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
09.03.04 Программная инженерия

Профиль подготовки:
Разработка программно-информационных систем,
Программное обеспечение вычислительной техники и автоматизированных систем
Разработка программного обеспечения и приложений искусственного интеллекта

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. Структура и содержание учебной практики. 5
1.6. Требования к оформлению отчета по учебной ознакомительной практике 6
1.7. Подготовка презентационного материала 10
1.8. Аттестация по итогам практики 11
II. Примерные перечень заданий по учебной практике 12
III. Примерный перечень контрольных вопросов при приеме материалов учебной практики 37
Приложение 1 39

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 09.03.04 Программная инженерия, профили – «Разработка программно-информационных систем», «Программное обеспечение вычислительной техники и автоматизированных систем», «Разработка программного обеспечения и приложений искусственного интеллекта».

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 09.03.04 «Программная инженерия», направленность (профили) «Разработка программно-информационных систем», «Программное обеспечение вычислительной техники и автоматизированных систем», «Разработка программного обеспечения и приложений искусственного интеллекта» 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 09.03.04 «Программная инженерия» являются также обеспечение непрерывности и последовательности овладения студентами 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 09.03.04 «Программная инженерия», направленность (профили) «Разработка программно-информационных систем», «Программное обеспечение вычислительной техники и автоматизированных систем», «Разработка программного обеспечения и приложений искусственного интеллекта»:

– закрепление и углубление теоретических знаний, полученных студентами при изучении дисциплин учебного плана, таких как «Инженерная графика»», «Функциональное и логическое моделирование», «Машинное обучение»;

– формирование умений использовать на практике математический аппарат, принципы и методы компьютерного решения сложных научно - технических задач получения, хранения и переработки информации;

– формирование навыков использования технологии, позволяющей описать сложные системы и явления в природе и обществе при решении современных и перспективных задач;

– изучение современного состояния и направлений развития компьютерной техники и информационных технологий;

– 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 результатовпроведенного 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 3 курсе во 2 семестре, срок проведения – 2 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.

1.5. 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, особенность и описание практических задач, выполненных студентом во время практики.

– Выполнение индивидуальной работы (составление тестов по дисциплине курса).

– 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.

1.6. 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 расположения приложений 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.

1.7. 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)

- последний лист «Благодарю Вас за внимание».

1.8. Аттестация по итогам практики

Защита учебной практики проводится после окончания практики в сроки, определяемые деканатом.

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».

Защита студентом учебной практики оценивается комиссией по модульно-рейтинговой системе (отлично, хорошо, удовлетворительно, неудовлетворительно).

Сроки прохождения практики по согласованию с деканатом факультета могут быть продлены студентам, нарушающим трудовую дисциплину.

Полностью оформленный отчет представляется руководителю практики от кафедры «Информационные системы и технологии программирования» для проверки и защиты.

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.

Защита отчетов проводится на кафедре ИСиТП комиссией, в состав которой входят руководители практик от кафедры, а также другие преподаватели кафедры.

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. Примерные перечень заданий по учебной практике
Задания, для развития профессиональных навыков по дисциплине «Машинное обучение»

Задание 1. Исследовательский анализ данных (EDA) и визуализация

Цель: развить навык первичного анализа структуры, качества и взаимосвязей в данных для обоснованного выбора методов предобработки и моделирования.

Техническое задание:

Дан набор данных о недвижимости (файл housing.csv): признаки — площадь, количество комнат, этаж, год постройки, район; целевая переменная — цена. Выполните:

- загрузку данных с проверкой на пропуски и выбросы;
- статистический анализ (среднее, медиана, квантили, дисперсия) для числовых признаков;
- визуализацию распределений (гистограммы, box-plot) и матрицу корреляций;
- вывод о наличии мультиколлинеарности и нелинейных зависимостей.

Требования к реализации:

Код в стиле PEP8, наличие type hints для функций.

Документация (docstrings) для каждой функции анализа.

Обработка возможных ошибок (например, отсутствие файла, некорректные типы данных).

Использование pandas, matplotlib/seaborn, numpy.

