МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«ДАГЕСТАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
Факультет информатики и информационных технологий

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

по организации и проведению учебной (ознакомительной) практики

для студентов направления подготовки

09. 04 .03 Прикладная информатика

Профиль подготовки:

Разработка и внедрение информационных систем

Махачкала 2026

Содержание

I. Общие положения 3

1.1. Цель учебной ознакомительной практики 3

1.2. Место проведения практики 4

1.3. Руководство практикой 4

1.4. Права и обязанности студентов в период практики 5

1.5. Требования к оформлению отчета по учебной ознакомительной практике 6

1.6. Подготовка презентационного материала 10

1.7. Аттестация по итогам практики 11

II. Примерный перечень заданий по учебной практике 12

III. Перечень контрольных вопросов при приеме материалов учебной практики 40

Приложение 1 41

I. Общие положения

Учебная практика является обязательным компонентом учебного процесса в высших учебных заведениях.

Учебные практики проводятся в соответствии с программами для студентов очной формы обучения по направлению подготовки 09.04.03 Прикладная информатика (магистратура), профиль – «Разработка и внедрение информационных систем».

Методические указания содержат требования к организации и порядку проведения учебной практики студентов, её основное содержание.

В методических указаниях сформулированы права и обязанности практикантов, рассмотрен перечень вопросов для изучения студентами, а также требования к формированию и защите отчета по практике.

Методические указания предназначены для студентов указанного направления подготовки, профессорско-преподавательского состава выпускающих кафедр, руководителей практики от кафедры.

1.1. Цель учебной ознакомительной практики

Целями учебной (ознакомительной) практики по направлению подготовки 09.04.03 «Прикладная информатика», направленность (профиль) «Разработка и внедрение информационных систем» являются закрепление и углубление теоретической подготовки обучающегося и приобретение ими практических навыков и компетенций в сфере профессиональной деятельности.

Кроме того, целями учебной ознакомительной практики по направлению подготовки 09.04.03 Прикладная информатика являются также обеспечение непрерывности и последовательности овладения магистрами умениями и навыками профессиональной деятельности, закрепление и углубление универсальных и общепрофессиональных компетенций в сфере программной инженерии, сформированных у студентов в процессе обучения и в результате самостоятельной работы по исследованию предметной области, исследованию информационно-вычислительной системы.

Основные задачи учебной практики:

Задачами учебной ознакомительной практики по направлению подготовки 09.04.03 Прикладная информатика, профиль - «Разработка и внедрение информационных систем»:

– изучение обучающимися опыта создания и применения информационных технологий в структурных подразделениях вуза;

– изучение обучающимися опыта применения технологий разработки программного обеспечения в структурных подразделениях вуза;

углубление знаний по дисциплинам, полученным за время обучения на курсе, таких как «Современные технологии разработки баз данных», «Нейронные сети», «Имитационное моделирование сложных информационных систем»;

– формирование навыков подготовки и систематизации необходимых материалов и научно-технической информации для выполнения задания;

– создание условий для практического применения знаний в области общепрофессиональных, специализированных компьютерных и математических дисциплин;

– формирование и совершенствование базовых профессиональных навыков и умений в области применения современных информационных технологий;

– формирование информационной компетентности с целью успешной работы в профессиональной сфере деятельности;

– анализ результатов проведенного исследования, подготовка и составление отчетов;

– приобретение навыков создания отчетов, в том числе и научно-технических, обеспечение успеха дальнейшей профессиональной карьеры.

1.2. Место проведения практики

Учебная (ознакомительная) практика проводится на 1 курсе во 2 семестре, срок проведения – 4 недели.

Местом проведения практики является факультет информатики и информационных технологий, кафедра информационных систем и технологий программирования, а также иные подразделения университета.

План-график прохождения практики разрабатывается руководителями от университета на основе баланса времени и с учетом особенностей базы практики и ее вида.

При наличии обучающихся по данной образовательной программе с ограниченными возможностями здоровья и инвалидов выбор мест прохождения практики осуществляется с учетом состояния здоровья и требований их доступности для данной категории.

1.3. Руководство практикой

Руководство практикой осуществляют факультет информатики и информационных технологий и кафедра информационных систем и технологий программирования (ИСиТП). Кафедра отвечает за выполнение программы и качество проведения учебной практики.

Руководитель практики от кафедры:

– осуществляет методическое руководство учебной (ознакомительной) практикой студентов;

– обеспечивает выполнение подготовительной и текущей работы по организации практики;

– разрабатывает программу учебной практики, индивидуальные задания студентам с указанием основной и дополнительной литературы;

– готовит и проводит совместно с деканатом организационные собрания студентов перед началом практики;

– при необходимости проводит консультации студентов в период учебной практики и перед защитой отчетов;

– совместно с факультетом в назначенный срок после окончания практики организует и принимает защиту отчетов по практике;

– представляет на кафедру итоги учебной практики;

– подводит итоги практики студентов кафедры;

– организует на кафедре хранение отчетов студентов по практике в течение установленного срока.

1.4. Права и обязанности студентов в период практики

С целью наилучшей подготовки к практике студент обязан ознакомиться с программой и содержанием предстоящих работ, собрать и изучить рекомендованную литературу.

В процессе прохождения практики студенты должны выполнять задания, предусмотренные планом, правила внутреннего распорядка организации, следовать указаниям руководителей практики, собрать материал для отчёта и оформить его. Порядок сбора и обработки материалов согласовывается с руководителем практики.

Перед завершением практики студент должен обсудить результаты своей работы над отчётом с руководителей практики.

Студент при прохождении практики обязан:

- в случае болезни обязательно поставить в известность руководителя практики и предоставить медицинскую справку;
- в течении последней недели практики представить отчёт руководителю практики для проверки и отзыва;
- в последние два дня практики представить руководителю практики от Университета отчёт и после проверки защитить его.

В случае затруднений с выполнением задания практики, связанных с характером работы, студент должен сообщить об этом руководителю практики от Университета.

1.3. Структура и содержание учебной практики»

При формировании отчёта по практике следует соблюдать определённую последовательность расположения материала:

– Титульный лист (см. Приложение 1)

– Содержание (перечень разделов и подразделов отчёта).

– Введение (цель учебной практики, особенности ее прохождения).

– Перечень, особенности и описание практических задач, выполненных студентом во время практики

– Выполнение индивидуальной работы (составление тестов).

– Заключение (итоги практики, выводы и предложения студента по практике).

– Список использованной литературы, технической и нормативной документации.

Руководитель практики должен проверить содержание отчета по практике с целью проверки грамотности его выполнения, допустимости разглашения отдельной информации.

1.5. Требования к оформлению отчета по учебной ознакомительной практике

При оформлении отчёта по практике необходимо соблюдать следующие требования:

– отчёт оформляется на одной стороне листа А4 (210х297 мм), белой бумаги с соблюдением следующих размеров полей: верхнее и нижнее - по 20 мм, левое поле – 30 мм, правое поле – 10 мм;

– таблицы и иллюстрации допускается представлять на листах формата А3 (297х420 мм);

– текст готовится к распечатке на одной стороне листа на компьютере;

– шрифт «Times New Roman» 14 pt, межстрочный интервал – полуторный;

– наименования разделов располагаются по центру строки без точки в конце и набирается прописными буквами без подчёркиваний;

– заголовки подразделов набираются с абзацного отступа строчными буквами (кроме первой) без точки в конце;

– после номера раздела и подраздела в тексте точка не ставится;

– не допускаются переносы в словах разделов и подразделов;

– расстояние между заголовками и текстом должен составлять два интервала;

– страницы следует нумеровать цифрами;

– номер проставляется внизу страницы по центру в пределах рабочего поля; на титульном листе, номер страницы не ставится. Формулы следует выделять в отдельную строку, отделяя её от текста свободной строкой (сверху и снизу). Оформление формул производится при помощи любого редактора формул. Номера формул указываются арабскими цифрами в круглых скобках и имеют сквозную нумерацию;

– при оформлении таблиц и иллюстраций следует руководствоваться следующими правилами.

Все иллюстрации (кроме таблиц) в тексте работы называются **рисунками,** их представляют сразу же после упоминания или на следующей странице.

Выполняют рисунки черными чернилами (пастой) или тушью, если они не могут быть представлены с помощью компьютерной графики или в виде фотографий.

Под изображением рисунка приводится слово «Рисунок» с указанием после него через интервал цифры (номера) далее (через интервал) указывается название рисунка с заглавной буквы. При этом номер и название располагаются на одной и той же строке.

Перед рисунком в тексте оставляют одну свободную строку. Рисунок должен быть расположен по центру страницы, в абзаце, содержащем рисунок, не должно быть заказано отступа или выступа первой строки. Выравнивание названия рисунка производится по левому краю. Нумерация рисунков в тексте должна быть сквозной.

Пример оформления рисунка:



Рис. 1. Функциональная схема котлоагрегата на экране монитора

Уравнения и формулы следует выделять из текста в отдельную строку. В формулах в качестве символов следует применять обозначения, установленные соответствующими государственными стандартами, и отраженные в формате формул. Образец оформления формулы представлен на рисунке 2.

