

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Андрей Драгомирович Хлудков
Должность: директор
Дата подписания: 17.06.2026 18:14:16
Уникальный программный ключ:
880f7c07c583b07b775f6604c39281b15e9f12

Федеральное государственное бюджетное образовательное учреждение
высшего образования

**РОССИЙСКАЯ АКАДЕМИЯ НАРОДНОГО ХОЗЯЙСТВА и
ГОСУДАРСТВЕННОЙ СЛУЖБЫ при ПРЕЗИДЕНТЕ РОССИЙСКОЙ
ФЕДЕРАЦИИ**

СЕВЕРО-ЗАПАДНЫЙ ИНСТИТУТ УПРАВЛЕНИЯ

Факультет среднего профессионального образования

УТВЕРЖДЕНА
решением цикловой (методической)
комиссии общепрофессиональных
дисциплин и профессиональных
модулей специальностей 09.02.07
Информатика и вычислительная
техника
Протокол от 31.10.2025 № 2

РАБОЧАЯ ПРОГРАММА ПРАКТИКИ

ПП.02.01 Производственная практика

Специальность – 09.02.11 Разработка и управление программным обеспечением

Профиль – на базе основного общего образования

Квалификация – программист

Форма обучения – очная

Год набора – 2026

Санкт-Петербург 2025 год

Автор-составитель: Вилков Владислав Евгеньевич, преподаватель 1 категории ФСПО СЗИУ РАНХиГС.

СОДЕРЖАНИЕ

1. Общие положения	4
1.1. Область применения программы	4
1.2. Место практики в структуре основной профессиональной образовательной программы	4
1.3. Цели и задачи производственной практики	4
1.4. Планируемые результаты обучения по производственной практике	5
2. Структура и содержание производственной практики.....	16
2.1. Объем производственной практики и виды работ	16
3. Материалы текущего контроля успеваемости и промежуточной аттестации обучающихся	20
3.1. Формы и методы текущего контроля успеваемости обучающихся и промежуточной аттестации.....	20
3.2. Оценочные средства текущей и промежуточной аттестации обучающихся.....	21
4. Методические указания для обучающихся по освоению производственной практики.....	26
5. Учебная литература и ресурсы информационно-телекоммуникационной сети «Интернет»	30
6. Материально-техническая база, информационные технологии, программное обеспечение и информационные справочные системы	31
7. Приложения.....	33

1 Общие положения

1.1 Область применения программы

Рабочая программа производственной практики ПП.02.01 является частью основной профессиональной образовательной программы в соответствии с ФГОС по специальности СПО 09.02.11 «Разработка и управление программным обеспечением».

1.2 Место практики в структуре основной профессиональной образовательной программы

Производственная практика по профессиональному модулю ПМ.02 «Разработка и интеграция модулей программного обеспечения» и представляет собой неотъемлемый элемент профессиональной подготовки обучающихся. Она базируется на освоении дисциплин «Разработка программных модулей», «Осуществление интеграции программных модулей», «Поддержка и тестирование программных модулей», «Математическое моделирование», «Численные методы» и «Безопасность программного обеспечения» и реализуется на 3 курсе обучения в 6 семестре.

1.3 Цели и задачи производственной практики

Цель производственной практики «ПП.02.01 Производственная практика»: формирование у студентов систематизированных знаний и практических навыков в области разработки и тестирования программного обеспечения, включая проектирование модулей с учётом технического задания и визуализацию архитектурных решений, определение интерфейсов и взаимодействия модулей в системе, создание программных модулей и работу с API и веб-сервисами для их интеграции. В рамках практики студенты осваивают работу с интеграционными платформами и инструментами, отладку и комплексное тестирование ПО — от формирования тестовых сценариев и подготовки тестовых платформ до выполнения тестовых процедур на тестовые данные и оценки объёма тестирования для определения необходимых ресурсов. Отдельное внимание уделяется навыкам технической документации: созданию документации для модулей, документированию кода, API и интерфейсов, а также работе со специализированным программным обеспечением для документирования программного кода и формированию отчётности о подготовке к выполнению заданий на тестирование в соответствии с установленными регламентами.

1.4 Планируемые результаты обучения по производственной практике

Перечень профессиональных компетенций

Код и наименование компетенции	Умения	Знания	Навыки
ПК 2.1 Проектировать модули программного обеспечения	<ul style="list-style-type: none"> – проектировать модули, соответствующие бизнес-задачам; – создавать архитектурные диаграммы и документацию; – определять структуру и интерфейсы модулей; – анализировать требования к модулю и определять его функциональность; – проектировать архитектуру модуля, включая выбор подходящих паттернов проектирования и структуры данных; – создавать диаграммы классов, последовательностей и прочих диаграмм для визуализации проектируемого модуля; – выбирать подходящие языки программирования и технологии для реализации модуля; – проектировать интерфейсы программного обеспечения для взаимодействия с другими модулями и системами; – учитывать требования к масштабируемости, производительности и безопасности при проектировании модуля; 	<ul style="list-style-type: none"> – основные принципы проектирования модулей программного обеспечения; – языки программирования и технологии для реализации модулей; – паттерны проектирования и структуры данных для создания эффективных и масштабируемых модулей; – методы анализа требований и способов определения функциональности модуля; – принципы создания интерфейсов для взаимодействия с другими модулями и системами; – принципы обеспечения безопасности, производительности и масштабируемости при проектировании модулей; – методы анализа и оптимизации проектируемых модулей для повышения их эффективности и качества. 	<ul style="list-style-type: none"> – проектирования модулей ПО с учетом требований заказчика; – создания архитектурных диаграмм и спецификаций модулей; определения интерфейсов и взаимодействия модулей в системе.

	<p>проводить анализ и оптимизацию проектируемого модуля для повышения его эффективности и качества</p>		
<p>ПК 2.2 Разрабатывать модули программного обеспечения</p>	<ul style="list-style-type: none"> – разрабатывать модули программного обеспечения с использованием различных языков программирования и технологий; – применять паттерны проектирования и структуры данных для создания эффективных и масштабируемых модулей; – анализировать требования и определять функциональность модуля; – создавать интерфейсы для взаимодействия с другими модулями и системами; – обеспечивать безопасность, производительность и масштабируемость при разработке модулей; – оптимизировать проектируемые модули для повышения их эффективности и качества; – работать с системой контроля версий; – улучшать производительность модулей, выявляя и устраняя узкие места; – проводить анализ и мониторинг 	<ul style="list-style-type: none"> – язык программирования, основные конструкции, синтаксис; – паттерны проектирования; – структуры данных; – принципы создания интерфейсов для взаимодействия с другими модулями и системами, таких как REST API, SOAP; – работу с инструментальным программным обеспечением; – методы оптимизации кода и алгоритмов; – эффективные алгоритмы и структуры данных для повышения производительности; – многопоточность в программных модулях; – методы оптимизации сетевых протоколов для ускорения обмена данными; – кэширование данных; – управление памятью; техники повышения производительности программного обеспечения 	<ul style="list-style-type: none"> – создания модулей программного обеспечения на различных языках программирования; – отладки и тестирования разработанных модулей; – применения структурного и объектно-ориентированного программирования; – оптимизации кода и алгоритмов программных модулей для увеличения производительности; мониторинга и анализа производительности приложений.

