

МИНИСТЕРСТВО СЕЛЬСКОГО ХОЗЯЙСТВА РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
СТАВРОПОЛЬСКИЙ ГОСУДАРСТВЕННЫЙ АГРАРНЫЙ УНИВЕРСИТЕТ

Факультет цифровых технологий
Кафедра информационных систем

Методические указания
по выполнению и защите курсового проекта по дисциплине «Разработка
серверных веб-приложений» для студентов очной и заочной форм
обучения направления подготовки
09.03.02 «Информационные системы и технологии»
(профиль «Инженерия информационных систем»)

Ставрополь 2026

Содержание

1. Цели и задачи проекта	3
2. Рекомендуемые темы курсовых проектов	5
3. Требования к структуре проекта.....	9
4. Требования к оформлению проекта	10
5. Список рекомендованных источников литературы.....	15
6. Требования к защите проекта.....	16
7. Критерии оценки проекта.....	17
Приложение 1	19
Приложение 2	20

1. Цели и задачи проекта

Выполнение курсового проекта является одним из важнейших видов учебного процесса, направленного на закрепление и углубление теоретических знаний и формирование практических навыков в области разработки серверных веб-приложений. Он способствует укреплению связи учебного процесса с научно-исследовательской деятельностью и выступает необходимым средством усиления целенаправленности профессиональной подготовки обучающегося.

Курсовой проект представляет собой вид учебной и научно-исследовательской работы, проводимой обучающимися самостоятельно под руководством преподавателя по определенным темам. Он призван углубить знания обучающихся по изучаемой дисциплине, полученные ими в ходе теоретических и практических занятий, привить им навыки самостоятельного изучения материала, исследовательской деятельности, а также сформировать у обучающихся навык подбирать, изучать и обобщать материалы источников информации, разрабатывать алгоритмы и программный код.

Цели выполнения курсового проекта:

- Закрепление, углубление и систематизация знаний, профессиональных умений и навыков в области проектирования и разработки серверной части веб-приложений.
- Овладение современными технологиями и инструментарием для создания бэкенд-компонентов информационных систем.
- Формирование навыков самостоятельной учебной и научно-исследовательской работы в области разработки программного обеспечения.
- Развитие навыков работы с технической документацией, научной литературой и открытыми источниками кода.
- Сформировать умение применять методы объектно-ориентированного анализа и проектирования для решения прикладных задач.

Таким образом, основной целью этого вида учебного процесса является

закрепление полученных теоретических знаний и практических навыков в процессе анализа, проектирования и реализации серверного веб-приложения для решения конкретной задачи. Она, в свою очередь, определяется содержанием и требованиями Федеральных государственных образовательных стандартов высшего образования.

Конкретные задачи, решаемые обучающимися при выполнении курсового проекта, состоят в следующем:

- Изучить научную и техническую литературу, документацию по выбранному стеку технологий.
- Провести анализ предметной области и сформулировать требования к разрабатываемому серверному приложению.
- Спроектировать архитектуру серверного приложения, выбрать и обосновать используемые технологии, фреймворки и инструменты.
- Разработать серверную часть веб-приложения, реализующую бизнес-логику, взаимодействие с базой данных и предоставляющую API для клиентской части.
- Выполнить тестирование разработанного приложения и оценить его производительность.
- Оформить результаты работы в виде пояснительной записки и представить их на защиту.

В процессе подготовки курсового проекта у обучающегося появляется серьезная возможность для расширения и углубления своих знаний, получения необходимого опыта самостоятельной разработки сложных программных систем.

2. Рекомендуемые темы курсовых проектов

Искусственный интеллект и рекомендательные системы

1. Разработка серверного API для интеллектуальной системы рекомендации сельскохозяйственных культур на основе данных о почве и климате.
2. Создание бэкенда для AI-ассистента фермера по прогнозированию урожайности с использованием моделей машинного обучения.
3. Разработка серверной части системы раннего обнаружения болезней растений по изображениям с использованием компьютерного зрения.
4. Создание серверного приложения для анализа состояния посевов по спутниковым снимкам (расчет NDVI, оценка вегетации).
5. Разработка бэкенда для системы прогнозирования оптимальных сроков посева и сбора урожая на основе погодных данных.
6. Создание серверной платформы для автоматической классификации сорняков и рекомендации гербицидов.
7. Разработка серверного приложения для оценки качества зерна и сельхозпродукции на основе изображений.
8. Создание бэкенда для системы мониторинга стресса растений (засуха, переувлажнение, недостаток питания) с использованием данных с дронов.

