

Содержание вопроса
<p>Прочитайте задание и выполните его.</p> <p>Укажите последовательность этапов верификации программного обеспечения:</p> <p>А. Анализ требований Б. Верификация дизайна В. Верификация кода Г. Верификация системы</p> <p>Формат ответа: А – Г – В – Б</p> <p>А – Б – В – Г</p>
<p>Прочитайте задание и выполните его.</p> <p>Укажите правильную последовательность уровней тестирования программного обеспечения:</p> <p>А. Интеграционное тестирование Б. Системное тестирование В. Модульное тестирование Г. Приемочное тестирование</p> <p>Формат ответа: А – Г – В – Б</p> <p>В – А – Б – Г</p>
<p>Прочитайте задание и выполните его.</p> <p>Укажите правильную последовательность этапов процесса тестирования программного обеспечения:</p> <p>А. Акцептование Б. Анализ результатов тестирования В. Повторное тестирование Г. Планирование тестирования Д. Разработка тест-кейсов Е. Выполнение тестов</p>

- А. Анализ требований
- Б. Верификация дизайна
- В. Верификация кода
- Г. Верификация системы

Формат ответа: А – Г – В – Б

А – Б – В – Г

- А. Интеграционное тестирование
- Б. Системное тестирование
- В. Модульное тестирование
- Г. Приемочное тестирование

Формат ответа: А – Г – В – Б

В – А – Б – Г

- А. Акцептование
- Б. Анализ результатов тестирования
- В. Повторное тестирование
- Г. Планирование тестирования
- Д. Разработка тест-кейсов
- Е. Выполнение тестов

Формат ответа: А – Г – В – Б

Г – Д – Е – Б – В – А

Прочитайте задание и выполните его.

Укажите правильную последовательность основных этапов жизненного цикла программного обеспечения согласно ГОСТ Р ИСО/МЭК 12207:

- А. Разработка
- Б. Эксплуатация
- В. Утилизация
- Г. Сопровождение
- Д. Анализ требований
- Е. Тестирование

Формат ответа: А – Г – В – Б

Д – А – Е – Б – Г – В

Прочитайте задание и выполните его.

Установите соответствие между определениями и терминами.

Определения:

1. Проверка соответствия кода техническому заданию
2. Подтверждение, что система решает бизнес-задачи пользователя
3. Поиск расхождений между фактическим и ожидаемым поведением

Термины:

- А. Верификация
- Б. Валидация
- В. Тестирование

Формат ответа:

- 1 – А
- 2 – Г
- 3 – В и т.д.

1 – А
2 – Б
3 – В

Прочитайте текст и установите соответствие.

Сопоставьте модели нейронных сетей с их особенностями:

Модель:

- 1) Многослойная перцептронная сеть
- 2) Свёрточная нейронная сеть (CNN)
- 3) Рекуррентная нейронная сеть (RNN)
- 4) Генеративно-сопоставительная сеть (GAN)

Особенность:

- А) Модель, состоящая из нескольких слоёв нейронов и способная обрабатывать сложные данные
- Б) Используется для анализа изображений, способна извлекать пространственные признаки
- В) Распознаёт временные зависимости и может обрабатывать последовательности данных
- Г) Состоит из генератора и дискриминатора для генерации новых образцов данных

Формат ответа:

- 1 – А
- 2 – Г
- 3 – В и т.д.

- 1 – А
- 2 – Б
- 3 – В
- 4 – Г

Прочитайте текст и установите соответствие.

Сопоставьте виды ошибок в нейронных сетях с их описанием:

Вид ошибки:

- 1) Ошибка обучения
- 2) Ошибка тестирования
- 3) Ошибка общего представления
- 4) Ошибка переобучения

Описание:

- А) Разница между предсказанными и реальными значениями в обучающем наборе данных
- Б) Разница между предсказанными и реальными значениями в тестовом наборе данных
- В) Неспособность модели обобщать знания на новые данные
- Г) Когда модель слишком сильно подстраивается под обучающий набор данных

Формат ответа:

1 – А

2 – Г

3 – В и т.д.

1 – А

2 – Б

3 – В

4 – Г

Прочитайте текст и установите соответствие.

Сопоставьте методы регуляризации с их описанием:

Метод регуляризации:

1) L1-регуляризация

2) L2-регуляризация

3) Dropout

4) Уменьшение размерности

Описание:

А) Применяет штраф в форме суммы абсолютных значений весов

Б) Применяет штраф в форме суммы квадратов весов, способствует разреженности

В) Случайно отключает часть нейронов во время обучения для предотвращения переобучения

Г) Использует сокращение размерности входных данных для улучшения обобщающей способности модели

Формат ответа:

- 1 – А
- 2 – Г
- 3 – В и т.д.

- 1 – А
- 2 – Б
- 3 – В
- 4 – Г

Прочитайте текст и установите соответствие.

Сопоставьте этапы обучения нейронной сети с их описанием.

Этап обучения:

- 1) Инициализация весов
- 2) Прямое распространение
- 3) Обратное распространение
- 4) Оценка модели

Описание:

- А) Установка начальных значений для весов нейронов
- Б) Процесс вычисления выходных значений на основе входных данных и весов
- В) Процедура обновления весов в соответствии с ошибками
- Г) Проверка производительности модели на тестовом наборе данных

Формат ответа:

- 1 – А
- 2 – Г
- 3 – В и т.д.

- 1 – А
- 2 – Б
- 3 – В
- 4 – Г

Прочитайте текст и установите соответствие.

Сопоставьте архитектуры нейронных сетей с их применением.

Архитектура:

- 1) Свёрточная сеть (CNN)
- 2) Рекуррентная сеть (RNN)
- 3) Автоэнкодер
- 4) Полносвязная нейронная сеть

Применение:

- А) Предсказание временных рядов, обработка последовательных данных
- Б) Обработка изображений, классификация и сегментация
- В) Сжатие данных и извлечение признаков
- Г) Общие задачи классификации и регрессии

Формат ответа:

- 1 – А
- 2 – Г
- 3 – В и т.д.

- 1 – Б
- 2 – А
- 3 – В
- 4 – Г

Прочитайте задание и выполните его.

При практической реализации построения модели процесса в нотации BPMN 2.0 необходимо реализовать несколько последовательных действий. Установите их последовательность:

- А. Определить границы процесса, тем самым выявив условие запуска процесса и результат его завершения.
- Б. Определить наличие каких-либо условий, которые могут приводить к ветвлению процесса, формированию циклов и т.п.
- В. Определить участников процесса, то есть исполнителей действий.
- Г. Выявить лиц и/или процессы, которые своими действиями или сообщениями могут влиять на исследуемый и моделируемый процесс.
- Д. Определить последовательность действий в процессе, закрепив за ними исполнителей.

Формат ответа: А – Г – В – Б

А-Г-В-Д-Б

При практической реализации построения процессно-событийной модели процесса необходимо реализовать несколько последовательных действий:

- А. Определить наличие каких-либо условий, которые могут приводить к ветвлению процесса.
- Б. Определить границы процесса, тем самым выявив иницирующее и завершающее события.
- В. Определить участников процесса, то есть исполнителей действий.
- Г. Определить последовательность действий в процессе, закрепив за ними исполнителей.

Формат ответа: А – Г – В – Б

Б-В-Г-А

Прочитайте задание и выполните его.

Нотариальная контора «Иванов и Ко» осуществляет частную деятельность в области нотариата. В конторе работают два сотрудника, нотариус и помощник. Процесс «Заверить копию документа об образовании» осуществляет нотариус. Процесс иницируется приходом клиента. Нотариус проверяет документ в ФРДО, делает копию документа, ставит печать, ставит подпись, принимает оплату, отдает документы, после чего процесс завершается. В случае, если клиенту недостаточно одной копии документа об образовании, то нотариус повторяет действие «делает копию». Необходимо осуществить переход от текстового описания к элементам BPMN, соотнеся элементы процесса с элементами нотации BPMN.

- 1) Заверить копию документа об образовании
- 2) Клиент
- 3) Отдать документы
- А) Название свернутого пула
- Б) Название развернутого пула
- В) Название элемента «Задача»

Формат ответа:

1 – А
2 – Г
3 – В и т.д.

1 – Б
2 – А
3 – В

Прочитайте задание и выполните его.

Городская детская библиотека, расположенная по адресу: г. Рязань, ул. Пушкина, д. 4, относится к объектам культуры и осуществляет просветительскую деятельность путем предоставления печатных изданий для чтения в читательском зале и на дому. Процесс «Выдать книгу» осуществляет библиотекарь Иванова В.В. Процесс начинается при приходе посетителя. Библиотекарь запрашивает паспорт, затем находит читательский формуляр, узнает потребности посетителя, находит книгу, заносит название книги в читательский формуляр, отдать книгу посетителю. Процесс заканчивается выдачей книги. Соотнесите элементы процесса с элементами методологии ARIS.

- 1) Паспорт
 - 2) Отдать книгу
 - 3) Библиотекарь
 - 4) Книга выдана
- А) Event
Б) Role
В) Activity
Г) Document

Формат ответа:

1 – А
2 – Г
3 – В и т.д.

1 – Г
2 – В
3 – Б

4 – А

Прочитайте задание и выполните его.

Соотнесите виды стрелок, отражаемых на структурно-функциональной диаграмме, с их сущностью.

- 1) Вход
 - 2) Управление
 - 3) Выход
 - 4) Механизм
- А) Непреобразуемые ресурсы, которые необходимы для осуществления преобразования Входов в Выходы
- Б) Материал или информация, которые производятся процессом
- В) Стандарты, приказы, регламенты и т.п., всё то, чем руководствуется процесс
- Г) Материал или информация, которые используются или преобразуются процессом для получения результата (выхода)

Формат ответа:

- 1 – А
- 2 – Г
- 3 – В и т.д.

1 – Г

2 – В

3 – Б

4 – А

Прочитайте задание и выполните его

Сопоставьте уровень RAID и минимальное количество дисков:

- 1) RAID 0
- 2) RAID 1
- 3) RAID 5
- 4) RAID 6
- 5) RAID 10

Варианты характеристик:

- А) 2
- Б) 3
- В) 4
- Г) 2

Формат ответа:

- 1 – А
- 2 – Г
- 3 – В и т.д.

- 1 – А
- 2 – Г
- 3 – Б
- 4 – В
- 5 – В

Прочитайте задание и выполните его

Сопоставьте уровень RAID и его поведение при отказе одного диска:

- 1) RAID 0
- 2) RAID 1
- 3) RAID 5
- 4) RAID 6

Варианты характеристик:

- А) Полный отказ массива, потеря данных
- Б) Работа продолжается в деградированном режиме без потери данных

Формат ответа:

- 1 – А
- 2 – Г
- 3 – В и т.д.

- 1 – А
- 2 – Б

3 – Б
4 – Б

Прочитайте задание и выполните его

Установите соответствие между типом СХД и используемыми протоколами.

- 1) DAS
- 2) SAN
- 3) NAS

Варианты:

- А) iSCSI, Fibre Channel (FC), FCoE, NVMe-oF
- Б) SMB/CIFS (Windows), NFS (Unix/Linux), FTP
- В) SATA, SAS, SCSI

Формат ответа:

- 1 – А
- 2 – Г
- 3 – В и т.д.

1 – В
2 – А
3 – Б

Прочитайте задание и выполните его

Сопоставьте уровень RAID и его принцип работы:

- 1) RAID 0
- 2) RAID 1
- 3) RAID 5
- 4) RAID 6
- 5) RAID 10

- А) Зеркалирование (дублирование данных)
- Б) Чередование без избыточности
- В) Чередование с контролем четности, "четность" распределено по всем дискам

- Г) Двойная распределенная четность
- Д) Комбинация зеркалирования и чередования

Формат ответа:

- 1 – А
- 2 – Г
- 3 – В и т.д.

Ответ:

- 1 – Б
- 2 – А
- 3 – В
- 4 – Г
- 5 – Д

Прочитайте задание и выполните его

Установите соответствие между типом СХД и ключевым преимуществом.

- 1) DAS
- 2) SAN
- 3) NAS

Варианты:

- А) Простота управления и невысокая стоимость развертывания
- Б) Высочайшая производительность и низкая задержка для критичных приложений
- В) Централизованное управление файлами и легкость общего доступа

Формат ответа:

- 1 – А
- 2 – Г
- 3 – В и т.д.

- 1 – А
- 2 – Б
- 3 – В

Прочитайте задание и выполните его.

Укажите правильную последовательность основных этапов тестирования программного обеспечения:

- А. Акцептование
- Б. Анализ результатов тестирования
- В. Повторное тестирование
- Г. Планирование тестирования
- Д. Разработка тест-кейсов
- Е. Выполнение тестов.

Формат ответа: А – Г – В – Б

Г-Д-Е-Б-В-А

Прочитайте задание и выполните его.

Установите последовательность по степени увеличения влияния ошибок на тестируемый программный продукт.

- А. Критическая ошибка
- Б. Незначительная ошибка
- В. Блокирующая ошибка
- Г. Тривиальная ошибка
- Д. Серьёзная ошибка

Формат ответа: А – Г – В – Б

Г – Б – Д – А – В

Прочитайте задание и выполните его.

Установите соответствие между описанием ошибки и категорией, к которой она относится.

Описание:

1. дефект, при котором пользователь не может продолжить выполнение сценария тестирования из-за полной недоступности функции

2. ошибка вызывает неправильное поведение системы при выполнении важной функции, но не мешает выполнению других задач
3. ошибка приводит к невозможности выполнения важных операций, таких как обработка транзакций или расчёты
4. ошибка, которая относится к эстетическим недостаткам или мелким отклонениям в визуальном представлении системы
5. ошибка возникает при некорректном отображении текста, не влияя при этом на работу приложения

Категория:

- А. Критическая
- Б. Блокирующая
- В. Тривиальная
- Г. Серьёзная
- Д. Незначительная

Формат ответа:

- 1 – А
- 2 – Г
- 3 – В и т.д.