	<p>производительности приложений; применять инструменты для рефакторинга и оптимизации программного кода.</p>		
<p>ПК 2.3 Выполнять интеграцию модулей и компонентов программного обеспечения</p>	<ul style="list-style-type: none"> – интегрировать модули и компоненты, обеспечивая их взаимодействие; – работать с API и устанавливать соединения между компонентами; – отслеживать и устранять конфликты и ошибки интеграции; – анализировать и определять зависимости между модулями и компонентами; – работать с различными форматами данных и протоколами передачи данных 	<ul style="list-style-type: none"> – общие принципы функционирования аппаратных, программных и программно-аппаратных средств администрируемой информационно-коммуникационной системы; – международные стандарты локальных вычислительных сетей; – методы и подходы к интеграции модулей и компонентов; – принципы версионирования и управления изменениями при интеграции; – принципы безопасности при интеграции модулей и компонентов 	<ul style="list-style-type: none"> – интеграции программных модулей и компонентов в единое программное решение; – работы с API и веб-сервисами для взаимодействия между модулями; – работы с интеграционными платформами и инструментами; – обеспечения совместимости и стабильности системы
<p>ПК 2.4 Выполнять тестирование и отладку программного обеспечения</p>	<ul style="list-style-type: none"> – анализировать требования к программному обеспечению и составлять планы тестирования; – создавать тестовые сценарии и тест-кейсы для проверки функциональности и соответствия требованиям; – выполнять тестирование программного обеспечения вручную и автоматизировать процесс тестирования; – анализировать результаты тестирования и 	<ul style="list-style-type: none"> – принципы и методы тестирования программного обеспечения; – основы программирования и архитектуры программного обеспечения; – основы баз данных и SQL-запросов; – инструменты для автоматизации тестирования; – основы разработки и отладки программного обеспечения на разных языках программирования; – понятие дефекта программного обеспечения; 	<ul style="list-style-type: none"> – отладки программного обеспечения на уровне программных модулей; – тестирования программного обеспечения; – формирования тестовых сценариев; – подготовки тестовых платформ (установка операционной системы, дополнительного ПО и другого по необходимости); – оценки объема тестирования ПО с целью определения необходимых ресурсов для его выполнения; – настройки тестовой среды и аппаратных средств для выполнения

	<p>документировать найденные ошибки;</p> <ul style="list-style-type: none"> – разрабатывать стратегии отладки и исправлять ошибки в программном обеспечении; – выполнять модульные тесты с использованием инструментов тестирования, в том числе автоматизированного тестирования; – использовать системы контроля дефектов ПО; <p>составлять отчет о выполнении тестирования ПО</p>	<ul style="list-style-type: none"> – критерии качества ПО; – виды и типы тестирования ПО; – техники ручного тестирования; – техники автоматизированного тестирования; – жизненный цикл дефекта ПО; – принципы работы в системе контроля дефектов; <p>основные понятия о качестве ПО</p>	<p>тестирования ПО в соответствии с заданием на тестирование в пределах своей компетенции;</p> <ul style="list-style-type: none"> – формирования и представления отчетности о подготовке к выполнению задания на тестирование ПО в соответствии с установленными регламентами; <p>выполнения тестовых процедур на тестовых данных</p>
<p>ПК 2.5 Осуществлять документирование программных модулей программного обеспечения</p>	<ul style="list-style-type: none"> – описывать функциональность модулей в документации; – создавать диаграммы для иллюстрации работы модулей; – программировать с использованием комментариев для документирования кода; – использовать специальные метки/теги для отметки важных частей кода в документации; – вести журнал изменений и фиксировать обновления программных модулей; – разбивать модули на логические блоки и описывать каждый блок отдельно; – включать в документацию особенности модулей, такие как ограничения, 	<ul style="list-style-type: none"> – стандарты технической документации; – принципы документирования программного обеспечения; <p>инструменты для создания технической документации и комментирования кода</p>	<ul style="list-style-type: none"> – создания технической документации для модулей; – документирования кода, API и интерфейсов; <p>работы со специализированным ПО по документированию программного кода</p>

	уязвимости или оптимальные настройки; проводить регулярное обновление документации при изменении модулей или добавлении нового функционала.		
--	--	--	--

В результате освоения производственной практики студент должен:

иметь практический опыт	<ul style="list-style-type: none"> – Проектировать модули ПО с учётом технического задания; – Визуализировать и описывать архитектурные решения; – Определять интерфейсы и взаимодействие модулей в системе; – Создавать модули программного обеспечения; – Работать с API и веб-сервисами для взаимодействия между модулями; – Работать с интеграционными платформами и инструментами; – Отлаживать ПО на уровне программных модулей; – Тестировать программное обеспечение; – Формировать тестовые сценарии; – Подготавливать тестовые платформы; – Оценивать объём тестирования ПО для определения ресурсов; – Формировать и представлять отчётность о подготовке к тестированию ПО; – Выполнять тестовые процедуры на тестовых данных; – Создавать техническую документацию для модулей; – Документировать код, API и интерфейсы; – Работать со специализированным ПО для документирования кода.
уметь	<ul style="list-style-type: none"> – Анализировать техническое задание для проектирования модулей ПО; – Применять архитектурные шаблоны (MVC, MVVM, MVP) при разработке модулей; – Использовать системы контроля версий при разработке модулей; – Реализовывать паттерны проектирования; – Разрабатывать модули для работы с массивами, коллекциями, датой и временем; – Организовывать асинхронный и параллельный вызов методов; – Работать с файловой системой (навигация, чтение/запись файлов, потоки); – Обращивать текстовые данные (кодирование/декодирование, регулярные выражения); – Проектировать GUI (окна, элементы управления, обработка событий); – Разрабатывать модули для взаимодействия с БД (CRUD-операции, запросы); – Оптимизировать код и алгоритмы для повышения производительности; – Обеспечивать безопасность при разработке модулей (аутентификация, защита от уязвимостей); – Интегрировать модули через REST API, WebSocket, микросервисы; – Настраивать логирование и мониторинг приложений; – Контейнеризировать приложения и развёртывать решения; – Проводить тестирование ПО (модульное, интеграционное, нагрузочное); – Анализировать качество модулей с использованием метрик; – Обращивать исключения и ошибки в коде; – Рефакторить и оптимизировать код; – Составлять техническую документацию и спецификации.

