

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ЛУГАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ ВЛАДИМИРА ДАЛЯ»

Экономический институт
Кафедра экономической кибернетики и прикладной статистики
(наименование кафедры)

УТВЕРЖДАЮ
Директор ЭКОНОМИЧЕСКИЙ ИНСТИТУТ
(подпись) Тхор Е.С.
«1» 15 2025 года

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
по практике**

Производственная (преддипломная) практика

(наименование учебной дисциплины, практике)

38.03.05 Бизнес-информатика

(код и наименование направления подготовки (специальности))

«Информационная бизнес-аналитика»

(наименование профиля подготовки (специальности, магистерской программы); при отсутствии ставится прочерк)

Разработчик (разработчики):

доцент

(подпись)

Воронова А.Г.

ФОС рассмотрен и одобрен на заседании кафедры экономической кибернетики и прикладной статистики от «15» 02 2025 г., протокол № 25

Заведующий кафедрой экономической кибернетики и прикладной статистики

(подпись)

Велигура А.В.

Луганск 2025 г.

Комплект оценочных материалов по практике Производственная (преддипломная) практика

Задания закрытого типа

Задания закрытого типа на выбор правильного ответа

1. *Выберите один правильный ответ.*

Какое из следующих утверждений о микросервисах является правильным?

- А) Микросервисы всегда должны быть написаны на одном языке программирования.
- Б) Микросервисы могут быть развернуты и масштабированы независимо друг от друга.
- В) Микросервисы являются монолитной архитектурой, где все компоненты связаны.
- Г) Микросервисы не могут взаимодействовать между собой.

Правильный ответ: Б

Компетенции (индикаторы): УК-1.1; УК-1.2; УК-1.3; УК-1.4; УК-2.1; УК-2.2; УК-2.3; УК-3.1; УК-3.2; УК-3.3; УК-4.1; УК-4.2; УК-4.3; УК-6.1; УК-6.2; УК-6.3; УК-10.1; УК-10.2; УК-10.3; ОПК-1.1; ОПК-1.2; ОПК-1.3; ОПК-2.1; ОПК-2.2; ОПК-2.3; ОПК-3.1; ОПК-3.2; ОПК-3.3; ОПК-4.1; ОПК-4.2; ОПК-4.3; ОПК-5.1; ОПК-5.2; ОПК-6.1; ОПК-6.2; ОПК-6.3; ПК-1.1; ПК-1.2; ПК-1.3; ПК-2.1; ПК-2.2; ПК-2.3; ПК-3.1; ПК-3.2; ПК-3.3; ПК-4.1; ПК-4.2

2. *Выберите один правильный ответ.*

Какое из следующих утверждений о PostgreSQL является правильным?

- А) PostgreSQL не поддерживает транзакции.
- Б) PostgreSQL является реляционной базой данных с поддержкой расширенных типов данных.
- В) PostgreSQL не может быть использован для обработки больших объемов данных.
- Г) PostgreSQL является проприетарной базой данных и не имеет открытого исходного кода. Модель бизнес-процессов

Правильный ответ: Б

Компетенции (индикаторы): УК-1.1; УК-1.2; УК-1.3; УК-1.4; УК-2.1; УК-2.2; УК-2.3; УК-3.1; УК-3.2; УК-3.3; УК-4.1; УК-4.2; УК-4.3; УК-6.1; УК-6.2; УК-6.3; УК-10.1; УК-10.2; УК-10.3; ОПК-1.1; ОПК-1.2; ОПК-1.3; ОПК-2.1; ОПК-2.2; ОПК-2.3; ОПК-3.1; ОПК-3.2; ОПК-3.3; ОПК-4.1; ОПК-4.2; ОПК-4.3; ОПК-5.1; ОПК-5.2; ОПК-6.1; ОПК-6.2; ОПК-6.3; ПК-1.1; ПК-1.2; ПК-1.3; ПК-2.1; ПК-2.2; ПК-2.3; ПК-3.1; ПК-3.2; ПК-3.3; ПК-4.1; ПК-4.2

Задания закрытого типа на установление соответствия

1. *Прочитайте текст и установите соответствие определения и термина.
Соответствие этапов разработки программного продукта и их*

определений. Каждому элементу левого столбца соответствует только один элемент правого столбца.

