

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Комарова Светлана Юриевна

Должность: Проректор по образовательной деятельности

Дата подписания: 30.07.2026 20:28:48

Уникальный программный ключ:

43ba42f5deae4116bbfcb9ac98e39108031227e81add207cbe41900987a

**Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Омский государственный аграрный университет
имени П.А. Столыпина»**

Университетский колледж агробизнеса

**ООП по специальности 09.02.11 Разработка и управление программным
обеспечением**

СОГЛАСОВАНО

Руководитель ООП

 Е.А. Поединок

« 17 » июня 2026 г.

УТВЕРЖДАЮ

Директор

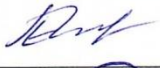
 А.П. Шевченко

« 17 » июня 2026 г.

РАБОЧАЯ ПРОГРАММА

ПП.02.01 Производственная практика (по профилю специальности)

ПМ.02 Разработка и интеграция модулей программного обеспечения

Выпускающее отделение	Инженерное отделение	
Разработчики РПУД (внутренние и внешние):		А.В. Кортусов
Внутренние эксперты:		
Заведующая методическим отделом		Г.А. Горелкина
Директор НСХБ		И.М. Демчукова
Омск, 2026		

СОДЕРЖАНИЕ

<u>1. ОБЩАЯ ХАРАКТЕРИСТИКА РАБОЧЕЙ ПРОГРАММЫ ПРОИЗВОДСТВЕННОЙ ПРАКТИКИ</u>	3
<u>2. СТРУКТУРА И СОДЕРЖАНИЕ ПРОИЗВОДСТВЕННОЙ ПРАКТИКИ</u>	5
<u>3. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОИЗВОДСТВЕННОЙ ПРАКТИКИ</u>	7
<u>4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ ПРОИЗВОДСТВЕННОЙ ПРАКТИКИ</u>	9

1. ОБЩАЯ ХАРАКТЕРИСТИКА РАБОЧЕЙ ПРОГРАММЫ ПРОИЗВОДСТВЕННОЙ ПРАКТИКИ ПП.02.01 Производственная практика

1.1 Цель и планируемые результаты освоения производственной практики

В результате производственной практики обучающийся должен освоить основной вид деятельности «Разработка и интеграция модулей программного обеспечения» и соответствующие ему общие и профессиональные компетенции.

1.1.1 Перечень общих компетенций

Код	Наименование общих компетенций
ОК 1	Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам
ОК 9	Пользоваться профессиональной документацией на государственном и иностранном языках

1.1.2. Перечень профессиональных компетенций

Код	Наименование видов деятельности и профессиональных компетенций
ПК 2.4	Выполнять тестирование и отладку программного обеспечения.
ПК 2.5	Осуществлять документирование программных модулей программного обеспечения.

1.1.3. В результате освоения профессионального модуля обучающийся должен:

Владеть навыками	Н 2.4.1	отладки программного обеспечения на уровне программных модулей
	Н 2.4.2	тестирования программного обеспечения и формирования тестовых сценариев подготовки тестовых платформ (установка операционной системы, дополнительного ПО и другого по необходимости)
	Н 2.4.3	настройки тестовой среды и аппаратных средств для выполнения тестирования ПО в соответствии с заданием на тестирование в пределах своей компетенции
	Н 2.4.4	формирования и представления отчетности о подготовке к выполнению задания на тестирование ПО в соответствии с документирования установленными регламентами
	Н 2.5.1	создания технической документации для модулей
	Н 2.5.2	кода, API и интерфейсов, работы со специализированным ПО по документированию программного кода
Уметь	У 2.4.1	анализировать требования к программному обеспечению и составлять планы тестирования
	У 2.4.2	выполнять тестирование программного обеспечения вручную и автоматизировать процесс тестирования, анализировать результаты тестирования и документировать найденные ошибки
	У 2.4.3	разрабатывать стратегии отладки и исправлять ошибки в программном обеспечении
	У 2.5.1	описывать функциональность модулей в документации, создавать диаграммы для иллюстрации работы модулей

	У 2.5.2	разбивать модули на логические блоки и описывать каждый блок отдельно включать в документацию особенности модулей, такие как ограничения, уязвимости или оптимальные настройки
	У 2.5.3	вести журнал изменений и фиксировать обновления программных модулей, проводить регулярное обновление документации при изменении модулей или добавлении нового функционала
Знать	З 2.4.1	принципы и методы тестирования программного обеспечения.
	З 2.4.2	инструменты для автоматизации тестирования
	З 2.4.3	основы разработки и отладки программного обеспечения на разных языках программирования
	З 2.4.4	виды и типы тестирования ПО, техники ручного тестирования и техники автоматизированного тестирования
	З 2.4.5	жизненный цикл дефекта ПО и принципы работы в системе контроля дефектов
	З 2.5.1	стандарты технической документации
	З 2.5.2	инструменты для создания технической документации и комментирования кода

1.2. Количество часов, отводимое на освоение программы производственной практики – 180 часов.