знать	<ul style="list-style-type: none"> – Принципы ООП и модульной архитектуры приложений; – Архитектурные шаблоны и паттерны проектирования; – Основы работы с системами контроля версий; – Принципы работы с массивами, коллекциями, строками, датой и временем; – Асинхронную и параллельную модели программирования; – Принципы работы с файловой системой и потоками данных; – Методы работы с текстовыми данными; – Технологии разработки GUI; – Принципы взаимодействия приложения с БД, основы ORM и CRUD; – Методы обеспечения безопасности, производительности и масштабируемости модулей; – Протоколы и стандарты интеграции (REST API, WebSocket и др.); – Инструменты логирования, мониторинга и контейнеризации; – Виды и методы тестирования ПО; – Метрики качества программных модулей; – Принципы отладки кода и обработки исключений; – Стандарты документирования ПО; – Основные угрозы безопасности (SQL-инъекции, XSS, CSRF) и методы защиты; – Основы математического моделирования и численных методов; – Принципы криптографии и безопасной аутентификации.
-------	---

2 Структура и содержание производственной практики

2.1 Объем производственной практики и виды работ

Виды работы	Объем работы, час.
Учебная нагрузка обучающихся всего, в том числе:	144 (4 недели)
практические занятия	144
промежуточная аттестация	-
Форма промежуточной аттестации	Зачет с оценкой

Производственная практика проводится концентрировано, в соответствии с графиком учебного процесса на текущий учебный год.

Производственная практика может быть организована на базе сторонних организаций, образовательных учреждений либо структурных подразделений учебного заведения, которые располагают достаточным кадровым и научно-техническим потенциалом.

Распределение студентов для прохождения практики осуществляется на основании:

1. договоров, которые университет заключает с организациями независимо от их организационно-правовой формы (включая коммерческие, некоммерческие, государственные и муниципальные структуры);

2. гарантийных писем от организаций, в которых фиксируются обязательства предоставить:

- условия, необходимые для реализации программы практики;
- квалифицированных специалистов для руководства практикой;
- условия, обеспечивающие безопасность труда студентов.

Если в организации имеются вакантные должности, студенты могут быть оформлены на них — при условии, что выполняемая работа соответствует задачам и требованиям программы практики.

Структура, содержание и виды работ производственной практики приведены в таблице «Тематический план и содержание производственной практики»

Тематический план и содержание производственной практики

п/п	Наименование тем (разделов)	Содержание и виды работ	Распределение часов	Формируемые компетенции
			ПР	
1	Тема 1.1. Разработка программных модулей	<p>Содержание:</p> <ul style="list-style-type: none"> — цели и задачи практики. Требования охраны труда. Инструктаж по технике безопасности во время прохождения практики. — модульная архитектура построения приложений: принципы, преимущества, примеры приложений; — архитектурные шаблоны, применяемые при разработке программных модулей (MVC, MVVM, MVP); — инструменты разработки приложений с модульной архитектурой, системы контроля версий; — работа с библиотеками: применение стандартных библиотек, создание библиотек; — базовые принципы работы с массивами, коллекциями, строками, датой и временем; — паттерны проектирования (отношения между классами и объектами: наследование, реализация, ассоциация, композиция, агрегация; интерфейсы, абстрактные классы; порождающие, структурные и поведенческие паттерны); — система ввода-вывода, средства доступа к файлам и папкам файловой системы (чтение/запись, сжатие потоков, изолированное хранение); — работа со строками, регулярными выражениями, кодирование/декодирование текста; — асинхронная модель программирования (пул потоков, шаблон асинхронного вызова методов, синхронизация вызывающего потока); — параллельное программирование (создание задачи, методы ожидания выполнения задачи, лямбда-выражения, создание продолжения задачи, возврат значений, отмена задачи). <p>Виды работ:</p> <p>1. Разработка программных модулей для работы с массивами (с использованием системы контроля версий).</p>	30	ПК 2.1 ПК 2.2 ПК 2.3 ПК 2.4 ПК 2.5

		<p>2. Разработка программных модулей для работы с коллекциями (с использованием системы контроля версий).</p> <p>3. Разработка программных модулей для работы с датой и временем (с использованием системы контроля версий).</p> <p>4. Разработка программных модулей с использованием паттернов проектирования (с использованием системы контроля версий).</p> <p>5. Навигация по файловой системе: чтение и запись файлов, работа с потоками, изолированным хранилищем.</p> <p>6. Работа с большими объёмами текста: кодирование/декодирование строк, построение регулярных выражений, чтение/запись файлов в разных кодировках.</p> <p>7. Организация асинхронного вызова методов.</p> <p>8. Создание программного модуля, выполняющего методы в рамках параллельных задач.</p>		
2	Тема 1.2. Осуществление интеграции программных модулей	<p>Содержание:</p> <ul style="list-style-type: none"> — алгоритмы и структуры данных, оценка сложности алгоритмов (асимптотическая оценка, большие O-нотации, временная и пространственная сложность); — основные структуры данных (массив, связный список, стек, очередь, операции вставки/поиска/удаления, представление данных в памяти); — алгоритмы сортировки и поиска, основы рекурсии; — хеш-таблицы и хеш-функции (коллизии, методы хеширования, сжатие данных, эффективность хеш-структур); — деревья и графы (представление, поиск в глубину/ширину, алгоритм Дейкстры, деревья поиска и обхода); — жадные алгоритмы и динамическое программирование; — алгоритмы работы с текстовыми данными (поиск подстроки, алгоритмы Кнута–Морриса–Пратта, Бойера–Мура, Рабина–Карпа, операции с динамическими строками, триальные деревья); — кучи и очереди (очереди с приоритетом, применение кучи). <p>Виды работ:</p> <ol style="list-style-type: none"> 1. Оценка сложности алгоритмов. 2. Применение рекурсивных алгоритмов. 3. Работа с алгоритмами сортировки и поиска. 4. Создание хеш-таблиц и использование для ускорения поиска данных. 5. Нахождение кратчайших путей в графах (алгоритм Дейкстры). 	28	