Образец оформления формулы представлен на рисунке 2.

$$S = P \times (1 + \frac{I \times j}{100 \times K})^n$$

Рис. 2. Образец оформления формулы

где S – общая сумма («тело» вклада + проценты), причитающаяся к возврату вкладчику по истечении срока действия вклада;

P – первоначальная величина вклада;

n - общее количество операций по капитализации процентов за весь срок привлечения денежных средств (в данном случае оно соответствует количеству лет);

I – годовая процентная ставка.

K – количество дней в текущем году,

j – количество дней в периоде, по итогам которого банком производится капитализация начисленных процентов (остальные обозначения – как и в предыдущей формуле).

Цифровой материал рекомендуется помещать в тексте в виде **таблиц**. Каждая из них должна иметь заголовок, точно отражающий её содержание и расположенный по центру. Помещается заголовок непосредственно над таблицей по центру. После слова таблица через интервал цифры (номера) далее (через интервал) указывается название таблицы с заглавной буквы. При этом номер и название располагаются на одной и той же строке.

Пример оформления таблицы:

Таблица 1

Доля государственного долга в ВВП в развитых странах G-20
в 2017–2025 гг., %

Страны	2017 (факт)	2019 (факт)	2020 (оценка)	2021 (прогноз)	2025 (прогноз)
G-20 – развитые экономики	77,9	96,9	104,4	108,8	117,1
Япония	187,7	217,7	227,1	234,6	250,0
Италия	103,4	115,8	118,6	120,5	124,7
США	62,1	83,2	92,6	97,4	109,7
Франция	63,8	77,4	84,2	88,6	94,8
Великобритания	44,1	68,2	78,2	84,9	90,6

Источник: IMF World Economic Outlook. 2020. April. [Электронный ресурс]. – Электр. данн. – Режим доступа: http://www.imf.org/external/pubs/ft/weo/

Располагаются таблицы в тексте сразу после первой ссылки на них или на следующей (за первой ссылкой) странице. Оформляются таблицы так, как показано на примере приведенной Таблицы 1. При переносе части таблицы на другую страницу название помещают только над первой её частью, а над второй справа пишут «Продолжение таблицы» с указанием её номера. Допускается располагать таблицу вдоль длинной стороны листа. Графу «Номер по порядку» в таблице не делают.

Если повторяющийся в разных строках таблицы текст состоит из одного слова, то после его первого написания далее допускается заменять его кавычками. Если текст состоит из двух или более слов, то его можно заменять словами «то же», а далее заменять кавычками. Ставить кавычки вместо повторяющихся цифр не допускается.

Если таблица заимствована или рассчитана по данным статистического ежегодника или другого литературного источника, надо обязательно делать ссылку на первоисточник (см. пример Таблица 1).

Если в тексте необходимо сослаться на таблицу, то следует указать номер таблицы и страницу, на которой она расположена.

Приложения помещают за списком литературы в порядке упоминания их в основном тексте. Каждое приложение следует начинать с новой страницы с указанием наверху справа слова «Приложение».

Приложения обозначаются заглавными буквами русского алфавита, начиная с А. После слова «Приложение» должна следовать буква, обозначающая его последовательность.

Разработанные Обучающимся документы, имеющие самостоятельное значение и подлежащие исполнению в соответствии с заданием и нормативно-закрепленными требованиями, помещают в прилагаемую к текстовой части папку с соответствующей описью. При наличии аудио - или видеоматериалов и материалов в электронном виде их носители размещают

в кармане этой же папки.

Наличие графического материала является необходимым условием её написания. Этот материал может быть представлен в виде схем, рисунков, таблиц, графиков и диаграмм, которые должны наглядно иллюстрировать и подтверждать основную часть работы. Графический материал может помещаться или в основном тексте, или (если он представлен в формате, большем, чем А4) выноситься в приложения к работе. Перечень графического материала устанавливается в задании на практику.

Список использованной литературы оформляется в соответствии с требованиями ГОСТ Р 7.0.100-2018, ГОСТ 7.32-2017.

Рекомендуется представлять единый список литературы к работе в целом. Список обязательно должен быть пронумерован. Каждый источник упоминается в списке один раз, вне зависимости от того, как часто на него делается ссылка в тексте работы.

Наиболее удобным является алфавитное расположение материала, так как в этом случае произведения собираются в авторских комплексах. Произведения одного автора расставляются в списке по алфавиту заглавий.
Официальные документы ставятся в начале списка в определенном порядке
Конституции; Кодексы; Законы; Указы Президента; Постановления Правительства; другие нормативные акты (письма, приказы и т.д.). Внутри каждой группы документы располагаются в хронологическом порядке.

Литература на иностранных языках ставится в конце списка после литературы на русском языке, образуя дополнительный алфавитный ряд. Для каждого документа предусмотрены следующие элементы библиографической характеристики: фамилия автора, инициалы; название; подзаголовочные сведения (учебник, учебное пособие, словарь и т.д.); выходные сведения (место издания, издательство, год издания); количественная характеристика (общее количество страниц в книге).

В каждой группе библиографические записи должны располагаться в алфавитном порядке. Нумерация всего **списка использованных источников** сквозная. Библиографическая запись должна выполняться согласно ГОСТ Р 7.0.100-2018, ГОСТ 7.32-2017. Как правило, на языке выходных сведений: Автор (ФИО). Название источника. –Место издания: Издательство, год издания, количество страниц (общее или же страницы того раздела, который был использован при написании работы). Сведения об используемом издании находятся на обороте титульного листа книги;

– ссылки формируются в квадратных скобках, внутри которых указывается номер источника литературы, который соответствует его порядковому номеру в списке использованных источников. Например, ссылка [5] означает, что ссылка по тексту работы приведена на литературный источник, расположенный под порядковым номером 5 в Списке использованных источников;

– в случае необходимости дословного цитирования фрагмента авторского произведения заимствованный текст должен быть взят в кавычки

и снабжён ссылкой на источник, содержащий данный текст. Если в тексте присутствует заключенная в кавычки цитата, то ссылка на источник должна приводиться с указанием не только номера источника в Списке использованных источников, но и номера страницы источника, где расположена цитата, например, [10, с.37];

– указанные в Списке использованных источников электронные источники должны быть актуальными. Это означает, что дата обращения к электронному ресурсу должна соответствовать текущему месяцу, в котором осуществляются написание и проверка работы;

– запрещается использование в работе множественных ссылок на одно смысловое предложение, что наводит на мысль об их бессистемном, произвольном использовании.

В раздел Приложение включаются материалы, не вошедшие в текст основной части работы. Каждое приложение должно быть пронумеровано в порядке расположенияприложений в тексте отчета практики. Заголовок «Приложение» располагается в правом верхнем углу с номером. Каждое приложение начинается с новой страницы.

– каждое Приложение имеет своё название, заголовок которого печатается в центре полужирным шрифтом, размер №16. Слово «Приложение» с номером (без знака № и точки на конце) печатается заглавными буквами в правом верхнем углу страницы, над названием. Каждое приложение начинается с новой страницы. Объем **Приложения** не входит в обязательное количество страниц работы и не должен превышать 1/3 всеготекста работы;

– рисунки и таблицы имеют сквозную нумерацию в каждом отдельном Приложении. Например, рисунок 5, расположенный в Приложении В, обозначается: Рисунок В.5, таблица 3, расположенная в Приложении А, обозначается: Таблица А.3;

– работа должна быть написана от третьего лица. Использование в работе личных местоимений запрещается. Необходимо использовать выражения, типа: «известно, что», «существует мнение», «учёные придерживаются следующей точки зрения», «необходимо заметить», «представляет интерес» и т.п. Не допускается сокращение слов, все используемые аббревиатуры необходимо расшифровывать;

В тексте рекомендуется чаще применять красную строку, выделяя законченную мысль в самостоятельный абзац. Не следует приводить слишком много цитат, которые используются как приём аргументации. Текст не должен представлять собой сплошные перечисления. Необходимо чётко и ясно излагать материал, применяя принятую научную терминологию.