2. СТРУКТУРА И СОДЕРЖАНИЕ ПРОИЗВОДСТВЕННОЙ ПРАКТИКИ

Название этапа практики	Содержание выполняемых работ	Объем, акад. час.	Код ПК, ОК	Код Н/У/З
1	2	3	4	5
1.1 Организационный этап (в УКАБ)	1.Инструкция по прохождению практики и подготовки документации. Знакомство с программой практики. Целеполагание и планирование собственных действий.	2	ОК 1, ОК 9	Уо 01.01, Уо 01.02, Уо 09.01, Зо 01.01, Зо 01.02, Зо 09.01
1.2 Организационный этап (на рабочем месте)	2.Инструктаж по технике безопасности и противопожарным мероприятиям. Знакомство с рабочим местом.	2		
2 Основной этап	3.Разработка алгоритмов решения поставленных задач в соответствии с требованиями технического задания или других принятых в организации нормативных документов	10	ПК 2.4, ПК 2.5, ОК 1, ОК 9	Н 2.4.1, Н 2.4.2, Н 2.4.3, Н 2.4.4, Н 2.5.1, Н 2.5.2, У 2.4.1, У 2.4.2, У 2.4.3, У 2.5.1, У 2.5.2, У 2.5.3, З 2.4.1, З 2.4.2, З 2.4.3, З 2.4.4, З 2.4.5, З 2.5.1, З 2.5.2, Уо 01.01, Уо 01.02, Уо 09.01, Зо 01.01, Зо 01.02, Зо 09.01
	4.Применение стандартных алгоритмов в соответствующих областях	10		
	5.Создание программного кода в соответствии с техническим заданием (готовыми спецификациями)	10		
	6.Наличие соответствия результатов выполнения ожидаемым значениям	10		
	7.Установление соответствия разработанных тестов используемой методике тестирования	10		
	8.Проверка полноты группы тестов	10		
	9.Проверка отсутствия избыточной вложенности циклов	10		
	10.Проверка отсутствия дублирующих действий	10		
	11.Установление разумного баланса по используемой памяти и быстродействию	10		
	12.Выбор платформы разработки для среды выполнения	10		
	13.Создание программного кода в соответствии с техническим заданием	10		

	14.Тестирование приложений с использованием эмулятора	10		
	15.Тестирование приложения на физическом устройстве	10		
	16. Настройка резервного копирования данных, обрабатываемых программным модулем.	10		
	17. Разработка установщика программного обеспечения и сценариев развертывания	10		
	18. Составление инструкций по установке и эксплуатации разработанного модуля	6		
	19. Обеспечение безопасности хранения и передачи конфиденциальных данных	6		
	20. Сопровождение тестовой эксплуатации и сбор обратной связи от пользователей	4		
3 Заключительный этап	21. Оформление отчетной документации по производственной практике (по профилю специальности) 22. Защита отчета.	10	ОК 1, ОК 9	Уо 01.01, Уо 01.02, Уо 09.01, Зо 01.01, Зо 01.02, Зо 09.01
Итого		180		

3. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОИЗВОДСТВЕННОЙ ПРАКТИКИ

3.1. Для реализации программы профессионального модуля должны быть предусмотрены следующие специальные помещения:

Лаборатории компьютерных систем; программного обеспечения для компьютерных систем в соответствии с образовательной программой по специальности 09.02.11 Разработка и управление программным обеспечением.

Оснащенные базы практики в соответствии с образовательной программой по специальности 09.02.11 Разработка и управление программным обеспечением.

3.2. Информационное обеспечение реализации программы

Для реализации программы библиотечный фонд образовательной организации имеет печатные и/или электронные образовательные и информационные ресурсы, для использования в образовательном процессе.