		<p>6. Решение задачи о рюкзаке (метод динамического программирования).</p> <p>7. Реализация строковых алгоритмов.</p> <p>8. Реализация приоритетных очередей для планирования задач.</p>		
3	Тема 1.3. Поддержка и тестирование программных модулей	<p>Содержание:</p> <ul style="list-style-type: none"> — основные принципы проектирования модулей ПО, методы анализа требований и определения функциональности модуля; — методы анализа и оптимизации проектируемых модулей; — декомпозиция задачи на подзадачи, создание спецификаций модуля; — принципы обеспечения безопасности, производительности и масштабируемости при проектировании модулей; — принципы проектирования классов (инкапсуляция, наследование, полиморфизм); — применение диаграмм классов и компонентов для проектирования внутренней структуры модуля. <p>Виды работ:</p> <ol style="list-style-type: none"> 1. Анализ требований к модулю и определение функциональности. 2. Создание спецификации программного модуля. 3. Проектирование требований к внутренней структуре модуля (диаграммы классов, паттерны проектирования). 4. Проектирование требований к организации компонентов модуля (диаграммы компонентов). 5. Проектирование интерфейсов ПО для взаимодействия с другими модулями и системами. 6. Анализ и оптимизация проектируемого модуля. 	22	
4	Тема 1.4. Математическое моделирование	<p>Содержание:</p> <ul style="list-style-type: none"> — понятие модели, классификация моделей, математическая модель, типы математических моделей; — принципы построения математических моделей, основные этапы математического моделирования. <p>Виды работ:</p> <ol style="list-style-type: none"> 1. Построение простейших математических моделей. 	6	
5	Тема 1.5. Численные методы	<p>Содержание:</p> <ul style="list-style-type: none"> — приближённые числа и действия над ними (погрешности, значащие цифры); — численные методы решения алгебраических и трансцендентных уравнений (метод половинного деления, метод простой итерации, методы Ньютона); — численные методы решения систем линейных алгебраических уравнений (метод Гаусса, метод простой итерации, метод Зейделя); 	22	

		<p>— интерполяция и экстраполяция функций (интерполяционный многочлен Лагранжа, интерполяционные формулы Ньютона, интерполяция сплайнами);</p> <p>— численное интегрирование (квадратурные формулы Ньютона–Котеса, формула Гаусса);</p> <p>— численные методы решения обыкновенных дифференциальных уравнений (метод Эйлера, метод Рунге–Кутты);</p> <p>— численное решение задач оптимизации (методы минимизации функции одной и двух переменных).</p> <p>Виды работ:</p> <ol style="list-style-type: none"> 1. Вычисление погрешностей приближённых значений. 2. Решение алгебраических и трансцендентных уравнений приближёнными методами. 3. Решение систем линейных алгебраических уравнений методом Гаусса. 4. Составление интерполяционных формул Лагранжа и Ньютона. 5. Вычисление интегралов при помощи формул Ньютона–Котеса и Гаусса. 6. Нахождение решений обыкновенных дифференциальных уравнений методами Эйлера и Рунге–Кутты. 7. Нахождение экстремумов функций одной и двух переменных приближёнными методами. 		
6	Тема 1.6. Безопасность программного обеспечения	<p>Содержание:</p> <ul style="list-style-type: none"> — основы кибербезопасности и уязвимости ПО; — модели угроз и анализ рисков; — уязвимости веб-приложений (OWASP Top 10); — безопасная аутентификация и авторизация; — криптография для разработчиков; — принципы безопасного проектирования архитектуры; — криптографические протоколы и их реализация; — криптография в мобильных приложениях, веб-приложениях, облачных средах. <p>Виды работ:</p> <ol style="list-style-type: none"> 1. Анализ кода на наличие уязвимостей (ручной review). 2. Эксплуатация и защита от SQL-инъекций. 3. Создание и предотвращение XSS-атак. 4. Реализация защиты от CSRF-атак. 5. Составление модели угроз для типового веб-приложения. 6. Настройка безопасной аутентификации (JWT, refresh-токены). 7. Реализация RBAC-системы с разделением привилегий. 8. Шифрование данных (AES, RSA). 9. Хэширование паролей (bcrypt, Argon2). 10. Анализ сетевого трафика (Wireshark). 	36	

		11. Сканирование уязвимостей (OWASP ZAP, Burp Suite). 12. Настройка HTTPS и создание самоподписанных сертификатов. 13. Защита от brute-force атак. 14. Безопасная работа с файлами и десериализацией данных.		
		Итого часов:	144	

3 Материалы текущего контроля успеваемости и промежуточной аттестации обучающихся

3.1 Формы и методы текущего контроля успеваемости и промежуточной аттестации обучающихся

Форма текущего контроля – практическая: проверка умений применять знания на практике.

Методы текущего контроля: наблюдение – фиксация руководителем практики от предприятия действий и поведения практиканта в процессе практики; письменный – отзыв руководителя практики от предприятия, дневник и отчёт по практике.

Форма промежуточной аттестации – зачёт с оценкой. Метод – устный – собеседование по отчёту по практике с руководителем практики от филиала.

Во время практики студент обязан вести дневник (ПРИЛОЖЕНИЕ В), фиксируя все этапы и задачи. Дневник должен быть оформлен согласно утверждённым требованиям и заверен подписью непосредственного руководителя практики от предприятия.

По окончании практики студент готовит отчёт, соблюдая установленную форму (титульный лист в ПРИЛОЖЕНИИ Б) и отражающий результаты прохождения практики. В отчёте необходимо систематизировать и представить знания и навыки, приобретённые в ходе практики в соответствии с поставленными задачами.

Форма отчёта определяется на основании методических указаний, а его содержание формируется в соответствии с программой практики и индивидуальным заданием (ПРИЛОЖЕНИЕ А).

По завершении практики руководители практики со стороны предприятия, оформляет отзыв (ПРИЛОЖЕНИЕ Г). В отзыве указывается:

степень выполнения программы практики;

отношение студента к работе;

соблюдение трудовой дисциплины;

уровень освоения производственных (профессиональных) навыков;

участие в рационализаторской деятельности и общественной жизни организации;

другие показатели сформированности общих и профессиональных компетенций, а также приобретённых умений и практического опыта по специальности.

Чтобы получить допуск к промежуточной аттестации по практике, студент должен предоставить следующий пакет документов:

- отзыв, подписанный руководителем практики от предприятия с печатью предприятия, или комплект выполненных индивидуальных заданий – в случае прохождения практики непосредственно в учебном заведении;
- заполненный дневник практики;
- отчёт о прохождении практики;
- приложения к отчёту, оформленные в соответствии с установленными требованиями.

На основании представленных документов руководитель практики принимает решение о том, допускается ли студент к промежуточной аттестации.

Аттестация по итогам производственной практики проходит в форме собеседования, при этом учитываются (или служат основой) результаты, подтверждённые документами от предприятия.

Промежуточная аттестация проходит в два этапа:

1. Студент выступает с кратким докладом (продолжительностью до 5 минут), в котором подводит итоги своей работы во время практики.

2. Студент отвечает на вопросы, касающиеся содержания отчёта по практике.

Оценка результатов практики осуществляется по следующим критериям:

- отзыв руководителя практики, включающий оценку работы студента;
- уровень подготовленности, инициативности и дисциплинированности студента, отражённый в отзыве руководителя;
- степень выполнения запланированной программы практики;
- качество и содержательность подготовленного студентом отчёта;
- продемонстрированный студентом уровень знаний при защите отчёта о прохождении практики.