1.6. Подготовка презентационного материала

Обучающийся должен подготовить презентационный материал для защиты отчета по практике перед членами кафедры, используя средства Power Point. Структура презентационного материала (5– 10 слайдов):

– титульный лист (фирменный логотип ДГУ; название практики; выполнил обучающийся Ф.И.О., руководитель практики Ф.И.О., с указанием

научного звания и степени, Махачкала 202_г»;

- Введение
- результаты практики (реферат)

1.7. Аттестация по итогам практики

2. Защита учебной практики проводится после окончания практики в сроки, определяемые деканатом.
3. К защите допускаются студенты, у которых материалы по практике оформлены надлежащим образом и собраны все необходимые документы, указанные в предыдущем подразделе «Требования к оформлению отчета по учебной ознакомительной практике».
4. Защита студентом учебной практики оценивается комиссией по модульно-рейтинговой системе (отлично, хорошо, удовлетворительно, неудовлетворительно).
5. Сроки прохождения практики по согласованию с деканатом факультета могут быть продлены студентам, нарушающим трудовую дисциплину.
6. Полностью оформленный отчет представляется руководителю практики от кафедры «Информационные системы и технологии программирования» для проверки и защиты.
7. На основании полученного отчета, руководитель практики от кафедры принимает решение о допуске студента к защите отчета. Допуск студента к защите указывается на отчете, который вместе с рецензией руководителя практики от кафедры передается студенту для защиты. При отсутствии отчета с соответствующими рецензиями комиссия вправе не допустить студента к защите.
8. Защита отчетов проводится на кафедре ИСиТП комиссией, в состав которой входят руководители практик от кафедры, а также другие преподаватели кафедры.
9. Отчет с указанием даты защиты передается руководителем практики от кафедры ИСиТП ответственному лицу (методисту) по кафедре ИСиТП. Отметки о защите отчета по практике проставляются руководителем практики от кафедры ИСиТП в зачетной книжке и экзаменационной ведомости. Студенты, не прошедшие практику в установленные учебным планом сроки, допускаются к прохождению практики только по решению деканата.

II. Примерный перечень заданий по учебной практике
Задания, для развития профессиональных навыков по дисциплине «Современные технологии разработки БД»

Каждое задание содержит:

1. Название и цель (акцент на профессиональный навык).
2. Техническое задание (ТЗ).
3. Требования к реализации (кодстайл, безопасность, работа с памятью).

Задание 1. Библиотека

Название: Управление библиотечным фондом и читателями

Цель: Освоить проектирование связей «многие-ко-многим», первичные и внешние ключи, операции JOIN для получения сводных данных.

Техническое задание (ТЗ)

Спроектировать базу данных для библиотеки, в которой хранится информация:

– о книгах (уникальный идентификатор, название, автор, год издания, количество экземпляров);

– о читателях (ID, ФИО, дата регистрации, контактный телефон);

– о выдаче книг (какой читатель, какую книгу, дата выдачи, дата возврата, фактическая дата возврата).

Требуемые возможности:

1. Добавление, редактирование, удаление книг и читателей.
2. Выдача книги читателю (если есть свободный экземпляр) и возврат.
3. Получение списка книг, находящихся на руках у конкретного читателя.
4. Получение отчёта о самых популярных книгах (по количеству выдач).

Требования к реализации

Кодстайл: имена таблиц и полей – snake_case, SQL-ключевые слова – заглавными буквами. Каждый запрос должен быть осмысленно отформатирован.

Безопасность: при встраивании в приложение использовать параметризованные запросы, никогда не подставлять значения напрямую (защита от SQL-инъекций).

Работа с памятью: явно закрывать соединение с БД после выполнения операций. Использовать транзакции для групп изменений (выдача/возврат книги). Установить PRAGMA foreign_keys = ON.

Задание 2. Интернет-магазин

Название: Учёт товаров, заказов и клиентов

Цель: Научиться применять агрегатные функции, группировки, подзапросы и транзакции при оформлении заказа.

Техническое задание (ТЗ)

Создать БД для интернет-магазина:

– таблица categories (ID, название);

– таблица products (ID, название, цена, остаток на складе, category_id);

– таблица customers (ID, email, дата регистрации);

– таблица orders (ID, customer_id, дата заказа, статус);

– таблица order_items (ID заказа, ID товара, количество, цена на момент заказа).

Функционал:

1. Корзина – добавление товаров, изменение количества, оформление заказа (списание остатков, расчёт суммы).

2. Получение общей суммы заказов по каждому клиенту.

3. Список самых продаваемых товаров (по количеству единиц).

4. Отчёт по выручке за выбранный период.

Требования к реализации

Кодстайл: для сложных запросов использовать CTE (Common Table Expressions) для повышения читаемости.

Безопасность: проверять, что остаток на складе не становится отрицательным (CHECK-ограничение или триггер). Для цен использовать тип DECIMAL.

Работа с памятью: каждое изменение данных (оформление заказа) выполнять внутри одной транзакции. При работе из нескольких потоков использовать WAL-режим.

Задание 3. Студенты и оценки

Название: Успеваемость студентов

Цель: Освоить оконные функции, ранжирование и сложные подзапросы для анализа успеваемости.

Техническое задание (ТЗ)

Схема:

– students (ID, ФИО, группа);

– subjects (ID, название предмета);

– teachers (ID, ФИО);

– subject_teacher (связь предмет-преподаватель);

– grades (ID студента, ID предмета, оценка, дата получения, ID преподавателя).

Задачи:

1. Рассчитать средний балл каждого студента по всем предметам.

2. Для каждого предмета вывести топ-3 студента по среднему баллу (с использованием оконной функции RANK()).

3. Найти студентов, у которых есть хотя бы одна двойка (оценка < 3).

4. Получить динамику успеваемости группы по месяцам.

Требования к реализации

Кодстайл: избегать SELECT * – перечислять только нужные поля. Комментировать сложные части запросов.

Безопасность: ограничить диапазон оценок через CHECK (grade BETWEEN 2 AND 5). Для дат использовать тип DATE.

Работа с памятью: при больших объёмах данных создавать индексы на

поля, участвующие в WHERE, JOIN и ORDER BY. Следить за размером журнала транзакций.

Задание 4. Сотрудники и отделы (иерархия)

Название: Управление организационной структурой

Цель: Научиться работать с самореференциальными связями и рекурсивными CTE для построения деревьев.

Техническое задание (ТЗ)

Таблицы:

- departments (ID, название);
- employees (ID, ФИО, должность, зарплата, department_id, manager_id – ссылка на другого сотрудника).

Необходимо:

1. Вывести полное подчинение: для каждого сотрудника – его начальника и начальника начальника (рекурсивный запрос).
2. Получить всех подчинённых заданного руководителя (на любую глубину).
3. Рассчитать среднюю зарплату по отделам, включая вложенные отделы (рекурсивная агрегация).
4. Найти сотрудников, у которых зарплата выше, чем у их непосредственного начальника.

Требования к реализации

Кодстайл: рекурсивные CTE оформлять с явным указанием RECURSIVE, выравнивать UNION ALL.

Безопасность: предусмотреть защиту от циклических ссылок (при вставке проверять, что manager_id не равен собственному ID).

Работа с памятью: для рекурсивных запросов следить за глубиной рекурсии (SQLite ограничивает по умолчанию, можно задать PRAGMA recursive_triggers). Использовать временные индексы на manager_id.

Задание 5. Блог (посты, комментарии)

Название: Система ведения блога

Цель: Изучить триггеры, представления и автоматическое обновление временных меток.

Техническое задание (ТЗ)

Схема:

- users (ID, логин, email, дата регистрации);
- posts (ID, заголовок, содержание, автор_id, дата_создания, дата_обновления);
- comments (ID, пост_id, автор_id, текст, дата_создания).

Требования:

1. При добавлении нового комментария автоматически обновлять date_updated соответствующего поста (триггер).
2. Создать представление post_summary, показывающее для каждого поста: заголовок, количество комментариев, дату последнего комментария.

3. Запретить удаление пользователя, если у него есть хотя бы один пост (триггер или внешний ключ с ON DELETE RESTRICT).

4. Выбрать посты, у которых количество комментариев больше среднего по всем постам.

Требования к реализации

Кодстайл: именовать триггеры по схеме trigger_{table}_{event}_{action}. Использовать NEW и OLD явно.

Безопасность: валидировать длину текста на уровне приложения. Для хранения паролей (если бы они были) использовать хеширование, но в учебной БД – просто строки.

Работа с памятью: триггеры не должны выполнять медленных подзапросов. Для представления post_summary можно создать материализованное представление с периодическим обновлением.

Задание 6. Авиабилеты (рейсы, бронирования)

Название: Система бронирования авиабилетов

Цель: Практика работы с датами и временем, проверка целостности данных (ограничения, внешние ключи).

Техническое задание (ТЗ)

Сущности:

– airports (код IATA, название, город);

– flights (номер рейса, аэропорт вылета, аэропорт прилёта, время вылета, время прилёта, максимальное количество мест);

– passengers (ID, ФИО, паспортные данные);

– bookings (ID, пассажир_id, рейс_id, дата бронирования, статус (подтверждён/отменён), место).

Запросы:

1. Найти все рейсы, отправляющиеся в ближайшие 24 часа от текущего момента.

2. Вывести список свободных мест на заданный рейс.

3. Для каждого пассажира – количество совершённых перелётов.

4. Проверить, что при бронировании не превышается лимит мест (использовать транзакцию с проверкой).

Требования к реализации

Кодстайл: для работы с датами использовать встроенные функции SQLite (strftime, julianday). Все временные поля хранить в формате ISO (YYYY-MM-DD HH:MM:SS).

Безопасность: на поле passport_data наложить ограничение UNIQUE. Для статуса бронирования использовать CHECK (подтверждён/отменён).

Работа с памятью: при частом поиске по дате вылета создать индекс на departure_time. Использовать BEGIN IMMEDIATE для блокировки записи при бронировании последнего места.

