Роль логирования в отслеживании выполнения пакетных сценариев.
16. Базовые принципы архитектуры REST и назначение HTTP-методов.
17. Различия между сетевыми ошибками и ошибками ответа сервера.
18. Механизм работы обещаний (Promises) и преимущества синтаксиса async/await.
19. Принципы безопасного хранения ключей авторизации на стороне клиента.
20. Назначение мокирования ответов сервера при разработке прототипов.
21. Концепция конечного автомата при проектировании логики интерфейса.
22. Методы изоляции состояний от обработчиков пользовательских событий.
23. Принцип применения регулярных выражений при валидации ввода.
24. Различия между паттернами debounce и throttle, сферы их применения.
25. Механизм динамического обновления списков без полной перерисовки DOM.
26. Назначение и принцип работы цикла отрисовки в библиотеке p5.js.
27. Методы привязки числовых метрик к визуальным параметрам объектов.
28. Принцип изоляции области видимости при интеграции графических скетчей в веб-страницу.
29. Способы оптимизации анимации при ограниченной частоте кадров устройства.
30. Различия между растровым холстом (canvas) и векторной графикой (SVG) в браузере.
31. Назначение изолированной среды выполнения (sandbox) для расширений редакторов.
32. Структура конфигурационного файла manifest.json и его роль в загрузке плагина.
33. Механизм обмена данными между основным сценарием и пользовательским окном.
34. Архитектурные ограничения при работе с деревом слоёв через прикладной интерфейс.
35. Принципы пакетной обработки операций для обхода лимитов производительности.
36. Различия между статическим анализом кода (линтинг) и автоматическим форматированием.
37. Назначение точек останова в процессе пошаговой отладки сценариев.
38. Стандарт документирования функций и его роль в передаче материалов.
39. Принципы культуры передачи кода от дизайнера к разработчику.
40. Назначение ветвления и тегирования в системе контроля версий при подготовке релиза.

Раздел 6. Оценочные средства промежуточной аттестации (с ключами)

1. Укажите один правильный ответ. Почему прямое изменение style-свойств через JavaScript считается антипаттерном?
а) Это нарушает семантику HTML-разметки
б) Это вызывает лишние перерисовки (reflow/repaint) и усложняет поддержку стилей
в) Браузеры блокируют выполнение таких скриптов из-за политики безопасности
г) Этот метод не работает в современных версиях Chrome и Firefox
Правильный ответ: б)

2. Установите соответствие между кодами состояния HTTP и их значением:

Код	**Значение**
А) 200	1) Запрос требует авторизации или доступ запрещён
Б) 404	2) Успешная обработка запроса
В) 500	3) Ресурс не найден на сервере
Г) 401	4) Внутренняя ошибка сервера

Правильный ответ: А-2, Б-3, В-4, Г-1

3. Запишите термин (с большой буквы, в именительном падеже), обозначающий шаблон проектирования, при котором поведение компонента определяется его текущим состоянием, а переходы между режимами явно описаны. _____ — инструмент предсказуемого управления логикой интерфейса. **Правильный ответ: Машина состояний**

4. Выберите два правильных ответа. Какие два преимущества предоставляет конструкция async/await по сравнению с цепочками .then()?
а) Синхронное выполнение кода без блокировки основного потока
б) Линейная читаемость, похожая на синхронный код
в) Встроенная обработка всех типов ошибок без дополнительных блоков
г) Упрощённая работа с вложенными асинхронными операциями
Правильный ответ: б) г)

5. Укажите один правильный ответ. Какую роль выполняет функция draw() в библиотеке p5.js?
а) Инициализирует холст и загружает внешние ресурсы один раз
б) Выполняется непрерывно с заданной частотой кадров, обеспечивая анимацию
в) Обрабатывает пользовательский ввод и передаёт данные в DOM
г) Конвертирует векторную графику в растровый формат
Правильный ответ: б)

6. Установите соответствие между инструментами контроля качества и их назначением:

Инструмент	**Назначение**
А) ESLint	1) Автоматическое выравнивание отступов и форматирование кода
Б) Prettier	2) Поиск синтаксических ошибок и соблюдение правил написания
В) JSDoc	3) Генерация документации из комментариев в коде
Г) Git tag	4) Фиксация стабильной версии проекта в репозитории

