ПРИЛОЖЕНИЕ

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«РЯЗАНСКИЙ ГОСУДАРСТВЕННЫЙ РАДИОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ
В.Ф. УТКИНА»

Кафедра «Систем автоматизированного проектирования вычислительных средств»

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ

Б1.В.16 «Технологии Интернета вещей»

Направление подготовки
09.03.01 Информатика и вычислительная техника

Направленность (профиль) подготовки
«Интеллектуальные системы и технологии»

Квалификация выпускника – бакалавр

Форма обучения – очная

Рязань 2026

Фонд оценочных средств – это совокупность учебно-методических материалов (контрольных заданий, описаний форм и процедур), предназначенных для оценки качества освоения обучающимися данной дисциплины как части основной образовательной программы.

Цель – оценить соответствие знаний, умений и уровня приобретенных компетенций, обучающихся целям и требованиям основной образовательной программы в ходе проведения текущего контроля и промежуточной аттестации.

Основная задача – обеспечить оценку уровня сформированности общекультурных и профессиональных компетенций, приобретаемых обучающимся в соответствии с этими требованиями.

Контроль знаний обучающихся проводится в форме текущего контроля и промежуточной аттестации.

Текущий контроль успеваемости проводится с целью определения степени усвоения учебного материала, своевременного выявления и устранения недостатков в подготовке обучающихся и принятия необходимых мер по совершенствованию методики преподавания учебной дисциплины (модуля), организации работы обучающихся в ходе учебных занятий и оказания им индивидуальной помощи.

К контролю текущей успеваемости относятся проверка знаний, умений и навыков обучающихся: по результатам выполнения обучающимися индивидуальных заданий в ходе выполнения практических занятий.

При оценивании (определении) результатов освоения дисциплины применяется шкала оценки "зачет — незачет".

В качестве оценочных средств на протяжении семестра используются практические задания.

Результат выполнения каждого практического занятия оценивается как "зачет" в случае выполнения обучающимся всех индивидуальных заданий.

Результат выполнения каждого индивидуального задания должен соответствовать всем критериям оценки в соответствии с компетенциями, установленными для заданного раздела дисциплины.

По итогам курса обучающиеся сдают теоретический зачет. Форма проведения теоретического зачета – устный ответ по списку вопросов, сформулированных с учетом содержания учебной дисциплины.

Паспорт фонда оценочных средств по дисциплине

№ п/п	Контролируемые разделы (темы) дисциплины	Код контролируемой компетенции (или её части)	Вид, метод, форма оценочного мероприятия
1	Введение в IoT: архитектура и стандарты	ПК-24.1 (LC-5.1, Б)	текущий контроль
2	Сенсорные технологии и устройства сбора данных	ПК-18.1 (PL-4.1, Б) ПК-24.1 (LC-5.1, Б)	текущий контроль, выполнение ПЗ
3	Протоколы связи в IoT	ПК-24.1 (LC-5.1, Б) ПК-24.2 (LC-5.2, Б)	текущий контроль, выполнение ПЗ

4	Платформы и облачные сервисы для IoT	ПК-18.1 (PL-4.1, Б) ПК-18.2 (PL-4.2, Б) ПК-24.1 (LC-5.1, Б) ПК-24.2 (LC-5.2, Б)	текущий контроль, выполнение ПЗ
5	Хранение и обработка IoT данных	ПК-24.1 (LC-5.1, Б) ПК-24.2 (LC-5.2, Б)	текущий контроль, выполнение ПЗ
6	Интеграция IoT с ИИ и машинным обучением	ПК-24.1 (LC-5.1, Б) ПК-24.2 (LC-5.2, Б)	текущий контроль, выполнение ПЗ
7	Безопасность и надёжность IoT систем	ПК-24.2 (LC-5.2, Б)	текущий контроль, выполнение ПЗ
8	Внедрение и сопровождение IoT решений	ПК-24.1 (LC-5.1, Б) ПК-24.2 (LC-5.2, Б)	текущий контроль, выполнение ПЗ
11	Промежуточная аттестация	ПК-18.1 (PL-4.1, Б) ПК-18.2 (PL-4.2, Б) ПК-24.1 (LC-5.1, Б) ПК-24.2 (LC-5.2, Б)	сдача экзамена