Задание 7. Финансовый учёт

Название: Учёт доходов и расходов

Цель: Освоить согласованность данных при помощи транзакций, сложные обновления с вычислениями.

Техническое задание (ТЗ)

База данных для личных финансов:

– accounts (ID, название, текущий баланс, валюта);

– cateories (ID, название, тип: доход/расход);

– transactions (ID, счёт_id, категория_id, сумма, дата, описание, тип (приход/расход)).

Функции:

1. Добавление транзакции с автоматическим изменением баланса счёта (в одной транзакции).

2. Получение отчёта: остаток на счетах, сумма доходов и расходов по категориям за период.

3. Перевод средств между счетами (списание с одного, зачисление на другой) с проверкой достаточности средств.

4. Выявление подозрительных транзакций (больше среднего за последние 30 дней).

Требования к реализации

Кодстайл: хранить суммы в DECIMAL(10,2). Для перевода использовать именованную транзакцию с SAVEPOINT.

Безопасность: при списании проверять balance >= amount через CHECK или явную проверку перед обновлением. Не допускать отрицательного баланса.

Работа с памятью: для длительных отчётов использовать курсоры (в приложении). Настроить PRAGMA journal_mode = WAL для конкурентных чтений и записи.

Задание 8. Складской учёт

Название: Управление запасами на складе

Цель: Оптимизация запросов, использование индексов и материализованных представлений (через отдельные таблицы).

Техническое задание (ТЗ)

Схема:

– warehouses (ID, название, адрес);

– products (ID, артикул, наименование, единица измерения);

– stock (склад_id, товар_id, количество);

– supply (приход: дата, товар_id, склад_id, количество, цена закупки);

– shipment (отгрузка: дата, товар_id, склад_id, количество, цена продажи).

Задачи:

1. Рассчитать текущий остаток каждого товара на каждом складе (сумма приходов минус сумма отгрузок) – оптимизировать запрос.

2. Создать таблицу stock_snapshot (материализованное представление), обновляемую раз в сутки, для быстрого отчёта по остаткам.

3. Найти товары, которые заканчиваются (остаток меньше

минимального порога, заданного в отдельной таблице).

4. Получить оборот (сумма продаж) по каждому товару за месяц.

Требования к реализации

Кодстайл: для сложных агрегаций использовать временные таблицы (CREATE TEMP TABLE). Имена индексов – idx_таблица_поле.

Безопасность: при вставке в supply или shipment проверять существование склада и товара через внешние ключи. Избегать целочисленного переполнения (суммы большие – использовать INTEGER с проверкой).

Работа с памятью: для материализованного представления создать триггеры, которые обновляют его при изменении исходных данных (или сделать ежедневный скрипт). Индексировать поля date в таблицах движения.

Задание 9. Социальная сеть (граф друзей)

Название: Управление связями между пользователями

Цель: Реализация графовых структур в реляционной БД, рекурсивные запросы для поиска путей и рекомендаций.

Техническое задание (ТЗ)

Сущности:

– users (ID, имя, дата регистрации);

– friends (user_id, friend_id, статус (подтверждён/ожидание), дата_дружбы);

– messages (ID, отправитель, получатель, текст, дата);

– likes (ID, пользователь, тип объекта (пост/комментарий), объект_id, дата).

Необходимо:

1. Найти всех друзей пользователя (прямые связи, статус подтверждён).

2. Построить список друзей второй степени (друзья друзей, исключая прямых друзей).

3. Найти кратчайший путь между двумя пользователями (рекурсивный CTE с ограничением глубины).

4. Предложить потенциальных друзей (друзья друзей, не являющиеся друзьями, с общим количеством общих друзей > 2).

Требования к реализации

Кодстайл: в таблице friends использовать составной первичный ключ (user_id, friend_id) и проверку user_id < friend_id для уникальности (или триггер).

Безопасность: при удалении пользователя каскадно удалять его связи в friends, messages, likes.

Работа с памятью: для рекурсивного поиска пути установить ограничение глубины (например, WHERE depth < 5) чтобы избежать бесконечных циклов. Индексировать friends(user_id, friend_id).

Задание 10. Медицинская клиника

Название: Учёт пациентов, врачей и приёмов

Цель: Обеспечение безопасности конфиденциальных данных (разграничение доступа через представления), хранение истории болезней.

Техническое задание (ТЗ)

Схема:

- patients (ID, ФИО, дата рождения, пол, СНИЛС, телефон, адрес);
- doctors (ID, ФИО, специализация, кабинет);
- appointments (ID, пациент_id, врач_id, дата_приёма, диагноз, лечение, рецепт);
- prescriptions (ID, приём_id, лекарство, дозировка, срок).

Задачи:

1. Создать представление public_patients, которое показывает только ФИО, дату рождения и телефон (скрывает СНИЛС и адрес) для роли «регистратура».
2. Получить список пациентов, которые не посещали врача более года.
3. Вывести топ-3 самых частых диагнозов у врача-терапевта.
4. Реализовать триггер, который не позволяет удалить врача, если у него есть приёмы за последние 3 месяца.

Требования к реализации

Кодстайл: для чувствительных полей использовать явное именование (например, snils). Представления именовать с префиксом v_ или в зависимости от роли.

Безопасность: в учебной БД не требуется шифрование, но обязательно демонстрировать разделение доступа через CREATE VIEW. Использовать CHECK для валидации дат (приём не раньше текущей даты).

Работа с памятью: для быстрого поиска по дате приёма создать индекс на appointments.appointment_date. При работе с представлением public_patients оптимизатор SQLite использует индексы базовой таблицы.

Задания, для развития профессиональных навыков по дисциплине «Нейронные сети»

Каждое задание содержит:

1. Название и цель (акцент на профессиональный навык).
2. Техническое задание (ТЗ).
3. Требования к реализации (кодстайл, безопасность, работа с памятью).

Задание 1. Реализация перцептрона с нуля

Цель: развитие навыка низкоуровневой реализации базовых алгоритмов нейронных сетей, понимание математической основы градиентного спуска и бинарной классификации.

Техническое задание:

Реализовать класс Perceptron, который обучается на датасете (например, искусственно сгенерированные точки двух классов, линейно разделимые). Класс должен содержать методы:

– fit(X, y, epochs, learning_rate) – обучение с использованием стохастического градиентного спуска;

– predict(X) – предсказание метки класса (0 или 1);

– score(X, y) – вычисление точности.

Запрещено использовать готовые библиотеки для нейронных сетей (можно NumPy). Реализовать функцию потерь (бинарную кросс-энтропию) и её производную. Провести эксперимент: обучить на 80% данных, оценить на 20%, построить график разделяющей линии.

Требования к реализации:

Кодстайл: следовать PEP 8, именование snake_case для функций и переменных, CamelCase для классов; каждый публичный метод снабдить docstring с описанием параметров и возвращаемого значения.

Безопасность: проверять размерности входных матриц (X должен быть 2D, y – 1D); обрабатывать случай пустых данных; выбрасывать исключения с понятными сообщениями.

Работа с памятью: для хранения весов использовать массив NumPy с типом float32; избегать копирования данных в циклах – использовать векторизованные операции; при большом количестве эпох использовать генераторы для батчей, но в базовой версии достаточно полных проходов.

Задание 2. Реализация обратного распространения ошибки для многослойного перцептрона

Цель: углубление навыков дифференцирования и программной реализации алгоритма backpropagation, отладка градиентных вычислений.

Техническое задание:

Создать класс MLP, позволяющий задать произвольное количество скрытых слоёв с функцией активации ReLU или Sigmoid, выходной слой – Sigmoid (бинарная классификация) или Softmax (многоклассовая). Реализовать методы:

– forward(X) – прямой проход;

– backward(y_true, learning_rate) – обратный проход с обновлением весов (вручную, без autograd).

Обучить на датасете make_moons из scikit-learn (2 класса) и на make_circles. Сравнить точность с перцептроном. Визуализировать границу решений.

Требования к реализации:

Кодстайл: модульная структура (отдельные функции для активаций и их производных); комментировать сложные математические операции (размерности тензоров).

Безопасность: проверять соответствие размерностей при умножении матриц; использовать np.clip для предотвращения переполнения в сигмоиде.

Работа с памятью: хранить промежуточные значения (активации, выходы до активации) только для одного прямого прохода; после обратного прохода удалять ненужные ссылки; использовать float32 для экономии

памяти.

Задание 3. Классификация изображений с использованием сверточной нейронной сети

Цель: освоение фреймворка глубокого обучения (PyTorch или TensorFlow), развитие навыков построения и обучения сверточных архитектур.

Техническое задание:

Построить CNN для классификации датасета CIFAR-10 (10 классов). Архитектура: два сверточных слоя (Conv2D + ReLU + MaxPooling) и два полносвязных. Обучить с использованием оптимизатора Adam, функцией потерь CrossEntropyLoss. Добиться точности не менее 65% на тестовой выборке за 20 эпох. Реализовать сохранение лучшей модели по валидационной точности и раннюю остановку (early stopping).

Требования к реализации:

Кодстайл: разбить код на ячейки в Jupyter Notebook или отдельные скрипты: подготовка данных, определение модели, цикл обучения, визуализация. Использовать type hints для функций.