3.2.1. Основные печатные издания

Печатных изданий нет

3.2.2. Основные электронные издания

1. Гагарина, Л. Г. Технология разработки программного обеспечения : учебное пособие / Л.Г. Гагарина, Е.В. Кокорева, Б.Д. Сидорова-Виснадул ; под ред. Л.Г. Гагариной. — Москва : ИНФРА-М, 2026. — 400 с. — (Среднее профессиональное образование). - Текст : электронный. - URL: <https://znanium.ru/catalog/product/2212387> – Режим доступа: по подписке.

2. Зайков, В. П. Организационные основы в управлении проектированием программных продуктов : учебное пособие / В. П. Зайков, Е. А. Шумков. — Краснодар : КубГТУ, 2024. — 239 с. — ISBN 978-5-8333-1324-4. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/478310> — Режим доступа: для авториз. пользователей.

3. Обоснование и разработка требований к программным системам : учебное пособие / А. А. Бирюкова, А. М. Володина, К. В. Гусев, А. Н. Миронов. — Москва : РТУ МИРЭА, 2022. — 157 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/240089> — Режим доступа: для авториз. пользователей.

3.2.3. Дополнительные источники

1. Операционные системы реального времени и технологии разработки кроссплатформенного программного обеспечения. Часть 4 : учебное пособие / Д.А. Беспалов, С. М. Гушанский, Н. М. Коробейникова, В. Е. Буглов ; Южный федеральный университет. - Ростов-на-Дону : Издательство Южного федерального университета, 2023. - 115 с. - ISBN 978-5-9275-4523-0. - Текст : электронный. - URL: <https://znanium.ru/catalog/product/2146701> – Режим доступа: по подписке.

2. Сонькин, М. А. Микропроцессорные системы. Разработка программного обеспечения для микроконтроллеров семейства AVR : учебное пособие / М. А. Сонькин, А. А. Шамин. - Москва ; Вологда : Инфра-Инженерия, 2023. - 96 с. - ISBN 978-5-9729-1211-7. - Текст : электронный. - URL: <https://znanium.ru/catalog/product/2095075> – Режим доступа: по подписке.

3. Федорова, Г. Н. Разработка, внедрение и адаптация программного обеспечения отраслевой направленности : учебное пособие / Г.Н. Федорова. — Москва : КУРС : ИНФРА-М, 2026. — 336 с. — (Среднее профессиональное образование). - ISBN

978-5-906818-41-6. - Текст : электронный. - URL: <https://znanium.ru/catalog/product/2213137>
– Режим доступа: по подписке.

4. Мякишев, Д. В. Разработка программного обеспечения АСУ ТП на основе объектно-ориентированного подхода : методическое пособие / Д. В. Мякишев. - 2-е изд. - Москва ; Вологда : Инфра-Инженерия, 2024. - 128 с. - ISBN 978-5-9729-2017-4. - Текст : электронный. - URL: <https://znanium.ru/catalog/product/2169721> – Режим доступа: по подписке.

5. Информационные технологии. – Москва : Новые технологии, 1995. – Выходит ежемесячно. – ISSN 1684-6400. – Текст : электронный. – URL: <https://eivis.ru/browse/publication/115066>.

6. Прикладная информатика. – Москва : Московский финансово-промышленный университет «Синергия», 2006. – . – Выходит 6 раз в год. – ISSN 1993-8314. – Текст : электронный. – URL: <https://eivis.ru/browse/publication/66410>.

7. Современные профессиональные базы данных (ИОС ОмГАУ-Moodle).

8. Справочная Правовая Система КонсультантПлюс.

9. Электронно-библиотечная система издательства «Лань».

10. Электронно-библиотечная система «Znanium».

11. Электронно-библиотечная система «Консультант студента».