- 1 – Б
- 2 – Г
- 3 – А
- 4 – Д
- 5 – В

Прочитайте задание и выполните его.

Даны три определения:

1. процесс проверки соответствия программы требованиям, спецификации или ожиданиям конечного пользователя
2. процесс проверки на соответствие заданным требованиям, оценке качества и выявление возможных дефектов или ошибок.
3. процесс проверки соответствия программного кода заданным спецификациям и требованиям

Сопоставьте их с терминами, которые они описывают.

- А. Верификация программного продукта
- Б. Тестирование программного продукта
- В. Валидация программного продукта

Формат ответа:

- 1 – А
- 2 – Г
- 3 – В и т.д.

- 1 – В
- 2 – Б
- 3 – А

Прочитайте задание и выполните его.

Установите правильную последовательность прохождения ключевых состояний ошибки в её жизненном цикле.

Состояния:

- А. Новая
- Б. Опубликована
- В. Неподтверждённая
- Г. Открытая
- Д. Проверена
- Е. Закрыта
- Ж. Исправленная

Формат ответа: А – Г – В – Б

А – В – Г – Б – Ж – Д – Е

Содержание вопроса

Прочитайте задание и дайте развёрнутый ответ.

В рамках подготовки к запуску нового веб-приложения необходимо провести нагрузочное тестирование. Перечислите действия для эффективного проведения нагрузочного тестирования.

- Определить цели и метрики тестирования
- Выбрать ключевые сценарии и точки измерения
- Подготовить тестовую среду
- Разработать и проверить тестовые сценарии
- Провести нагрузочные тесты
- Проанализировать результаты
- Оптимизировать систему и повторить тестирование

Прочитайте задание и дайте развёрнутый ответ.

Команда разработчиков работает над большим проектом. Чтобы избежать конфликтов при слиянии кода и повысить стабильность системы, архитектор предложил внедрить практику непрерывной интеграции (CI). Запишите суть практики непрерывной интеграции.

Непрерывная интеграция — это практика, при которой разработчики часто вносят изменения в общий репозиторий и каждая интеграция автоматически проверяется с помощью сборки и тестов. Цель CI — как можно раньше обнаруживать ошибки, предотвращать

конфликтные слияния кода и поддерживать основной код в стабильном, рабочем состоянии.

Прочитайте задание и дайте развёрнутый ответ.

В команде тестирования за последний спринт накопилось множество дефектов. Руководитель просит наладить процесс управления ошибками, чтобы ускорить их устранение и повысить качество продукта. Запишите действия для эффективного управления дефектами?

Для эффективного управления дефектами необходимо:

1. централизованно регистрировать каждый дефект с полным описанием и шагами воспроизведения;
2. классифицировать и подтверждать дефект (установить серьёзность и приоритет);
3. назначать ответственного разработчика и сроки исправления;
4. обеспечивать оперативное исправление с анализом первопричины;
5. выполнять повторное тестирование и регрессионное тестирование после исправления;
6. закрывать дефект только после подтверждения устранения;
7. регулярно анализировать накопленные дефекты и метрики (время исправления, повторные баги, плотность дефектов), чтобы улучшать качество продукта и процесс разработки.

Прочитайте задание и дайте развёрнутый ответ.

Опишите, каким образом проверить корректность работы механизмов аутентификации и авторизации?

Корректность механизмов аутентификации и авторизации проверяется через целый набор тестов:

1. Для аутентификации:

- проверка успешного входа с корректными учётными данными и отказа при неверных;
- проверка ограничений по числу неудачных попыток (защита от перебора паролей);
- проверка, что данные входа передаются по защищённому каналу и не попадают в открытый вид;
- проверка корректной обработки сессии/токена (создание после входа, завершение при выходе, невозможность использовать старую сессию после logout);
- проверка восстановления пароля с контролем личности.

2. Для авторизации:

- проверка, что пользователь имеет доступ только к тем функциям и данным, которые соответствуют его роли;
- попытка получить доступ к чужим данным или административным возможностям без соответствующих прав;
- проверка, что даже прямые вызовы API не дают лишних прав;
- проверка реакции системы на запросы без входа (неаутентифицированный пользователь должен получить отказ);
- проверка отзыва прав при смене роли.

3. Дополнительно:

- проверка, не оставляет ли система конфиденциальные данные в открытом виде;
- проверка ведения журналов попыток несанкционированного доступа.

Если все эти проверки пройдены, можно считать, что механизмы аутентификации (кто входит) и авторизации (что разрешено после входа) работают корректно и безопасно.

Прочитайте задание и дайте развёрнутый ответ.

Вы участвуете в разработке крупной программной системы. Архитектор настаивает на том, чтобы проект был построен по принципу модульности. Запишите суть принципа модульности в программной инженерии

Принцип модульности в программной инженерии заключается в том, что программная система должна строиться как совокупность отдельных модулей, каждый из которых реализует ограниченную, понятную область функциональности и имеет чётко определённый интерфейс взаимодействия с другими модулями. Такая структура уменьшает взаимные зависимости, упрощает разработку, тестирование, сопровождение и модернизацию системы, а также повышает надёжность и управляемость проекта.

Прочитайте задание и дайте развёрнутый ответ.

Во время ревью тестов разработчик утверждает, что «у нас хорошее тестовое покрытие, значит, всё протестировано». Чтобы объективно оценить это утверждение, команда решает использовать метрику тестового покрытия. Запишите суть и значение тестового покрытия.

Тестовое покрытие — это метрика, которая показывает, какая доля кода системы была выполнена при запуске тестов (например, сколько строк, ветвей или функций реально отработали во время тестирования). Значение этой метрики — в выявлении непроверенных участков кода, в оценке полноты автоматизированного тестирования и в снижении риска скрытых дефектов. При этом высо-

кая величина покрытия сама по себе не гарантирует качество: она не оценивает корректность поведения, полноту проверок и соответствие бизнес-требованиям, а лишь фиксирует факт, что код выполнялся под контролем тестов.

Прочитайте задание и дайте развёрнутый ответ.

Команда готовится к выпуску нового релиза сложной информационной системы. Перед началом тестирования необходимо сформировать эффективную тестовую стратегию, запишите основные моменты.

Тестовая стратегия — это документ верхнего уровня, который определяет, как будет проводиться тестирование релиза. В неё должны входить:

- цели тестирования и ожидаемое качество;
- область тестирования (что проверяется и что не проверяется);
- перечень типов тестирования (функциональное, интеграционное, регрессионное, нагрузочное, безопасность и т. д.);
- критические бизнес-сценарии, которые должны быть обязательно покрыты;
- критерии входа и выхода из тестирования;
- описание тестовой среды и тестовых данных;
- распределение ролей и ответственности;
- процесс управления дефектами;
- ключевые метрики качества; а также риски, которые могут повлиять на итоговую оценку готовности системы к релизу.

Прочитайте задание и дайте развёрнутый ответ.

Вы участвуете в разработке корпоративной системы, и архитектор просит вас ознакомиться с архитектурой ПО проекта. Для этого вы должны

понимать её состав. Запишите ключевые элементы архитектуры программного обеспечения.

Ключевые элементы архитектуры программного обеспечения включают:

1. структуру системы — набор модулей, подсистем и сервисов с указанием их ответственности;
2. взаимодействие между компонентами — используемые интерфейсы, протоколы обмена, направления зависимостей;
3. модель данных — хранилища данных, распределение владения данными, правила доступа к ним и интеграцию данных между частями системы;
4. интерфейсы системы — внешние API, пользовательские интерфейсы и точки интеграции с другими системами;
5. инфраструктуру и модель развёртывания — среду выполнения, способ развёртывания, масштабирования и обеспечения отказоустойчивости;
6. нефункциональные характеристики (требования к производительности, надёжности, безопасности, масштабируемости, поддерживаемости и наблюдаемости);
7. архитектурные стандарты и ограничения, определяющие допустимые технологии и правила разработки;
8. описание ключевых потоков данных и критичных бизнес-сценариев, которые показывают, как система работает целиком.

Прочитайте задание и дайте развёрнутый ответ.

Компания-разработчик веб-приложения, предназначенного для онлайн-покупок, обнаружила, что при большой нагрузке на сайт возникают проблемы с производительностью, а некоторые пользователи жалуются на задержки при оформлении заказа. Для того, чтобы решить эту проблему, компания решает провести Стрессовое тестирование (Stress Testing), чтобы определить причину

проблем и улучшить производительность сайта. В процессе работ выявляется, что база данных неэффективно обрабатывает запросы при высокой нагрузке, что приводит к задержкам при оформлении заказа. Укажите дальнейшие действия компании

После того как стрессовое тестирование выявило проблемы с производительностью базы данных, компания должна:

1. Проанализировать результаты теста и определить конкретные запросы и операции, вызывающие задержки.
2. Провести оптимизацию базы данных: улучшить SQL-запросы, добавить необходимые индексы, оптимизировать схему данных и конфигурацию СУБД.
3. Оптимизировать уровень приложения — уменьшить количество обращений к БД, внедрить кэширование и пулы соединений.
4. При необходимости масштабировать инфраструктуру (вертикально — увеличить ресурсы, горизонтально — репликация, шардирование).
5. Повторно провести стрессовое тестирование для проверки улучшений.
6. Настроить постоянный мониторинг и систему предупреждений, чтобы отслеживать производительность в реальном времени.

Такой подход позволит устранить текущие проблемы, повысить устойчивость системы к нагрузкам и обеспечить стабильную работу веб-приложения при росте числа пользователей.

Прочитайте задание и дайте развёрнутый ответ.

При тестировании был определен дефект, представленный на рисунке.

Нет кнопки выхода из приложения	1. Запустить ПО 2. Выйти из ПО Фактический результат: нет кнопки выхода из приложения	удобство	Minor	Trivial	Ссылка
---------------------------------	---	----------	-------	---------	------------------------

В данном отчете о тестировании приложения отсутствует ...

В этом описании **не указано ожидаемое поведение.**

То есть не написано, что должно было быть, например: «Ожидаемый результат: у пользователя должна быть доступна явная кнопка выхода, позволяющая корректно завершить работу приложения».

Это поле является обязательным в хорошем баг-репорте, потому что именно оно фиксирует, почему мы считаем ситуацию дефектом, а не просто особенностью дизайна.

Прочитайте задание и дайте развёрнутый ответ.

Команда тестирования готовится провести проверку производительности нового веб-приложения. Руководитель просит вас определить, какие параметры необходимо измерять для объективной оценки производительности системы.

Запишите их.

Для объективной оценки производительности веб-приложения необходимо измерять:

1. Время отклика (среднее и на перцентилях — например, 95-й/99-й перцентиль).
2. Пропускную способность системы (число запросов или транзакций в секунду).
3. Нагрузку на ресурсы инфраструктуры — использование процессора, памяти, диска и сети.
4. Долю неуспешных запросов и типы возникающих ошибок под нагрузкой.

5. Устойчивость системы во времени при длительной нагрузке (нет ли деградации и утечек).
6. Время выполнения критически важных бизнес-сценариев (например, оформление заказа).
7. Масштабируемость — как система ведёт себя при росте числа пользователей и при добавлении ресурсов.
8. Поведение системы в момент пиков и после них (способность корректно восстановиться).

Прочитайте задание и дайте развёрнутый ответ.

Вы участвуете в начале проекта по разработке корпоративного программного продукта. Перед командой стоит задача собрать максимально полные и точные требования к системе от заинтересованных сторон. Запишите стандартные методы сбора требований к программному обеспечению.

Стандартные методы сбора требований к программному обеспечению включают:

1. **Интервью с заинтересованными сторонами** — проведение индивидуальных бесед с заказчиками, будущими пользователями, владельцами процессов для выявления целей, ожиданий и ограничений.
2. **Опросы и анкеты** — массовый сбор мнений и потребностей от широкой группы пользователей, когда важно получить статистическую картину.
3. **Рабочие встречи, семинары и воркшопы** — совместная проработка требований несколькими стейкхолдерами одновременно, выявление конфликтов интересов и согласование приоритетов.

4. **Наблюдение за пользователями в реальной работе (наблюдение, job shadowing)** — анализ того, как сейчас выполняются бизнес-процессы, какие шаги занимают много времени и где возникают ошибки.
5. **Анализ существующих документов и текущих систем** — изучение регламентов, инструкций, форм отчётности, протоколов безопасности, а также функционала уже действующих систем.
6. **Прототипирование (мокапы, интерактивные прототипы интерфейса)** — демонстрация будущего решения в виде набросков экранов и сценариев, с последующим сбором обратной связи для уточнения требований.
7. **Описание пользовательских сценариев и use cases** — фиксация типичных последовательностей действий пользователя и реакции системы, включая альтернативные потоки.
8. **Моделирование бизнес-процессов** — построение диаграмм процессов компании, чтобы понять роли, потоки данных, точки согласования и автоматизации.
9. **Консультации с безопасностью, юристами и комплаенсом** — сбор нефункциональных требований по защите данных, аудиту действий, надёжности, соответствию нормативам.
10. **Анализ аналогичных решений и конкурентного рынка** — извлечение требований из уже существующих продуктов и стандартов отрасли.

Прочитайте задание и дайте развёрнутый ответ.

Вы работаете в команде разработки распределённой системы. Проект близится к завершению, и начинается этап интеграции всех модулей, разработанных разными участниками команды. Запишите основные этапы процесса интеграции программных модулей.