Безопасность: обрабатывать ошибки загрузки данных (если нет локальной копии – скачать); использовать try/except при чтении/записи модели.

Работа с памятью: загружать данные батчами через DataLoader (размер батча 32–64); после обучения удалять ненужные тензоры; при необходимости использовать torch.no_grad() при валидации.

Задание 4. Регуляризация и борьба с переобучением

Цель: развитие навыков диагностики и устранения переобучения, применение различных техник регуляризации.

Техническое задание:

Взять модель из задания 3 (или простую полносвязную сеть для MNIST) и продемонстрировать переобучение (например, тренировка на малом количестве данных или слишком сложная модель). Применить последовательно: L2-регуляризацию (weight decay), Dropout, Batch Normalization, Data Augmentation. Для каждого метода измерить точность на тренировочной и валидационной выборках, построить графики. Написать краткий отчёт о том, какой метод дал наибольший прирост обобщающей способности.

Требования к реализации:

Кодстайл: использовать конфигурационный словарь для гиперпараметров (L2-коэффициент, dropout rate); логировать метрики в таблицу (pandas).

Безопасность: проверять, что при включении Dropout в режиме оценки (eval) он отключается; использовать model.train() и model.eval().

Работа с памятью: при аугментации данных использовать генераторы (ImageDataGenerator или torchvision.transforms), чтобы не хранить все

преобразованные изображения в памяти.

Задание 5. Обучение рекуррентной нейронной сети для генерации текста

Цель: приобретение навыков работы с последовательными данными, реализация RNN/LSTM, генерация новых последовательностей.

Техническое задание:

Используя символьный уровень (character-level RNN), обучить модель генерировать текст в стиле заданного корпуса (например, произведения Шекспира или русские народные сказки). Задачи:

1. Загрузить текст, построить словарь уникальных символов.
2. Создать RNN/LSTM с одним слоем и размером скрытого состояния 128.
3. Реализовать генерацию: подавать на вход начальную строку (seed), предсказывать следующий символ с температурой (softmax с коэффициентом).
4. Обучить с помощью кросс-энтропии, оптимизатор RMSprop.
5. Сгенерировать 500 символов от разных seed-ов.

Требования к реализации:

Кодстайл: функции для создания последовательностей (X, y) как сдвинутых окон; отдельный класс генератора; документация по параметрам температуры.

Безопасность: проверять, что seed-строка полностью состоит из символов словаря; иначе приводить к ближайшим или выдавать ошибку.

Работа с памятью: не загружать весь датасет как тензор сразу – использовать генератор батчей, который читает текст порциями; при длинных текстах использовать усечённое обратное распространение (truncated backpropagation).

Задание 6. Трансферное обучение с использованием предобученной модели

Цель: освоение техники переноса знаний (transfer learning), fine-tuning предобученных сетей, адаптация под новый домен.

Техническое задание:

Взять предобученную на ImageNet модель (VGG16, ResNet50 или MobileNetV2) из torchvision.models или tensorflow.keras.applications. Адаптировать её для классификации датасета из 5 классов (например, цветы, собаки vs кошки). Реализовать два подхода:

– Feature extraction (заморозить все слоё, обучить только новую голову);

– Fine-tuning (разморозить последние несколько слоёв и дообучить с маленькой скоростью обучения).

Сравнить точность и время обучения. Визуализировать confusion matrix.

Требования к реализации:

Кодстайл: использовать аргументы командной строки или конфигурационный файл для выбора модели, размера изображения, learning

rate.

Безопасность: при загрузке предобученных весов проверить доступность интернета, при отсутствии – использовать локальную копию; нормализовать входные изображения в соответствии со статистикой ImageNet.

Работа с памятью: применять аугментацию on-the-fly; при fine-tuning загружать модель в память один раз, использовать requires_grad=False для замороженных слоёв, чтобы не хранить их градиенты.

Задание 7. Создание простого автокодировщика для сжатия данных

Цель: развитие навыков обучения без учителя (unsupervised learning), понимание скрытых представлений и реконструкции данных.

Техническое задание:

Реализовать полносвязный автокодировщик (сжатие до 32 нейронов) для датасета рукописных цифр MNIST. Состоит из энкодера (уменьшение размерности) и декодера (восстановление). Обучить с MSE-лоссом. Визуализировать:

– исходные и восстановленные изображения;

– скрытое пространство в 2D (после применения PCA или t-SNE к 32-мерным векторам).

Дополнительно: сделать денуазирующий автокодировщик (вход – изображение с шумом, выход – чистое).

Требования к реализации:

Кодстайл: разделение на модули encoder, decoder; функция добавления гауссовского шума; логирование потерь.

Безопасность: проверять, что входные изображения нормализованы к [0,1]; для денуазинга – диапазон шума не должен выводить значения за пределы допустимого.

Работа с памятью: при обучении использовать генератор батчей; для визуализации выбирать небольшое количество изображений (например, 10), чтобы не перегружать память.

Задание 8. Визуализация и интерпретация нейронных сетей

Цель: приобретение навыков анализа внутреннего представления сети, отладки и повышения доверия к модели.

Техническое задание:

Для обученной на CIFAR-10 или ImageNet модели (можно взять предобученную из задания 6) реализовать:

– Визуализацию фильтров первого сверточного слоя.

– Градиентный метод получения карт значимости (Grad-CAM) для выбранного класса на конкретном изображении.

– Анализ активаций скрытых слоёв (построить гистограммы распределений).

– Написать интерпретацию: какие признаки выделяет сеть.

Требования к реализации:

Кодстайл: использовать готовые библиотеки (например, torchcam или написать свой Grad-CAM); комментировать этапы.

Безопасность: проверять, что модель находится в режиме eval; изображения нормализованы.

Работа с памятью: для визуализации фильтров использовать только веса, не запуская прямой проход; для Grad-CAM – загружать одно изображение, не хранить все промежуточные градиенты одновременно.

Задание 9. Оптимизация гиперпараметров с использованием библиотек

Цель: развитие навыков автоматического подбора гиперпараметров, организации экспериментов, использования фреймворков оптимизации.

Техническое задание:

Для задачи классификации (например, MNIST) и модели MLP с 1-2 скрытыми слоями провести поиск оптимальных гиперпараметров с помощью Optuna или Hyperopt. Параметры для настройки: количество нейронов в скрытом слое, learning rate, коэффициент L2, batch size, функция активации. Цель – максимизировать точность на валидационной выборке. Использовать 5-fold кросс-валидацию. Построить графики важности гиперпараметров (optuna.visualization). Сохранить лучшую конфигурацию.

Требования к реализации:

Кодстайл: целевая функция должна быть чистой (без побочных эффектов); логировать все попытки в файл.

Безопасность: ограничить количество итераций (trials) разумным числом; обрабатывать ошибки в целевой функции (например, расхождение размерностей).

Работа с памятью: после каждой итерации удалять модель и данные из памяти (использовать del, gc.collect); для кросс-валидации не загружать все фолды одновременно.

Задание 10. Развертывание нейронной сети в веб-приложении

Цель: получение навыков инжиниринга ML-моделей, создания API, контейнеризации и интеграции с веб-интерфейсом.

Техническое задание:

Взять обученную модель (например, классификатор MNIST из задания 3 или 7). Разработать веб-сервис на Flask или FastAPI, который:

– принимает POST-запрос на /predict с изображением (base64 или загруженный файл);

– предобрабатывает изображение (ресайз, нормализация);

– возвращает предсказанный класс и вероятности.

Дополнительно: создать простую HTML-страницу с формой загрузки изображения и отображением результата. Обеспечить обработку ошибок (неверный формат, слишком большой файл). Упаковать приложение в Docker-контейнер.

Требования к реализации:

Кодстайл: разделение на модули: model.py (загрузка модели, препроцессинг), app.py (эндпоинты); использовать pydantic для валидации входных данных.

Безопасность: ограничить максимальный размер загружаемого файла; проверять MIME-тип; не выполнять произвольные операции с путями; использовать переменные окружения для секретов.

Работа с памятью: загружать модель один раз при старте приложения (глобальный объект); после предсказания очищать тензоры; при необходимости использовать асинхронные очереди для больших нагрузок.

Задания, для развития профессиональных навыков по дисциплине «Имитационное моделирование сложных информационных систем»

Каждое задание содержит:

4. Название и цель (акцент на профессиональный навык).
5. Техническое задание (ТЗ).
6. Требования к реализации (кодстайл, безопасность, работа с памятью).

Задание 1. Разработка дискретно-событийного ядра симуляции

Цель: Формирование навыка проектирования и реализации дискретно-событийного движка с управлением очередью событий и виртуальным временем.

Техническое задание

Реализовать класс EventSimulationEngine, поддерживающий:

– Добавление событий с заданным временем срабатывания.

– Обработку событий в хронологическом порядке (без задержек).

– Возможность отмены запланированного события.

– Автоматическое завершение при достижении заданного модельного времени simulation_time.

Входные данные: поток событий (ID события, время, тип).