IoT, сенсорики и телеметрия

9. Разработка серверного приложения для сбора, обработки и визуализации телеметрии с полевых IoT-датчиков (влажность, температура, pH почвы).
10. Создание бэкенда для системы управления автоматизированным поливом на основе данных с датчиков влажности и прогнозов погоды.
11. Разработка серверной части для мониторинга микроклимата в теплицах с интеграцией MQTT-брокера и WebSocket.
12. Создание серверного приложения для управления группой автономных сельскохозяйственных дронов (планирование маршрутов, телеметрия).
13. Разработка бэкенда для системы инвентаризации и отслеживания состояния сельхозтехники на основе IoT-трекеров.
14. Создание серверной платформы для сбора данных с RFID-меток

животных (умное животноводство).

15. Разработка серверного API для системы мониторинга уровня воды в водоемах и ирригационных каналах.
16. Создание бэкенда для системы контроля за состоянием зерна в хранилищах (температура, влажность, уровень CO₂).

Цифровые платформы и маркетплейсы

17. Разработка серверной части B2B-платформы для прямых продаж сельхозпродукции от фермера к переработчику.
18. Создание бэкенда для краудфандинговой платформы агростартапов (сбор инвестиций, отслеживание проектов).
19. Разработка серверного приложения для платформы аренды сельхозтехники и оборудования между фермерскими хозяйствами.
20. Создание бэкенда для агрегатора цен на агропродукцию с аналитикой рыночных трендов.
21. Разработка серверной части биржи труда для сезонных сельскохозяйственных работ.
22. Создание серверного приложения для платформы агротуризма (бронирование экскурсий, ферм, дегустаций).
23. Разработка бэкенда для цифровой доски объявлений по обмену семенами и посадочным материалом.
24. Создание серверной платформы для коллективного закупа удобрений и средств защиты растений (кооперация фермеров).
25. Разработка серверного API для системы сертификации и отслеживания происхождения органической продукции (Farm-to-Table).

Блокчейн, прослеживаемость и безопасность

26. Разработка серверного приложения для системы прослеживаемости цепочки поставок агропродукции на основе блокчейна.
27. Создание бэкенда для системы смарт-контрактов по страхованию урожая от погодных рисков.
28. Разработка серверной части децентрализованной системы сертификации органического земледелия.

- 29.Создание серверного приложения для управления цифровыми паспортами полей и севооборотами на основе распределенного реестра.
- 30.Разработка бэкенда для системы токенизации прав на использование земельных участков.

Управление фермой и Big Data

- 31.Разработка серверного API для комплексной ERP-системы управления сельскохозяйственным предприятием.
- 32.Создание бэкенда для системы планирования севооборотов с учетом состояния почвы и предшественников.
- 33.Разработка серверного приложения для расчета экономической эффективности выращивания различных культур.
- 34.Создание бэкенда для системы управления складскими запасами семян, удобрений и ГСМ на ферме.
- 35.Разработка серверной части для системы кадрового учета и расчета заработной платы сезонных работников.
- 36.Создание серверного приложения для мониторинга и анализа себестоимости производства продукции.
- 37.Разработка бэкенда для системы управления ветеринарными записями и лечением животных.
- 38.Создание серверной платформы для сбора и анализа Big Data с нескольких ферм для выявления общих трендов.

Экология, устойчивое развитие и климат

- 39.Разработка серверного приложения для системы учета и торговли углеродными кредитами в агросекторе.
- 40.Создание бэкенда для системы мониторинга выбросов парниковых газов от сельскохозяйственной деятельности.
- 41.Разработка серверной части для платформы оценки экологического следа агропродукции.
- 42.Создание серверного приложения для моделирования влияния изменений климата на урожайность в регионе.
- 43.Разработка бэкенда для системы управления водными ресурсами в

сельском хозяйстве (умное водопользование).

- 44.Создание серверной платформы для поддержки решений по восстановлению деградированных почв.