Этапы разработки программного продукта		Определение
1)	Анализ требований	А) Процесс, в котором осуществляется сбор и анализ информации о потребностях пользователей.
2)	Проектирование	Б) Этап, на котором создается программный код и реализуются функциональные возможности.
3)	Разработка	В) Оценка и проверка работоспособности программного продукта перед его запуском.
4)	Тестирование	Г) Обеспечение функционирования продукта после его выхода на рынок, включая обновления и исправления.
5)	Внедрение	Д) Процесс создания архитектуры и дизайна программного обеспечения.
6)	Поддержка	Е) Установка и настройка программного продукта в рабочей среде пользователя.

Правильный ответ: 1 – А; 2 – Д; 3 – Б; 4 – В; 5 – Е; 6 - Г

Компетенции (индикаторы): УК-1.1; УК-1.2; УК-1.3; УК-1.4; УК-2.1; УК-2.2; УК-2.3; УК-3.1; УК-3.2; УК-3.3; УК-4.1; УК-4.2; УК-4.3; УК-6.1; УК-6.2; УК-6.3; УК-10.1; УК-10.2; УК-10.3; ОПК-1.1; ОПК-1.2; ОПК-1.3; ОПК-2.1; ОПК-2.2; ОПК-2.3; ОПК-3.1; ОПК-3.2; ОПК-3.3; ОПК-4.1; ОПК-4.2; ОПК-4.3; ОПК-5.1; ОПК-5.2; ОПК-6.1; ОПК-6.2; ОПК-6.3; ПК-1.1; ПК-1.2; ПК-1.3; ПК-2.1; ПК-2.2; ПК-2.3; ПК-3.1; ПК-3.2; ПК-3.3; ПК-4.1; ПК-4.2

Задания закрытого типа на установление правильной последовательности

1. Прочитайте текст и установите последовательность. Какие основные этапы включает в себя процесс разработки программного продукта и в каком порядке они должны быть реализованы? Запишите правильную последовательность букв слева направо.

А) Поддержка и сопровождение: продукт нуждается в обновлениях и исправлениях.

- Б) Тестирование: программный продукт тестируется на наличие ошибок и несоответствий требованиям.
- В) Анализ требований: на этом этапе собираются и анализируются требования пользователей и заинтересованных сторон.
- Г) Разработка: на этом этапе происходит написание кода.
- Д) Внедрение: продукт устанавливается на рабочие серверы и передается пользователям.
- Е) Проектирование: здесь создается архитектура системы и разрабатываются технические спецификации.

Правильный ответ: В,Е,Г,Б,Д,А

Компетенции (индикаторы): УК-1.1; УК-1.2; УК-1.3; УК-1.4; УК-2.1; УК-2.2; УК-2.3; УК-3.1; УК-3.2; УК-3.3; УК-4.1; УК-4.2; УК-4.3; УК-6.1; УК-6.2; УК-6.3; УК-10.1; УК-10.2; УК-10.3; ОПК-1.1; ОПК-1.2; ОПК-1.3; ОПК-2.1; ОПК-2.2; ОПК-2.3; ОПК-3.1; ОПК-3.2; ОПК-3.3; ОПК-4.1; ОПК-4.2; ОПК-4.3; ОПК-5.1; ОПК-5.2; ОПК-6.1; ОПК-6.2; ОПК-6.3; ПК-1.1; ПК-1.2; ПК-1.3; ПК-2.1; ПК-2.2; ПК-2.3; ПК-3.1; ПК-3.2; ПК-3.3; ПК-4.1; ПК-4.2

Задания открытого типа

Задания открытого типа на дополнение

1. *Напишите пропущенные слова (с маленькой буквы).* На каждом из этапов разработка программных продуктов важно учитывать требования _____(1), проводить тестирование, обеспечивать качественное проектирование и внедрение, а также заниматься дальнейшей _____(2).

Правильный ответ: (1) пользователей, (2) поддержкой

Компетенции (индикаторы): УК-1.1; УК-1.2; УК-1.3; УК-1.4; УК-2.1; УК-2.2; УК-2.3; УК-3.1; УК-3.2; УК-3.3; УК-4.1; УК-4.2; УК-4.3; УК-6.1; УК-6.2; УК-6.3; УК-10.1; УК-10.2; УК-10.3; ОПК-1.1; ОПК-1.2; ОПК-1.3; ОПК-2.1; ОПК-2.2; ОПК-2.3; ОПК-3.1; ОПК-3.2; ОПК-3.3; ОПК-4.1; ОПК-4.2; ОПК-4.3; ОПК-5.1; ОПК-5.2; ОПК-6.1; ОПК-6.2; ОПК-6.3; ПК-1.1; ПК-1.2; ПК-1.3; ПК-2.1; ПК-2.2; ПК-2.3; ПК-3.1; ПК-3.2; ПК-3.3; ПК-4.1; ПК-4.2

Задания открытого типа с развернутым ответом

1. *Прочитайте текст задания. Продумайте логику и полноту ответа. Запишите развернутый и обоснованный ответ.*

Задание: Разработать программное обеспечение для автоматизации процесса обработки заказов в интернет-магазине.