Выходные данные: лог обработки событий с отметкой реального (модельного) времени.

Требования к реализации

Кодстайл: Google C++ Style Guide или PEP 8 (Python). Имена сущностей отражают их роль (EventQueue, Scheduler).

Безопасность:

Проверка корректности времени (неотрицательное, не меньше текущего).

Защита от бесконечного цикла при некорректных данных.

Работа с памятью (C++):

Использовать std::unique_ptr для событий.

Приоритетная очередь через std::priority_queue.

Запрещены сырые new/delete вне RAII-обёрток.

Тестирование: Юнит-тесты для проверки порядка событий и отмены.

Задание 2. Моделирование многоканальной системы массового обслуживания (СМО)

Цель: Развитие навыка сбора и статистического анализа выходных характеристик системы в процессе имитации.

Техническое задание

Построить имитационную модель M/M/c (пуассоновский входящий поток, экспоненциальное обслуживание, c каналов).

Параметры:

- λ (интенсивность поступления заявок),
- μ (интенсивность обслуживания на канале),
- c (число каналов),
- максимальная длина очереди L (опционально).

Собрать статистику:

1. Среднее время ожидания в очереди.
2. Вероятность отказа (если очередь ограничена).
3. Коэффициент использования каналов.
4. Доверительный интервал по результатам прогонов.

Требования к реализации

Кодстайл: Документирование публичных методов (Doxygen/ docstrings).

Безопасность:

Контроль переполнения счётчиков.

Защита от отрицательных интенсивностей.

Память (C++):

Хранение заявок в очереди через std::queue<Request> без динамического выделения каждой заявки (можно пул объектов).

Дополнительно: Реализовать множественные прогоны (например, 1000) и расчёт доверительного интервала.

Задание 3. Имитационная модель распределённой системы с сетевыми задержками

Цель: Освоение приёмов моделирования асинхронного взаимодействия и синхронизации времени в распределённых узлах.

Техническое задание

Смоделировать систему из N узлов, обменивающихся сообщениями через сеть с задержкой (нормальное распределение). Каждый узел:

- Периодически отправляет heartbeat-сообщения всем другим.
- Фиксирует время получения heartbeat от соседа.
- Если от узла нет сообщения дольше T_timeout – объявляет его отключённым.

Входные данные: N, средняя задержка и дисперсия, T_timeout, время симуляции.

Выходные: лог событий (отправка/получение, таймауты, восстановление).

Требования к реализации

Кодстайл: Модульная структура (классы Node, Network, Message).

Безопасность:

Корректная обработка ситуаций, когда сообщение приходит после таймаута.

Отсутствие гонок данных (в однопоточной событийной модели).

Память:

Эффективное планирование событий (сообщений) – без дублирования.

Освобождение ресурсов после завершения симуляции.

Проф. навык: Написание краткого отчёта о влиянии вариативности задержек на ложные срабатывания.

Задание 4. Оптимизация параметров методом имитационного отжига

Цель: Получение навыка сопряжения имитационной модели с алгоритмом оптимизации и управления стохастичностью.

Техническое задание

Для модели из Задания 2 (M/M/c) найти оптимальное число каналов c при ограничениях:

– Среднее время ожидания в очереди ≤ W_max.

– Минимизация суммарных затрат: cost(c) = c * C_channel + C_wait * W(c).

– Использовать метод имитационного отжига (Simulated Annealing) с динамическим уменьшением "температуры". Каждое значение c оценивается по результатам K прогонов модели.

Требования к реализации

Кодстайл: Разделение кода модели и оптимизатора (интерфейс SimulationEvaluator).

Безопасность:

Защита от бесконечного цикла оптимизации (макс. итерации).

Проверка, что c ≥ 1.

Память:

Переиспользование одного экземпляра модели между прогонами (сброс состояния).

Отсутствие утечек при повторных прогонах.

Проф. навык: Визуализация сходимости (график значения целевой функции от итерации).

Задание 5. Верификация имитационной модели через сравнение с аналитикой

Цель: Развитие навыка верификации и валидации имитационных моделей.

Техническое задание

Разработать модель M/M/1 (один канал).

Запрограммировать дискретно-событийную реализацию.

Рассчитать теоретические значения:

1. Среднее число заявок в системе: $\rho/(1-\rho)$, где $\rho = \lambda/\mu$.

2. Среднее время пребывания: $1/(\mu-\lambda)$.

3. Сравнить результаты модели (по 10 прогонам) с аналитическими при $\lambda=0.6$, $\mu=1.0$.

4. Выполнить t-тест для проверки гипотезы о равенстве средних.

Требования к реализации

Кодстайл: Чёткое разделение расчёта статистики и логики модели.

Безопасность:

Проверка условия стационарности ($\lambda < \mu$) – иначе предупреждение.

Отбрасывание переходного режима (warm-up period).

Память:

Сбор данных в std::vector<double> фиксированного размера.

Использование итераторов для вычислений.

Документация: Описание процедуры верификации в коде (комментарии).

Задание 6. Параллельное имитационное моделирование (распределение прогонов)

Цель: Освоение парадигмы независимых прогонов и ускорения за счёт многопоточности.

Техническое задание

Взять модель из Задания 2 (M/M/c). Реализовать:

– Функцию run_simulation(params, seed), возвращающую результат.

– Пул потоков (или std::async), выполняющий N прогонов параллельно.

– Сбор и агрегацию результатов (среднее, дисперсия, квантили).

Входные данные: число потоков, N прогонов, параметры модели.

Выходные: те же статистики, но время выполнения должно уменьшаться с ростом числа потоков.

Требования к реализации

Кодстайл: Использование идиомы "map-reduce" для прогонов.

Безопасность:

Каждый прогон использует собственный генератор случайных чисел (std::mt19937 с уникальным seed).

Отсутствие гонок при записи результатов (локальные векторы → слияние).

Память:

Минимизация копирований (перемещение результатов).

Освобождение памяти потоков после выполнения.

Проф. навык: Измерение speedup (ускорения) и эффективности распараллеливания в отчёте.

2.2.1. Индивидуальные задания по учебной практике
Инструкция по оформлению тестовых заданий для электронной базы ДГУ.

Формат файла заданий

Задания набираются в редакторе **Word** и сохраняются в формате **DOC.** Шрифт Times New Roman, размер шрифта 14.

Написание ключевых слов

Ключевое слово – это набор символов, расположенный в начале каждого файла заданий и каждого раздела тестового задания, необходимый для автоматической обработки заданий. По ключевым словам программа распознает и обрабатывает каждое тестовое задание и весь файл заданий в целом.

***Внимание*!** Если Вы не отследите правильность написания ключевых слов, то Ваш файл заданий будет обработан с ошибками.

Правила написания ключевых слов:

- каждое ключевое слово пишется на новой строке
- ключевое слово начинается с символа № в начале строки без пробела (набирается одновременным нажатием клавиши SHIFT и цифры 3 в кириллической раскладке клавиатуры)
- ключевые слова пишутся исключительно русскими буквами

При оформлении комплектов тестовых заданий используйте следующие ключевые слова:

для заголовка файла

№автор=
№дисциплина=
№блок=
№модуль=
№тема=

для оформления отдельного задания

№вопрос<форма>
№да
№нет

Внутри ключевого слова не должно быть пробелов!

Заголовок файла

Обязательный набор ключевых слов в заголовке файла:
№дисциплина= <название дисциплины>
№блок= <1, 2, 3>

Формы тестовых заданий и их кодировка

Предусмотрены следующие формы тестовых заданий:

- 1 – с выбором одного правильного ответа из предложенных и развернутым обоснованным ответом (1-5 предложений).
- 2 – с выбором нескольких правильных ответов из предложенных
- 3 – на установление правильной последовательности
- 4 – на установление соответствия
- 5 – с кратким ответом и обоснованием ответа (1-5 предложений)

Форма задания указывается в ключевом слове №вопрос<1, 2, 3, 4, 5>.

Например: №вопрос1 - означает, что за ней следует вопрос с выбором одного правильного ответа из нескольких предложенных.

***Внимание*!** Код формы задания **не отделяется** от ключевого слова пробелом.

Обязательно! После №вопрос<1, 2, 3, 4, 5> указать инструкцию по выполнению задания.

Для каждого типа заданий имеется своя типовая инструкция по выполнению

Варианты оформления тестовых заданий.

№вопрос1 – задания с выбором одного правильного ответа из списка

Перед каждым тестовым заданием должно быть ключевое слово **№вопрос1**. На следующей строке после текста задания должно находиться ключевое слово **№да**, а сточкой ниже – правильный ответ. На следующей строке – ключевое слово **№нет**, затем неправильный ответ, далее – опять ключевое слово **№нет** и следующий неправильный ответ и так далее. В конце тестового задания с новой строки ключевое слово **№ключ,** строкой ниже обоснование выбранного ответа (3-7 предложений)**.**

Количество предложенных ответов может быть в разных заданиях от 4 до 6.