Интеграция, API и микросервисы

- 45.Разработка серверного шлюза API для интеграции различных агро-сервисов (погода, карты, анализ почв).
- 46.Создание микросервисной архитектуры для системы доставки агропродукции с оптимизацией маршрутов.
- 47.Разработка серверного приложения для интеграции с внешними платформами (государственные субсидии, аналитические отчеты).
- 48.Создание бэкенда для системы уведомлений и оповещений фермеров о погодных аномалиях и рисках.
- 49.Разработка серверного API для мобильного приложения по управлению личным подсобным хозяйством.
- 50.Создание серверной платформы для сбора обратной связи от потребителей агропродукции с интеграцией в CRM.

3. Требования к структуре проекта

Структура курсового проекта должна включать следующие элементы:

- Титульный лист (Приложение 1).
- Содержание (оглавление).
- Введение.
- Основную часть (должна состоять из трех глав).
- Заключение с указанием основных результатов работы.
- Список использованных источников литературы.
- Приложения (при необходимости, например, листинги кода).

Важным этапом подготовки курсового проекта является разработка его плана. Основной задачей плана является структурирование проекта, формулировка заголовков глав и разделов. Названия глав формулируются на основании вопросов, подлежащих разработке. Подобный подход обеспечивает выполнение требования к курсовому проекту о соответствии его содержания теме. Аналогичный подход применим к формулировке разделов глав, которые должны раскрывать содержание каждой главы по тому заголовку, в котором они сформулированы.

Практика показывает, что наиболее характерными ошибками при разработке плана являются:

1. Совпадение названия глав (разделов) с темой курсового проекта.
2. Названия глав (разделов) не раскрывают реального содержания темы курсового проекта.

Обе ошибки недопустимы, особенно вторая, поскольку она приводит к несоответствию содержания курсового проекта его теме.

4. Требования к оформлению проекта

Курсовой проект оформляется в соответствии с общими правилами оформления научно-исследовательских и проектных работ.

Титульный лист курсового проекта содержит следующие элементы: полное наименование вышестоящего органа, университета, института/факультета и кафедры; название дисциплины; тему курсового проекта; сведения об исполнителе (Ф.И.О. обучающегося, группа, подпись); сведения о преподавателе (Ф.И.О., ученая степень, ученое звание); наименование места и год выполнения; сведения о регистрации на кафедре, количество баллов (по БРС) и оценка, даты и подпись ведущего преподавателя.

Содержание включает порядковые номера и наименование структурных элементов курсового проекта с указанием номера страницы, на которой они помещены.

Содержание		
	Введение	3
1	Теоретические аспекты и обзор технологий	5
	1.1 Анализ предметной области и постановка задачи	5
	1.2 Обзор современных технологий и инструментов разработки серверных приложений	8
	1.3 Выбор и обоснование архитектуры и стека технологий	10
2	Проектирование и разработка серверного приложения	15
	2.1 Проектирование архитектуры приложения и базы данных	15
	2.2 Реализация серверной логики и API-методов	20
	2.3 Разработка системы безопасности и аутентификации	25
3	Тестирование и оценка эффективности	28
	3.1 Методы и инструменты тестирования серверных приложений	28
	3.2 Проведение тестирования и анализ полученных результатов	30
	Заключение	33
	Список использованных источников литературы	35
	Приложение	37

Введение характеризует:

- Актуальность темы исследования — обоснование важности разработки серверной части для современных информационных систем.
- Цель и задачи проекта — краткая и четкая формулировка цели (например, "разработать серверное веб-приложение...") и нескольких задач, решение которых необходимо для достижения поставленной цели.
- Объект и предмет исследования (например, объект — процесс обработки данных в веб-приложении, предмет — методы и технологии разработки серверной части).
- Методы исследования (например, анализ, проектирование, моделирование, тестирование).
- Структуру работы — краткое содержание глав и параграфов основной части работы.

Последовательность рубрик должна соответствовать приведенному перечню, наименование каждой рубрики выделяется в тексте жирным шрифтом.

Основная часть курсового проекта состоит из трех глав. Каждая глава должна состоять из отдельных параграфов, каждый из которых посвящен отдельному аспекту изучаемой проблемы.