Время решения – 40 мин.

Ожидаемый результат

1. Анализ требований

Цель: Понять, какие функции и возможности необходимы системе.

- **Документация требований:** Создание документа с описанием функциональных и нефункциональных требований. Например, система должна поддерживать автоматическую обработку заказов, интеграцию с платежными системами и уведомления клиентов по электронной почте.
Пример: В ходе анализа было выявлено, что пользователи хотят получать уведомления о статусе заказа через SMS и электронную почту.

2. Проектирование архитектуры системы

Цель: Определить структуру системы и технологии, которые будут использоваться.

- **Выбор архитектуры:** Определение, будет ли система монолитной, микросервисной или основанной на облачных технологиях.
Пример: Выбор микросервисной архитектуры, где отдельные сервисы отвечают за обработку заказов, управление запасами и уведомления.

3. Разработка прототипа

Цель: Создать минимально жизнеспособный продукт (MVP), чтобы протестировать основные функции.

- **Функциональность:** Реализация базовых функций, таких как создание заказа, просмотр статуса и управление корзиной.
Пример: Прототип включает в себя веб-интерфейс, где пользователи могут добавлять товары в корзину, оформлять заказ и просматривать историю заказов.

4. Разработка и тестирование

Цель: Написание кода и проверка его работоспособности.

- **Кодирование:** Реализация функциональности с использованием выбранных технологий (например, JavaScript для фронтенда и Python для бэкенда).
- **Тестирование:** Проведение юнит-тестов, интеграционных тестов и функциональных тестов для проверки работоспособности системы.
Пример: Разработка модуля для обработки платежей, который проходит тестирование с использованием различных сценариев, таких как успешные и неуспешные транзакции.

5. Внедрение

Цель: Перенос системы в рабочую среду и ее запуск.

- **Подготовка инфраструктуры:** Настройка серверов, баз данных и облачных сервисов.
- **Развертывание:** Перенос кода на продакшн-сервер и настройка системы.
Пример: Внедрение системы на облачном хостинге, таком как AWS или Azure, с использованием контейнеров Docker для упрощения развертывания.

6. Обучение пользователей и поддержка

Цель: Обучить сотрудников и пользователей работать с новой системой.

- **Создание документации:** Подготовка руководств и обучающих материалов.
- **Тренинги:** Проведение обучающих сессий для сотрудников и пользователей.

Пример: Организация вебинаров для сотрудников службы поддержки, чтобы они могли эффективно отвечать на вопросы клиентов о новом процессе обработки заказов.

7. Поддержка и обновление

Цель: Обеспечить бесперебойную работу системы и вносить улучшения.

- Мониторинг производительности: Использование инструментов для отслеживания производительности системы и выявления проблем.
- Обновления: Регулярное добавление новых функций и исправление ошибок на основе отзывов пользователей.

Пример: Внедрение функции автоматического уведомления клиентов о статусе заказа через SMS, основанной на отзывах пользователей о предпочтениях.

Критерии оценивания: наличие в ответе указания основных этапов разработки: анализ требований, проектирование архитектуры системы (с обоснованием выбора типа), функциональность (с указанием хотя бы двух функций решения), разработка (с указанием выбранных технологий), тестирование (указание необходимости проведения тестов), внедрение (указание необходимости перенос системы в рабочую среду и ее запуск), обучение пользователей и поддержка

Компетенции (индикаторы): УК-1.1; УК-1.2; УК-1.3; УК-1.4; УК-2.1; УК-2.2; УК-2.3; УК-3.1; УК-3.2; УК-3.3; УК-4.1; УК-4.2; УК-4.3; УК-6.1; УК-6.2; УК-6.3; УК-10.1; УК-10.2; УК-10.3; ОПК-1.1; ОПК-1.2; ОПК-1.3; ОПК-2.1; ОПК-2.2; ОПК-2.3; ОПК-3.1; ОПК-3.2; ОПК-3.3; ОПК-4.1; ОПК-4.2; ОПК-4.3; ОПК-5.1; ОПК-5.2; ОПК-6.1; ОПК-6.2; ОПК-6.3; ПК-1.1; ПК-1.2; ПК-1.3; ПК-2.1; ПК-2.2; ПК-2.3; ПК-3.1; ПК-3.2; ПК-3.3; ПК-4.1; ПК-4.2

2. Прочитайте текст задания. Продумайте логику и полноту ответа. Запишите развернутый и обоснованный ответ.