Показатели и критерии обобщенных результатов обучения

Результаты обучения по дисциплине	Показатели оценки результата	Критерии оценки результата
ПК-18.1 (PL-4.1, уровень Б) Разрабатывает и отлаживает эффективные многопоточные решения на C++, тестирует, испытывает и оценивает качество таких решений. **Знать** - модель памяти C++ и атомарные операции, семантики упорядочивания для корректной работы с данными в потоках; - примитивы синхронизации (мьютексы, условные переменные, семафоры) и типичные ошибки их применения; - особенности IoT-среды (ограниченные ресурсы, работа с периферией, требования к энергопотреблению) для адаптации решений под реальные устройства. **Уметь** - проектировать потокобезопасные структуры данных и выбирать подходящие механизмы синхронизации для гарантированной и корректной	• Понимание модели памяти C++ и атомарных операций, примитивов синхронизаци, специфики IoT-среды. • Умение проектировать потокобезопасные структуры данных, декомпозировать задачи и балансировать нагрузку. • Владение инструментами профилирования и диагностики, типовыми паттернами для IoT, практиками безопасной	*Удовлетворительно***:** код работает на типовых сценариях, есть базовые тесты и понимание примитивов. Есть отдельные ошибки в синхронизации, но студент способен их исправить при подсказке. *Хорошо***:** решение корректно, устойчиво к большинству ошибок, есть профилирование и стресс-тесты. Студент обоснованно выбирает механизмы синхронизации и адаптирует решение под IoT-ограничения. *Отлично***:** архитектура надёжна, код устойчив к сбоям и деградации ресурсов, есть автоматизация проверок, глубокий анализ производительности и энергопотребления, чёткое описание компромиссов и стратегий восстановления.

параллельной работы; - декомпозировать задачи на параллельные подзадачи и балансировать нагрузку между потоками для повышения эффективности и масштабируемости; - локализовать и воспроизводить многопоточные ошибки (в т. ч. гонки данных),интерпретировать отчёты инструментов диагностики для надёжной отладки сложных сценариев. **Владеть** - инструментами профилирования и диагностики и методиками стресс-тестирования для объективной оценки производительности и устойчивости; - типовыми паттернами для IoT для создания надёжной архитектуры; - практиками безопасной разработки и автоматизации проверки качества для поддержания стабильности и предсказуемости работы системы.	разработки и автоматизации проверки качества.	
ПК-18.2 (PL-4.2, уровень Б) Разрабатывает и отлаживает системы ИИ на C++ под конкретные аппаратные платформы с ограничениями по вычислительной мощности, в том числе для встроенных систем **Знать** - особенности выполнения ИИ-алгоритмов на ограниченных платформах (встроенные системы, микроконтроллеры, SoC с малым объёмом RAM/ROM, низкой тактовой частотой); - базовые структуры и оптимизации для C++ в условиях ограничений (фиксированные массивы вместо	• Знание особенностей выполнения ИИ-алгоритмов на ограниченных платформах, базовых структур и оптимизации C++ для ограниченных сред, аппаратных возможностей целевых платформ. • Умение адаптировать ИИ-модели под ограниченные	*Удовлетворительно***:** код работает на целевой платформе, есть базовые тесты и понимание ограничений. Присутствуют отдельные неоптимальные решения (лишние аллокации, избыточные копирования), но студент способен их исправить при подсказке. *Хорошо***:** решение стабильно, оптимизировано по памяти и времени, есть профилирование и стресс-тесты. Студент обоснованно выбирает методы адаптации моделей и использует аппаратные возможности платформы. *Отлично***:** архитектура устойчива к сбоям и деградации ресурсов, реализована автоматизация проверок, глубокий анализ производительности и энергопотребления, чёткое описание компромиссов и стратегий