Основные этапы процесса интеграции программных модулей включают:

1. **Подготовка к интеграции:** согласование интерфейсов и контрактов между модулями, проверка совместимости версий, подготовка общего окружения.
2. **Сборка и развёртывание модулей в общей интеграционной среде:** размещение всех компонентов системы в едином стенде, настройка взаимодействия между ними.
3. **Интеграционное тестирование:** проверка взаимодействия модулей, прохождение сквозных бизнес-сценариев, выявление ошибок совместной работы.
4. **Устранение найденных несовместимостей:** анализ дефектов интеграции, доработка модулей, повторная сборка.
5. **Повторная проверка (регрессионное тестирование интеграции):** подтверждение, что исправления работают и не ломают другие части системы.
6. **Проверка нефункциональных характеристик совместной работы модулей:** производительность, устойчивость, масштабируемость, отказоустойчивость, целостность данных.
7. **Актуализация документации и контрактов:** фиксация итоговых интерфейсов, зависимостей и ограничений для последующей эксплуатации.
8. **Передача интегрированной системы на следующий этап тестирования:** системные и приёмочные испытания.

Такой процесс позволяет не просто “склеить код”, а получить работоспособную, согласованную и управляемую систему, готовую к релизу и сопровождению.

Прочитайте задание и дайте развёрнутый ответ.

В проекте с высоким уровнем критичности команда решила применять автоматизированный подход к созданию тестов, основанный на формальных описаниях поведения системы. Запишите суть практики тестирования на основе моделей.

Тестирование на основе моделей — это метод, при котором тесты автоматически выводятся из формального описания поведения системы. Сначала строится модель (например, в виде состояний и переходов, правил и ограничений), отражающая требуемую логику работы системы. Затем на основе этой модели автоматически генерируются тестовые сценарии, которые исполняются и сравниваются с ожидаемыми результатами, также заданными в модели.

Ценность подхода:

- обеспечивает систематическое и измеряемое покрытие требований;
- помогает обнаруживать противоречия и пробелы в спецификациях;
- уменьшает ручную разработку тест-кейсов;
- облегчает обновление тестов при изменении требований;
- особенно подходит для критичных систем, где нельзя допускать непроверенные ветви поведения.

Прочитайте задание и дайте развёрнутый ответ.

Запишите пунктом пропущенный сценарий ввода в эксплуатацию функционального модуля «Регистрация издания в библиотеке» автоматизированной информационной системы «Библиотека МИРЭА». Информационно-логическая схема базы данных представлена на рисунке 1.

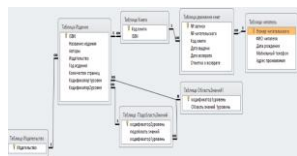


Рисунок 1-Схема базы данных

Автоматизированная информационная система «Библиотека МИРЭА» является локальной однопользовательской системой для автоматизации

работы библиотекаря. Функции АИС «Библиотека МИРЭА»: регистрация изданий в библиотеке, регистрация движения книг, подготовка отчетности.

1. Ознакомление с целями и сроками внедрения АИС.
 2. Утверждение ИТ специалиста и ключевого пользователя.
 3. Анализ технических параметров компьютера библиотекаря на соответствие требованиям, заявленным в спецификации на приложение к ИС.
 4. Анализ программного обеспечения на соответствие требованиям, заявленным в спецификации на приложение к ИС.
 5. Подготовка оборудования и программного обеспечения.
 6. Подготовка данных для занесения в систему.
 7. Установка приложения на компьютер библиотекаря.
 8. Запуск приложения.
 - 9.
 10. Заполнение справочной информации:
 11. Справочника «Издательство»
 12. Справочника «Кодификатор 1 уровень»
 13. Справочника «Кодификатор 2 уровень»
 14. Проведение тестовой (опытной эксплуатации).
 15. Обучение ключевого пользователя-библиотекаря.
 16. Тестовая эксплуатация.
 17. Валидация данных.
 18. Промышленная эксплуатация.
 19. Анализ работы модуля.
- Запишите 9 пункт.

9. Создание и настройка учётной записи ключевого пользователя (библиотекаря) и назначение прав доступа в системе

Прочитайте текст и запишите развернутый обоснованный ответ.

Спроектируйте архитектуру нейронной сети, которую вы бы использовали для классификации изображений размером 28×28 пикселей (например, цифр от 0 до 9). Укажите типы слоёв и их порядок.

1 Вход: изображение $1 \times 28 \times 28$

2 Свёрточный слой 3×3 , 32 фильтра

3 ReLU

4 MaxPooling 2×2

5 Свёрточный слой 3×3 , 64 фильтра

6 ReLU

7 MaxPooling 2×2

8 Flatten

9 Полносвязный слой $3136 \rightarrow 128$

10 ReLU

11 (Dropout 0.5 — при обучении)

12 Полносвязный слой $128 \rightarrow 10$

13 Softmax (для получения вероятности классов)

Напишите ответ и ход решения задачи.

Вычислить последовательность сигналов линейной свертки при сигналах: $a(m) = [2, 1, -1, 2]$ и $b(m) = [-1, -1, -1]$. Записать последовательность чисел, чему будут равны $s(0)$, $s(1)$, $s(2)$, $s(3)$, $s(4)$, $s(5)$ (через запятую).

		$a(m)$	2	1	-1	2		
$b(0-m)$	-1	-1	-1					
		-1	-1	-1				
			-1	-1	-1			
				-1	-1	-1		
					-1	-1		
						-1		
							-1	
								-1

Проверено

$$s(k) = \sum_{i=0}^{N-1} a(i) \cdot b(k-i)$$

3. Пошаговое вычисление

s(0):
только a(0)b(0)

$$s(0) = 2 \cdot (-1) = -2$$

s(1):

$$s(1) = a(0)b(1) + a(1)b(0) = 2 \cdot (-1) + 1 \cdot (-1) = -2 - 1 = -3$$

s(2):

$$\begin{aligned} s(2) &= a(0)b(2) + a(1)b(1) + a(2)b(0) \\ &= 2 \cdot (-1) + 1 \cdot (-1) + (-1) \cdot (-1) = -2 - 1 + 1 = -2 \end{aligned}$$

s(3):

$$\begin{aligned} s(3) &= a(1)b(2) + a(2)b(1) + a(3)b(0) \\ &= 1 \cdot (-1) + (-1) \cdot (-1) + 2 \cdot (-1) = -1 + 1 - 2 = -2 \end{aligned}$$

s(4):

$$\begin{aligned} s(4) &= a(2)b(2) + a(3)b(1) \\ &= (-1) \cdot (-1) + 2 \cdot (-1) = 1 - 2 = -1 \end{aligned}$$

s(5):

$$s(5) = a(3)b(2) = 2 \cdot (-1) = -2$$

✔ Ответ:

$$s(0), s(1), s(2), s(3), s(4), s(5) = -2, -3, -2, -2, -1, -2$$

Прочитайте текст и запишите развернутый обос-
нованный ответ.

Инженеру-разработчику необходимо построить архитектуру нейронной сети встречного распространения для решения задачи распознавания образов. Помимо входного и выходного слоёв, необходимо определить, какие скрытые слои следует включить в архитектуру, а также кратко описать их назначение и роль в обработке данных.

архитектуру сети прямого распространения (feedforward neural network, сеть встречного распространения в классической русской терминологии — это многослойный перцептрон с обучением методом обратного распространения ошибки). Задача: распознавание образов. Нам нужно указать слои, порядок слоёв и объяснить, зачем каждый слой нужен. Я опишу типичную архитектуру и роль каждого слоя.

1. Входной слой (input layer)

Назначение: принимает сырые данные.

- Если мы распознаём образы (например, изображения), то входной слой получает вектор признаков.
- Например, картинка 28×28 пикселей преобразуется в вектор длиной 784 чисел (28×28).

Этот слой сам по себе обработки не делает, он просто передаёт значения дальше.

2. Скрытые слои (hidden layers)

Чтобы сеть не просто “смотрела на пиксели”, а понимала структуру объекта, нужны несколько скрытых слоёв. Обычно выделяют такие типы скрытых слоёв:

2.1. Свёрточные слои (Convolutional layers)

Роль: обнаружение локальных признаков (контуры, углы, линии, фрагменты формы).

- Каждый свёрточный слой применяет набор фильтров (ядер свёртки) к изображению.
- На выходе получается набор карт признаков: каждая карта активируется там, где найдена определённая визуальная структура.

Почему это важно:

- Ранние слои учатся простым паттернам (горизонтальные/вертикальные линии).

- Более глубокие свёрточные слои начинают замечать более сложные элементы (дуги цифр, пересечения штрихов, куски букв).

Без таких слоёв сеть хуже понимает геометрию и намного хуже распознаёт изображения.

Свёрточные слои — это основной механизм выделения информативных визуальных признаков.

2.2. Нелинейная активация (ReLU или похожая функция)

Роль: внести нелинейность, чтобы сеть могла приближать сложные зависимости.

Обычно после каждой свёртки идёт функция активации, например ReLU (Rectified Linear Unit):

$$\text{ReLU}(x) = \max(0, x)$$

Зачем:

- Если бы не было нелинейности, вся сеть сводилась бы к одному линейному преобразованию, то есть была бы слишком “тупой”.
- Нелинейность позволяет модели отличать сложные классы образов.

2.3. Слой субдискретизации / пуллинга (Pooling layer)

Чаще всего Max Pooling.

Роль: уменьшить размер карты признаков и оставить самое важное.

Как он работает:

- Делит карту признаков на маленькие области (например, блоки 2×2).
- из каждой области берёт максимум.

Что это даёт:

1. Уменьшается размер данных → меньше параметров дальше → меньше переобучения.
2. Становится менее важно, сместился ли объект на пару пикселей (устойчивость к сдвигам).
3. Остаются самые сильные признаки.

То есть этот слой действует как “сжатие смысла”: оставляет главное, убирает лишнее.

2.4. Полносвязные скрытые слои (Fully Connected / Dense layers)

После нескольких блоков “Свёртка → Активация → Пуллинг” мы уже не храним картинку как изображение. У нас есть набор признаков высокого уровня (например: “петля как у 8”, “палочка как у 1”, “перекладина как у А”). Дальше эти признаки нужно собрать в решение “к какому классу относится образ”.

Для этого:

1. Мы выпрямляем (flatten) тензор признаков в вектор.
2. Подаём его в один или два полносвязных слоя.

Роль полносвязных скрытых слоёв:

- комбинировать найденные признаки и учить абстрактные концепты класса;
- например, понять, что “две вертикальные палочки + нижняя дуга” → это скорее цифра 5.

Обычно делают, например:

- Полносвязный слой на 128–512 нейронов.
- Активация ReLU.
- Dropout (см. ниже).

2.5. Dropout (как вариант скрытого слоя регуляризации)

Dropout — не “вычислительный” слой, а защитный.

Роль: уменьшение переобучения.

Как он работает:

- Во время обучения случайно выключает (зануляет) часть нейронов слоя.
- Это заставляет сеть не полагаться на один “любимый нейрон”, а учить более устойчивые признаки.

Практический смысл:

- Улучшает обобщающую способность сети на новых данных.
- Особенно важен в полносвязных слоях, потому что именно там обычно больше всего параметров и выше риск переобучения.

3. Выходной слой (output layer)

Это последний слой, который выдаёт класс.

- Если у нас задача классификации на (K) классов (например, 10 цифр), то последний слой — полносвязный слой из (K) нейронов.
- Поверх него применяется Softmax.

Softmax:

превращает выходы в набор вероятностей по классам (каждое число от 0 до 1, сумма равна 1).

Таким образом сеть говорит: “я считаю, что это цифра 3 с вероятностью 0.91, а цифра 5 — с вероятностью 0.04” и т.д.

4. Итого: архитектура по порядку

Я опишу типовой конвейер, используемый в задаче распознавания образов:

1. Входной слой

Принимает изображение (например, $1 \times 28 \times 28$).

2. Свёрточный слой №1

Извлекает базовые признаки (края, линии).

↓

ReLU

↓

Max Pooling

3. Свёрточный слой №2

Извлекает более сложные признаки (фрагменты фигур, дуги, пересечения).

↓

ReLU

↓

Max Pooling

4. Выпрямление признаков (Flatten)

Превращаем карты признаков в один длинный вектор чисел.

5. Полносвязный скрытый слой

Учится понимать композицию признаков как целостный образ (“это семёрка, а не единица”).

↓

ReLU

↓

(возможен Dropout для уменьшения переобучения)

6. Выходной слой + Softmax

Даёт вероятности классов.

5. Почему именно такая структура?

- Свёртки: учат глаза сети видеть локальные элементы образа.
- Пуллинг: уменьшает размерность и делает сеть устойчивой к сдвигам.

- Полносвязные скрытые слои: принимают решение на уровне “к какому классу относится объект”.
- Dropout: помогает, чтобы сеть не просто запоминала обучающие примеры.
- Softmax в конце: переводит всё в вероятности для классификации.

Такую архитектуру обычно считают базовым стандартом для распознавания образов, потому что она сочетает:

1. выделение визуальных признаков,
2. сжатие и обобщение,
3. принятие решения по классам.

Эта последовательность слоёв и есть осмысленный ответ на задачу инженера-разработчика.

Прочитайте текст и запишите развернутый обоснованный ответ.

Определите, какой нейрон победит на слое конкурирующих нейронов при обучении сети Кохонена.

$$X = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}, \quad \text{№1 } W_1^T = \begin{bmatrix} 0,1 \\ 0,2 \\ 0,1 \\ 0,5 \end{bmatrix}, \quad \text{№2 } W_2^T = \begin{bmatrix} 0,4 \\ 0,2 \\ 0,3 \\ 0,1 \end{bmatrix}$$

как это делается в самоорганизующихся картах Кохонена (Self-Organizing Map, SOM).

В слое конкурирующих нейронов побеждает тот нейрон, весовой вектор которого наиболее похож на входной вектор X . Обычно похожесть измеряют через евклидово расстояние (чем меньше расстояние — тем лучше совпадение).