Задание: Разработать и внедрить систему управления проектами для команды разработки программного обеспечения.

Время решения – 40 мин.

Ожидаемый результат

1. Анализ требований

Цель: Определить потребности команды и функциональные требования к системе.

- Документация требований: Создание документа, который будет содержать как функциональные, так и нефункциональные требования. Например, система должна поддерживать создание задач, отслеживание времени, управление ресурсами и интеграцию с системами контроля версий.

Пример: В ходе анализа выясняется, что команда хочет иметь возможность отслеживать прогресс задач в реальном времени и получать уведомления о сроках выполнения.

2. Проектирование системы

Цель: Определить архитектуру системы и ее компоненты.

- Выбор архитектуры: Решение о том, будет ли система монолитной или микросервисной. Определение технологий, которые будут использоваться (языки программирования, базы данных, фреймворки).
- Создание прототипов: Разработка макетов пользовательского интерфейса (UI) для визуализации взаимодействия пользователей с системой.

Пример: Выбор архитектуры на основе микросервисов с использованием React для фронтенда и Node.js для бэкенда, а также PostgreSQL в качестве базы данных.

3. Разработка прототипа

Цель: Создать минимально жизнеспособный продукт (MVP) для тестирования основных функций.

- Создание интерфейса: Разработка пользовательского интерфейса, который включает в себя основные функции, такие как создание и редактирование задач, назначение ответственных и отслеживание статуса.
- Функциональность: Реализация базовых функций, таких как создание проектов и задач, назначение исполнителей и установка сроков.

Пример: Прототип включает в себя страницу для создания задач с полями для ввода названия, описания, сроков и выбора исполнителя.

4. Разработка и тестирование

Цель: Написание кода и проверка его работоспособности.

- Кодирование: Реализация всех функций системы в соответствии с требованиями и проектом.
- Тестирование: Проведение юнит-тестов, интеграционных тестов и функциональных тестов для проверки работоспособности системы. Важно также проводить тестирование с участием конечных пользователей для получения обратной связи.

Пример: Разработка модуля для отслеживания времени, который позволяет пользователям регистрировать время, затраченное на выполнение задач. Проведение тестов для проверки корректности расчетов.

5. Внедрение

Цель: Перенос системы в рабочую среду и ее запуск.

- Подготовка инфраструктуры: Настройка серверов, баз данных и облачных сервисов для развертывания системы.
- Развертывание: Перенос кода на продакшн-сервер и настройка системы, включая конфигурацию баз данных и интеграцию с другими системами.

Пример: Развертывание системы на облачном хостинге, таком как AWS, с использованием контейнеров Docker для упрощения развертывания и управления зависимостями.

6. Обучение пользователей

Цель: Обучить сотрудников и пользователей работать с новой системой.

- Создание документации: Подготовка руководств пользователя и технической документации для администраторов системы.
- Тренинги: Проведение обучающих сессий для команды, чтобы они могли эффективно использовать новую систему управления проектами.

Пример: Организация семинаров и вебинаров для команды разработки, где демонстрируется использование системы и ее основные функции.

7. Поддержка и обновление

Цель: Обеспечить бесперебойную работу системы и вносить улучшения.

- Мониторинг производительности: Использование инструментов мониторинга для отслеживания производительности системы и выявления проблем.
- Обновления: Регулярное добавление новых функций и исправление ошибок на основе отзывов пользователей.

Пример: Внедрение функции автоматического уведомления о приближающихся сроках выполнения задач на основе обратной связи от команды.