- Первая глава посвящена теоретическим аспектам и обзору технологий. Она должна содержать анализ предметной области, обзор существующих решений, а также обоснование выбора конкретных технологий, фреймворков и инструментов для разработки (например, выбор между Django и Spring Boot, реляционной и NoSQL базой данных, REST и GraphQL). Глава должна состоять из 3 параграфов.
- Вторая глава содержит описание процесса проектирования и разработки серверного приложения. Это основная практическая часть проекта. В ней должны быть представлены разработанные диаграммы, схемы базы данных, архитектурные решения, а также описан процесс реализации

ключевых компонентов (API-эндпоинтов, бизнес-логики, системы аутентификации). Глава должна состоять из 3 параграфов.

- Третья глава посвящена тестированию и оценке эффективности разработанного приложения. В ней описываются используемые методы тестирования (модульное, интеграционное, нагрузочное), приводятся результаты тестов и анализ производительности. Также могут быть предложены пути дальнейшего развития и оптимизации системы. Глава должна включать 2 параграфа.

Каждая глава заканчивается выводами, где выделяется существенное, главное как результат проделанной работы.

Заключение — краткое изложение основных, наиболее существенных результатов проведенного анализа и разработки, сформулированных в виде выводов, соответствующих цели и поставленным во введении задачам.

В списке использованных источников литературы должны быть представлены основные источники по теме:

- Нормативно-правовые документы (ГОСТы, стандарты в области ИТ).
- Учебники и учебные пособия по разработке ПО, базам данных, сетевым технологиям.
- Официальная документация на используемые технологии (Spring, Django, Node.js, .NET и т.д.).
- Научные статьи и публикации из профильных журналов (например, «Программирование», «Информационные технологии и вычислительные системы»).
- Материалы научных конференций.
- Интернет-ресурсы (официальные сайты, блоги, Stack Overflow, GitHub).
- Список должен содержать **не менее 10** современных источников, преимущественно даты издания не более 5 лет относительно года написания проекта.

На все приведенные в списке источники должны быть ссылки в тексте курсового проекта. Они проставляются в квадратных скобках с указанием

номера источника, под которым он значится в списке литературы.

Приложения — вспомогательные материалы, которые нецелесообразно приводить в основном тексте. В случае курсового проекта по разработке серверных приложений это могут быть: полные листинги ключевых модулей, конфигурационные файлы, примеры входных/выходных данных API (скриншоты тестирования в Postman), ER-диаграммы.

Курсовой проект должен быть напечатан на стандартном листе писчей бумаги в формате A4 с соблюдением следующих требований:

- Поля: левое — 30 мм, правое — 15 мм, верхнее — 20 мм, нижнее — 20 мм.
- Шрифт: размером 14 пт, гарнитурой Times New Roman.
- Межстрочный интервал: полуторный.
- Отступ красной строки: 1,25.
- Выравнивание текста: по ширине.

Цифровой материал и результаты тестирования оформляются в виде таблиц, которые нумеруются последовательно арабскими цифрами сквозной нумерацией.

Пример таблицы:

Таблица 1 — Сравнительный анализ технологических стеков для разработки серверного приложения

Критерий сравнения	Node.js (Express)	Python (Django)	Java (Spring Boot)	Go (Gin)
Производительность	Высокая	Средняя	Высокая	Очень высокая
Скорость разработки	Высокая	Очень высокая	Средняя	Средняя
Порог входа	Низкий	Низкий	Высокий	Средний
Экосистема библиотек	Обширная	Обширная	Обширная	Развивающаяся
Поддержка микросервисов	Хорошая	Удовлетворительная	Отличная	Отличная
Наличие ORM	Sequelize, Prisma	Django ORM	Hibernate	GORM
Подходит для стартапов	Да	Да	Да	Да

Иллюстрации (блок-схемы алгоритмов, диаграммы классов, компонентов, развертывания, ER-диаграммы, графики производительности)

обозначаются словом «Рисунок» и нумеруются последовательно арабскими цифрами сквозной нумерацией.

Пример рисунка:



Рисунок 1 — Обобщенная архитектура разрабатываемого серверного приложения

Рекомендуемый общий объем курсового проекта не менее 30 страниц. Рекомендуемый объем введения: 2-3 страницы, заключения: 1-2 страницы, остальной объем составляет основная часть.