То есть мы ищем нейрон с минимальной дистанцией до X .

Дано

Входной вектор:

$$X = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

Весовой вектор нейрона №1:

$$W_1 = \begin{bmatrix} 0,1 \\ 0,2 \\ 0,1 \\ 0,5 \end{bmatrix}$$

Весовой вектор нейрона №2:

$$W_2 = \begin{bmatrix} 0,4 \\ 0,2 \\ 0,3 \\ 0,1 \end{bmatrix}$$

Шаг 1. Считаем расстояние между X и W_1

Евклидово расстояние:

$$D_1 = \sqrt{(0 - 0,1)^2 + (1 - 0,2)^2 + (1 - 0,1)^2 + (0 - 0,5)^2}$$

Посчитаем по компонентам:

- $(0 - 0,1)^2 = (-0,1)^2 = 0,01$
- $(1 - 0,2)^2 = (0,8)^2 = 0,64$
- $(1 - 0,1)^2 = (0,9)^2 = 0,81$
- $(0 - 0,5)^2 = (-0,5)^2 = 0,25$

Складываем:

$$0,01 + 0,64 + 0,81 + 0,25 = 1,71$$

Корень:

$$D_1 = \sqrt{1,71} \approx 1,3077$$

Шаг 2. Считаем расстояние между X и W_2

$$D_2 = \sqrt{(0 - 0,4)^2 + (1 - 0,2)^2 + (1 - 0,3)^2 + (0 - 0,1)^2}$$

Компоненты:

- $(0 - 0,4)^2 = (-0,4)^2 = 0,16$
- $(1 - 0,2)^2 = (0,8)^2 = 0,64$
- $(1 - 0,3)^2 = (0,7)^2 = 0,49$
- $(0 - 0,1)^2 = (-0,1)^2 = 0,01$

Сумма:

$$0,16 + 0,64 + 0,49 + 0,01 = 1,30$$

Корень:

$$D_2 = \sqrt{1,30} \approx 1,1402$$

Шаг 3. Сравниваем

- $D_1 \approx 1,3077$
- $D_2 \approx 1,1402$

Побеждает нейрон с меньшим расстоянием.

Меньше расстояние \rightarrow лучше совпадение с входом \rightarrow он "узнал" образ.

$$D_2 < D_1 \Rightarrow \text{побеждает нейрон №2.}$$

Итоговый вывод:

На слое конкурирующих нейронов в сети Кохонена победит **нейрон №2**, потому что его весовой вектор W_2 находится ближе к входному вектору X (евклидово расстояние до X у нейрона №2 меньше).

Напишите ответ и ход решения задачи.

Какие значения будут размещены в матрице 3x3 в результате проведения операции свертки окном

2x2 с коэффициентом 0,7 по представленной матрице? (ответ записать в виде: a_{11} ; a_{12} ; a_{13} ; a_{21} ; a_{22} ; a_{23} ;

2	4	-1	-2
-2	-1	0	4
1	1	1	0
4	1	-4	2

a_{31} ; a_{32} ; a_{33})

Нужно получить матрицу 3×3 с помощью свёртки окном 2×2 и коэффициентом 0,7.

Что это значит:

- Мы берем каждое возможное окно 2×2 в исходной матрице.
- Складываем числа внутри этого окна.
- Умножаем сумму на коэффициент 0,7.
- Результат кладем в соответствующую позицию выходной матрицы a_{ij} .

Так как исходная матрица 4×4 , а окно 2×2 , возможные верхние левые углы окон: (1,1), (1,2), (1,3); (2,1), (2,2), (2,3); (3,1), (3,2), (3,3). Это даёт нам как раз 3×3 результата.

Разберём все элементы.

a_{11} (верхний левый блок)

Окно:

$$\begin{pmatrix} 2 & 4 \\ -2 & -1 \end{pmatrix}$$

Сумма: $2 + 4 + (-2) + (-1) = 3$

Умножаем на 0,7:

$$a_{11} = 3 \cdot 0,7 = 2,1$$

a_{12} (сдвигаем окно вправо на 1)

Окно:

$$\begin{pmatrix} 4 & -1 \\ -1 & 0 \end{pmatrix}$$

Сумма: $4 + (-1) + (-1) + 0 = 2$

$$a_{12} = 2 \cdot 0,7 = 1,4$$

a₁₃

ОКНО:

$$\begin{pmatrix} -1 & -2 \\ 0 & 4 \end{pmatrix}$$

Сумма: $(-1) + (-2) + 0 + 4 = 1$

$$a_{13} = 1 \cdot 0,7 = 0,7$$

a₂₁ (второй ряд выходной матрицы, сдвиг окна вниз)

ОКНО:

$$\begin{pmatrix} -2 & -1 \\ 1 & 1 \end{pmatrix}$$

Сумма: $(-2) + (-1) + 1 + 1 = -1$

$$a_{21} = (-1) \cdot 0,7 = -0,7$$

a₂₂

ОКНО:

$$\begin{pmatrix} -1 & 0 \\ 1 & 1 \end{pmatrix}$$

Сумма: $(-1) + 0 + 1 + 1 = 1$

$$a_{22} = 1 \cdot 0,7 = 0,7$$

a₂₃

ОКНО:

$$\begin{pmatrix} 0 & 4 \\ 1 & 0 \end{pmatrix}$$

Сумма: $0 + 4 + 1 + 0 = 5$

$$a_{23} = 5 \cdot 0,7 = 3,5$$

a₃₁

ОКНО:

$$\begin{pmatrix} 1 & 1 \\ 4 & 1 \end{pmatrix}$$

Сумма: $1 + 1 + 4 + 1 = 7$

$$a_{31} = 7 \cdot 0,7 = 4,9$$

a_{32}

Окно:

$$\begin{pmatrix} 1 & 1 \\ 1 & -4 \end{pmatrix}$$

Сумма: $1 + 1 + 1 + (-4) = -1$

$$a_{32} = (-1) \cdot 0,7 = -0,7$$

a_{33}

Окно:

$$\begin{pmatrix} 1 & 0 \\ -4 & 2 \end{pmatrix}$$

Сумма: $1 + 0 + (-4) + 2 = -1$

$$a_{33} = (-1) \cdot 0,7 = -0,7$$

Готовый ответ в требуемом формате;

2.1; 1.4; 0.7; -0.7; 0.7; 3.5; 4.9; -0.7; -0.7

Прочитайте текст и запишите развернутый обоснованный ответ.

Руководитель команды разрабатывает интеллектуальную обучающую систему, имитирующую обучение на основе взаимодействия между учащимся и материалом. Для этого он решил использовать принцип Хебба для моделирования механизма "обучения через опыт". Чтобы правильно смоделировать поведение синаптической связи, необходимо определить ключевые свойства, на которых основано обучение по Хеббу.

Ключевые свойства обучения по Хеббу, которые должен смоделировать руководитель:

1. Совместная активация усиливает связь. Если пресинаптический и постсинаптический нейроны

активны одновременно, их синаптический вес увеличивается.

2. **Локальность правила.** Изменение веса зависит только от активности этих двух нейронов и их текущей связи; не нужен внешний учитель и не нужен расчёт глобальной ошибки.

3. **Направление изменения — усиление.** Повторяющиеся совпадения активности ведут к всё более сильным связям, то есть к закреплению ассоциаций (“память опыта”).

4. **Корреляционная природа.** Правило Хебба фактически запоминает статистические совпадения сигналов, формируя устойчивые связи стимул ↔ реакция.

5. **Обучение без учителя.** Это модель самоорганизующегося обучения: система сама находит и усиливает значимые связи из наблюдаемого поведения учащегося.

Эти свойства и есть основа "обучения через опыт", которую команда хочет имитировать.

Напишите ответ и ход решения задачи.

Какие значения будут размещены на выходе в результате проведения алгоритма по архитектуре математического нейрона с порогом, продемонстрированного ниже? (ответ – последовательность из четырех чисел)

$X_1=0, X_2=0; X_1=0, X_2=1; X_1=1, X_2=0;$

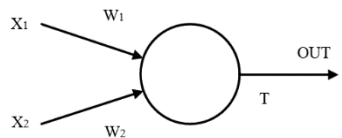
$X_1=1, X_2=1; W_1 = 1, W_2 = 1, T = 1$. Пороговая

функция:

$OUT =$

$$\begin{cases} 1, \text{ если } NET \geq 1 \\ 0, \text{ если } NET < 1 \end{cases}.$$

Сигнал NET – единичная комбинация скорректированного значения.



На изображении показана архитектура простейшего математического нейрона с двумя входами X_1 и X_2 , весами W_1 и W_2 , и порогом T

Такой нейрон вычисляет взвешенную сумму входных сигналов и применяет пороговую (ступенчатую) функцию активации, чтобы выдать результат.

◆ **Ход решения:**

1. Вычисляем суммарный сигнал (NET):

$$NET = X_1 \cdot W_1 + X_2 \cdot W_2$$

2. Сравниваем NET с порогом T:

$$OUT = \begin{cases} 1, & \text{если } NET \geq T \\ 0, & \text{если } NET < T \end{cases}$$

◆ **Подставляем данные:**

- $W_1 = 1$
- $W_2 = 1$
- $T = 1$

Комбинации входов:

№	X_1	X_2	$NET = X_1 \cdot W_1 + X_2 \cdot W_2$	Сравнение с T	OUT
1	0	0	0	$0 < 1$	0
2	0	1	1	$1 \geq 1$	1
3	1	0	1	$1 \geq 1$	1
4	1	1	2	$2 \geq 1$	1

✓ **Ответ (последовательность выходных значений):**

0,1,1,1

◆ **Пояснение:**

Такой нейрон реализует логическую операцию ИЛИ (OR)

он выдаёт 1, если хотя бы один из входов равен 1.

Напишите ответ и ход решения задачи.

Определить субдискретизацию с шагом 1 и 2 для сверточных нейронных сетей с помощью операций взятия максимума. (ответ записать в виде: a_{11} ; a_{12} ; a_{13} ; a_{21} ; a_{22} ; a_{23} ; a_{31} ; a_{32} ; a_{33} . b_{11} ; b_{12} ; b_{21} ; b_{22})

0	1	2	1
4	1	0	1
2	0	1	1
1	2	3	1

(можно думать как карта признаков после свёртки в CNN)

Нужно выполнить субдискретизацию (pooling), то есть операцию подвыборки с помощью **max pooling** — берём максимум в окнах и уменьшаем размерность.

Нам нужно:

1. субдискретизация с шагом 1
2. субдискретизация с шагом 2

И записать оба результата в виде строки значений:

- сначала результат для шага 1 (это будет матрица 3×3 : a_{ij}).
- затем результат для шага 2 (это будет матрица 2×2 : b_{ij}).

Часть 1. Max pooling с шагом 1 (stride = 1), окно 2×2

Мы берём каждое возможное окно 2×2 , двигаем его на один шаг вправо и вниз, и берём максимум внутри окна.

Позиции окон и максимумы:

a_{11}

Окно по верхнему левому углу (1,1):

0	1
4	1

Максимум: $\max(0, 1, 4, 1) = 4$

→ $a_{11} = 4$

a₁₂

Сдвигаем окно вправо (1,2):

$$\begin{pmatrix} 1 & 2 \\ 1 & 0 \end{pmatrix}$$

Максимум: $\max(1,2,1,0) = 2$

→ $a_{12} = 2$

a₁₃

Позиция (1,3):

$$\begin{pmatrix} 2 & 1 \\ 0 & 1 \end{pmatrix}$$

Максимум: $\max(2,1,0,1) = 2$

→ $a_{13} = 2$

a₂₁

Теперь строка ниже, позиция (2,1):

$$\begin{pmatrix} 4 & 1 \\ 2 & 0 \end{pmatrix}$$

Максимум: $\max(4,1,2,0) = 4$

→ $a_{21} = 4$

a₂₂

Позиция (2,2):

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

Максимум: $\max(1,0,0,1) = 1$

→ $a_{22} = 1$

a₂₃

Позиция (2,3):

$$\begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}$$

Максимум: $\max(0,1,1,1) = 1$

→ $a_{23} = 1$

a₃₁

Строка ниже, позиция (3,1):

$$\begin{pmatrix} 2 & 0 \\ 1 & 2 \end{pmatrix}$$

Максимум: $\max(2, 0, 1, 2) = 2$

$$\rightarrow a_{31} = 2$$

a_{32}

Позиция (3,2):

$$\begin{pmatrix} 0 & 1 \\ 2 & 3 \end{pmatrix}$$

Максимум: $\max(0, 1, 2, 3) = 3$

$$\rightarrow a_{32} = 3$$

a_{33}

Позиция (3,3):

$$\begin{pmatrix} 1 & 1 \\ 3 & 1 \end{pmatrix}$$

Максимум: $\max(1, 1, 3, 1) = 3$

$$\rightarrow a_{33} = 3$$

Итак, матрица после субдискретизации с шагом 1 (stride=1) равна:

$$A = \begin{pmatrix} 4 & 2 & 2 \\ 4 & 1 & 1 \\ 2 & 3 & 3 \end{pmatrix}$$

Часть 2. Max pooling с шагом 2 (stride = 2), окно 2×2

Теперь мы берём непересекающиеся окна 2×2 и двигаем их сразу на два шага вправо и вниз.

Из матрицы 4×4 получится матрица 2×2.