12. Универсальная База Данных ИВИС: <https://eivis.ru/>

4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ ПРОИЗВОДСТВЕННОЙ ПРАКТИКИ

Код и наименование профессиональных и общих компетенций, формируемых в рамках модуля	Критерии оценки	Методы оценки
ОК 1 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам	- обоснованность планирования учебной и профессиональной деятельности; - соответствие результата выполнения профессиональных задач эталону (стандартам, образцам, алгоритму, условиям, требованиям или ожидаемому результату); - степень точности выполнения поставленных задач.	- решение и проверка ситуационных задач; - наблюдение и экспертная оценка преподавателя; - анализ отчетной документации.
ОК 9 Пользоваться профессиональной документацией на государственном и иностранном языках	- правильность чтения и понимания содержания профессиональной документации на государственном языке; - правильность чтения и понимания содержания профессиональной документации на иностранном языке; - точность перевода и толкования основных терминов и понятий на иностранном языке; - самостоятельность в поиске необходимой информации в документации на разных языках; - соблюдение правил оформления документов в соответствии с установленными требованиями; - полнота и точность составления профессиональной документации на государственном языке; - полнота и точность составления профессиональной документации на иностранном языке.	- решение и проверка ситуационных задач; - наблюдение и экспертная оценка преподавателя; - анализ отчетной документации.

ПК 2.4. Выполнять тестирование и отладку программного обеспечения.	Правильность проведения анализа требований к программному обеспечению и составление планов 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 документировать найденные ошибки; 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 составлять отчет о выполнении тестирования ПО	- решение и проверка ситуационных задач; - наблюдение и экспертная оценка преподавателя; - анализ отчетной документации.
ПК 2.5. Осуществлять документирование программных модулей программного обеспечения.	Правильность использования системы контроля версий; использование методов для получения кода с заданной функциональностью и степенью качества; выявление ошибок в системных компонентах на основе спецификаций.	- решение и проверка ситуационных задач; - наблюдение и экспертная оценка преподавателя; - анализ отчетной документации.

**Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Омский государственный аграрный университет
имени П.А. Столыпина»**

Университетский колледж агробизнеса

09.02.11 Разработка и управление программным обеспечением

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
по ПП.02.01 Производственная практика (по профилю специальности)**

Обеспечивающее
подразделение

преподавание

дисциплины

Инженерное отделение

Разработчик:

Преподаватель

А.В. Кортусов

**Омск
2026**

СОДЕРЖАНИЕ

1. ОБЩИЕ ПОЛОЖЕНИЯ	3
2. ОЖИДАЕМЫЕ РЕЗУЛЬТАТЫ ИЗУЧЕНИЯ	4
3. РАСПРЕДЕЛЕНИЕ ОЦЕНИВАНИЯ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ И ТИПОВ ОЦЕНОЧНЫХ МАТЕРИАЛОВ ПО ЭЛЕМЕНТАМ ЗНАНИЙ И УМЕНИЙ	6
4. ПОКАЗАТЕЛИ ОЦЕНКИ РЕЗУЛЬТАТОВ ОСВОЕНИЯ ПРОИЗВОДСТВЕННОЙ ПРАКТИКИ	7

1. ОБЩИЕ ПОЛОЖЕНИЯ

1. Фонд оценочных средств (далее – ФОС) предназначен для контроля и оценки образовательных достижений обучающихся, освоивших программу ПП.02.01 Производственная практика (по профилю специальности).
2. ФОС включает оценочные материалы для проведения текущего контроля и промежуточной аттестации в форме зачета.
3. ФОС позволяет оценивать знания, умения, направленные на формирование компетенций.
4. ФОС разработан на основании положений основной образовательной программы по специальности 09.02.11 Разработка и управление программным обеспечением и рабочей программы ПП.02.01 Производственная практика (по профилю специальности).
5. ФОС является обязательным обособленным приложением к рабочей программе.