№вопрос1
<Текст задания>
№да
<правильный ответ>
№нет
<дистрактор>
№нет
<дистрактор>
№нет
<дистрактор>
№ключ
<Обоснование выбора ответа. От 3 до 7 предложений.>

Примеры заданий формы 1:

№вопрос1

В модели кругооборота продуктов и доходов в состав денежных утечек включаются

№да

чистые налоги

№нет

государственные закупки

№нет

валовые внутренние инвестиции

№нет

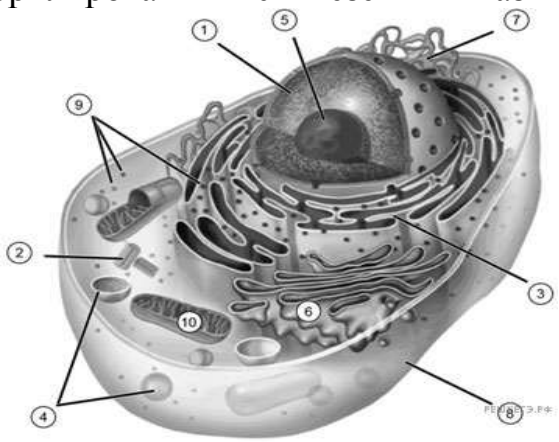
экспортная выручка

№ключ

Совокупные денежные утечки в модели кругооборота продуктов и доходов включают: сбережения, чистые налоги, расходы на импорт.

№вопрос1

Рассмотрите рисунок. Органоид, участвующий в окислительном фосфорилировании и синтезе АТФ называется



№да

митохондрии

№нет

лизосома

№нет

рибосома

№нет

ядрышко

№ключ

Органоид, который участвует в окислительном фосфорилировании и синтезе АТФ, называется: ядро, аппарат Гольджи, вакуоль, митохондрии.

№вопрос2 – задания с выбором нескольких правильных ответов из предложенных

Перед каждым тестовым заданием данной формы должно быть ключевое слово **№вопрос2.**

Количество ответов должно быть от 4 до 6, также обязателен хотя бы один неправильный ответ. Каждый правильный ответ предваряется

ключевым словом **№да**, все они последовательно располагаются после текста задания, каждый неправильный ответ предваряется ключевым словом **№нет**, все неправильные ответы располагаются после правильных.

№вопрос2
<Текст задания>
№да
<правильный ответ>
№да
<правильный ответ>
№да
<правильный ответ>
№нет
<дистрактор>
№нет
<дистрактор>
№нет
<дистрактор>

Примеры задания формы 2:

№вопрос2

Мероприятиями Петра I, направленными на «европеизацию» страны, являлись

№да

упразднение патриаршества и создание Синода

№да

создание регулярной армии

№нет

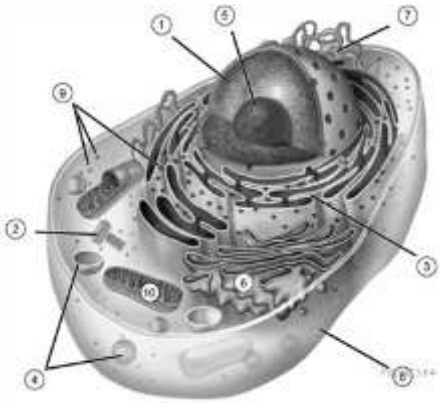
введение общерусского свода законов – Судебника

№нет

освобождение дворянства от обязательной службы

№вопрос2

Перечислить основные признаки подтверждающие, что на рисунке изображена животная клетка.



№да

отсутствие центральной крупной вакуоли

№да

отсутствие пластид
№да
клеточный центр с центриолями
№да
отсутствие толстой клеточной стенки
№нет
наличие ядрышка
№нет
наличие рибосом

№вопрос3 – задания на установление правильной последовательности

Перед каждым тестовым заданием данной формы должно быть ключевое слово **№вопрос3**. Для этой формы допустимы только правильные ответы, при этом последовательность расположения ответов соответствует правильной последовательности (последовательность должна содержать от 4 до 6 ответов).

№вопрос3
<Текст задания>
№да
<правильный ответ>
№да
<правильный ответ>
№да
<правильный ответ>
№да
<правильный ответ>

Пример задания формы 3:

№вопрос3
Расположите туристские районы России с запада на восток
№да
Республика Крым
№да
Верхневолжский
№да
Кольско-Карельский
№да
Уральский
№да
Саянский

№вопрос3
Установите, в какой последовательности происходят процессы митоза:
№да

спирализация хромосом
№да
образование метафазной пластинки
№да
расхождение сестринских хроматид - анафаза
№да
деление цитоплазмы - телофаза

№вопрос4 – задания на установление соответствия

Перед тестовым заданием данной формы обязательно наличие ключевого слова **№вопрос4.**

В данном типе заданий необходимо установить соответствие между двумя списками. Первый список входит в текст самого задания. Второй список относится к ответам, при этом последовательность расположения ответов соответствует правильному соответствию по порядку перечисления.

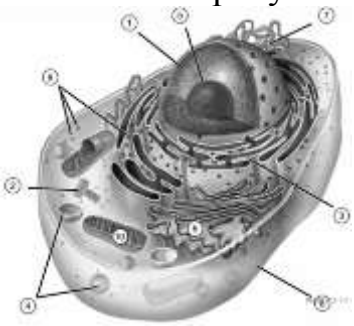
№вопрос4
<Текст задания>
№да
<правильный ответ>
№да
<правильный ответ>
№да
<правильный ответ>
№да
<правильный ответ>

Примеры задания формы 4:

№вопрос4
Установите соответствие между музеем и его местонахождением
1. Галерея Уффици
2. Центр Помпиду
3. Музей Шерлока Холмса
4. Прадо
5. Бельведер
№да
Флоренция
№да
Париж
№да
Лондон
№да
Мадрид
№да
Вена

№вопрос4

Рассмотрите рисунок и установите соответствие между цифрами, обозначенными на рисунке и органоидами клетки:



1. 2.
2. 3.
3. 4.
4. 5.
5. 7.
6. 9.

№да
центриоли
№да
гранулярный ЭПС
№да
лизосомы
№да
ядрышко
№да
гладкий ЭПС
№да
Рибосомы

№вопрос5 – задания с кратким ответом

В данном задании предполагается только правильный ответ. Ответ должен быть представлен одним словом или числом.

№вопрос5
<Текст задания>
№да
<правильный ответ>
№ключ
<Обоснование выбора ответа. От 3 до 7 предложений>

Примеры задания формы 5

№вопрос5

Цена товара А равна 22,5, цена товара В равна 18 доход потребителя равен 400 у.е. если потребитель приобрел товаров А в 5 раз меньше, чем товаров В, то объем приобретенных товаров А составит _____. (Ответ округлите с точностью до десятых.)

№да
3,5
№ключ
Так как покупатель приобрел товаров А в 5 раз меньше, чем товара В, то $Q_b=5Q_a$.

Подставим данные значения в формулу дохода $M = Q_aP_a + Q_bP_b$. Получим $400 = 22,5Q_a + 18*5Q_a$. $Q_a = 3,5$

№вопрос5
*Моносахарид, входящий в состав ДНК, называется*_____
№да
дезоксирибоза
№ключ
Нуклеиновые кислоты это – высокомолекулярные соединения, которые являются хранителями и носителями наследственной информации. Известно два вида нуклеиновых кислот – ДНК и РНК. По своей структуре нуклеиновые кислоты являются полимерами, мономерами которых являются нуклеотиды. В состав нуклеотида входит: азотистое основание, моносахарид и остаток фосфорной кислоты. В состав ДНК входят азотистые основания: аденин, гуанин, цитозин и тимин. В состав РНК входят аденин, гуанин, цитозин и урацил, вместо тимина. Моносахаридом, входящим в состав ДНК, является дезоксирибоза, в составе РНК – рибоза. Обязательным компонентом всех нуклеиновых кислот является фосфорная кислота, через которую происходит соединение нуклеотидов в длинные линейные цепи.

№вопрос5
Мера государственного принуждения, назначаемая по приговору суда к лицу, признанному виновным в совершении преступления, заключающаяся в лишении или ограничении прав и свобод этого лица, является _____ (Введите слово в форме соответствующего падежа.)
№да
наказание
№ключ
УК РФ Статья 43. Понятие и цели наказания
1. Наказание есть мера государственного принуждения, назначаемая по приговору суда. Наказание применяется к лицу, признанному виновным в совершении преступления, и заключается в предусмотренных настоящим Кодексом лишении или ограничении прав и свобод этого лица.
2. Наказание применяется в целях восстановления социальной справедливости, а также в целях исправления осужденного и предупреждения совершения новых преступлений.

Кейс – задание обозначается с помощью ключевого слова **№вопроскейс.**

№вопроскейс
<Текст задания>

№ключ
<Ответ. ½ листа формата А4.>

Примеры оформления кейс – задания:

№вопроскейс

Прочитайте текст и запишите развернутый обоснованный ответ.