Окна:

b_{11}

Верхний левый блок:

$$\begin{pmatrix} 0 & 1 \\ 4 & 1 \end{pmatrix}$$

Максимум: $\max(0, 1, 4, 1) = 4$

$$\rightarrow b_{11} = 4$$

b_{12}

Правая верхняя часть:

$$\begin{pmatrix} 2 & 1 \\ 0 & 1 \end{pmatrix}$$

Максимум: $\max(2, 1, 0, 1) = 2$
 $\rightarrow b_{12} = 2$

b₂₁

Левая нижняя часть:

$$\begin{pmatrix} 2 & 0 \\ 1 & 2 \end{pmatrix}$$

Максимум: $\max(2, 0, 1, 2) = 2$
 $\rightarrow b_{21} = 2$

b₂₂

Правая нижняя часть:

$$\begin{pmatrix} 1 & 1 \\ 3 & 1 \end{pmatrix}$$

Максимум: $\max(1, 1, 3, 1) = 3$
 $\rightarrow b_{22} = 3$

Значит, итоговая матрица после субдискретизации с шагом 2:

$$B = \begin{pmatrix} 4 & 2 \\ 2 & 3 \end{pmatrix}$$

Финальный ответ в требуемом формате:

Сначала элементы матрицы A (3×3) построчно, затем элементы матрицы B (2×2) построчно:

4; 2; 2; 4; 1; 1; 2; 3; 3. 4; 2; 2; 3 ✓

Прочитайте текст и запишите развернутый обоснованный ответ.

Разработчику необходимо спроектировать интеллектуальную систему, использующую искусственные нейронные сети для обработки данных. В зависимости от характера задачи и требуемой динамики обработки информации, необходимо опреде-

лить, к какому из двух основных классов по архитектуре связей может принадлежать нейросеть. Укажите эти два класса с небольшим пояснением.

1. Сети с прямым распространением сигнала

(англ. feedforward neural networks)

Суть:

- Сигнал идёт только вперёд: от входного слоя → через скрытые слои → к выходному слою.
- Связи не образуют циклов, нет обратных петель.
- Выход в данный момент зависит только от текущего входа.

Почему это важно:

Такие сети хороши там, где каждый вход можно рассматривать как отдельный независимый пример. То есть, нет понятия "память о прошлых шагах" во времени.

2. Рекуррентные сети (с обратными связями)

(англ. recurrent neural networks, RNN-класс)

Суть:

- В сети есть циклы: выходы некоторых нейронов возвращаются обратно как входы на следующем шаге.
- Сеть имеет внутреннее состояние (память).
- Текущий выход зависит не только от текущего входа, но и от предыдущих входов во времени.

Почему это важно:

Эти сети моделируют динамику во времени и могут хранить контекст. Они ближе к "мышлению во временной последовательности", а не к разовой классификации.

Итог для разработчика

Если задача — разово обработать статичный набор признаков (картинка, описание объекта, состояние системы на текущем шаге), то это сеть **с прямым распространением**.

Если задача — понимать последовательности, учитывать историю и контекст (речь пользователя, поведение со временем, учебный диалог), то нужна сеть с **обратными связями (рекуррентная сеть)**.

То есть два основных класса по архитектуре связей:

1. Сети прямого распространения (без петель)

2. Рекуррентные сети (с обратными связями, с памятью состояния)

Прочитайте текст и запишите развернутый обоснованный ответ.

Спроектируйте архитектуру нейронной сети, подходящей для классификации коротких текстов (например, отзывов). Укажите типы слоёв и их порядок.

вход — короткий текст (например, отзыв пользователя), выход — класс (положительный, отрицательный, нейтральный и т.д.).

архитектура по порядку

Можно выписать как последовательность слоёв:

1. Вход (последовательность токенов)

Текст преобразован в индексы слов фиксированной длины.

2. Embedding слой

Превращает индексы слов в обучаемые векторы признаков (семантические представления слов).

3. Bidirectional LSTM (или GRU)

Улавливает контекст и порядок слов в тексте в обе стороны.

4. Global Max Pooling (или Average Pooling по времени)

Сжимает последовательность скрытых состояний в один вектор фиксированной длины.

5. Полносвязный слой (Dense + ReLU)

Обработывает извлечённые признаки, учится распознавать паттерны, связанные с классом текста. (Здесь обычно применяется Dropout как регуляризация.)

6. Выходной слой (Dense + Softmax)

Даёт итоговые вероятности классов текста.

Почему такая сеть подходит именно для коротких отзывов

- Embedding даёт смысл слов.
- LSTM даёт понимание порядка слов (важно для языка).
- Pooling даёт компактный "отпечаток" смысла текста.
- Dense принимает решение.

То есть эта архитектура не просто ищет ключевые слова (“ужасно”, “классно”), а учитывает формулировку, смягчения, отрицания (“не так уж и плохо”), что критично для реальной тональности отзывов.

Прочитайте текст и запишите развернутый обоснованный ответ.

Какой тип нейросетевой архитектуры вы бы использовали для задачи кластеризации изображений, если обучающие метки отсутствуют? Спроектируйте сеть и кратко опишите принцип.

Для задачи кластеризации изображений без обучающих меток (то есть в полностью ненадзорном режиме) разумный выбор — использовать **автоэнкодер** (autoencoder), в частности свёрточный автоэнкодер, и кластеризовать объекты в его скрытом пространстве признаков.

Архитектура сети (свёрточный автоэнкодер)

1. Входной слой

- Вход: изображение (например, $28 \times 28 \times 1$ для чёрно-белых или $64 \times 64 \times 3$ для цветных).
- Просто передаём его в энкодер.

2. Энкодер

Энкодер — это несколько свёрточных слоёв (Conv2D), каждый из которых снижает пространственное разрешение и увеличивает число каналов признаков.

Примерная последовательность:

- Свёртка (Conv2D) + ReLU
- Max Pooling (уменьшение размера карты признаков)
- Ещё одна Conv2D + ReLU

- Ещё Pooling
- Flatten (разворачиваем тензор в вектор)
- Полносвязный слой → получаем компактный латентный вектор (например, размером 16, 32, 64 чисел)

Смысл энкодера:

- он выучивает, какие особенности изображения важны для описания структуры (например, форма цифры, очертания объекта и т.д.),
- и «сжимает» изображение до этих признаков.

Эта последняя сжатая часть называется **латентное пространство** или **вектор признаков**.

3. Декодер

Декодер делает обратное:

- Полносвязный слой: разворачивает компактный вектор обратно в объём признаков (например, $8 \times 8 \times N$ карт признаков).
- Transposed Convolution (Conv2DTranspose) или Upsampling + Conv2D: постепенно восстанавливаем пространственное разрешение.
- В конце — слой Conv2D, дающий изображение исходного размера.

Смысл декодера:

- попытаться из «сжатого смысла» снова собрать картинку.

4. Функция потерь

Обучаем автоэнкодер так, чтобы выход был как можно ближе к входу.

Обычно берут среднеквадратичную ошибку (MSE) по пикселям, или L1-потерю.

Важно: на этом этапе нам не нужны метки класса.

Модель просто учится качественно кодировать визуальные паттерны.

Что делать после обучения

1. Берём только **энкодер** (первую половину сети).
2. Прогоняем через него все изображения.
3. Для каждого изображения получаем его латентный вектор (например, размером 32).
4. В этом пространстве векторов запускаем метод кластеризации, например:
 - k-means.

- агломеративную кластеризацию.
- DBSCAN (если не знаем число кластеров заранее).

Почему это работает:

- Похожие изображения получают похожие латентные векторы.
- То есть изображения одного скрытого типа (например, «круглые объекты» против «угловатых объектов») естественно группируются.

Интуитивно: почему автоэнкодер хорош именно для кластеризации

- Он обучается без меток (то есть подходит под условие задачи).
- Он сам извлекает осмысленные визуальные признаки.
- Он выбрасывает шумные детали и подтягивает общую структуру объекта.
- Он сводит изображение к компактному признаковому вектору — а кластеризовать вектора проще, чем сырые пиксели.

Кластеризация прямо по пикселям плоха, потому что:

- пиксели чувствительны к сдвигу, повороту, освещению;
- расстояния в пиксельном пространстве не соответствуют «похожести смысла».

Кластеризация в латентном пространстве автоэнкодера лучше отражает «смысловое» сходство.

Прочитайте текст и запишите развернутый обоснованный ответ.

Какие функции активации вы бы использовали в скрытых и выходном слоях нейросети для задачи многоклассовой классификации и почему?

В скрытых слоях нейронной сети для многоклассовой классификации обычно используют нелинейные функции активации семейства ReLU (например, $\text{ReLU} = \max(0, x)$), потому что они не

насыщаются в положительной области, дают разреженные активации и обеспечивают стабильное и быстрое обучение без сильного затухания градиента.

В выходном слое используется функция Softmax, которая преобразует выходы последнего линейного слоя в распределение вероятностей по классам: сумма вероятностей по всем классам равна 1, и каждый класс конкурирует с остальными. Это соответствует постановке задачи, где объект должен быть отнесён ровно к одному из нескольких возможных классов.

Прочитайте текст и запишите развернутый обоснованный ответ.

Инженер по машинному обучению обучает нейронную сеть на ограниченном наборе данных. Чтобы повысить обобщающую способность модели и избежать переобучения, он должен применить эффективные методы регуляризации. Необходимо назвать два таких метода и кратко объяснить принцип их действия.

1. L2-регуляризация (weight decay).

В функцию потерь добавляется штраф за большие значения весов (сумма квадратов весов). Это не даёт отдельным весам расти слишком сильно, делает модель более гладкой и устойчивой к шуму, тем самым снижая переобучение и улучшая обобщение.

2. Dropout.

Во время обучения случайно “выключается” часть нейронов скрытых слоёв. Модель учится не зависеть от конкретных нейронов, а распределять знание по всей сети. Это действует как ансамбль многих подсетей и уменьшает переобучение на малом количестве данных.

Прочитайте текст и запишите развернутый обоснованный ответ.

Разработчику необходимо спроектировать сверточную нейронную сеть для задачи классификации изображений. Чтобы корректно сформировать архитектуру, он должен понимать назначение скрытых слоев сети. Укажите пять типов скрытых слоев, кроме входного и выходного, которые обычно используются в сверточных нейросетях, и кратко объясните их функции.

1. Свёрточный слой (Conv2D).

Извлекает пространственные признаки изображения (границы, текстуры, формы объектов).

2. Слой активации (например, ReLU).

Вносит нелинейность, помогает сети учить сложные зависимости и ускоряет обучение.

3. Слой пуллинга (Max Pooling / Average Pooling).

Уменьшает размер карт признаков, убирает шум, делает модель устойчивее к сдвигам.

4. Слой нормализации (Batch Normalization).

Стабилизирует распределение активаций, ускоряет и делает обучение более надёжным, снижает переобучение.

5. Полносвязный слой (Dense).

Комбинирует высокоуровневые признаки из свёрточной части и подготавливает решение для классификации.

Можно дополнительно упомянуть Dropout как шестой тип: он снижает переобучение за счёт случайного отключения нейронов, действует как регуляризация.

Прочитайте текст и запишите развернутый обоснованный ответ.

Спроектируйте архитектуру нейронной сети для задачи регрессии (предсказание числовых значений)

1. **Входной слой:** принимает вектор признаков объекта.
2. **Полносвязный слой (например, 64 нейрона) + ReLU:** находит первые нелинейные зависимости между признаками.
3. **Полносвязный слой (32 нейрона) + ReLU:** уточняет внутреннее представление, учит взаимодействия признаков.
4. **(Опционально) Полносвязный слой (16 нейронов) + ReLU + Dropout/L2:** уменьшает переобучение и делает модель более устойчивой.
5. **Выходной слой (1 нейрон, линейная активация):** выдаёт конечное числовое предсказание целевой величины.
6. **Функция потерь:** MSE, чтобы обучать сеть приближать реальные значения.

Прочитайте задание и выполните его.

В практике моделирования, реинжиниринга, управления бизнес-процессами применяются классификация процессов по признаку типа реализуемых целей и задач. В данной классификации выделяют 4 типа бизнес-процессов, перечислите их.

- 1 Основные (ключевые, операционные) процессы
- 2 Вспомогательные (обеспечивающие) процессы
- 3 Управляющие процессы
- 4 Развивающие (инновационные) процессы

Прочитайте задание и выполните его.

При построении структурно-функциональной диаграммы на основе группы технологий IDEF0 используются два основных элемента: блоки и стрелки. Перечислите все наименования стрелок на русском языке, рядом в скобках на английском языке.

Формат ответа:

Название на русском языке (название на английском языке);...;...;...

Вход (Input); Управление (Control); Выход (Output); Механизм (Mechanism)

Прочитайте задание и выполните его.

Перечислите четыре типа диаграмм, которые присутствуют в нотации BPMN.

1. Оркестровка
2. Диаграмма взаимодействия
3. Схема диалога
4. Хореография

Прочитайте задание и выполните его.

Перечислите четыре элемента, которые применяются при построении Data Flow Diagram, в нотации Гейна-Сарсона или Йордана-де Марко

Процесс (Process); Поток данных (Data Flow); Хранилище данных (Data Store); Внешняя сущность (External Entity).

Прочитайте задание и выполните его.

При построении Data Flow Diagram необходимо подписывать функциональные блоки и необходимо подписывать стрелки, отражающие информационные или материальные потоки.

1. Укажите, какое требование к названию работы в DFD предъявляют?
2. Укажите, какое требование к названию стрелок в DFD предъявляют?

Формат ответа:

1 Требование к названию работы:...

2 Требование к названию стрелок:...

1. Требование к названию работы:

Название функционального блока (работы, процесса) должно быть записано **глаголом в неопределённой форме (инфинитиве)**, отражающим действие или преобразование данных.

👉 Например: «Обработать заказ», «Сформировать отчёт», «Проверить данные».

2. Требование к названию стрелок:

Название стрелки, обозначающей поток данных, должно быть **существительным или словосочетанием, обозначающим информацию или объект**, который передаётся.

👉 Например: «Заказ клиента», «Отчёт о продажах», «Данные о сотрудниках».

Прочитайте задание и выполните его.