3.2 Оценочные средства текущей и промежуточной аттестации обучающихся

Для текущей аттестации каждому студенту выдается индивидуальное задание на практику, содержащее общие задачи практики. Руководитель практики от предприятия на основе общих задач формулирует и выдаёт конкретные задания для выполнения их практикантом.

Типовые задачи производственной практики ПП.02.01 для текущей аттестации

Задачей производственной практики является ознакомление студентов непосредственно в организации с производственными процессами с целью закрепления теоретических знаний и получение практических навыков в проектирования и разработки программного обеспечения.

Студент должен ознакомиться с разработкой, тестированием и исправлением ошибок кода. Программой практики предусмотрена работа студентов в ИТ-компании или подразделении ведущим работу в области информационных технологий.

В процессе прохождения практики студенту необходимо выполнить ряд задач:

1. разработать модуль информационной системы;
2. составить план тестирования модуля, провести модульное тестирование;
3. исправить обнаруженные ошибки;
4. подготовить документацию по эксплуатации.

Типовые вопросы для промежуточной аттестации по производственной практике ПП.02.01

1. Опишите пошаговый процесс проектирования модуля ПО с учётом бизнес-задач. Какие этапы вы считаете ключевыми?
2. Какие инструменты и нотации (UML, BPMN и т.д.) вы использовали для создания архитектурных диаграмм? Приведите пример диаграммы классов для гипотетического модуля.
3. Как вы анализируете требования к модулю и определяете его функциональность? Опишите метод сбора и формализации требований.
4. Какие паттерны проектирования вы применяли на практике? Приведите конкретный пример использования одного паттерна и объясните, почему выбрали именно его.
5. Как вы обеспечиваете масштабируемость и производительность модуля на этапе проектирования? Приведите примеры технических решений.
6. Какие критерии вы используете для выбора языка программирования и технологий реализации модуля? Сравните два языка/фреймворка для одной задачи.
7. Как проектируются интерфейсы взаимодействия модуля с другими компонентами системы? Опишите подход и приведите пример контракта API.
8. Расскажите о случае, когда вам пришлось оптимизировать архитектуру модуля. Какие изменения внесли и какой эффект получили?

9. Какие методы анализа требований вы считаете наиболее эффективными? Опишите опыт работы с пользовательскими историями (User Stories) или Use Case.
10. Как учитывать требования безопасности при проектировании модуля? Приведите 2–3 примера мер защиты на уровне архитектуры.
11. Опишите процесс разработки модуля от получения ТЗ до сдачи. Какие инструменты контроля версий вы использовали? Приведите пример workflow в Git.
12. Как вы оптимизируете код для повышения производительности? Приведите пример оптимизации алгоритма или структуры данных.
13. Расскажите о работе с многопоточностью. Какие проблемы синхронизации возникали и как их решали?
14. Как вы организуете кэширование данных в модуле? Опишите сценарий использования кэша и выбор стратегии инвалидации.
15. Какие методы отладки и профилирования производительности вы применяли? Назовите инструменты и их ключевые метрики.
16. Опишите опыт рефакторинга кода. Какие признаки «плохого кода» вы исправляли и какие паттерны применяли?
17. Как вы обеспечиваете безопасность модуля при разработке? Приведите примеры защиты от SQL-инъекций или XSS.
18. Расскажите о взаимодействии с БД в разрабатываемом модуле. Как использовали ORM или писали SQL-запросы? Приведите пример оптимизированного запроса.
19. Как организуется обработка исключений в вашем коде? Опишите иерархию исключений и стратегию логирования ошибок.
20. Какие инструменты автоматизации тестирования вы использовали при разработке? Опишите пример unit-теста для метода модуля.
21. Опишите процесс интеграции двух модулей с разными API. Какие протоколы и форматы данных вы использовали (REST, SOAP, JSON, XML)?
22. Как устранять конфликты зависимостей между модулями? Приведите пример проблемы и её решения.
23. Расскажите о работе с микросервисами. Как обеспечивали их взаимодействие и мониторинг?
24. Какие инструменты CI/CD вы применяли для автоматизации интеграции? Опишите pipeline сборки и деплоя.

25. Как тестируется совместимость модулей перед интеграцией? Опишите стратегию интеграционного тестирования.
26. Расскажите о проблеме сетевого взаимодействия между компонентами. Как диагностировали и устраняли задержки или потери данных?
27. Как управлять версиями API при интеграции? Приведите пример версионирования и обратной совместимости.
28. Какие меры безопасности вы внедрили при интеграции модулей? Опишите использование OAuth, JWT или шифрования трафика.
29. Расскажите о работе с интеграционными платформами (например, Apache Kafka, RabbitMQ). Как настраивали обмен сообщениями между сервисами?
30. Как обеспечиваете стабильность системы после интеграции нового модуля? Опишите методы мониторинга и отката изменений.
31. Составьте план тестирования для модуля авторизации. Какие виды тестирования включите (функциональное, нагрузочное и т. д.)?
32. Приведите пример тест-кейса для проверки CRUD-операции. Какие входные данные и ожидаемые результаты укажите?
33. Как автоматизируете тестирование? Опишите фреймворк (Selenium, JUnit и т. п.) и пример автотеста.
34. Расскажите о жизненном цикле дефекта. Как документируете и отслеживаете баги (Jira, Bugzilla)?
35. Какие метрики качества ПО вы отслеживаете? Приведите формулу расчёта покрытия кода тестами.
36. Как проводите нагрузочное тестирование? Опишите инструмент (JMeter, Gatling) и сценарий теста.
37. Расскажите о сложном баге, который отлаживали. Какие инструменты (отладчик, логи) использовали и как локализовали проблему?
38. Как оцениваете объём тестирования для нового модуля? Приведите методику расчёта ресурсов (времени, персонала).
39. Опишите стратегию регрессионного тестирования. Как часто его запускаете и какие тесты включаете?
40. Как готовите тестовую среду? Опишите процесс настройки ОС, БД и зависимостей для тестирования модуля.
41. Какие стандарты документации (ГОСТ, IEEE, Markdown) вы применяли? Приведите структуру технического описания модуля.

42. Как документируете API? Опишите использование Swagger/OpenAPI и приведите фрагмент спецификации.
43. Расскажите о практике комментирования кода. Какие инструменты (Doxygen, Javadoc) использовали для генерации документации?
44. Как ведёте журнал изменений (Changelog)? Опишите формат записи и частоту обновлений.
45. Приведите пример описания уязвимости в документации. Какие рекомендации даёте разработчикам по её устранению?
46. Как синхронизируете код и документацию при изменениях? Опишите процесс обновления комментариев и спецификаций.
47. Какие инструменты для создания диаграмм (PlantUML, Draw.io) вы использовали в документации? Приведите пример схемы взаимодействия модулей.
48. Как структурируете документацию для сложных модулей? Опишите разбиение на разделы и логику навигации.
49. Расскажите о стандартах именования и оформления кода. Как это влияет на читаемость и поддержку?
50. Как обеспечиваете актуальность документации после релиза? Опишите цикл ревизий и ответственных.