Критерии оценивания: наличие в ответе указания основных этапов разработки: анализ требований, проектирование архитектуры системы (с обоснованием выбора типа), функциональность (с указанием хотя бы двух функций решения), разработка (с указанием выбранных технологий), тестирование (указание необходимости проведения тестов), внедрение (указание необходимости перенос системы в рабочую среду и ее запуск), обучение пользователей и поддержка

Компетенции (индикаторы): УК-1.1; УК-1.2; УК-1.3; УК-1.4; УК-2.1; УК-2.2; УК-2.3; УК-3.1; УК-3.2; УК-3.3; УК-4.1; УК-4.2; УК-4.3; УК-6.1; УК-6.2; УК-6.3; УК-10.1; УК-10.2; УК-10.3; ОПК-1.1; ОПК-1.2; ОПК-1.3; ОПК-2.1; ОПК-2.2; ОПК-2.3; ОПК-3.1; ОПК-3.2; ОПК-3.3; ОПК-4.1; ОПК-4.2; ОПК-4.3; ОПК-5.1; ОПК-5.2; ОПК-6.1; ОПК-6.2; ОПК-6.3; ПК-1.1; ПК-1.2; ПК-1.3; ПК-2.1; ПК-2.2; ПК-2.3; ПК-3.1; ПК-3.2; ПК-3.3; ПК-4.1; ПК-4.2

3. Прочитайте текст задания. Продумайте логику и полноту ответа. Запишите развернутый и обоснованный ответ.

Задание: Создать CRM-систему для улучшения взаимодействия с клиентами в компании, занимающейся продажами.

Время решения – 40 мин.

Ожидаемый результат

1. Анализ требований

Цель: Определение потребностей бизнеса и функциональных требований к CRM-системе.

- Важно понять, какие проблемы существуют в текущих процессах взаимодействия с клиентами и какие функции должны быть включены в систему.
- Документирование требований: Создание документа, содержащего функциональные требования (например, управление контактами, отслеживание продаж, автоматизация маркетинга) и нефункциональные требования (например, безопасность данных, производительность).
Пример: В ходе анализа выясняется, что команда хочет иметь возможность отслеживать историю взаимодействия с клиентами, управлять лидами и автоматизировать процессы отправки коммерческих предложений.

2. Проектирование системы

Цель: Определение архитектуры системы и ее компонентов.

- Выбор архитектуры: Решение о том, будет ли система монолитной или микросервисной. Определение технологий, которые будут использоваться (языки программирования, базы данных, фреймворки).
- Создание прототипов: Разработка макетов пользовательского интерфейса (UI) и проектирование пользовательского опыта (UX) для визуализации взаимодействия пользователей с системой.

Пример: Выбор архитектуры на основе микросервисов с использованием Angular для фронтенда и Django для бэкенда, а также PostgreSQL в качестве базы данных.

3. Разработка прототипа

Цель: Создание минимально жизнеспособного продукта (MVP) для тестирования основных функций.

- Создание интерфейса: Разработка пользовательского интерфейса, который включает в себя основные функции, такие как управление контактами, создание сделок и отслеживание взаимодействия с клиентами.
- Функциональность: Реализация базовых функций, таких как создание карточек клиентов, добавление комментариев и назначение ответственных менеджеров.

Пример: Прототип включает в себя страницу для управления контактами, где пользователи могут добавлять новые контакты, редактировать существующие и просматривать историю взаимодействий.

4. Разработка и тестирование

Цель: Написание кода и проверка его работоспособности.

- Кодирование: Реализация всех функций системы в соответствии с требованиями и проектом.
- Тестирование: Проведение юнит-тестов, интеграционных тестов и функциональных тестов для проверки работоспособности системы. Важно также проводить тестирование с участием конечных пользователей для получения обратной связи.

Пример: Разработка модуля для отслеживания сделок, который позволяет пользователям регистрировать стадии продаж, а также проводить анализ конверсии лидов в клиентов.

5. Внедрение

Цель: Перенос системы в рабочую среду и ее запуск.

- Подготовка инфраструктуры: Настройка серверов, баз данных и облачных сервисов для развертывания системы.
- Развертывание: Перенос кода на продакшн-сервер и настройка системы, включая конфигурацию баз данных и интеграцию с другими системами (например, почтовыми сервисами и платформами для автоматизации маркетинга).

Пример: Развертывание системы на облачном хостинге, таком как AWS или Azure, с использованием контейнеров Docker для упрощения развертывания и управления зависимостями.

6. Обучение пользователей

Цель: Обучение сотрудников и пользователей работать с новой системой.

- Создание документации: Подготовка руководств пользователя и технической документации для администраторов системы.
- Тренинги: Проведение обучающих сессий для команды, чтобы они могли эффективно использовать новую систему управления взаимоотношениями с клиентами.

Пример: Организация семинаров и вебинаров для команды продаж, где демонстрируется использование системы и ее основные функции, а также ответы на вопросы пользователей.

7. Поддержка и обновление

Цель: Обеспечение бесперебойной работы системы и вносить улучшения.