2. ОЖИДАЕМЫЕ РЕЗУЛЬТАТЫ ИЗУЧЕНИЯ

Результаты обучения (освоенные умения, навыки)	Показатели оценки образовательных результатов
анализировать требования к программному обеспечению и составлять планы тестирования	Обучающийся умеет анализировать требования к программному обеспечению и составлять планы тестирования
выполнять тестирование программного обеспечения вручную и автоматизировать процесс тестирования, анализировать результаты тестирования и документировать найденные ошибки	Обучающийся умеет выполнять тестирование программного обеспечения вручную и автоматизировать процесс тестирования, анализировать результаты тестирования и документировать найденные ошибки
разрабатывать стратегии отладки и исправлять ошибки в программном обеспечении	Обучающийся умеет разрабатывать стратегии отладки и исправлять ошибки в программном обеспечении
описывать функциональность модулей в документации, создавать диаграммы для иллюстрации работы модулей	Обучающийся умеет описывать функциональность модулей в документации, создавать диаграммы для иллюстрации работы модулей
разбивать модули на логические блоки и описывать каждый блок отдельно включать в документацию особенности модулей, такие как ограничения, уязвимости или оптимальные настройки	Обучающийся умеет разбивать модули на логические блоки и описывать каждый блок отдельно включать в документацию особенности модулей, такие как ограничения, уязвимости или оптимальные настройки
вести журнал изменений и фиксировать обновления программных модулей, проводить регулярное обновление документации при изменении модулей или добавлении нового функционала	Обучающийся умеет вести журнал изменений и фиксировать обновления программных модулей, проводить регулярное обновление документации при изменении модулей или добавлении нового функционала
отладки программного обеспечения на уровне программных модулей	Обучающийся владеет навыками отладки программного обеспечения на уровне программных модулей
тестирования программного обеспечения и формирования тестовых сценариев подготовки тестовых платформ (установка операционной системы, дополнительного ПО и другого по необходимости)	Обучающийся владеет навыками тестирования программного обеспечения и формирования тестовых сценариев подготовки тестовых платформ (установка операционной системы, дополнительного ПО и другого по необходимости)
настройки тестовой среды и аппаратных средств для выполнения тестирования ПО в соответствии с заданием на тестирование в пределах своей компетенции	Обучающийся владеет навыками настройки тестовой среды и аппаратных средств для выполнения тестирования ПО в соответствии с заданием на тестирование в пределах своей компетенции
формирования и представления отчетности о подготовке к выполнению задания на тестирование ПО в соответствии с документирования установленными регламентами	Обучающийся владеет навыками формирования и представления отчетности о подготовке к выполнению задания на тестирование ПО в соответствии с документирования установленными регламентами
создания технической документации для модулей	Обучающийся владеет навыками создания технической документации для модулей

кода, API и интерфейсов, работы со специализированным ПО по документированию программного кода	Обучающийся владеет навыками кода, API и интерфейсов, работы со специализированным ПО по документированию программного кода
--	---

3. РАСПРЕДЕЛЕНИЕ ОЦЕНИВАНИЯ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ И ТИПОВ ОЦЕНОЧНЫХ МАТЕРИАЛОВ ПО ЭЛЕМЕНТАМ УМЕНИЙ И НАВЫКОВ

Содержание курса	Форма контроля	Умения	Навыки
Текущий контроль			
Раздел 1. Организационный этап			
Прохождение вводного инструктажа. Получение и обсуждение задания на практику	Проверка отчета по учебной практике	Уо 01.01, Уо 01.02, Уо 09.01	-
Раздел 2. Основной этап			
Выполнение работ:	наблюдение и оценка в процессе практики; анализ отчетной документации; экспертная оценка выполнения индивидуальных заданий.	У 2.4.1, У 2.4.2, У 2.4.3, У 2.5.1, У 2.5.2, У 2.5.3, Уо 01.01, Уо 01.02, Уо 09.01	Н 2.4.1, Н 2.4.2, Н 2.4.3, Н 2.4.4, Н 2.5.1, Н 2.5.2
Раздел 3. Заключительный этап			
Оформление введения. Оформление основной части. Оформление заключения. Оформление списка использованных источников и приложений. Оформление отчета и приложений. Прохождение собеседования (зачет)	Проверка отчета по учебной практике, собеседование	Уо 01.01, Уо 01.02, Уо 09.01	-
Промежуточный контроль			
Зачет	Проверка отчета по учебной практике, собеседование	У 2.4.1, У 2.4.2, У 2.4.3, У 2.5.1, У 2.5.2, У 2.5.3, Уо 01.01, Уо 01.02, Уо 09.01	Н 2.4.1, Н 2.4.2, Н 2.4.3, Н 2.4.4, Н 2.5.1, Н 2.5.2

4. МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ ДЛЯ ОЦЕНКИ УМЕНИЙ, НАВЫКОВ

4.1. Оценочные средства, применяемые для текущего контроля.

4.1.1. Разработка алгоритмов и программных модулей

Разработайте программный модуль для учета сотрудников предприятия на языке Python:

1. Создайте класс `Employee` с полями: `id`, `full_name`, `birth_date`, `position`, `salary`, `department`.
2. Реализуйте функции для выполнения CRUD-операций с использованием SQLite.
3. Разработайте модуль валидации данных (ФИО - минимум 2 слова, дата рождения - возраст от 16 до 65 лет, оклад - положительное число).
4. Реализуйте функцию поиска сотрудников по ФИО, должности и диапазону оклада.