Критерии оценивания промежуточной аттестации:

Оценка «зачтено-отлично»: письменный отчёт оформлен без ошибок: соблюдены все требования к структуре (титульный лист, введение, основная часть, заключение, список литературы, приложения), отсутствуют ошибки в оформлении и содержании. Материал глубоко раскрывает все аспекты практики, включая индивидуальное задание. Выводы логичны, обоснованы, подкреплены фактами. Используются первоисточники, приложения (схемы, таблицы, фотоматериалы) уместны и информативны. Отчёт сдан в срок. При защите студент демонстрирует превосходное владение материалом: чётко и грамотно излагает содержание отчёта, уверенно использует профессиональную терминологию. Даёт исчерпывающие ответы на все вопросы, включая дополнительные, показывает глубокое понимание темы, способность к анализу и формулировке выводов. Речь структурирована, убедительна, при необходимости сопровождается иллюстративными материалами.

Оценка «зачтено-хорошо»: письменный отчёт в целом соответствует методическим указаниям, однако допущены 1–2 незначительных недочёта в оформлении (например, мелкие ошибки в приложениях). Содержание отражает ключевые аспекты практики, но анализ

некоторых вопросов проведён не в полной мере. Выводы корректны, но недостаточно развёрнуты. Используются необходимые первоисточники, однако их анализ не всегда глубок. Отчёт сдан своевременно или с незначительным опозданием, согласованным с руководителем. Во время защиты студент грамотно излагает суть отчёта, но допускает 1–2 небольшие речевые или логические ошибки. На основные вопросы отвечает правильно, однако при ответе на дополнительные вопросы испытывает небольшие затруднения. Показывает хорошее понимание темы, но не всегда способен глубоко проанализировать отдельные аспекты. Терминологию использует уверенно, но иногда допускает неточности.

Оценка «зачтено-удовлетворительно»: письменный отчёт имеет существенные недостатки в оформлении: ошибки в структуре, форматировании, оформлении приложений. Содержание неполное: упущены отдельные важные аспекты практики, индивидуальное задание выполнено поверхностно или не выполнено. Выводы формальные, не отражают сути работы. Первоисточники использованы фрагментарно, анализ поверхностный. Отчёт сдан с опозданием, не согласованным с руководителем. При защите студент излагает содержание отчёта неполно, сбивчиво, с существенными логическими или речевыми ошибками. Ответы на вопросы неполные, содержат ошибки; на дополнительные вопросы ответить не может. Понимание темы поверхностное, анализ формальный. Терминология используется ограниченно, с ошибками. Тем не менее, базовый уровень понимания материала и выполнения заданий прослеживается.

Оценка «не зачтено-неудовлетворительно»: письменный отчёт не соответствует методическим указаниям, содержит множественные ошибки в оформлении и содержании. Ключевые аспекты практики не раскрыты, анализ отсутствует, выводы формальны или отсутствуют. Первоисточники не использованы, приложения либо отсутствуют, либо не соответствуют теме. Отчёт не сдан в срок или представлен в неполном объёме. Возможны случаи плагиата или использования недостоверных данных. Во время защиты студент не может чётко изложить содержание отчёта: речь несвязная, содержит множественные ошибки. Не отвечает на поставленные вопросы или даёт ошибочные ответы. Не демонстрирует понимания темы, не способен анализировать и делать выводы. Профессиональную терминологию не использует или использует неправильно. Материал не освоен, практика фактически не выполнена или выполнена ненадлежащим образом.

4 Методические указания для обучающихся по освоению дисциплины

Для того чтобы практика была успешно пройдена, студент должен в полном объёме реализовать программу практики. Необходимо придерживаться утверждённого графика

посещения – присутствовать на практике все запланированные дни. Важно регулярно фиксировать в дневнике практики все выполняемые задачи и полученные результаты. Также студент обязан в установленные сроки подготовить и сдать всю необходимую отчётную документацию, подтвердив, тем самым, освоение требуемых профессиональных компетенций.

Комплект отчётной документации должен включать следующие документы:

- титульный лист дневника практики;
- заполненный дневник прохождения практики;
- отзыв руководителя практики от предприятия/организации;
- титульный лист отчёта;
- оглавление (содержание) отчёта;
- основной текст отчёта;
- приложения к отчёту.

Структура и содержание отчёта

Титульный лист должен содержать:

- полное название учебного заведения;
- наименование специальности;
- наименование практики;
- ФИО студента и руководителей практики;
- информацию о месте и сроках прохождения практики.

Содержание (оглавление) должно включать:

- перечень всех разделов и подразделов отчёта;
- указание страниц, на которых расположены соответствующие разделы;
- обеспечение точного соответствия заголовков в оглавлении и в основном

тексте отчёта.

Введение (объёмом 1 – 2 страницы) должно охватывать следующие аспекты:

- актуальность практики – объяснение значимости выбранного направления для будущей профессиональной деятельности, а также его соответствия современным трендам в отрасли;
- цели и задачи практики.

Основная часть (8 – 11 страниц) должна быть структурирована по этапам практики и включать:

раздел 1. Организационная структура предприятия:

- организационная структура предприятия;

- перечисление основных направлений деятельности предприятия;
- описание подразделения, в котором проходила практика;
- должностная инструкция сотрудника (одна – на выбор).

Раздел 2. Характеристика проделанной работы

- содержание выполненных заданий;
- методы и технологические подходы, которые применялись в работе;
- оборудование и ПО, использованные в ходе практики,

Раздел 3. Результаты практической деятельности

- разбор сложностей и проблемных ситуаций, с которыми столкнулся практикант;
- варианты решений выявленных проблем;
- сопоставление теоретических знаний с реальными рабочими задачами;
- примеры решения профессиональных задач на практике.
- основные достижения за время прохождения практики.

5. Заключение (объём: 1 – 1,5 страницы). В этом разделе необходимо отразить:

- итоги практики: оценить, были ли выполнены поставленные задачи, соответствуют ли результаты целям практики;
- наметить шаги для дальнейшего профессионального роста.

6. Библиографический список (используемые источники) Следует включить:

- нормативные акты и документы;
- учебные и научные публикации;
- внутренние регламенты и документы организации;
- интернет-ресурсы (если использовались);
- оформление списка источников – в соответствии с ГОСТ Р 7.0.100-2018.

7. Приложения. В раздел приложений можно добавить:

- копии рабочих документов, с которыми практикант работал в ходе практики;
- фото, схемы и диаграммы;
- примеры выполненных работ;
- скриншоты интерфейсов используемых программ;
- образцы и шаблоны рабочих документов.