- Мониторинг производительности: Использование инструментов мониторинга для отслеживания производительности системы и выявления проблем.
- Обновления: Регулярное добавление новых функций и исправление ошибок на основе отзывов пользователей.

Пример: Внедрение функции автоматической генерации отчетов по продажам на основе запросов от менеджеров, а также регулярное обновление системы для улучшения пользовательского опыта.

Критерии оценивания: наличие в ответе указания основных этапов разработки: анализ требований, проектирование архитектуры системы (с обоснованием выбора типа), функциональность (с указанием хотя бы двух функций решения), разработка (с указанием выбранных технологий), тестирование (указание необходимости проведения тестов), внедрение (указание необходимости перенос системы в рабочую среду и ее запуск), обучение пользователей и поддержка

Компетенции (индикаторы): УК-1.1; УК-1.2; УК-1.3; УК-1.4; УК-2.1; УК-2.2; УК-2.3; УК-3.1; УК-3.2; УК-3.3; УК-4.1; УК-4.2; УК-4.3; УК-6.1; УК-6.2; УК-6.3; УК-10.1; УК-10.2; УК-10.3; ОПК-1.1; ОПК-1.2; ОПК-1.3; ОПК-2.1; ОПК-2.2; ОПК-2.3; ОПК-3.1; ОПК-3.2; ОПК-3.3; ОПК-4.1; ОПК-4.2; ОПК-4.3; ОПК-5.1; ОПК-5.2; ОПК-6.1; ОПК-6.2; ОПК-6.3; ПК-1.1; ПК-1.2; ПК-1.3; ПК-2.1; ПК-2.2; ПК-2.3; ПК-3.1; ПК-3.2; ПК-3.3; ПК-4.1; ПК-4.2

4. Прочитайте текст задания. Продумайте логику и полноту ответа. Запишите развернутый и обоснованный ответ.

Задание: Разработать систему управления запасами для склада, которая позволит отслеживать наличие товаров и оптимизировать заказы.

Время решения – 40 мин.

Ожидаемый результат

1. Анализ требований

Цель: Определить потребности бизнеса и функциональные требования к системе управления запасами.

- Сбор информации: Важно понять текущие процессы управления запасами и выявить их недостатки.
- Документирование требований: Создание документа, содержащего функциональные требования (например, отслеживание уровня запасов, управление поставками и заказами) и нефункциональные требования (например, безопасность, производительность).

Пример: В ходе анализа выясняется, что команда нуждается в функции автоматического уведомления о низком уровне запасов, а также в возможности интеграции с существующей системой учета.

2. Проектирование системы

Цель: Определить архитектуру системы и ее компоненты.

- Выбор архитектуры: Решение о том, будет ли система монолитной или микросервисной. Определение технологий, которые будут использоваться (языки программирования, базы данных, фреймворки).
- Создание прототипов: Разработка макетов пользовательского интерфейса (UI) и проектирование пользовательского опыта (UX) для визуализации взаимодействия пользователей с системой.

Пример: Выбор архитектуры на основе микросервисов с использованием React для фронтенда и Node.js для бэкенда, а также PostgreSQL в качестве базы данных.

3. Разработка прототипа

Цель: Создание минимально жизнеспособного продукта (MVP) для тестирования основных функций.

- Создание интерфейса: Разработка пользовательского интерфейса, который включает в себя основные функции, такие как управление запасами, создание заказов на пополнение и отслеживание поставок.

- **Функциональность:** Реализация базовых функций, таких как добавление новых товаров, редактирование информации о запасах и отображение текущих уровней запасов.

Пример: Прототип включает в себя страницу для управления запасами, где пользователи могут добавлять новые товары, редактировать существующие и просматривать историю перемещений товаров на складе.

4. Разработка и тестирование

Цель: Написание кода и проверка его работоспособности.

- **Кодирование:** Реализация всех функций системы в соответствии с требованиями и проектом.
- **Тестирование:** Проведение юнит-тестов, интеграционных тестов и функциональных тестов для проверки работоспособности системы. Важно также проводить тестирование с участием конечных пользователей для получения обратной связи.

Пример: Разработка модуля для управления заказами, который позволяет пользователям создавать заказы на пополнение запасов и отслеживать их статус.

5. Внедрение

Цель: Перенос системы в рабочую среду и ее запуск.