5. Список рекомендованных источников литературы

1. Фаулер, М. Архитектура корпоративных программных приложений : пер. с англ. — М. : Вильямс, 2021. — 544 с.
2. Ньюман, С. Создание микросервисов. — СПб. : Питер, 2022. — 304 с.
3. Робертсон, С. Spring в действии : пер. с англ. — М. : ДМК Пресс, 2023. — 800 с.
4. Ларман, К. Применение UML и шаблонов проектирования : пер. с англ. — М. : Вильямс, 2021. — 736 с.
5. Хансен, К. Django. Практика создания веб-сайтов на Python : пер. с англ. — СПб. : БХВ-Петербург, 2022. — 528 с.
6. Бэнкс, А., Порселло, Е. RESTful Web API : пер. с англ. — М. : ДМК Пресс, 2020. — 440 с.
7. Мартин, Р. Чистая архитектура. Искусство разработки программного обеспечения : пер. с англ. — М. : Питер, 2022. — 384 с.
8. Фаулер, М. Шаблоны корпоративных приложений : пер. с англ. — М. : Вильямс, 2021. — 576 с.
9. Тейлор, Р. Node.js в действии : пер. с англ. — М. : ДМК Пресс, 2023. — 496 с.
10. Официальная документация фреймворка Spring Boot. — URL: <https://spring.io/projects/spring-boot>.
11. Официальная документация Django. — URL: <https://www.djangoproject.com/>.
12. Официальная документация Node.js. — URL: <https://nodejs.org/>.
13. Рекомендации по проектированию REST API. — URL: <https://restfulapi.net/>.
14. Электронная библиотека научных статей по информационным технологиям. — URL: <https://www.elibrary.ru>.
15. Git: система контроля версий. — URL: <https://git-scm.com/>.

6. Требования к защите проекта

- Курсовой проект допускается к защите при выполнении следующих условий:

- Разработанное приложение сдано на проверку и успешно функционирует.

- Пояснительная записка полностью оформлена и проверена на соответствие требованиям.

- Степень оригинальности текста пояснительной записки (по решению кафедры) составляет не ниже 25%.

- Наличие рецензии преподавателя, принимающего курсовой проект (Приложение 2).

- Защита курсовых проектов относится к промежуточной аттестации и проводится в конце семестра. Защита курсовых проектов назначается кафедрой, вносится в расписание промежуточной аттестации и отражается в расписании учебных занятий.

- Защиту курсовых проектов проводит ведущий преподаватель, а в случае возникновения спорных ситуаций создается комиссия, в состав которой входит заведующий кафедрой и преподаватели кафедры.

- Защита проекта проходит в форме публичного выступления (доклад на 7-10 мин.) с представлением результатов работы в виде презентации (7-10 слайдов), включающей демонстрацию разработанного серверного приложения (API, интерфейс, результаты тестирования), и ответов на вопросы преподавателя/комиссии.

- В тексте выступления (доклада) отражается:

- Актуальность выбранной темы.

- Цель и основные задачи курсового проекта.

- Используемые технологии и архитектура.

- Основное содержание и ключевые результаты работы (демонстрация).

- Основные выводы и рекомендации по дальнейшему развитию.

7. Критерии оценки проекта

Выполненный и защищенный курсовой проект оценивается в соответствии с учетом балльно-рейтинговой системы оценивания и критериями оценки, указанными в рабочей программе дисциплины.

В соответствии с Положением о балльно-рейтинговой системе оценки знаний студентов, курсовой проект необходимо оценить по следующим критериям с учетом установленных максимальных баллов:

Критерий	Максимальное значение в баллах	Набранных баллов
Оформление курсового проекта	10	
Содержание курсового проекта	60	
Защита курсового проекта	30	
ИТОГО	100	

Содержание критериев оценки курсовой проекта:

Оформление курсового проекта (до 10 баллов):

- 10 баллов — курсовая работа соответствует всем требованиям к ее оформлению, использованы современные средства визуализации информации (диаграммы, графики, UML-схемы).
- 5 баллов — курсовая работа частично соответствует требованиям к оформлению, визуализация материала выполнена некачественно, современные средства визуализации использованы не в полной мере.
- 0 баллов — оформление не соответствует требованиям.