По ниже представленному описанию будет построена модель бизнес-процесса в нотации BPMN.

Процесс «Сформировать договор поставки» осуществляет менеджер. Процесс инициируется письмом клиента. Менеджер готовит форму договора поставки (пользовательская задача), заполняет позиции договора (пользовательская задача), проверяет договор (задача «Сценарий»), подписывает договор (ручная задача), после чего процесс завершается. В случае, если по результатам проверки договора выявлена ошибка, то менеджер заполняет договор поставки заново.

В модели бизнес-процесса будут применены элементы:



A)

Б)



В)



Укажите, какие названия из текстового описания будут иметь элементы А, Б, В с учетом правил наименования элемента «Задача».

Формат ответа:

А)

Б)

В);...

А) Подготовить форму договора поставки; Заполнить позиции договора

Б) Проверить договор

В) Подписать договор




Прочитайте задание и выполните его.

По ниже представленному описанию будет построена организационная модель и процессно-событийная модель.

ООО «Увлеченный турист», расположенный по адресу: г. Москва, Воробьевы горы, д. 3, реализует процесс «Выполнить заказ на перевозку пассажиров», который включает в себя следующие действия: принять заказ, определить маршрут, рассчитать стоимость, перевезти пассажиров. Заказ принимает менеджер отдела перевозок Ива-

нов В.В. Маршрут определяет менеджер транспортного отдела Петров К.К., стоимость рассчитывает бухгалтер Трепова Л.В., водитель Кузякин П.В. перевозит пассажиров.

В организационной модели и процессно-событийной модели будут применены элементы:

- А)  ???
- Б)  ???
- В)  ???

Укажите, какие названия из текстового описания будут иметь элементы А, Б, В.

Формат ответа:

- А)
- Б);...;...
- В);...;...;...




(

- А) Местоположение
- Б) Должность
- В) Внутренний сотрудник)

А) г. Москва, Воробьевы горы, д. 3

Б) Менеджер отдела перевозок; Менеджер транспортного отдела; Бухгалтер; Водитель

В) Иванов В.В.; Петров К.К.; Трепова Л.В.; Кузякин П.В.

ARIS Express и SILA UNION		
Элемент	ARIS Express	SILA Union
Location		
Organizational unit		
Role		
Person		

Прочитайте задание и выполните его.

В нотации BPMN есть элемент стартовое событие-таймер, в каких случаях его применяют для отражения начала процесса?

Стартовое событие-таймер (*Timer Start Event*) в нотации BPMN применяется в тех случаях, когда начало процесса определяется временем или расписанием, а не действием пользователя или внешнего события.

ОБОЗНАЧЕНИЕ СТАРТОВОГО
СОБЫТИЯ-ТАЙМЕР



Прочитайте задание и выполните его.

В нотации BPMN есть элемент конечное событие-сообщение, в каких случаях его применяют для отражения завершения процесса?

Элемент конечное событие-сообщение (Message End Event) в нотации BPMN используется тогда,

когда завершение процесса сопровождается отправкой сообщения внешнему участнику или другой системе.

Прочитайте задание и выполните его.

Перечислите пять маркеров действий в нотации BPMN, которые могут быть использованы для моделирования процесса.

- 1 Маркер цикла
- 2 Маркер Ad-hoc
- 3 Маркер Многоэкземплярности
- 4 Маркер Компенсации
- 5 Маркер подпроцесса

Прочитайте задание и выполните его.

Перечислите наименование логических операторов в методологии ARIS.

И, ИЛИ, Исключающий ИЛИ

Прочитайте задание и выполните его.

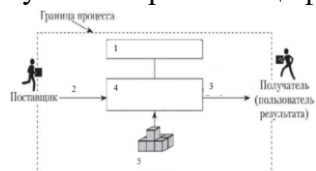
Укажите логические операторы, которые нельзя применить в данной схеме при построении процессно-событийной модели, и поясните почему нельзя.



Нельзя применять логическое ИЛИ, Исключающее ИЛИ, так как событие это свершившийся факт и событие не может принимать решение, то после наступления события при ветвлении может применяться только логическое И

Прочитайте задание и выполните его.

Перечислите 6 основных характеристик процесса (бизнес-процесса), пять из шести отражены на рисунке и скрыты за цифрами.



1 Владелец процесса

2 Вход

3 Выход(результат)

4 Процесс «название процесса»

5 Ресурсы

6 Цель процесса



Прочитайте задание и выполните его.

Перечислите наименование элементов нотации BPMN, отражённых на панели меню инструментального средства Stormbpmn



(проверено лично)

События (кружочки)

Начальное событие

Промежуточное/граничное событие

Конечное событие

Артефакты(справа)

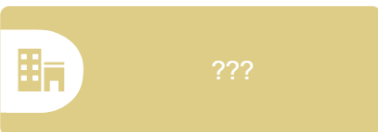
Объект данных

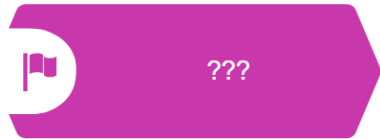
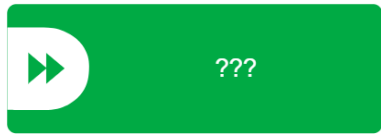
Хранилище данных

Пул/участник

Прочитайте задание и выполните его.

Последовательно перечислите название элементов методологии ARIS на русском языке, отраженных на панели меню инструментального средства SILA Union





Местоположение

Организационная единица

Функция

Событие

Прочитайте задание и выполните его.

Имеется 8 дисков по 4 ТБ каждый. Рассчитайте полезную емкость для конфигурации RAID 10. Опишите решение.

полезная ёмкость RAID10 = (количество дисков ÷ 2) × размер одного диска.
(8/2)*4=16

Нужно посчитать полезную ёмкость в RAID 10.

Что такое RAID 10

RAID 10 = RAID 1 + RAID 0.

1. Сначала диски объединяются в зеркальные пары (RAID 1). В паре данные дублируются: один диск — рабочий, второй — его полная копия. То есть ёмкость пары равна ёмкости одного диска, потому что второй тратится на зеркало.
2. Затем эти пары объединяются в RAID 0 (полосы/striping), чтобы сложить их объём и получить суммарную скорость.

Шаг 1. Разбей 8 дисков на пары

8 дисков → можно сделать 4 пары по 2 диска.

Каждая пара:

- 2×4 ТБ физически
- но из-за зеркалирования полезно только 4 ТБ

То есть одна пара = 4 ТБ полезно.

Шаг 2. Объедини все пары в RAID 0

Теперь у нас 4 пары по 4 ТБ полезно каждая.

RAID 0 просто складывает объём:

Общий объём = 4 ТБ + 4 ТБ + 4 ТБ + 4 ТБ

Посчитаем:

4 ТБ \times 4 пары = 16 ТБ

Ответ

Полезная ёмкость массива RAID 10 из 8 дисков по 4 ТБ составляет **16 ТБ**.

Прочитайте задание и выполните его.

Имеется 6 дисков по 2 ТБ каждый. Пять из них используются для создания массива RAID 5. Один диск добавляется в массив как горячий резерв (hot spare). Рассчитайте полезную ёмкость, доступную для записи данных. Опишите ход решения.

- Было 6×2 ТБ = 12 ТБ «железа» суммарно.
- Один диск — резерв \rightarrow остаётся 5 в RAID 5.
- В RAID 5 из 5 дисков полезны 4.
- 4×2 ТБ = 8 ТБ доступно для данных.

Прочитайте задание и выполните его.

Есть 6 дисков по 1 ТБ. Сравните полезную ёмкость для конфигураций RAID 5 и RAID 10. Опишите ход решения.

Давай сравним RAID 5 и RAID 10 на одинаковом «железе».

Исходные данные:

- Всего дисков: 6 штук
- Размер каждого диска: 1 ТБ

- Считаем полезную (доступную пользователю) ёмкость:
 - для RAID 5
 - для RAID 10

Часть 1. RAID 5

Как работает RAID 5:

- Данные «раскидываются» по дискам (чередование, striping).
- Плюс считается паритет (контрольная информация), который распределяется по всем дискам.
- Потеря одного диска допускается без потери данных.

Формула полезной ёмкости RAID 5:

Полезная ёмкость = $(N - 1) \times \text{размер диска}$,

где N — число дисков в массиве.

У нас $N = 6$.

Тогда:

- Полезная ёмкость = $(6 - 1) \times 1 \text{ ТБ}$
- Полезная ёмкость = $5 \times 1 \text{ ТБ}$
- Полезная ёмкость = **5 ТБ**

То есть в RAID 5 объём, эквивалентный одному диску (1 ТБ), уходит на паритет.

Часть 2. RAID 10

Как работает RAID 10:

- Сначала диски объединяются попарно в зеркала (RAID 1). Это дублирование: из двух дисков остаётся ёмкость одного.
- Потом эти пары объединяются в RAID 0 (чередование), то есть объёмы зеркальных пар складываются.

Важно: RAID 10 корректно строится из чётного количества дисков, разбитых на пары.

У нас 6 дисков → это 3 зеркальные пары.

Рассчитаем:

1 пара (2 диска по 1 ТБ):

- физически: 2 ТБ
- полезно: 1 ТБ (потому что второй диск — копия первого)

У нас 3 такие пары:

- Пара 1 → 1 ТБ полезно
- Пара 2 → 1 ТБ полезно
- Пара 3 → 1 ТБ полезно

Объединяем пары в RAID 0 → объём просто складывается:

1 ТБ + 1 ТБ + 1 ТБ = **3 ТБ**

Итого для RAID 10 с 6 дисками по 1 ТБ:

- Полезная ёмкость = **3 ТБ**

Часть 3. Сравнение

- RAID 5 из 6×1 ТБ → полезно **5 ТБ**
- RAID 10 из 6×1 ТБ → полезно **3 ТБ**

Вывод:

- RAID 5 даёт больше полезного пространства (5 ТБ против 3 ТБ), потому что он хранит только один диск «в паритете», а не делает полную копию каждого диска.
- RAID 10 жертвует половиной общего объёма ради большей отказоустойчивости и скорости записи.

Прочитайте задание и выполните его.

Опишите последовательность действий, чтобы в ОС Astra Linux сконфигурировать службу удаленного доступа SSH.

Укажите, какие именно файлы необходимо отредактировать и какие параметры в них прописать чтобы разрешить прослушивать порт 22, и прослушивать IP-адрес 10.0.10.15

Какую команду надо ввести, чтобы изменения применились? (Вы действуете уже как root-пользователь)

Для конфигурирования службы удалённого доступа SSH в ОС Astra Linux необходимо:

1. Убедиться, что установлен пакет **openssh-server** (команда `apt install openssh-server`).
2. Отредактировать файл конфигурации **/etc/ssh/sshd_config**.
3. Указать в нём параметры:
4. Port 22
5. ListenAddress 10.0.10.15

Эти строки разрешают прослушивание порта 22 и IP-адреса 10.0.10.15.

6. Сохранить изменения и перезапустить службу SSH командой:

<div>7. systemctl restart ssh</div> <div>После этого служба SSH будет активна и доступна по адресу 10.0.10.15:22.</div>	
<div>Прочитайте задание и выполните его.</div> <div>Напишите, какие команды необходимо ввести в командную строку в ОС Astra Linux, чтобы перезагрузить фаервол ufw командами (Вы действуете уже как root-пользователь)</div> <div>Ответ: Чтобы перезагрузить фаервол ufw в ОС Astra Linux (действуя от имени root), нужно выполнить следующие команды: ufw disable ufw enable</div>	
<div>Пояснение:</div> <div><ul style="list-style-type: none">• Команда ufw disable — отключает фаервол и сбрасывает активные правила.• Команда ufw enable — снова включает фаервол и применяет текущие правила из конфигурации.</div> <div>Таким образом, последовательность этих двух команд приводит к перезапуску (перезагрузке) ufw.</div>	
<div>Прочитайте задание и выполните его.</div> <div>Опишите фундаментальное различие в механизме выделения дискового пространства между толстыми LUN и тонкими LUN с точки зрения процедуры записи данных.</div> <div>Фундаментальное различие между толстыми (thick) и тонкими (thin) LUN заключается в том, когда и как выделяется физическое дисковое пространство при записи данных.</div>	
<div>◆ Толстый LUN (Thick LUN)</div>	

- Всё дисковое пространство выделяется **заранее** в момент создания LUN.
- Даже если пользователь записал только часть данных, система уже **резервирует весь объём** под этот LUN.
- При записи данные помещаются **в заранее выделенные блоки**, поэтому задержки минимальны.
- Преимущество: высокая производительность и предсказуемое поведение.
- Недостаток: низкая эффективность использования дискового пространства (часть может оставаться неиспользованной).

◆ Тонкий LUN (Thin LUN)

- Физическое пространство выделяется **динамически, только при фактической записи данных**.
- Изначально LUN создаётся “логически” большого размера, но физически занимает очень мало места.
- При каждой новой записи система **дополнительно выделяет** необходимые блоки.
- Преимущество: экономия пространства, эффективное использование дисков.
- Недостаток: возможны **задержки при записи** (из-за выделения блоков “на лету”) и риск переполнения пула хранения, если резерв недостаточен.

✓ Итог:

- **Thick LUN** — пространство выделяется **сразу**, запись происходит в заранее зарезервированные блоки.
- **Thin LUN** — пространство выделяется **по мере записи**, что экономит место, но может снижать скорость.

Прочитайте задание и выполните его.

Почему использование массивов RAID любого уровня (кроме RAID 0) не отменяет необходимости создавать резервные копии данных? Приведите не менее трех причин.