Правильный ответ: А-2, Б-1, В-3, Г-4

7. Запишите термин (с маленькой буквы), обозначающий механизм, при котором обработчик события привязывается к родительскому элементу вместо отдельных дочерних узлов.
_____ событий — способ оптимизации обработки кликов в длинном списке.
Правильный ответ: делегирование

8. Разместите по порядку этапы безопасного выполнения сетевого запроса к внешнему API:
1. Парсинг ответа в формат JSON
2. Проверка статуса ответа (response.ok)
3. Отправка запроса через fetch с указанием заголовков
4. Обработка ошибок в блоке catch или ветке else

Правильный ответ: 3, 2, 1, 4

9. Укажите один правильный ответ. Для чего расширениям дизайн-инструментов требуется изолированная среда выполнения (sandbox)?
а) Для повышения скорости рендеринга графических примитивов
б) Для ограничения доступа плагина к системным файлам и предотвращения конфликтов
в) Для автоматической публикации плагина в официальном магазине
г) Для конвертации кода JavaScript в нативный машинный язык
Правильный ответ: б)

10. Выберите два правильных ответа. Какие два действия входят в понятие «graceful degradation» (устойчивое выполнение) сценария?
а) Полная остановка скрипта при любой ошибке ввода
б) Возврат значений по умолчанию при отсутствии данных
в) Логирование проблемы без прерывания основного потока обработки
г) Автоматическое удаление всех файлов при некорректном формате
Правильный ответ: б) в)

11. Разместите по порядку этапы подготовки репозитория к передаче смежным специалистам:
1. Запуск линтера и форматтера, устранение предупреждений
2. Создание тега релиза и фиксация изменений в changelog.md
3. Написание инструкций по запуску в README.md
4. Добавление JSDoc-комментариев к ключевым функциям

Правильный ответ: 1, 4, 3, 2

12. Укажите один правильный ответ. Какова основная цель использования локальной заглушки (json-server) при разработке прототипа?
а) Замена реального бэкенда для ускорения загрузки сайта в продакшене
б) Имитация ответов сервера для автономной проверки логики интерфейса
в) Шифрование передаваемых данных перед отправкой на сервер
г) Автоматическая генерация ключей авторизации для API
Правильный ответ: б)

13. Запишите термин (с маленькой буквы), обозначающий метод обмена сообщениями между изолированным интерфейсом плагина и основным сценарием в дизайн-редакторе.
_____ — механизм связи iframe с родительским окном.
Правильный ответ: postMessage

14. Выберите два правильных ответа. Какие два приёма помогают оптимизировать производительность анимации на canvas?
а) Бесконечное увеличение разрешения холста
б) Ограничение frameRate до оптимального значения
в) Использование requestAnimationFrame для синхронизации с экраном
г) Полное отключение очистки холста для накопления кадров

Правильный ответ: б) в)

15. Установите соответствие между методом управления стилями и его характеристикой:

Метод	**Характеристика**
А) element.style.property	1) Прямое переопределение, высокий приоритет, сложно поддерживать
Б) element.classList.add()	2) Переключение правил из таблицы стилей, рекомендуемый подход
В) CSS Custom Properties (--var)	3) Динамическая передача значений из JS в CSS без перезаписи классов
Г) getComputedStyle()	4) Чтение итоговых вычисленных значений, заданных таблицей стилей

Правильный ответ: А-1, Б-2, В-3, Г-4

16. Разместите по порядку этапы обработки пользовательского ввода с применением валидации:

1. Проверка соответствия регулярному выражению
2. Ввод данных пользователем в поле формы
3. Вывод сообщения об ошибке или блокировка кнопки
4. Разблокировка отправки и переход к следующему шагу

Правильный ответ: 2, 1, 3, 4

17. Укажите один правильный ответ. Какой файл используется для исключения определённых папок и файлов из отслеживания системой контроля версий Git?
а) .gitconfig
б) .gitignore
в) package.json
г) README.md
Правильный ответ: б)

18. Выберите два правильных ответа. Какие два ограничения характерны для API дизайн-инструментов при работе с большими файлами?
а) Блокировка асинхронных операций
б) Лимит на количество одновременных изменений дерева слоёв
в) Требование пакетной обработки (batching) для повышения производительности
г) Автоматическое увеличение оперативной памяти редактора
Правильный ответ: б) в)

19. Установите соответствие между функцией библиотеки p5.js и её назначением:

Функция	**Назначение**
А) setup()	1) Выполняется один раз при запуске для начальной настройки
Б) draw()	2) Выполняется в цикле для отрисовки кадров
В) mouseX / mouseY	3) Переменные, хранящие текущие координаты указателя
Г) createCanvas()	4) Инициализация области рисования заданного размера

Правильный ответ: А-1, Б-2, В-3, Г-4

20. Укажите один правильный ответ. Для чего применяется приём debounce при проверке ввода в поисковой строке?
а) Для мгновенной отправки запроса после каждого нажатия клавиши

б) Для задержки выполнения функции до прекращения ввода на заданное время
в) Для блокировки всех событий ввода до перезагрузки страницы
г) Для автоматического исправления орфографических ошибок
Правильный ответ: б)

21. Запишите термин (с маленькой буквы), обозначающий текстовый формат обмена данными, основанный на паре «ключ: значение» и часто используемый для конфигураций и API-ответов.
_____ — стандарт представления структурированной информации в веб-разработке. **Правильный ответ: json**

22. Разместите по порядку действия разработчика при поиске причины ошибки в браузере:
1. Установка точки останова (breakpoint) в строке кода
2. Открытие вкладки Sources в DevTools
3. Пошаговое выполнение (step over) и анализ переменных
4. Перезагрузка страницы для активации отладчика

Правильный ответ: 2, 1, 4, 3

23. Выберите два правильных ответа. Какие два правила безопасности необходимо соблюдать при установке сторонних npm-пакетов?
а) Проверка популярности пакета и наличия обновлений
б) Установка всех зависимостей с правами администратора системы
в) Аудит зависимостей через npm audit на наличие уязвимостей
г) Отключение антивирусного ПО для ускорения установки
Правильный ответ: а) в)

24. Установите соответствие между HTTP-методом и его типичным применением в REST API:

Метод	**Применение**
А) GET	1) Обновление существующего ресурса
Б) POST	2) Получение данных с сервера
В) PUT	3) Создание нового ресурса
Г) DELETE	4) Удаление указанного ресурса

Правильный ответ: А-2, Б-3, В-1, Г-4

25. Укажите один правильный ответ. Какое действие браузера считается наиболее затратным по производительности при динамическом обновлении страницы?
а) Перерасчёт геометрии элементов (reflow)
б) Изменение цвета фона (repaint)
в) Загрузка шрифта из кэша
г) Обработка события mousemove
Правильный ответ: а)

16. Запишите термин (с большой буквы, в именительном падеже), обозначающий стандарт комментирования кода на JavaScript, позволяющий автоматически генерировать документацию.
_____ — формат описания функций, параметров и возвращаемых значений. **Правильный ответ: JSDoc**

Раздел 7. Перечень учебной литературы, необходимой для освоения дисциплины

7.1. Обязательная литература

1. Адамс, Д. Р. Основы работы с XHTML и CSS : учебник / Д. Р. Адамс, К. С. Флойд. — 4-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2025. — 567 с. — ISBN 978-5-4497-0907-3. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: https://www.iprbookshop.ru/146372.html

2. Борисов, Р. С. Информатика и программирование : учебное пособие для среднего профессионального образования / Р. С. Борисов, А. С. Скотченко. — Москва : Российский государственный университет правосудия, 2023. — 334 с. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: https://www.iprbookshop.ru/133635.html

3. Боровков, В. А. Информатика и программирование. Текстовый редактор MS Word : учебное пособие для СПО / В. А. Боровков, С. М. Колмогорова. — Москва : Ай Пи Ар Медиа, 2023. — 136 с. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: https://www.iprbookshop.ru/129311.html

4. Кашевский, П. А. Типографика : учебное пособие / П. А. Кашевский. — Минск : Республиканский институт профессионального образования (РИПО), 2022. — 298 с. — ISBN 978-985-895-072-9. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: https://www.iprbookshop.ru/134164.html

5. Киренберг, А.Г. Основы информатики, организации ЭВМ, вычислительных и информационных систем : учебное пособие / А. Г. Киренберг, В. О. Коротин. — Кемерово : Кузбасский государственный технический университет имени Т.Ф. Горбачева, 2023. — 165 с. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: https://www.iprbookshop.ru/135106.html