- **Подготовка инфраструктуры:** Настройка серверов, баз данных и облачных сервисов для развертывания системы.
- **Развертывание:** Перенос кода на продакшн-сервер и настройка системы, включая конфигурацию баз данных и интеграцию с другими системами (например, системами учета и ERP).

Пример: Развертывание системы на облачном хостинге, таком как AWS, с использованием контейнеров Docker для упрощения развертывания и управления зависимостями.

6. Обучение пользователей

Цель: Обучение сотрудников и пользователей работать с новой системой.

- **Создание документации:** Подготовка руководств пользователя и технической документации для администраторов системы.
- **Тренинги:** Проведение обучающих сессий для команды, чтобы они могли эффективно использовать новую систему управления запасами.

Пример: Организация семинаров и вебинаров для сотрудников склада, где демонстрируется использование системы и ее основные функции, а также ответы на вопросы пользователей.

7. Поддержка и обновление

Цель: Обеспечение бесперебойной работы системы и вносить улучшения.

- **Мониторинг производительности:** Использование инструментов мониторинга для отслеживания производительности системы и выявления проблем.

- Обновления: Регулярное добавление новых функций и исправление ошибок на основе отзывов пользователей.

Пример: Внедрение функции автоматического прогнозирования потребностей в запасах на основе исторических данных о продажах и текущих трендах, а также регулярное обновление системы для улучшения пользовательского опыта.

Критерии оценивания: наличие в ответе указания основных этапов разработки: анализ требований, проектирование архитектуры системы (с обоснованием выбора типа), функциональность (с указанием хотя бы двух функций решения), разработка (с указанием выбранных технологий), тестирование (указание необходимости проведения тестов), внедрение (указание необходимости перенос системы в рабочую среду и ее запуск), обучение пользователей и поддержка

Компетенции (индикаторы): УК-1.1; УК-1.2; УК-1.3; УК-1.4; УК-2.1; УК-2.2; УК-2.3; УК-3.1; УК-3.2; УК-3.3; УК-4.1; УК-4.2; УК-4.3; УК-6.1; УК-6.2; УК-6.3; УК-10.1; УК-10.2; УК-10.3; ОПК-1.1; ОПК-1.2; ОПК-1.3; ОПК-2.1; ОПК-2.2; ОПК-2.3; ОПК-3.1; ОПК-3.2; ОПК-3.3; ОПК-4.1; ОПК-4.2; ОПК-4.3; ОПК-5.1; ОПК-5.2; ОПК-6.1; ОПК-6.2; ОПК-6.3; ПК-1.1; ПК-1.2; ПК-1.3; ПК-2.1; ПК-2.2; ПК-2.3; ПК-3.1; ПК-3.2; ПК-3.3; ПК-4.1; ПК-4.2

5. Прочитайте текст задания. Продумайте логику и полноту ответа. Запишите развернутый и обоснованный ответ.

Задание: Разработать систему для автоматизации документооборота в компании, чтобы сократить время на обработку документов.

Время решения – 40 мин.

Ожидаемый результат

1. Анализ требований

Цель: Определить потребности бизнеса и функциональные требования к системе автоматизации документооборота.

- Сбор информации: Выявление текущих проблем и потребностей.
- Документирование требований: Создание спецификации требований, которая включает функциональные (например, создание, редактирование, хранение и поиск документов) и нефункциональные требования (например, безопасность, производительность, интеграция с другими системами).

Пример: Выясняется, что сотрудники часто теряют документы, и требуется функция централизованного хранения, а также возможность отслеживания статуса документов на всех этапах их обработки.

2. Проектирование системы

Цель: Определить архитектуру системы и ее компоненты.

- Выбор архитектуры: Определение, будет ли система монолитной или основанной на микросервисах, выбор технологий (языки программирования, базы данных, фреймворки).
- Создание прототипов: Разработка макетов пользовательского интерфейса (UI) и проектирование пользовательского опыта (UX) для визуализации взаимодействия пользователей с системой.

Пример: Решение использовать микросервисную архитектуру с использованием Angular для фронтенда и Python (Django) для бэкенда, а также PostgreSQL для хранения данных.

3. Разработка прототипа

Цель: Создание минимально жизнеспособного продукта (MVP) для тестирования основных функций.

- Создание интерфейса: Разработка пользовательского интерфейса, который включает основные функции, такие как загрузка документов, создание шаблонов и управление версиями.
- Функциональность: Реализация базовых функций, таких как создание и редактирование документов, а также их хранение в централизованной базе данных.