динамических аллокаций, статические буферы, избегание исключений и RTTI, работа с фиксированной точкой вместо float/double); - аппаратные возможности целевых платформ и способы их использования для ИИ (NEON/SIMD, DSP-ядра, аппаратные ускорители, DMA, кэши, выравнивание данных). **Уметь** - адаптировать ИИ-модели под ограниченные платформы (квантование, обрезка весов, уменьшение размерности, выбор лёгких архитектур типа TinyML) и оценивать компромисс между точностью и ресурсами; - реализовывать и профилировать критические участки кода (векторизация, контроль аллокаций, минимизация копирований и памяти) и интерпретировать метрики (задержки, пропускная способность, энергопотребление); - отлаживать и диагностировать проблемы на целевых платформах (логирование с учётом ограничений, дампы состояний, тесты на детерминизм, проверка граничных условий и переполнений). **Владеть** - инструментами кросс-разработки и профилирования под целевые платформы (CMake, GDB-multiarch, perf/trace); - типичными паттернами и фреймворками для ИИ во встроенных системах (TensorFlow Lite Micro, ONNX Runtime (micro), статическая компиляция моделей); - практиками верификации и тестирования ИИ-систем в условиях ограничений (детерминированные тесты,	платформы, реализовывать и профилировать критические участки кода, осуществлять отлаживать и диагностировать проблемы на целевых платформах. • Владение инструментами кросс-разработки и профилирования, паттернами и фреймворками для ИИ во встроенных системах • Иметь практику верификации и тестирования ИИ-систем в условиях ограничений	восстановления. Код предсказуем по времени выполнения (real-time-friendly), есть верификация детерминизма и отказоустойчивости.

регрессионные проверки точности после оптимизаций, стресс-тесты на граничных ресурсах, проверка отказоустойчивости при нехватке памяти).		
ПК-24.1 (LC-5.1, уровень Б) Осуществляет выбор инструментов и инженерных практик промышленной разработки систем ИИ, развертывания и сопровождения моделей МО в продуктивной среде Знать • Архитектурные паттерны развёртывания ML моделей в IoT системах (edge, fog, cloud) и критерии их выбора. • Инструменты контейнеризации (Docker) и оркестрации (Kubernetes) для промышленного деплоя ML сервисов. • Методики мониторинга и логирования ML систем в продуктивной среде (Grafana, Prometheus, MLflow). Уметь • Развёртывать ML модель на edge устройстве (например, Raspberry Pi) с учётом ограничений по CPU/RAM и latency. • Настраивать CI/CD пайплайны для автоматического переобучения и деплоя моделей по триггеру (новые данные, ухудшение метрик). • Диагностировать и устранять типичные проблемы эксплуатации ML систем (дрейф данных, деградации качества модели). Владеть • Практиками Edge AI: квантование моделей (TensorFlow Lite), оптимизация под аппаратные ускорители. • Инструментами A/B тестирования моделей в продуктивной среде. • Методиками резервного копи-	• Знание архитектурных паттернов развёртывания ML в IoT системах, инструменты контейнеризации и оркестрации, методики мониторинга и логирования ML-систем. • Умение развёртывать ML-модели на edge-устройстве с учётом ограничений, настраивать CI/CD-пайплайны для автоматического переобучения и деплоя, диагносцировать и устранять проблемы эксплуатации ML-систем • Владение практиками Edge AI: квантованием и оптимизацией, инструментами A/B-тестирования моделей в продуктивной среде, методиками резервного	*Удовлетворительно:* решение работает в целевой среде, есть базовые конфигурации и метрики. Присутствуют отдельные неоптимальные практики (например, ручное управление версиями, отсутствие guardrails в CI/CD). Студент способен исправить недочёты при подсказке. *Хорошо***:** решение стабильно, автоматизировано, есть мониторинг и механизмы отказоустойчивости. Студент обоснованно выбирает инструменты и стратегии (деплой, тестирование, бэкапы), умеет диагностировать и устранять типовые проблемы. *Отлично***:** архитектура масштабируема и отказоустойчива, процессы полностью автоматизированы, реализованы guardrails и A/B-тестирование, есть чёткие SLA и процедуры реагирования. Код и конфигурации соответствуют индустриальным best practices, стратегия бэкапа проверена на практике, инциденты документируются и анализируются.