6. Кучерова, А. В. Типографика: основы верстки : учебное пособие / А. В. Кучерова. — Омск : Омский государственный технический университет, 2023. — 97 с. — ISBN 978-5-8149-3649-3. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: https://www.iprbookshop.ru/140872.html

7. Кучерова, А. В. Типографика : учебное пособие / А. В. Кучерова. — Омск : Омский государственный технический университет, 2022. — 144 с. — ISBN 978-5-8149-3460-4. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: https://www.iprbookshop.ru/131234.html

8. Петракова, Н. В. Веб-программирование. Основы HTML : учебное пособие для СПО / Н. В. Петракова. — Саратов : Профобразование, 2026. — 49 с. — ISBN 978-5-4488-2823-2. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: https://www.iprbookshop.ru/160948.html

9. Савельев, А. О. HTML5. Основы клиентской разработки : учебное пособие / А. О. Савельев, А. А. Алексеев. — 4-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2024. — 270 с. — ISBN 978-5-4497-2398-7. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: https://www.iprbookshop.ru/133910.html

10. Сычев, А. В. Web-технологии : учебное пособие / А. В. Сычев. — 4-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2024. — 407 с. — ISBN 978-5-4497-2429-8. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: https://www.iprbookshop.ru/133914.html

11. Фролов, А. Б. Web-сайт. Разработка, создание, сопровождение : учебное пособие / А. Б. Фролов, И. А. Нагаева, И. А. Кузнецов ; под редакцией И. А. Нагаевой. — 2-е изд. — Саратов : Вузовское образование, 2024. — 355 с. — ISBN 978-5-4487-1025-4. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: https://www.iprbookshop.ru/142801.html

7.2. Дополнительная литература

1. Бурьков, Д. В. Информатика и программирование : учебное пособие / Д. В. Бурьков. — Ростов-на-Дону, Таганрог : Издательство Южного федерального университета, 2022. — 215 с. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: https://www.iprbookshop.ru/131449.html

2. Кисленко, Н. П. Информатика и программирование : учебное пособие / Н. П. Кисленко, И. Н. Мухина. — Новосибирск : Новосибирский государственный архитектурно-строительный университет (Сибстрин), ЭБС АСВ, 2022. — 105 с. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: https://www.iprbookshop.ru/129325.html

3. Кузьменко, И. П. Информатика и программирование : учебник для иностранных студентов / И. П. Кузьменко, С. В. Богданова. — Ставрополь : Ставропольский государственный аграрный университет, 2022. — 184 с. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: https://www.iprbookshop.ru/129581.html

4. Моренкова, О. И. Введение в курс информатики : учебное пособие / О. И. Моренкова, Т. И. Парначева. — Новосибирск : Сибирский государственный университет телекоммуникаций и информатики, 2020. — 158 с. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: https://www.iprbookshop.ru/117092.html

7.3. Перечень ресурсов информационно-телекоммуникационной сети «Интернет»

1. http: //elibrary.ru/
2. https://habr.com/
3. Справочно-правовая система «Консультант Плюс».
4. «Гарант»
5. ПО для организации конференций

Раздел 8. Материально-техническая база и информационные технологии

Материально-техническое обеспечение дисциплины «**Программирование для дизайна**» включает в себя учебные аудитории для проведения лекционных, лабораторных занятий, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, самостоятельной работы обучающихся.

Учебные аудитории укомплектованы специализированной мебелью и техническими средствами обучения. Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к сети Интернет.

Дисциплина может реализовываться с применением дистанционных технологий обучения. Специфика реализации дисциплины с применением дистанционных технологий обучения устанавливается дополнением к рабочей программе. В части не противоречащей специфике, изложенной в дополнении к программе, применяется настоящая рабочая программа.

Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине с применением дистанционных образовательных технологий включает в себя:
Компьютерная техника, расположенная в учебном корпусе Института (ул.Качинцев, 63, кабинет Центра дистанционного обучения):

1) Intel i3 3.4Ghz\ОЗУ 4Gb\500GB\RadeonHD5450
2) Intel PENTIUM 2.9GHz\ОЗУ 4GB\500GB
3) личные электронные устройства (компьютеры, ноутбуки, планшеты и иное), а также средства связи преподавателей и студентов.

Информационные технологии, необходимые для осуществления образовательного процесса по дисциплине с применением дистанционных образовательных технологий включают в себя:

- система дистанционного обучения (СДО) (Learning Management System) (LMS) Moodle (Modular Object-Oriented Dynamic Learning Environment);
- электронная почта;
- система компьютерного тестирования;
- Цифровой образовательный ресурс IPR SMART;
- система интернет-связи skype;
- телефонная связь;
- ПО для проведения конференций.