Использование RAID-массивов (кроме RAID 0) действительно повышает отказоустойчивость и надёжность хранения данных, однако **не заменяет резервное копирование**. Основные причины следующие:

1 Вероятность логических ошибок и удаления данных
RAID защищает только от **физического отказа дисков**, но не от ошибок пользователя:

- случайное удаление или перезапись файлов,
- повреждение файловой системы,
- действие вирусов или вредоносного ПО.

При таких сбоях данные удаляются **на всех дисках массива одновременно**, и восстановить их невозможно без резервной копии.

2 Отказ контроллера или самого массива

Если выйдет из строя **RAID-контроллер**, произойдёт сбой прошивки или ошибка конфигурации массива, доступ ко всем данным может быть потерян, даже если все диски физически исправны.

Резервная копия позволяет восстановить данные независимо от оборудования.

3 Массовые сбои оборудования или катастрофы

RAID не защищает от внешних факторов:

- пожар, затопление, электроперепады, кража,
- вирус-шифровальщик, уничтожающий все данные, включая зеркала.

В этих случаях теряется весь массив целиком, и восстановить данные можно только из резервной копии, хранящейся отдельно.

✓ Итог:

RAID обеспечивает **непрерывность работы и защиту от аппаратных сбоев**, но **резервное копирование** остаётся необходимым для защиты от **логических, программных и катастрофических потерь данных**.

Прочитайте задание и выполните его.

Опишите принципиальное различие между сетевыми архитектурами SAN и NAS с точки зрения уровня доступа к данным

Главное различие между архитектурами **SAN (Storage Area Network)** и **NAS (Network Attached Storage)** заключается в **уровне доступа к данным** — то есть, как именно сервер или клиент взаимодействует с хранилищем.

◆ SAN — блочный доступ

- SAN предоставляет **блочный доступ** к данным.
- Операционная система сервера видит LUN (Logical Unit Number) как **локальный диск**.
- Управление файловой системой, создание разделов и файлов — выполняется **на стороне сервера**.
- Протоколы: **Fibre Channel (FC)**, **iSCSI**, **FCoE**, **NVMe-oF**.
- Используется в высокопроизводительных и критичных системах (БД, виртуализация).

■ *Пример:* сервер обращается к хранилищу как к «сырым» дисковым блокам (например, /dev/sdb), сам создаёт на них файловую систему (ext4, XFS и т.д.).

◆ NAS — файловый доступ

- NAS предоставляет **файловый доступ** к данным.
- Сервер или пользователь получает доступ к **файлам и каталогам**, а не к физическим блокам.
- Управление файловой системой выполняется **на стороне самого NAS**.
- Протоколы: **NFS (Linux/Unix)**, **SMB/CIFS (Windows)**, **FTP**.
- Удобен для общего доступа пользователей и хранения документов.

■ *Пример:* клиент монтирует сетевую папку вроде \\NAS\Shared или /mnt/nas, и работает с файлами через сетевой протокол.

✓ Итоговое различие:

- SAN — блочный уровень доступа (как к диску).

- **NAS** — файловый уровень доступа (как к сетевой папке).

Прочитайте задание и выполните его.

Опишите два основных физических интерфейса, используемых для подключения DAS-системы к серверу. Какой из них обеспечивает лучшую производительность и масштабируемость?

Ответ:

Для подключения **DAS (Direct Attached Storage)** — напрямую подключаемых систем хранения — к серверу используются два основных физических интерфейса:

◆ 1. SATA (Serial ATA)

- Применяется в недорогих и массовых системах.
- Обеспечивает скорость передачи данных до **6 Гбит/с (SATA III)**.
- Подходит для подключения одиночных жёстких дисков или SSD.
- Прост в реализации, но имеет **ограниченные возможности масштабирования** — каждый диск требует отдельного порта.

◆ 2. SAS (Serial Attached SCSI)

- Используется в серверных и корпоративных системах.
- Скорость передачи данных достигает **12 Гбит/с и выше**.
- Поддерживает **многоадресное подключение (multidrop)** — к одному контроллеру можно подключать **до сотен устройств**.
- Обеспечивает более надёжную передачу данных, возможность дублирования путей (multipath) и лучшую интеграцию с RAID-контроллерами.

✓ **Вывод:**

Из двух интерфейсов SAS обеспечивает более высокую производительность и масштабируемость, чем SATA, поэтому чаще используется в корпоративных DAS-системах.

Прочитайте задание и выполните его.

Для работы с групповыми политиками необходимо, чтобы компьютер был членом домена. Опишите, как добавить компьютер в домен (название домена: admins.com)

Чтобы добавить компьютер в домен **admins.com**, необходимо выполнить следующие шаги (действия выполняются от имени администратора):

◆ 1. Открыть настройки системы

1. Щёлкните правой кнопкой мыши по пункту «Этот компьютер» → выберите «Свойства».
2. В открывшемся окне нажмите «Изменить параметры» (справа, в разделе «Имя компьютера, домен и рабочая группа»).

◆ 2. Изменить параметры имени компьютера

1. В окне «Свойства системы» нажмите кнопку «Изменить».
2. В разделе «Член домена» (Domain) выберите пункт «Домен» и введите имя домена:
3. admins.com

◆ 3. Ввести учетные данные администратора домена

После нажатия ОК появится запрос имени и пароля пользователя, который имеет права добавлять компьютеры в домен (например, *Administrator* домена).

Введите соответствующие учётные данные.

◆ 4. Перезагрузить компьютер

После успешного присоединения к домену появится сообщение:

Добро пожаловать в домен admins.com.

Нажмите **ОК** и **перезагрузите компьютер** для применения изменений.

◆ 5. Проверка

После перезагрузки при входе в систему можно выбрать вход в домен **admins.com** (например, **ADMINS\username**) вместо локальной учётной записи.

☑ Итог:

Компьютер добавляется в домен **admins.com** через свойства системы → «Изменить параметры» → «Домен» → ввод имени домена и учётных данных администратора → перезагрузка системы.

Прочитайте задание и выполните его.

При настройке сервера часто требуется статический IP-адрес. Опишите последовательность действий для настройки статического IP-адреса в ОС Astra Linux через редактирование конфигурационных файлов. Укажите, какие именно файлы необходимо изменить и какие параметры в них прописать для интерфейса eth0 (адрес: 192.168.1.10/24, шлюз: 192.168.1.1)
(Вы действуете уже как root-пользователь)

Ответ:

Для задания **статического IP-адреса** в ОС **Astra Linux** (на базе Debian) необходимо отредактировать сетевой конфигурационный файл и задать нужные параметры вручную.

◆ 1. Открыть конфигурационный файл сети

Основной файл конфигурации сетевых интерфейсов:

`/etc/network/interfaces`

Открыть его можно любым текстовым редактором от имени root, например:

`nano /etc/network/interfaces`

◆ 2. Прописать настройки для интерфейса eth0

Удалите (или закомментируйте) старые строки, если они есть, и добавьте следующие параметры:

```
auto eth0
iface eth0 inet static
    address 192.168.1.10
    netmask 255.255.255.0
    gateway 192.168.1.1
```

Пояснение параметров:

- auto eth0 — интерфейс будет подниматься автоматически при загрузке системы;
- iface eth0 inet static — указывает, что используется статическая настройка IPv4;
- address — IP-адрес интерфейса;
- netmask — маска подсети (для /24 это 255.255.255.0);
- gateway — основной шлюз по умолчанию.

3. (Необязательно) указать DNS-серверы

Для работы с интернетом желательно добавить DNS-серверы в файл:
/etc/resolv.conf
Например:
nameserver 8.8.8.8
nameserver 1.1.1.1

4. Применить изменения

После сохранения конфигурации нужно перезапустить сетевые службы:
systemctl restart networking
или вручную перезапустить интерфейс:
ifdown eth0 && ifup eth0

Итог:

Этап	Действие	Команда / файл
1	Открыть конфигурацию сети	/etc/network/interfaces
2	Прописать параметры	address 192.168.1.10net
3	(Опционально) добавить DNS	/etc/resolv.conf
4	Применить настройки	systemctl restart netwo

После выполнения этих действий интерфейс **eth0** будет иметь статический IP-адрес **192.168.1.10/24** и шлюз **192.168.1.1**.

Прочитайте задание и выполните его.

Есть 6 дисков по 2 ТБ. Сравните полезную емкость для конфигураций RAID 0 и RAID 10. Опишите ход решения.

Исходные данные:

- Количество дисков: **6**
- Объем каждого диска: **2 ТБ**

Нужно сравнить **полезную ёмкость** (доступную для записи данных) для двух конфигураций: **RAID 0** и **RAID 10**.

◆ 1. RAID 0 — чередование без избыточности

Принцип работы:

Данные распределяются (чередуются) между всеми дисками без резервирования.

Нет защиты от отказа дисков, но достигается максимальная скорость и 100% использование объема.

Расчёт:

Полезная ёмкость = количество дисков × объем одного диска
 $6 \times 2 \text{ ТБ} = 12 \text{ ТБ}$

✓ RAID 0: полезная ёмкость = 12 ТБ

◆ 2. RAID 10 — комбинация зеркалирования и чередования (RAID 1 + RAID 0)

Принцип работы:

- Сначала диски объединяются в зеркальные пары (RAID 1), где половина объема уходит на дублирование.
- Затем пары объединяются в RAID 0, чтобы увеличить общую скорость и суммарный объем.

Так как RAID 10 требует чётное количество дисков, при 6 дисках мы получаем **3 пары**.

Расчёт:

Каждая пара: 2 диска по 2 ТБ → полезно 2 ТБ

3 пары × 2 ТБ = 6 ТБ

✓ RAID 10: полезная ёмкость = 6 ТБ

⚖ Сравнение:

Конфигурация

Защита данных

RAID 0

Нет

RAID 10

Есть (может выдержать отказ 1 диска в ка

Вывод:

- RAID 0 даёт максимальный объём (12 ТБ), но не защищает данные — при отказе одного диска теряется всё.
- RAID 10 обеспечивает отказоустойчивость, но полезная ёмкость составляет только половину от общего объёма — 6 ТБ.

Прочитайте задание и выполните его.

Есть 2 диска по 2 ТБ. Сравните полезную ёмкость для конфигураций RAID 0 и RAID 1. Опишите ход решения.

Разберём по шагам 📌

Исходные данные:

- Количество дисков: 2
- Объём каждого диска: 2 ТБ

Нужно сравнить полезную ёмкость (доступный объём для данных) в конфигурациях RAID 0 и RAID 1.

◆ 1. RAID 0 — чередование без избыточности

Принцип работы:

RAID 0 распределяет данные между всеми дисками (striping).

Нет зеркалирования и резервирования — используется весь объём.

При отказе одного диска теряются все данные.

Расчёт:

Полезная ёмкость = Количество дисков × Объём одного диска

$$2 \times 2\text{ТБ} = 4\text{ТБ}$$

✓ RAID 0: полезная ёмкость = 4 ТБ

◆ 2. RAID 1 — зеркалирование

Принцип работы:

RAID 1 хранит полную копию данных на каждом из двух дисков.

При выходе из строя одного диска данные сохраняются, но объём уменьшается вдвое.

Расчёт:

Полезная ёмкость = (Количество дисков ÷ 2) × Объём одного диска

$$(2 \div 2) \times 2\text{ТБ} = 2\text{ТБ}$$

✓ RAID 1: полезная ёмкость = 2 ТБ

⚖ Сравнение

Конфигурация	Защита данных	Полезная ёмкость
RAID 0	Нет	4 ТБ
RAID 1	Да (зеркало)	2 ТБ

Вывод:

- RAID 0 обеспечивает максимальную ёмкость (4 ТБ) и скорость, но не защищает данные.
- RAID 1 уменьшает объём наполовину (2 ТБ), но обеспечивает надёжность и сохранность информации при отказе одного диска.

Прочитайте задание и выполните его.

Опишите последовательность действий для временного отключения сетевого интерфейса eth0 и его последующего включения с уже обновленной конфигурацией из файла /etc/network/interfaces. Какие команды для этого используются? (вы вводите команды от root-пользователя)

Ответ:

Для временного отключения и повторного включения сетевого интерфейса **eth0** с применением обновлённой конфигурации из файла `/etc/network/interfaces` выполняются следующие шаги (от имени **root**):

◆ 1. Отключить интерфейс **eth0**

Команда:

```
ifdown eth0
```

○ Эта команда **выключает (деактивирует)** сетевой интерфейс и разрывает сетевое соединение.

◆ 2. Внести (или убедиться, что внесены) изменения в конфигурацию

Файл конфигурации сети:

```
/etc/network/interfaces
```

В нём можно изменить IP-адрес, шлюз, DNS и другие параметры для интерфейса **eth0**.

◆ 3. Включить интерфейс заново

Команда:

```
ifup eth0
```

○ Эта команда **включает интерфейс** и применяет все изменения, прописанные в `/etc/network/interfaces`.

◆ 4. (Альтернатива — одной командой)

Чтобы перезапустить интерфейс одной строкой:

```
ifdown eth0 && ifup eth0
```

☑ **Итог:**

Команды для выполнения:

```
ifdown eth0
```

```
ifup eth0
```

или одной строкой:

```
ifdown eth0 && ifup eth0
```

Эти действия временно отключают сетевой интерфейс **eth0** и затем включают его снова с обновлённой конфигурацией.

Прочитайте задание и выполните его.

Вам необходимо выбрать тип СХД (DAS, SAN, NAS). Сформулируйте ключевые преимущества, которые помогут осуществить выбор.

Ответ:

Выбор типа системы хранения данных (СХД) — DAS, SAN или NAS — зависит от требований к производительности, масштабируемости и способу доступа к данным. Ниже приведены ключевые преимущества каждой архитектуры, которые позволяют сделать правильный выбор.