Пример кода:

```
# models.py
from dataclasses import dataclass
from typing import Optional, List
import sqlite3
from datetime import datetime

@dataclass
class Employee:
    id: Optional[int] = None
    full_name: str = ""
    birth_date: str = ""
    position: str = ""
    salary: float = 0.0
    department: str = ""

class EmployeeValidator:
    @staticmethod
    def validate_full_name(name: str) -> bool:
        return len(name.strip().split()) >= 2

    @staticmethod
    def validate_birth_date(date: str) -> bool:
        try:
            age = (datetime.now() - datetime.strptime(date, '%Y-%m-%d')).days // 365
            return 16 <= age <= 65
        except:
            return False

    @staticmethod
    def validate_salary(salary: float) -> bool:
        return salary > 0

class EmployeeRepository:
    def __init__(self, db_path='employees.db'):
        self.db_path = db_path
        self._init_db()
```

```

def _init_db(self):
    with sqlite3.connect(self.db_path) as conn:
        conn.execute("""
            CREATE TABLE IF NOT EXISTS employees (
                id INTEGER PRIMARY KEY AUTOINCREMENT,
                full_name TEXT NOT NULL,
                birth_date TEXT NOT NULL,
                position TEXT NOT NULL,
                salary REAL NOT NULL,
                department TEXT
            )
        """)

def add_employee(self, emp: Employee) -> Optional[int]:
    if not all([
        EmployeeValidator.validate_full_name(emp.full_name),
        EmployeeValidator.validate_birth_date(emp.birth_date),
        EmployeeValidator.validate_salary(emp.salary)
    ]):
        return None

    with sqlite3.connect(self.db_path) as conn:
        cursor = conn.execute("""
            INSERT INTO employees (full_name, birth_date, position, salary, department)
            VALUES (?, ?, ?, ?, ?)
        """, (emp.full_name, emp.birth_date, emp.position, emp.salary, emp.department))
        return cursor.lastrowid

def get_employee(self, emp_id: int) -> Optional[Employee]:
    with sqlite3.connect(self.db_path) as conn:
        row = conn.execute('SELECT * FROM employees WHERE id = ?',
            (emp_id,)).fetchone()
        return Employee(*row) if row else None

def update_employee(self, emp: Employee) -> bool:
    with sqlite3.connect(self.db_path) as conn:
        cursor = conn.execute("""
            UPDATE employees SET full_name=?, birth_date=?, position=?, salary=?,
            department=?
            WHERE id=?
        """, (emp.full_name, emp.birth_date, emp.position, emp.salary, emp.department,
            emp.id))
        return cursor.rowcount > 0

def delete_employee(self, emp_id: int) -> bool:
    with sqlite3.connect(self.db_path) as conn:
        cursor = conn.execute('DELETE FROM employees WHERE id = ?', (emp_id,))
        return cursor.rowcount > 0

```

```

def search_employees(self, full_name=None, position=None, salary_from=None,
salary_to=None) -> List[Employee]:
    conditions, params = [], []
    if full_name:
        conditions.append("full_name LIKE ?")
        params.append(f"%{full_name}%")
    if position:
        conditions.append("position = ?")
        params.append(position)
    if salary_from:
        conditions.append("salary >= ?")
        params.append(salary_from)
    if salary_to:
        conditions.append("salary <= ?")
        params.append(salary_to)

    where = f"WHERE {' AND '.join(conditions)}" if conditions else ""
    with sqlite3.connect(self.db_path) as conn:
        rows = conn.execute(f"SELECT * FROM employees {where} ORDER BY
full_name", params).fetchall()
    return [Employee(*row) for row in rows]

```

Форма выполнения: Исходный код модуля, примеры вызова функций, скриншоты работы.