Требования к приложениям к отчёту по практике:

Каждое приложение необходимо размещать на отдельной странице. Приложения должны иметь сквозную нумерацию (например: Приложение 1, Приложение 2, Приложение 3 и т. д.). В основном тексте отчёта должны присутствовать ссылки на приложения. Названия

приложений должны быть понятными, также требуется добавить краткие пояснения к каждому из них.

Объём отчёта:

- введение – 1 – 2 страницы;
- основная часть – 12 – 25 страниц;
- заключение — 1–1,5 страницы.

Общий объём отчёта (без учёта приложений) должен составлять 16 – 30 страниц.

Ключевые требования к отчёту:

отчёт должен:

- содержать логическую связь между разделами;
- включать конкретные утверждения с доказательствами;
- соответствовать программе практики;
- содержать практические примеры и аналитические материалы;
- быть грамотно оформлен и структурирован.

Требования к оформлению отчёта:

технические параметры:

- формат бумаги: А4 (210 × 297 мм), ориентация — книжная;
- шрифт: Times New Roman, размер — 14 pt;
- межстрочный интервал: 1,5;
- абзацный отступ: 1,25 см;
- выравнивание текста: по ширине;
- поля: левое — 30 мм, правое — 10 мм, верхнее и нижнее — по 20 мм.

Структурные требования:

- каждый раздел отчёта должен начинаться с новой страницы;
- заголовки разделов следует оформлять прописными буквами, используя полужирный шрифт;
- нумерация страниц должна быть сквозной, с использованием арабских цифр;
- таблицы и формулы должны иметь сквозную нумерацию по всему тексту отчёта;
- приложения следует располагать в порядке их упоминания в тексте отчёта.

Особые указания:

- список используемой литературы нужно оформлять в соответствии с ГОСТ Р 7.0.100–2018;
- не допускается использование произвольных сокращений;

- в тексте отчёта нельзя использовать цветные выделения и подчёркивания;
- математические знаки разрешается использовать только в сочетании с цифровыми значениями.

Порядок защиты отчёта по практике:

1. Студент выступает с кратким докладом (продолжительностью до 5 минут), в котором подводит итоги своей работы во время практики.

2. Студент отвечает на вопросы, касающиеся содержания отчёта по практике.

Студенты, которые не прошли практику, не будут допущены к промежуточной аттестации.

5. Учебная литература и ресурсы информационно-телекоммуникационной сети «Интернет»

Основная литература

1. Агальцов, В. П. Математические методы в программировании: учебник / В. П. Агальцов. — 2-е изд., перераб. и доп. — Москва: ФОРУМ : ИНФРА-М, 2023. — 240 с. — (Среднее профессиональное образование). - ISBN 978-5-8199-0410-7. - Текст: электронный. - URL: <https://znanium.ru/catalog/product/1896458> – Режим доступа: по подписке.
2. Емелина Е.И. Поддержка и тестирование программных модулей: учебник / Е.И. Емелина. – Москва: КНОРУС, 2024. – 272 с. – (Среднее профессиональное образование).
3. Колдаев, В. Д. Численные методы и программирование: учебное пособие / В.Д. Колдаев; под ред. Л.Г. Гагариной. — Москва: ФОРУМ : ИНФРА-М, 2025. — 336 с. — (Среднее профессиональное образование). - ISBN 978-5-8199-0779-5. - Текст: электронный. - URL: <https://znanium.ru/catalog/product/2139606> – Режим доступа: по подписке.
4. Лапчик М.П. Численные методы: учебное издание / Лапчик М.П., Рагулина М.И., Хеннер Е. К. - Москва: Академия, 2024. - 256 с. (Специальности среднего профессионального образования). - URL: <https://academia-moscow.ru> - Режим доступа: Электронная библиотека «Academiamoscow». - Текст: электронный
5. Рогачева О.А. Разработка программных модулей: учебное издание / Рогачева О.А. - Москва: Академия, 2024. - 272 с. (Профессии среднего профессионального образования). - URL: <https://academia-moscow.ru> - Режим доступа: Электронная библиотека «Academia-moscow». - Текст: электронный
6. Слабнов, В. Д. Численные методы и программирование: учебное пособие для СПО / В. Д. Слабнов. — 2-е изд., стер. — Санкт-Петербург: Лань, 2022. — 460 с. — ISBN 978-5-8114-9250-3. — Текст: электронный // Лань: электронно-библиотечная система. — URL: <https://e.lanbook.com/book/189402> — Режим доступа: для авториз. пользователей.
7. Федорова Г.Н. Осуществление интеграции программных модулей: учебное издание / Федорова Г.Н. - Москва: Академия, 2023. - 288 с. (Специальности среднего профессионального образования). - URL: <https://academia-moscow.ru> - Режим доступа: Электронная библиотека «Academia-moscow». - Текст: электронный
8. Федорова Г.Н. Разработка модулей программного обеспечения для компьютерных систем: учебное издание / Федорова Г.Н. - Москва: Академия, 2024. - 384 с. (Специальности среднего профессионального образования). - URL: <https://academia-moscow.ru> - Режим доступа: Электронная библиотека «Academia-moscow». - Текст: электронный

Дополнительная литература

1. ГОСТ 19.001-77. Государственный стандарт Союза ССР. Единая система программной документации. Общие положения (введен в действие Постановлением Госстандарта СССР от 20.05.1977 N 1268). - URL: <https://www.consultant.ru> - Режим доступа: Правовой сервер КонсультантПлюс. - Текст: электронный
2. ГОСТ 19.101-77. Государственный стандарт Союза ССР. Единая система программной документации. Виды программ и программных документов (введен Постановлением Госстандарта СССР от 20.05.1977 N 1268). - URL: <https://www.consultant.ru> - Режим доступа: Правовой сервер КонсультантПлюс. - Текст: электронный
3. ГОСТ 19.102-77. Государственный стандарт Союза ССР. Единая система программной документации. Стадии разработки (введен в действие Постановлением Госстандарта СССР от 20.05.1977 N 1268). - URL: <https://www.consultant.ru> - Режим доступа: Правовой сервер КонсультантПлюс. - Текст: электронный
4. ГОСТ 19.201-78. Государственный стандарт Союза ССР. Единая система программной документации. Техническое задание. Требования к содержанию и оформлению (введен в действие Постановлением Госстандарта СССР от 18.12.1978 N3351). - URL: <https://www.consultant.ru> - Режим доступа: Правовой сервер КонсультантПлюс. - Текст: электронный
5. ГОСТ 19.701-90. Единая система программной документации. Схемы алгоритмов, программ, данных и систем. Обозначения условные и правила выполнения (утв. Постановлением Госстандарта СССР от 26.12.1990 N 3294). - URL: <https://www.consultant.ru> - Режим доступа: Правовой сервер КонсультантПлюс. - Текст: электронный
6. ГОСТ Р ИСО/МЭК 25023-2021. Национальный стандарт Российской Федерации. Системная и программная инженерия. Требования и оценка качества систем и программной продукции (SQaRE). Измерения качества системы и программной продукции (утв. и введен в действие Приказом Росстандарта от 19.11.2021 N 1524-ст). - URL: <https://www.consultant.ru> - Режим доступа: Правовой сервер КонсультантПлюс. - Текст: электронный
7. Акопов, А. С. Имитационное моделирование: учебник и практикум для вузов / А. С. Акопов. — 2-е изд., перераб. и доп. — Москва: Издательство Юрайт, 2024. — 426 с. — (Высшее образование). — ISBN 978-5-534-18379-5. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/534885>