◆ 1. DAS (Direct Attached Storage) — напрямую подключаемое хранилище

Ключевые преимущества:

- Простота установки и настройки — подключается напрямую к серверу.
- Низкая стоимость оборудования и обслуживания.
- Высокая производительность при работе с одним сервером (нет сетевой задержки).
- Оптимально для небольших систем, рабочих станций или локальных серверов.

■ Выбор в пользу DAS:

если требуется простое, недорогое и быстрое локальное хранилище без сетевого доступа.

◆ 2. NAS (Network Attached Storage) — файловое сетевое хранилище

Ключевые преимущества:

- Предоставляет файловый доступ (SMB/CIFS, NFS) — удобно для общего доступа пользователей.
- Централизованное хранение данных и простое управление.
- Поддержка резервного копирования, прав доступа и многопользовательской работы.
- Подходит для офисных сетей, рабочих групп и организаций, где важен общий доступ к файлам.

■ Выбор в пользу NAS:

если приоритет — совместный доступ и простота администрирования в локальной сети.

◆ 3. SAN (Storage Area Network) — сетевая система блочного уровня

Ключевые преимущества:

- Высочайшая производительность и минимальные задержки при передаче данных.
- Масштабируемость — можно подключать десятки серверов и сотни устройств хранения.
- Блочный доступ — ОС видит тома как локальные диски (оптимально для баз данных, виртуализации, критичных систем).
- Поддержка отказоустойчивости и распределения нагрузки.

■ Выбор в пользу SAN:

если требуется высокопроизводительное, масштабируемое и отказоустойчивое хранилище для дата-центра или корпоративной инфраструктуры.

✓ Итоговое сравнение:

Тип СХД	Уровень доступа	Основные преимущества
DAS	Блочный (локальный)	Простота, низкая цена, высокая производительность
NAS	Файловый (через сеть)	Централизованный доступ, простота использования
SAN	Блочный (через сеть)	Высокая производительность, отказоустойчивость

Вывод:

Выбор типа СХД определяется задачей:

- Для простых и недорогих решений — DAS.
- Для общего доступа к файлам — NAS.
- Для высоконагруженных корпоративных систем — SAN.

Прочитайте задание и выполните его

Опишите поэтапно правильную последовательность действий при разработке программного

продукта методологией Test Driven Development (TDD).

1. Сформулировать требование и написать автоматизированный тест, описывающий ожидаемое поведение, до написания реализации.
2. Запустить тесты и убедиться, что новый тест падает (то есть он действительно улавливает отсутствие нужной функциональности).
3. Написать минимально достаточную реализацию кода, чтобы удовлетворить этот тест.
4. Повторно запустить тесты и убедиться, что теперь все тесты проходят (новый и существующие).
5. Выполнить рефакторинг кода, улучшая его структуру без изменения внешнего поведения, при постоянном контроле тестами.
6. Перейти к следующему требованию и повторить цикл.

Это и есть классический цикл TDD «Red → Green → Refactor».

Прочитайте задание и выполните его

Опишите поэтапно правильную последовательность действий при разработке программного продукта методологией Behavior Driven Development (BDD).

1. Совместно с заказчиком и тестированием уточняется ожидаемое поведение системы на уровне бизнес-сценариев.
2. Это поведение формализуется в виде читаемых спецификаций (обычно в формате Given–When–Then).
3. Сценарии преобразуются в автоматизируемые тесты: для каждого шага пишется код шага.
4. Сценарии запускаются и изначально падают, подтверждая, что они действительно проверяют требуемую функциональность.
5. Реализуется функциональность приложения, чтобы удовлетворить этим сценариям.

6. Сценарии запускаются повторно и должны пройти успешно, подтверждая корректное поведение системы.

7. Выполняется рефакторинг кода приложения и тестовой базы при сохранении зелёных сценариев.

8. Сценарии сохраняются как часть регрессионного набора и живой спецификации системы.

Таким образом, в BDD разработка идёт “от поведения, понятного бизнесу” к коду, а не наоборот.

Прочитайте задание и выполните его

Раскройте понятие «тестовое покрытие» или «покрытие тестами» (test coverage)

Тестовое покрытие — это показатель полноты тестирования, выражающий, какая часть кода, функций или требований системы была проверена с помощью тестов. Оно помогает оценить качество тестирования и определить, какие участки программы остаются непроверенными.

Прочитайте задание и выполните его

Опишите, в чем заключается характеристика пригодности продукта для автоматического тестирования.

Пригодность продукта для автоматического тестирования — это характеристика качества программного обеспечения, отражающая, насколько система позволяет эффективно разрабатывать, запускать и поддерживать автоматические тесты.

Продукт считается пригодным для автоматизации, если он имеет чётко определённые интерфейсы, детерминированное поведение, доступные средства наблюдения и управления, возможность изоляции компонентов и повторяемости результатов.

Прочитайте задание и дайте развернутый ответ.

В чем заключается принцип работы «monkey»-тестирования?

Принцип работы monkey-тестирования заключается в том, что приложению намеренно подаются случайные, непредсказуемые и зачастую некорректные пользовательские действия и входные данные с высокой интенсивностью, без заранее подготовленных тест-кейсов. Цель — проверить устойчивость системы: не падает ли она, не зависает ли интерфейс, не возникают ли утечки памяти и неконтролируемые состояния при хаотичном использовании. По сути, этот подход имитирует “абсолютно неквалифицированного и неряшливого пользователя”, выявляя дефекты надёжности и обработки ошибок, которые не всегда обнаруживаются традиционным сценарным тестированием.

Прочитайте задание и дайте развернутый ответ.

Опишите разницу тестирования альфа-версии и бета-версии продукта.

Альфа-тестирование — это внутреннее тестирование продукта, проводимое командой разработчиков и тестировщиков в контролируемой среде на раннем этапе. Цель альфа-тестирования — выявить критические дефекты, нестабильность и логические ошибки, убедиться, что основные функции в принципе работают. На этом этапе продукт может быть неполным, с черновой реализацией и вспомогательными заглушками.

Бета-тестирование — это внешний этап тестирования, выполняемый реальными пользователями или ограниченной группой клиентов в условиях, приближённых к боевой эксплуатации. Цель бета-тестирования — проверить удобство и применимость продукта в реальной среде, собрать обратную связь от пользователей, выявить проблемы совместимости, производительности и сценариев использования, о которых разработчики могли не подумать. К моменту беты продукт

обычно уже почти готов к релизу и должен работать достаточно стабильно.

Короче: альфа — проверка “работает ли вообще и не падает?”, бета — “подходит ли это живым людям и можно ли это выпускать в реальный мир?”.

Прочитайте задание и дайте развернутый ответ.

Раскройте понятие блокирующих ошибок (blocker).

Блокирующие ошибки (blocker) — это дефекты максимального уровня серьезности, из-за которых невозможно продолжать тестирование или эксплуатацию системы. Такая ошибка делает недоступным ключевой функционал или полностью выводит приложение из работоспособного состояния (например, приложение не запускается, нельзя войти, нельзя выполнить основной бизнес-сценарий, происходит немедленный краш). Пока блокирующая ошибка не устранена, сборка не может считаться пригодной для дальнейшего тестирования и тем более для выпуска.

Пример 1. Приложение не стартует

Вы запускаете десктопное приложение, и оно тут же падает с ошибкой “Fatal error” и закрывается.

Тестирование всего приложения невозможно.

Пример 2. Бесконечный логин

После ввода логина и пароля — вечная загрузка без входа внутрь.

Если без входа в систему нельзя тестировать остальной функционал, это blocker.

Прочитайте задание и дайте развернутый ответ.

Опишите, в чем заключается метод тестирования совместимости программных продуктов.

Метод тестирования совместимости заключается в проверке того, что программный продукт корректно работает в различных целевых средах эксплуатации, предусмотренных требованиями. На практике это означает прогон ключевых

пользовательских сценариев и проверку отображения интерфейса на разных платформах: различных версиях операционных систем, браузеров, устройств, драйверов, а также при взаимодействии с разными версиями внешних сервисов, библиотек и баз данных.

Тестирование совместимости включает подготовку матрицы поддерживаемых окружений, развёртывание продукта в каждой из этих сред и сравнение поведения, чтобы выявить платформенно-зависимые дефекты — случаи, когда функционал доступен в одном окружении, но не работает или искажён в другом.

Ценность этого метода в том, что он подтверждает, что продукт может использоваться реальными пользователями с разными конфигурациями, уменьшает риски инцидентов после релиза и доказывает соответствие заявленным требованиям по поддерживаемым платформам.

Прочитайте задание и дайте развернутый ответ.

Опишите, что необходимо оценить для определения уровня тестирования ПП.

Чтобы определить уровень тестирования программного продукта, необходимо оценить:

1. **Степень готовности продукта** — что у нас есть: отдельный модуль, связка модулей, целая система или почти готовое решение для заказчика.
2. **Границы тестируемого объекта** — тестируем ли мы изолированный компонент, их взаимодействие, систему целиком или поведение с точки зрения бизнеса.
3. **Цель тестирования на данном шаге** — ищем ошибки реализации, ошибки интеграции, ошибки системы в целом или подтверждаем соответствие ожиданиям заказчика.
4. **Наличие и зрелость тестового окружения** — можем ли мы развернуть один модуль, несколько модулей вместе, всю систему или приближённую к боевой среду заказчика.
5. **Уровень требований, с которым сравнивается результат** — технические (логика кода), интер-

фейсные (контракты между компонентами), системные (функциональность и надёжность), бизнес-требования (готовность к вводу в эксплуатацию).

6. Критичные риски текущего этапа разработки — что сейчас нужно снять в первую очередь: сбой алгоритма, сбой взаимодействия, сбой всей системы, несоответствие бизнесу.

7. Состав участников тестирования — только разработчики, несколько команд, централизованная QA-команда или уже заказчик/конечный пользователь.

Прочитайте задание и дайте развернутый ответ.

О чём свидетельствует повышение показателя KIVD (оценка качества выявления дефектов), который характеризует уровень работ по ликвидации обнаруженных при тестировании дефектов.

Повышение показателя KIVD свидетельствует о росте эффективности и качества работ по устранению дефектов, найденных в ходе тестирования. Это означает, что команда не просто фиксирует ошибки, но своевременно и корректно их исправляет; дефекты реже переоткрываются; снижается количество нерешённых и повторяющихся проблем; улучшается взаимодействие между тестировщиками и разработчиками. В результате повышается общая стабильность программного продукта и снижается риск выпуска версии с известными или плохо исправленными ошибками.

Прочитайте задание, приведите формулы для расчёта и решение

Произведите расчёт оценки полноты совокупности тестов (IPSTS), если

Nts – количество дефектов, обнаруженных при выполнении тестовых случаев равно 75

N_{pts} — количество дефектов, обнаруженных при выполнении всего тестового цикла 80

N_r - количество дефектов, обнаруженных при эксплуатации ПП равно 20

При необходимости использовать калькулятор.

$$IPSTS = \frac{N_{ts} + N_{pts}}{N_{ts} + N_{pts} + N_r} \times 100\%$$

где:

- N_{ts} — количество дефектов, найденных при выполнении тестовых случаев;
- N_{pts} — количество дефектов, обнаруженных в ходе всего тестового цикла (включая дополнительные проверки);
- N_r — количество дефектов, выявленных уже **после тестирования**, при эксплуатации программного продукта.

$$N_{ts} = 75, \quad N_{pts} = 80, \quad N_r = 20$$

$$IPSTS = \frac{75 + 80}{75 + 80 + 20} \times 100\%$$

$$IPSTS = \frac{155}{175} \times 100\%$$

= 88.6%

Прочитайте задание и дайте ответ.

Раскройте сущность тестирования программного обеспечения

Тестирование программного обеспечения — это процесс проверки и оценки качества программы, направленный на выявление дефектов, проверку соответствия системы требованиям и обеспечение её надёжности и корректной работы.

Оно проводится систематически на всех этапах разработки для того, чтобы минимизировать ошибки, снизить риски и обеспечить выпуск качественного продукта, удовлетворяющего потребности пользователя.

Прочитайте задание и дайте ответ.

Перечислите основные этапы процесса тестирования программного обеспечения.

перечисление этапов

1. Планирование тестирования
2. Проектирование и разработка тестов
3. Подготовка тестового окружения
4. Выполнение тестов
5. Анализ и документирование результатов
6. Повторное и регрессионное тестирование
7. Завершение тестирования и отчётность

Прочитайте задание, приведите формулы для расчёта и решение

Найдите значение коэффициента ошибок $K_{\text{опп}}$, пропущенных на продуктив, если после выпуска релиза в программе было обнаружено 4 ошибки, а всего во время тестирования программы и после ее выпуска было обнаружено 10 ошибок.

При необходимости использовать калькулятор.

$$K_{\text{опп}} = \frac{N_p}{N_{\text{общ}}} \times 100\%$$

где:

- N_p — количество дефектов, обнаруженных после выпуска релиза (на продуктиве);
- $N_{\text{общ}}$ — общее количество дефектов, найденных во время тестирования и после выпуска, то есть

$$N_{\text{общ}} = N_{\text{тест}} + N_p$$

$$K_{\text{опт}} = \frac{4}{10} \times 100\%$$

Прочитайте задание и выполните его

В процессе тестирования была выявлена следующая серьезная ошибка: при сохранении документа в формате PDF вместо исходного формата (например, DOCX), текст и форматирование документа теряются. Это означает, что пользователи могут потерять важные данные и форматирование документов, когда они попытаются сохранить их в PDF.

Опишите процесс исправления дефекта.

Процесс исправления дефекта включает следующие шаги:

- (1) регистрация дефекта →
- (2) анализ причины →
- (3) внесение исправлений →
- (4) код-ревью →
- (5) повторное тестирование →
- (6) регрессионное тестирование →
- (7) документирование и закрытие дефекта.

После успешного прохождения всех этапов функция сохранения в PDF становится устойчивой и безопасной для пользователей.