рования и восстановления ML сервисов при сбоях.	копирования и восстановления ML сервисов.	
ПК-24.2 (LC-5.2, уровень Б) **Знать** • Типы СУБД для IoT данных: временные ряды (InfluxDB, TimescaleDB), потоковые (Kafka), кэши (Redis). • Принципы ETL/ELT пайплайнов: очистка, агрегация, трансформация данных с датчиков. • Механизмы обеспечения отказоустойчивости (репликация, шардирование) и резервного копирования IoT данных. **Уметь** • Проектировать схему хранения данных для IoT системы с учётом требований к скорости записи/чтения и сроку хранения. • Реализовывать правила валидации и очистки данных (удаление выбросов, заполнение пропусков) в потоковом режиме. • Настраивать системы мониторинга качества данных (алерты при аномалиях, метрики полноты/актуальности). **Владеть** • Инструментами потоковой обработки (Apache Kafka + Flink/ Spark Streaming) для агрегации данных в реальном времени. • Методиками разграничения доступа к данным (RBAC, шифрование на уровне полей). • Практиками оптимизации запросов к временным рядам (индексы, партиционирование, downsampling).	• Знание типов СУБД для IoT данных, принципов ETL/ELT пайплайнов, механизмов обеспечения отказоустойчивости. • Умение проектировать схему хранения данных для IoT, реализовывать правила валидации и очистки данных. • Владение инструментами потоковой обработки данных, методиками разграничения доступа к данным, практиками оптимизации запросов.	*Удовлетворительно***:** решение работает в тестовой среде, есть основные компоненты и базовые метрики. Присутствуют отдельные неоптимальные практики (например, отсутствие партиционирования по времени, ручное управление бэкапами). Студент способен исправить недочёты при подсказке. *Хорошо***:** система масштабируема, процессы автоматизированы, есть мониторинг и механизмы отказоустойчивости. Студент обоснованно выбирает инструменты и стратегии (шардирование, репликация, downsampling, RBAC), умеет диагностировать и устранять типовые проблемы. *Отлично***:** архитектура устойчива к сбоям и всплескам нагрузки, процессы полностью автоматизированы, реализованы guardrails качества и политики безопасности. Код и конфигурации соответствуют индустриальным best practices, стратегия бэкапа проверена на практике, инциденты документируются и анализируются, оптимизация запросов подтверждена замерами и планами выполнения.