Обучение обучающихся инвалидов и обучающихся с ограниченными возможностями здоровья осуществляется посредством применения специальных технических средств в зависимости от вида нозологии.

При проведении учебных занятий по дисциплине используются мультимедийные комплексы, электронные учебники и учебные пособия, адаптированные к ограничениям здоровья обучающихся.

Лекционные аудитории оборудованы мультимедийными кафедрами, подключенными к звуковым колонкам, позволяющими усилить звук для категории слабослышащих обучающихся, а также проекционными экранами, которые увеличивают изображение в несколько раз и позволяют воспринимать учебную информацию обучающимся с нарушениями зрения.

При обучении лиц с нарушениями слуха используется усилитель слуха для слабослышащих людей Cyber Ear модель HAP-40, помогающий обучаемым лучше воспринимать учебную информацию.

Обучающиеся с ограниченными возможностями здоровья, обеспечены печатными и электронными образовательными ресурсами (программы, учебники, учебные пособия, материалы для самостоятельной работы и т.д.) в формах, адаптированных к ограничениям их здоровья и восприятия информации:

для лиц с нарушениями зрения:

- в форме электронного документа;
- в форме аудиофайла;

для лиц с нарушениями слуха:

- в печатной форме;
- в форме электронного документа;

для лиц с нарушениями опорно-двигательного аппарата:
- в печатной форме;
- в форме электронного документа;
- в форме аудиофайла.

Программное обеспечение, используемое на занятиях**:**

- Операционная система Windows,
- Архиватор 7-zip,
- Система тестирования,
- Microsoft Office 2007,
- Антивирус Касперский 6,
- Консультант+,
- Виртуальная машина VirtualBox,
- Виртуальная машина VirtualPC,
- Internet Explorer.
-

Раздел 9. Методические указания для обучающихся по освоению дисциплины

Для успешного усвоения материала дисциплины требуются значительное время, концентрация внимания и усилия: посещение лекционных занятий и конспектирование преподаваемого материала, работа с ним дома, самостоятельная проработка материала рекомендуемых учебников и учебных пособий при самостоятельной подготовке. Особое внимание следует обратить на выполнение практических работ, практических задач по СРО, тестовых вопросов.

При самостоятельной работе с учебниками и учебными пособиями полезно иметь под рукой справочную литературу (энциклопедии) или доступ к сети Интернет, так как могут встречаться новые термины, понятия, которые раньше обучающиеся не знали.

Цель практических занятий по дисциплине - закрепление знаний по определенной теме, приобретенных в результате прослушивания лекций, получения консультаций и самостоятельного изучения различных источников литературы. При выполнении данных работ обучающиеся должны будут глубоко изучить состав и принцип работы современных информационных систем. Получить практические навыки работы с современными ИС.

Перед практическим занятием обучающийся должен детально изучить теоретические материалы вопросов практики в учебниках, конспектах лекций, периодических журналах и прочее. Если при выполнении практического задания у обучающегося остаются неясности, то ему необходимо оперативно обратиться к преподавателю за уточнением.

После выполнения практического задания обучающиеся должны выполнить самостоятельную работу. Самостоятельная работа включает в себя индивидуальное задание по пройденной теме. Таким образом, каждый обучающийся выполняет только свой вариант задания.

При дистанционном выполнении практических работ обучающийся может самостоятельно приобрести операционные системы Windows XP, Windows 7, Windows 8, Windows 10 и пакет Microsoft Office или Open Office. Ответственность за установку и настройку программного обеспечения в данном случае ложится на обучающегося. Следует воспользоваться методическими указаниями по установке данных программных систем.

Результаты выполненных заданий оцениваются с учетом теоретических знаний по соответствующим разделам дисциплины, техники выполнения работы, объективности и обоснованности принимаемых решений в процессе работы с данными, качества оформления. Переход к выполнению следующего практического задания допускается только после отчета выполненной работы.

Учебно-методическое издание

Рабочая программа учебной дисциплины

Программирование для дизайна

(Наименование дисциплины в соответствии с учебным планом)

Сафонова Елена Владимировна

(Фамилия, Имя, Отчество составителя)