4.1.2. Отладка и тестирование программного модуля

Выполните отладку и тестирование разработанного модуля с использованием PyTest:

```

python
# test_module.py
import pytest
import tempfile
import os
from models import Employee, EmployeeRepository, EmployeeValidator

class TestEmployeeValidator:
    def test_valid_full_name(self):
        assert EmployeeValidator.validate_full_name("Иванов Иван") is True
        assert EmployeeValidator.validate_full_name("Иванов") is False

    def test_valid_birth_date(self):
        assert EmployeeValidator.validate_birth_date("1990-01-01") is True
        assert EmployeeValidator.validate_birth_date("2010-01-01") is False

    def test_valid_salary(self):
        assert EmployeeValidator.validate_salary(50000) is True
        assert EmployeeValidator.validate_salary(-1000) is False

class TestEmployeeRepository:
    @pytest.fixture
    def repo(self):
        fd, path = tempfile.mkstemp(suffix='.db')
        os.close(fd)

```

```
yield EmployeeRepository(path)
os.unlink(path)
```

```
def test_add_employee(self, repo):
    emp = Employee(full_name="Иванов Иван", birth_date="1990-01-01",
                    position="Программист", salary=50000)
    emp_id = repo.add_employee(emp)
    assert emp_id is not None
    assert repo.get_employee(emp_id).full_name == "Иванов Иван"

def test_add_invalid_employee(self, repo):
    emp = Employee(full_name="Иван", birth_date="2010-01-01",
                    position="", salary=-1000)
    assert repo.add_employee(emp) is None

def test_search_employees(self, repo):
    emp1 = Employee(full_name="Иванов Иван", birth_date="1990-01-01",
                    position="Программист", salary=50000)
    emp2 = Employee(full_name="Петров Петр", birth_date="1985-06-15",
                    position="Программист", salary=60000)
    repo.add_employee(emp1)
    repo.add_employee(emp2)

    results = repo.search_employees(position="Программист")
    assert len(results) == 2
    results = repo.search_employees(full_name="Иванов")
    assert len(results) == 1
```

Форма выполнения: Код тестов, отчет о результатах тестирования с описанием выявленных ошибок.

4.1.3. Оптимизация и рефакторинг кода

Выполните оптимизацию и рефакторинг программного кода:

До рефакторинга:

python

```
def process_employees():
    conn = sqlite3.connect('employees.db')
    cursor = conn.cursor()
    cursor.execute('SELECT * FROM employees')
    employees = cursor.fetchall()
    conn.close()

    for emp in employees:
        if emp[4] < 30000:
            bonus = emp[4] * 0.1
        elif emp[4] >= 30000 and emp[4] < 50000:
            bonus = emp[4] * 0.15
        elif emp[4] >= 50000 and emp[4] < 80000:
            bonus = emp[4] * 0.2
        else:
            bonus = emp[4] * 0.25
```

```

conn = sqlite3.connect('employees.db')
cursor = conn.cursor()
cursor.execute("UPDATE employees SET bonus = ? WHERE id = ?", (bonus, emp[0]))
conn.commit()
conn.close()

```

После рефакторинга:

python

```
class BonusCalculator:
```

```
    @staticmethod
```

```
    def calculate(salary: float) -> float:
```

```
        if salary < 30000: return salary * 0.1
```

```
        if salary < 50000: return salary * 0.15
```

```
        if salary < 80000: return salary * 0.2
```

```
        return salary * 0.25
```

```
def process_employees():
```

```
    with sqlite3.connect('employees.db') as conn:
```

```
        employees = conn.execute("SELECT * FROM employees").fetchall()
```

```
        for emp in employees:
```

```
            bonus = BonusCalculator.calculate(emp[4])
```

```
            conn.execute("UPDATE employees SET bonus = ? WHERE id = ?", (bonus, emp[0]))
```

```
        conn.commit()
```

Форма выполнения: Код до и после рефакторинга с пояснением выполненных изменений.