2. Содержание курсового проекта (до 60 баллов):

- 60 баллов — в проекте корректно подобраны информационные источники, проведен полный анализ, все этапы разработки освещены полностью. Архитектура и код реализованы в соответствии с современными требованиями, для выводов приведены достаточные обоснования и результаты тестирования.
- 40 баллов — источники подобраны не в полном объеме, информация использована не везде корректно, не все разделы освещены полностью. В

архитектуре или реализации есть существенные недочеты, обоснования выводов недостаточны.

- 20 баллов — отсутствуют некоторые разделы, их название не отвечает содержанию. Код разработанного приложения имеет критические ошибки или не работает.
- 0 баллов — содержание не соответствует теме или проект не выполнен.

3. Защита курсового проекта (до 30 баллов):

- 30 баллов — студент продемонстрировал полное понимание всех положений защищаемого проекта, четкость и правильность изложения ответов на все вопросы, уверенную защиту принятых технических решений.
- 20 баллов — студент продемонстрировал понимание основных положений защищаемого проекта, ответил на большинство вопросов.
- 10 баллов — студент дал недостаточно полные ответы на вопросы, на некоторые дал ошибочные ответы или не ответил.
- 0 баллов — студент не ответил на вопросы или не явился на защиту.

Перевод оценки из 100-балльной в пятибалльную систему осуществляется следующим образом:

- 89-100 — оценка «отлично».
- 77-88 — оценка «хорошо».
- 65-76 — оценка «удовлетворительно».
- Менее 64 — оценка «неудовлетворительно».

При неудовлетворительной оценке курсового проекта обучающийся имеет право на повторную защиту после доработки и внесения исправлений.

МИНИСТЕРСТВО СЕЛЬСКОГО ХОЗЯЙСТВА РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«СТАВРОПОЛЬСКИЙ ГОСУДАРСТВЕННЫЙ АГРАРНЫЙ УНИВЕРСИТЕТ»
Факультет цифровых технологий
Кафедра информационных систем

Курсовой проект

по дисциплине «Разработка серверных веб-приложений»

Тема: «Название»

Выполнил:

Студент _____ курса _____ группы

ФИО _____

Направление подготовки: _____

Форма обучения: _____

Проверил:

уч. степень, должность

ФИО _____

Зарегистрирована

« _____ » _____ 20 ____ г.

Критерий	Максимальное значение в баллах	Набранных баллов
Оформление курсовой работы (проекта)	10	
Содержание курсовой работы (проекта)	60	
Защита курсовой работы (проекта)	30	
ИТОГО	100	

Оценка « _____ » Дата _____ Подпись _____

Ставрополь, 20 _____

Кафедра информационных систем

РЕЦЕНЗИЯ

на курсовой проект

Тема _____

Обучающийся (Ф.И.О.) _____

Курс _____ Группа _____

Преподаватель (Ф.И.О.) _____

Выполнение общих требований к курсовому проекту

1	Объем проекта соответствует установленным требованиям	Да/нет
2	Степень оригинальности курсового проекта соответствует установленным требованиям	Да/нет (указать %)
3	Функциональность разработанного программного обеспечения соответствует техническому заданию	Да/Нет

Критерии оценивания курсового проекта

Критерии	Количество баллов	Содержание критерия оценки	Итоговый балл
Оформление курсового проекта	10	Проект соответствует всем требованиям к оформлению. Используются современные средства визуализации.	
	5	Проект частично соответствует требованиям к оформлению, представленный материал проиллюстрирован не качественно.	
Содержание курсового проекта	60	Корректно подобраны источники, все разделы освещены, код работает, для выводов приведены достаточные обоснования.	
	40	Подобраны не все источники, информация использована не везде корректно, не все разделы освещены полностью, в коде есть недочеты.	
	20	Отсутствуют некоторые разделы, или их название не отвечает содержанию. Разработанное ПО не работает.	
Защита курсового проекта	30	Студент продемонстрировал полное понимание работы, четко и правильно ответил на все вопросы.	
	20	Студент продемонстрировал понимание основных положений работы, ответил на большую часть вопросов.	
	10	Студент дал недостаточно полные ответы на вопросы, на некоторые дал ошибочные ответы или не ответил.	
Итого			

Рекомендации:

Ведущий преподаватель _____ / _____
 (ФИО) (подпись)