Министерство внутренних дел Российской Федерации (МВД России) является

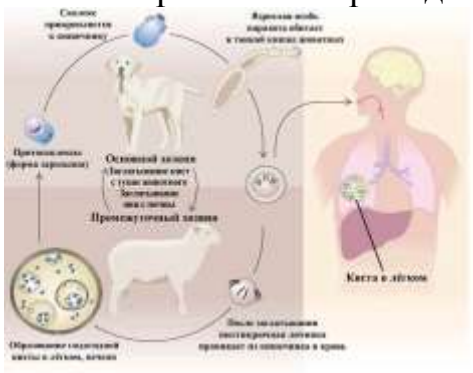
№ключ

федеральным органом исполнительной власти, осуществляющим функции по выработке и реализации государственной политики и нормативно-правовому регулированию в сфере внутренних дел, в сфере контроля за оборотом наркотических средств, психотропных веществ и их прекурсоров, в сфере миграции (далее - сфера внутренних дел), а также правоприменительные функции по федеральному государственному контролю (надзору) в сфере внутренних дел

№вопроскейс

Прочитайте текст и запишите развернутый обоснованный ответ.

Одним из самых опасных для человека червей является эхинококк, заражение которым часто приводит к смертельному исходу.



Рассмотрите схему жизненного цикла эхинококка и ответьте, каким из хозяев является человек. Ответ поясните.

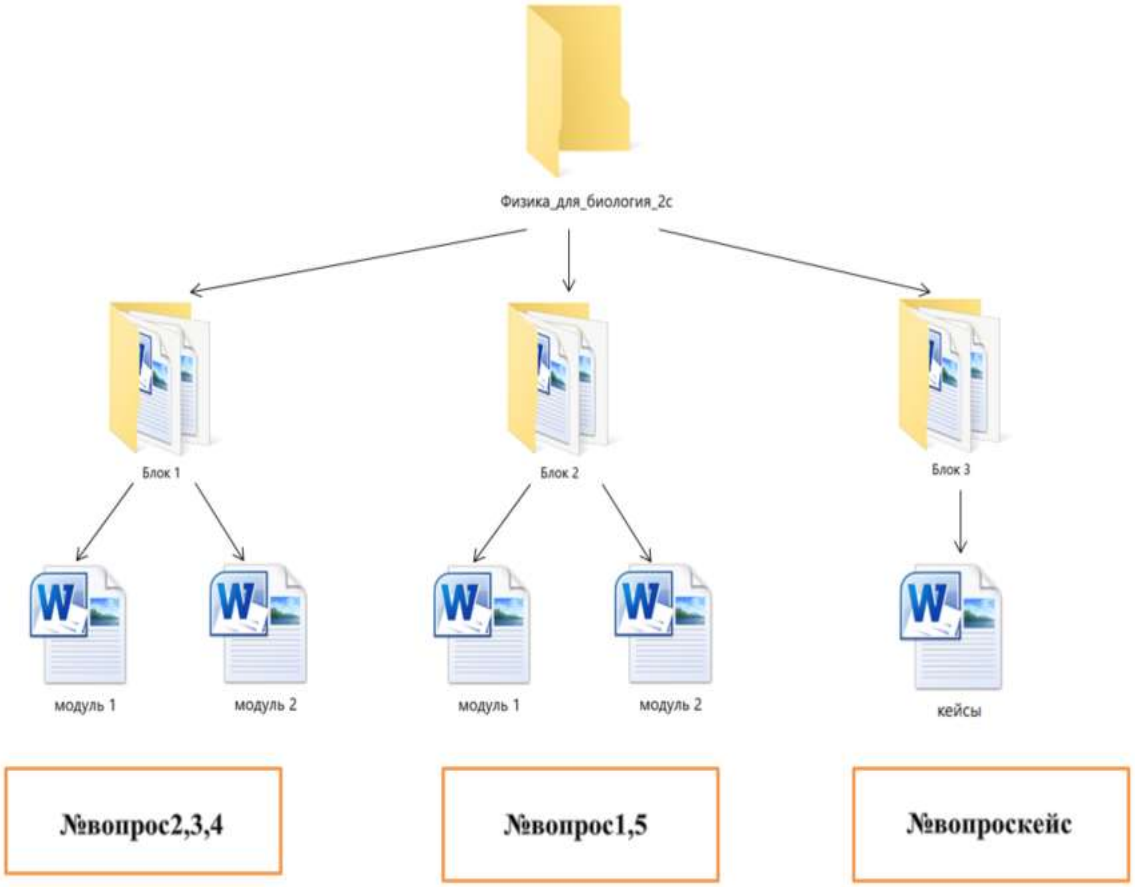
№ключ

Эхинококк - опасный паразит, не щадит человека - использует его только как промежуточного хозяина. В роли промежуточного хозяина эхинококка может выступать домашний скот (коровы, овцы, свиньи) или человек. Заражение происходит перорально. В кишечнике из яйца выходит личинка с шестью подвижными крючьями на заднем конце тела (онкосфера). С помощью крючьев она проникает через стенку кишечника в систему воротной вены и с кровью заносится в печень, реже в лёгкие, мышцы, кости или другие органы.

Человек может заразиться эхинококкозом от собак при несоблюдении правил личной гигиены, при употреблении немытых овощей, которые контаминированы яйцами эхинококка. Окончательный хозяин заражается, поедая органы (финнозное мясо) пораженных животных. Профилактика заключается в соблюдении правил гигиены, мытье рук после соприкосновения с собакой, дегельминтизации домашних животных

(выведение червей из организма).

Структура электронной базы тестовых заданий по дисциплине



Тест состоит из 16 ТЗ и 1 кейса

	Кол-во ТЗ по формам	Время, мин	Кол-во баллов за 1 ТЗ	16 ТЗ+ 1 кейс	Из 100 баллов	За 1 ТЗ по 100 балльной шкале
Блок 1						
№вопрос2	**4**	1-3	1	12	50	4,2
№вопрос3	**4**	1-3	1			4,2
№вопрос4	**4**	1-3	1			4,2
Блок 2						
№вопрос1	**2**	5-7	1	4	35	8,75
№вопрос5	**2**	5-7	1			8,75
Блок 3						
№вопроскейс	**1**	15	3	1	15	15,00



Требования к количеству тестовых заданий

№	*Кол-во л/ч*	*Минимальное кол-во ТЗ*	*Количество Т/З по формам**					
			Блок 1			**Блок 2**		**Блок 3**
			№вопрос2	*№вопрос3*	*№вопрос4*	*№вопрос1*	*№вопрос5*	*№вопроскейс*
1	**34-36**	**500**	**125**	**75**	**75**	**125**	**100**	**50**
2	**28-32**	**400**	**100**	**60**	**60**	**100**	**80**	**40**
3	**<26**	**300**	**75**	**45**	**45**	**75**	**60**	**30**

*№вопрос2 - 25% от общего количества ТЗ
№вопрос3 - 15% от общего количества ТЗ
№вопрос4 - 15% от общего количества ТЗ
№вопрос1 - 25% от общего количества ТЗ
№вопрос5 - 20% от общего количества ТЗ
№вопроскейс – фиксированное число

III. Перечень контрольных вопросов при приеме материалов учебной практики

1. Что такое нейронная сеть и каковы ее основные свойства?
2. Какова структура нейрона?
3. Какие функции активации могут быть использованы в нейронных сетях?
4. Какие требования предъявляются к функциям активации?
5. Какие функции выполняет входной слой в многослойной сети?
6. Можно ли обучить нейронную сеть без скрытого слоя?
7. В чем заключается обучение нейронных сетей?
8. Почему один из алгоритмов обучения получил название «алгоритм обратного распространения»?
9. Чем отличается обучение с учителем от обучения без учителя?
10. Почему входные и выходные сигналы нейронной сети должны быть нормированы?
11. Архитектура нейронных сетей.
12. Проектирования нейросетевых моделей.
13. Этапы проектирования и разработки нечетких продукционных систем
14. Метод градиентного спуска и обратного распространения ошибки
15. Полносвязные нейронные сети
16. Сверточные нейронные сети
17. Рекуррентные нейронные сети
18. Анализ изображений
19. Анализ естественного языка
20. Анализ временных рядов
21. Инструментарий разработки систем искусственного интеллекта
22. Модели и слои Keras.
23. Основы работы с последовательной моделью Keras
24. Основы работы в TensorFlow.
25. Задача кластеризации. Карты Кахоннена.
26. История развития теории искусственных нейронных сетей.
27. Сеть Кохонена. Стахостические сети. Обучение Больцмана.
28. Сети с обратными связями.
29. Сеть Хопфилда. Сеть Хэминга.
30. Локализация объектов, сегментация.
31. Сверточные нейронные сети и автоэнкодеры
32. Векторизация слов. Embedding слой Keras
33. Рекуррентные нейронные сети. LSTM сети.

Приложение 1

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«ДАГЕСТАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
Факультет информатики и информационных технологий

Отчет

по учебной (ознакомительной) практике

Выполнил: магистр 1 курса,
очной формы обучения
направления 09.04.03.
Прикладная информатика
Магомедов А.А.

Руководитель: к.э.н.,
доцент кафедры ИСиТП
Баммаева Г.А.

Махачкала 2026