4.1.4. Интеграция модулей в приложение

Объедините все модули в единое приложение:

```
# main.py
```

```
import logging
```

```
from models import Employee, EmployeeRepository
```

```
logging.basicConfig(level=logging.INFO)
```

```
class EmployeeApp:
```

```
    def __init__(self, db_path='employees.db'):
```

```
        self.repo = EmployeeRepository(db_path)
```

```
    def run(self):
```

```
        while True:
```

```
            print("\n===== Управление сотрудниками =====")
```

```
            print("1. Добавить сотрудника")
```

```
            print("2. Поиск сотрудников")
```

```
            print("3. Показать всех сотрудников")
```

```
            print("4. Удалить сотрудника")
```

```
            print("0. Выход")
```

```
            choice = input("Выберите действие: ").strip()
```

```
            if choice == '1':
```

```
                emp = Employee(
```

```
                    full_name=input("ФИО: ").strip(),
```

```

        birth_date=input("Дата рождения (ГГГГ-ММ-ДД): ").strip(),
        position=input("Должность: ").strip(),
        salary=float(input("Оклад: ").strip()),
        department=input("Отдел: ").strip()
    )
    emp_id = self.repo.add_employee(emp)
    print(f"Сотрудник добавлен с ID={emp_id}" if emp_id else "Ошибка
добавления")

    elif choice == '2':
        results = self.repo.search_employees(
            full_name=input("ФИО (Enter для пропуска): ").strip() or None,
            position=input("Должность (Enter для пропуска): ").strip() or None
        )
        print(f"\nНайдено {len(results)} сотрудников:")
        for emp in results:
            print(f"{emp.id}: {emp.full_name} - {emp.position} - {emp.salary}")

    elif choice == '3':
        results = self.repo.search_employees()
        print(f"\nВсего {len(results)} сотрудников:")
        for emp in results:
            print(f"{emp.id}: {emp.full_name} - {emp.position} - {emp.salary} руб.")

    elif choice == '4':
        emp_id = int(input("Введите ID сотрудника: "))
        print("Сотрудник удален" if self.repo.delete_employee(emp_id) else "Сотрудник
не найден")

    elif choice == '0':
        print("До свидания!")
        break

if __name__ == '__main__':
    EmployeeApp().run()

```

Форма выполнения: Полностью работающее приложение, скриншоты работы всех функций.

4.2. Оценочные средства, применяемые для промежуточной аттестации по итогам изучения дисциплины

Зачет проводится по завершении производственной практики.

Промежуточная аттестация в форме зачета осуществляется по результатам сдачи отчета по практике и с учетом текущего контроля успеваемости при выполнении всех видов текущего контроля.

Обучающиеся, не выполнившие виды работ, предусмотренные рабочей программой; пропустившие более 50% практики без уважительной причины, не допускаются к зачету.

Промежуточная аттестация таких лиц проводится только после прохождения ими всех видов текущего контроля.

**V. ПОКАЗАТЕЛИ ОЦЕНКИ РЕЗУЛЬТАТОВ ОСВОЕНИЯ
ПРОИЗВОДСТВЕННОЙ ПРАКТИКИ**

Уровень сформированности компетенций	Оценка	Критерии оценивания по видам работ	
		тестирование (процент правильных ответов)	прочие виды работ по практике
Высокий	Отлично	90-100%	Обучающийся глубоко и прочно усвоил теоретический и освоил практический материал. Дает логичные и грамотные ответы. Демонстрирует знание не только основного, но и дополнительного материала, быстро ориентируется, отвечая на дополнительные вопросы. Свободно справляется с поставленными задачами, аргументировано и верно обосновывает принятые решения.
Повышенный	Хорошо	70-89%	Обучающийся твердо знает программный материал, грамотно и по существу излагает его. Не допускает существенных неточностей при ответах на вопросы, правильно применяет теоретические положения при решении практических задач, владеет навыками и приемами их выполнения.
Базовый	Удовлетворительно	50-69%	Обучающийся демонстрирует знания только основного материала, но не усвоил его детали, испытывает затруднения при решении практических задач. В ответах на поставленные вопросы допускает неточности. Дает определения понятий, искажающие их смысл. Нарушает последовательность изложения программного материала.
Не сформирована	Неудовлетворительно	0-49%	Обучающийся не знает, не выполняет или неправильно выполняет большую часть учебного материала. Допускает ошибки в формулировке определений, искажающие их смысл, беспорядочно и неуверенно излагает материал. Ответы на дополнительные вопросы отсутствуют. Не выполняет задания.

ЛИСТ РАССМОТРЕНИЙ И ОДОБРЕНИЙ

рабочей программы

ПП 02.01 Производственная практика (по профилю специальности)

в составе ООП 09.02.11 Разработка и управление программным обеспечением

1) <u>Рассмотрена и одобрена:</u>
а) На заседании предметно-цикловой методической комиссии протокол № 6 от 10.03.2026 г. Председатель ПЦМК  С.М. Нурбаева
б) На заседании методического совета протокол №4 от 21.04.2026 г. Председатель методического совета  М.В. Иваницкая
2) <u>Рассмотрена и одобрена внешним экспертом</u>
а) директор ООО «Сатори Партнер» А.Б. Мальцев