Типовые контрольные задания или иные материалы

Вопросы к зачету по дисциплине

1). Перечислите основные типы СУБД, подходящие для хранения IoT-данных (временные ряды, потоковые, кэши).
2). Объясните разницу между подходами ETL и ELT применительно к обработке телеметрии с датчиков. В каких случаях предпочтительнее ETL, а в каких — ELT?
3). Как проектирование схемы хранения в time-series СУБД (например, TimescaleDB или InfluxDB) влияет на скорость записи и чтения? Какие приёмы позволяют увеличить эту скорость?
4). Что такое downsampling и агрегация временных рядов? Приведите пример, когда хранение только агрегированных данных (минутные/часовые агрегаты) оправдано, а когда нужно сохранять сырые данные. Как это отражается в политике хранения (TTL)?
5). Раскройте понятие «водяные знаки» (watermarks) в потоковой обработке (на примере Apache Flink). Зачем они нужны и как помогают корректно выполнять оконные агрегации при задержках и неупорядоченности событий?
6). Какие механизмы используются для обеспечения отказоустойчивости в распределённых системах хранения и потоковой обработки данных (репликация, шардирование, партиционирование топиков Kafka)? Опишите, как каждый из них решает конкретные проблемы (сбой узла, всплеск нагрузки, потеря данных).
7). Что означают метрики RPO и RTO в контексте резервного копирования IoT-данных? Приведите пример стратегии бэкапа (полные/инкрементальные, хранение в другом регионе и т. д.) и соотнесите её с целевыми RPO/RTO.
8). Опишите типовой пайплайн обработки IoT-данных: от источника (датчики) до хранилища и потребителей. Укажите, какие компоненты отвечают за буферизацию, очистку/валидацию, агрегацию и хранение, и какие инструменты для этого обычно применяют (Kafka, Flink/Spark, TimescaleDB/InfluxDB, Redis).
9). Какие методы применяются для валидации и очистки данных в потоковом режиме? Приведите 3 примера правил (диапазонные проверки, удаление выбросов, заполнение пропусков) и объясните, как их реализовать в Flink или Spark Streaming.
10). Что такое семантика доставки сообщений в Kafka (at-least-once, exactly-once, at-most-once)? Как выбор семантики влияет на корректность агрегаций и подсчётов в потоковых приложениях?
11). Как организовать мониторинг качества данных в IoT-системе? Перечислите ключевые метрики (полнота, актуальность, доля аномалий, задержки и т. п.) и опишите, как на их основе настроить алерты и правила реагирования.
12). Раскройте принципы разграничения доступа к данным (RBAC) в системах хранения IoT-данных. Приведите пример ролей (администратор, аналитик, оператор) и соответствующих им прав (чтение/запись, доступ к отдельным таблицам/колонкам, ограничения по времени).
13). В чём разница между шифрованием на уровне полей и прозрачным шифрованием хранилища? Опишите, когда целесообразно применять каждый из подходов и какие компромиссы (производительность, удобство управления ключами) при этом возникают.
14). Объясните, как партиционирование по времени и выбор ключа распределения (например, device_id) влияют на производительность запросов к временным рядам. Приведите

пример запроса, который выигрывает от такой схемы, и покажите, как это видно в плане выполнения (EXPLAIN).

15). Какие проблемы могут возникать при масштабировании потоковой обработки (backpressure, рост состояния, неравномерная нагрузка по партициям)? Опишите способы их решения (балансировка нагрузки, управление состоянием, настройка параллелизма).
16). Что такое «дрейф данных» в контексте IoT-систем? Приведите признаки дрейфа и опишите шаги диагностики (анализ распределения признаков, сравнение с историческими данными, проверка источников) и возможные меры (переобучение, корректировка правил валидации).
17). Опишите, как реализовать стратегию восстановления данных после сбоя. Включите в ответ этапы: обнаружение проблемы, выбор точки восстановления, проверка целостности данных, валидация качества восстановленных данных. Какие артефакты (скрипты, манифесты, политики) для этого нужны?
18). В чём заключаются особенности хранения и обработки телеметрии с тысяч устройств (высокая кардинальность device_id, частая запись, разный формат данных)? Опишите архитектурные приёмы (шардирование по device_id, использование отдельных измерений/таблиц, сжатие данных) и инструменты, которые помогают с этим справиться.
19). Сравните InfluxDB и TimescaleDB для хранения временных рядов: укажите ключевые отличия (модель данных, язык запросов, масштабирование, интеграция с экосистемой) и приведите сценарий, где один из них предпочтительнее.
20). Представьте, что вам нужно спроектировать систему хранения и обработки данных для сети датчиков температуры и влажности (10 000 устройств, 1 запись в секунду на устройство). Опишите архитектуру (компоненты, топологию, механизмы отказоустойчивости и контроля качества), обоснуйте выбор инструментов и перечислите метрики, по которым вы будете оценивать производительность и надёжность системы.

Оператор ЭДО ООО "Компания "Тензор"

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ

ПОДПИСАНО ЗАВЕДУЮЩИМ КАФЕДРЫ	**ФГБОУ ВО "РГРТУ", РГРТУ,** Корячко Вячеслав Петрович, Заведующий кафедрой САПР	**20.10.26** 15:14 (MSK)	Простая подпись
ПОДПИСАНО ЗАВЕДУЮЩИМ ВЫПУСКАЮЩЕЙ КАФЕДРЫ	**ФГБОУ ВО "РГРТУ", РГРТУ,** Корячко Вячеслав Петрович, Заведующий кафедрой САПР	**20.10.26** 15:14 (MSK)	Простая подпись