Пример: Прототип включает в себя интерфейс для загрузки документов, где пользователи могут добавлять файлы, указывать метаданные (например, тип документа, дата создания) и назначать ответственных за обработку.

4. Разработка и тестирование

Цель: Написание кода и проверка его работоспособности.

- Кодирование: Реализация всех функций системы в соответствии с требованиями и проектом.
- Тестирование: Проведение юнит-тестов, интеграционных тестов и функциональных тестов для проверки работоспособности системы. Обязательно проводить тестирование с участием конечных пользователей для получения обратной связи.

Пример: Разработка модуля для отслеживания статуса документов, который позволяет пользователям видеть, на каком этапе обработки находится каждый документ, и получать уведомления о необходимости действий.

5. Внедрение

Цель: Перенос системы в рабочую среду и ее запуск.

- Подготовка инфраструктуры: Настройка серверов, баз данных и облачных сервисов для развертывания системы.
- Развертывание: Перенос кода на продакшн-сервер и настройка системы, включая конфигурацию баз данных и интеграцию с другими системами (например, ERP, CRM).

Пример: Развертывание системы на облачном хостинге, таком как AWS или Azure, с использованием контейнеров для упрощения управления зависимостями и масштабируемости.

6. Обучение пользователей

Цель: Обучение сотрудников и пользователей работать с новой системой.

- Создание документации: Подготовка руководств пользователя и технической документации для администраторов системы.
- Тренинги: Проведение обучающих сессий для команды, чтобы они могли эффективно использовать новую систему автоматизации документооборота.

Пример: Организация семинаров и вебинаров для сотрудников, где демонстрируется использование системы, ее основные функции, а также ответы на вопросы пользователей.

7. Поддержка и обновление

Цель: Обеспечение бесперебойной работы системы и внесение улучшений.

- Мониторинг производительности: Использование инструментов мониторинга для отслеживания производительности системы и выявления проблем.
- Обновления: Регулярное добавление новых функций и исправление ошибок на основе отзывов пользователей.

Пример: Внедрение функции автоматического создания отчетов по обработке документов и регулярное обновление системы для улучшения пользовательского опыта.

Критерии оценивания: наличие в ответе указания основных этапов разработки: анализ требований, проектирование архитектуры системы (с обоснованием выбора типа), функциональность (с указанием хотя бы двух функций решения), разработка (с указанием выбранных технологий), тестирование (указание необходимости проведения тестов), внедрение (указание необходимости перенос системы в рабочую среду и ее запуск), обучение пользователей и поддержка

Компетенции (индикаторы): УК-1.1; УК-1.2; УК-1.3; УК-1.4; УК-2.1; УК-2.2; УК-2.3; УК-3.1; УК-3.2; УК-3.3; УК-4.1; УК-4.2; УК-4.3; УК-6.1; УК-6.2; УК-6.3; УК-10.1; УК-10.2; УК-10.3; ОПК-1.1; ОПК-1.2; ОПК-1.3; ОПК-2.1; ОПК-2.2; ОПК-2.3; ОПК-3.1; ОПК-3.2; ОПК-3.3; ОПК-4.1; ОПК-4.2; ОПК-4.3; ОПК-5.1; ОПК-5.2; ОПК-6.1; ОПК-6.2; ОПК-6.3; ПК-1.1; ПК-1.2; ПК-1.3; ПК-2.1; ПК-2.2; ПК-2.3; ПК-3.1; ПК-3.2; ПК-3.3; ПК-4.1; ПК-4.2

Экспертное заключение

Представленный фонд оценочных средств (далее – ФОС) по «Производственная (преддипломная) практика» соответствует требованиям ФГОС ВО.

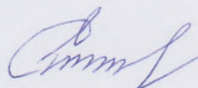
Предлагаемые формы и средства текущего и промежуточного контроля адекватны целям и задачам реализации основной профессиональной образовательной программы по направлению подготовки 38.03.05 Бизнес-информатика

Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины представлены в полном объеме.

Виды оценочных средств, включенные в представленный фонд, отвечают основным принципам формирования ФОС.

Разработанный и представленный для экспертизы фонд оценочных средств рекомендуется к использованию в процессе подготовки обучающихся по указанному направлению

Председатель учебно-методической
комиссии экономического института



Шаповалова Е.Н.

Лист изменений и дополнений

№ п/п	Виды дополнений и изменений	Дата и номер протокола заседания кафедры (кафедр), на котором были рассмотрены и одобрены изменения и дополнения	Подпись (с расшифровкой) заведующего кафедрой (заведующих кафедрами)