8. Згода Ю. Н. Проектирование программного обеспечения: учебно-методическое пособие / Ю. Н. Згода. – СПб.: Научно-технологические технологии, 2024. – 74 с. URL:<https://publishing.intelgr.com/archive/Proektirovanie-programmnogo-obespecheniya.pdf>. - Текст: электронный

9. Поколодина Е. В. Ревьюирование программных модулей: учебное издание / Поколодина Е. В., Долгова Н. А., Ананьев Д. В. - Москва: Академия, 2024. - 208 с. (Специальности среднего профессионального образования). - URL: <https://academia-moscow.ru> - Режим доступа: Электронная библиотека «Academia-moscow». - Текст: электронный

10. Библиотека профессионала №1 <https://profspo.ru/>

6 Материально-техническая база, информационные технологии, программное обеспечение и информационные справочные системы

Реализация образовательной программы предполагает обязательную учебную и производственную практику (по профилю специальности).

Производственная практика (по профилю специальности) для студентов проходит в ИТ-отделах, департаментах разработки ПО, центрах кибербезопасности или других подразделениях ИТ-компаний различного профиля. Эти подразделения обеспечивают погружение обучающихся в профессиональную сферу информационных технологий — от программирования и системного администрирования до анализа данных и обеспечения информационной безопасности.

Техническое оснащение рабочих мест во время производственной практики должно соответствовать реальным задачам профессиональной деятельности ИТ-специалиста. Это означает, что студенты должны работать с:

- современными средствами разработки (IDE, например, IntelliJ IDEA, Visual Studio Code);
- системами контроля версий (Git, SVN);
- базами данных (MySQL, PostgreSQL, MongoDB);
- инструментами DevOps и автоматизации (Docker, Kubernetes, Ansible);
- средствами обеспечения кибербезопасности (антивирусы, системы обнаружения вторжений IDS/IPS, криптографические инструменты);
- операционными системами (Windows, Linux, macOS) и виртуальными машинами.

Практика должна дать возможность студентам:

- закрепить навыки программирования на изучаемых языках (Python, Java, C++, JavaScript и др.);
- закрепить навыки работы с базами данных;
- освоить работу с реальными проектными задачами (разработка модулей ПО, тестирование, отладка кода);
- научиться взаимодействовать с API и веб-сервисами;
- познакомиться с процессами развёртывания и поддержки IT-инфраструктуры;
- развить навыки работы в команде (Scrum, Agile) и документирования проектов.

Допускается использование виртуальных лабораторий и облачных платформ (например, AWS, Google Cloud, Yandex Cloud) в качестве альтернативы физическому оборудованию. Это позволяет имитировать реальные рабочие сценарии — от развёртывания серверов до тестирования сетевых конфигураций — без необходимости наличия собственного дорогостоящего оборудования.

Федеральное государственное бюджетное образовательное учреждение
высшего образования

**РОССИЙСКАЯ АКАДЕМИЯ НАРОДНОГО ХОЗЯЙСТВА и
ГОСУДАРСТВЕННОЙ СЛУЖБЫ при ПРЕЗИДЕНТЕ РОССИЙСКОЙ
ФЕДЕРАЦИИ**

СЕВЕРО-ЗАПАДНЫЙ ИНСТИТУТ УПРАВЛЕНИЯ

Факультет среднего профессионального образования

ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ

по _____ практике
(индекс и наименование практики по уч. плану)

обучающегося _____ курса, учебной группы № _____

(фамилия, имя, отчество)

По специальности _____
(код и наименование)

Место прохождения практики _____
(указывается полное наименование профильной организации/структурного подразделения Академии)

Срок прохождения практики: с «___» _____ 202__ г. по «___» _____ 202__ г.

№ п\п	Содержание индивидуального задания	Планируемые результаты

Руководитель практики от филиала:

Руководитель практики от предприятия:

_____ И. О. Фамилия
_____ 2026

_____ И. О. Фамилия
_____ 2026

Задание получил: «___» _____ 202__ г. _____ И. О. Фамилия студента
дата подпись

Федеральное государственное бюджетное образовательное учреждение
высшего образования
**РОССИЙСКАЯ АКАДЕМИЯ НАРОДНОГО ХОЗЯЙСТВА и
ГОСУДАРСТВЕННОЙ СЛУЖБЫ при ПРЕЗИДЕНТЕ РОССИЙСКОЙ
ФЕДЕРАЦИИ**
СЕВЕРО-ЗАПАДНЫЙ ИНСТИТУТ УПРАВЛЕНИЯ

Факультет среднего профессионального образования

Специальность 09.02.11 Разработка и управление программным обеспечением

ОТЧЕТ

по производственной практике ПП.02.01

по ПМ.02 Разработка и интеграция модулей программного обеспечения

Студента __ курса группы _____
номер группы

Иванов Иван Иванович
(Ф.И.О. студента)

Наименование базы практики: _____
название предприятия/организации

Сроки прохождения практики: с «__» _____ 202__ г. по «__» _____ 202__ г.

Руководитель практики от предприятия: _____
(должность)
_____ С. С. Сидоров
(подпись)

Руководитель практики от ФСПО: _____ П. П. Петров
(подпись)

Санкт-Петербург 202__ год

Федеральное государственное бюджетное образовательное учреждение
высшего образования
**РОССИЙСКАЯ АКАДЕМИЯ НАРОДНОГО ХОЗЯЙСТВА и
ГОСУДАРСТВЕННОЙ СЛУЖБЫ при ПРЕЗИДЕНТЕ РОССИЙСКОЙ
ФЕДЕРАЦИИ**
СЕВЕРО-ЗАПАДНЫЙ ИНСТИТУТ УПРАВЛЕНИЯ

Факультет среднего профессионального образования

Специальность 09.02.11 Разработка и управление программным обеспечением

ДНЕВНИК
прохождения производственной практики ПП.02.01
по ПМ.02 Разработка и интеграция модулей программного обеспечения

Студента __ курса, группы _____
номер группы

Иванов Иван Иванович
(Ф.И.О. студента)

Наименование базы практики: _____
наименование предприятия и наименование структурного подразделения

Санкт-Петербург 202_ год