Экономия памяти: при работе с большими данными использовать чанковую загрузку (если файл >100 МБ).

Задание 2. Предобработка данных и создание пайплайна

Цель: освоить построение воспроизводимых пайплайнов обработки признаков с учётом защиты от утечек данных (data leakage).

Техническое задание:

Создайте Pipeline с помощью sklearn.pipeline.Pipeline и ColumnTransformer, который:

- заполняет пропуски в числовых признаках медианой, а в категориальных — константой 'missing';
- масштабирует числовые признаки (StandardScaler);
- кодирует категориальные признаки (OneHotEncoder с handle_unknown='ignore');
- выделяет целевой признак и разделяет данные на train/test (stratify для классификации).

Проверьте пайплайн на датасете titanic.csv.

Требования к реализации:

Наличие unit-тестов (pytest) для проверки работы трансформеров.

Запрет на использование fit_transform на тестовой выборке — только transform.

Обработка памяти: использование sparse-матриц для one-hot признаков при большом количестве категорий.

Логирование основных шагов (logging).

Задание 3. Линейная регрессия и регуляризация (Ridge, Lasso)

Цель: сформировать навык выбора и настройки линейных моделей с регуляризацией для борьбы с переобучением и отбором признаков.

Техническое задание:

На синтетическом датасете (1000 объектов, 50 признаков, из которых только 10 информативных) обучите:

– линейную регрессию без регуляризации;

– Ridge и Lasso регрессии с подбором alpha по кросс-валидации (TimeSeriesSplit или KFold).

– Сравните метрики (MAE, RMSE, R^2) и коэффициенты моделей. Визуализируйте путь регуляризации (коэффициенты vs alpha). Сделайте вывод о том, какая модель лучше отбирает признаки.

Требования к реализации:

Все функции должны быть аннотированы (вход/выход).

Использование sklearn.linear_model.RidgeCV, LassoCV.

Сохранение обученной модели в файл (joblib) с проверкой целостности.

Безопасность: валидация входных данных на соответствие ожидаемой размерности.

Задание 4. Деревья решений и визуализация переобучения

Цель: научиться диагностировать переобучение и управлять сложностью дерева через параметры и пост-обрезку.

Техническое задание:

– На датасете wine (классификация) обучите дерево решений (DecisionTreeClassifier) с разными глубинами (1–15).

– Постройте графики зависимости ошибки на train и validation от глубины.

– Выберите оптимальную глубину по критерию F1-macro. Визуализируйте лучшее дерево с помощью plot_tree. Примените cost-complexity pruning (ccp_alpha) и сравните результат.

Требования к реализации:

Разделение данных на train/validation/test в соотношении 60/20/20.

Функция для расчёта метрик и построения графиков.

Проверка на утечку: параметры валидации не должны влиять на тестовую выборку.

Использование graphviz для красивой визуализации (опционально).

Задание 5. Ансамблевые методы: Random Forest и Gradient Boosting

Цель: приобрести навык настройки ансамблей, работы с out-of-bag оценками и анализа важности признаков.

Техническое задание:

На датасете credit_risk (бинарная классификация) обучите:

– Random Forest с подбором n_estimators, max_depth, min_samples_split;

– Gradient Boosting (например, XGBoost или CatBoost) с ранней остановкой.

– Сравните ROC-AUC, время обучения, важность признаков (встроенная важность и SHAP). Постройте калибровочную кривую. Реализуйте бэггинг поверх бустинга для улучшения устойчивости.

Требования к реализации:

Использование GridSearchCV с n_jobs=-1 для эффективного параллелизма. Обработка памяти: контроль размера модели, удаление неиспользуемых деревьев. Логирование процесса обучения (прогресс-бар, время итераций). Наличие файла requirements.txt с версиями библиотек.

Задание 6. Кластеризация и снижение размерности (PCA, t-SNE)

Цель: развить навыки поиска скрытых групп объектов и визуализации многомерных данных.

Техническое задание:

Для датасета digits (рукописные цифры, 64 признака) выполните:

– стандартизацию признаков;

– PCA (выберите число компонент, объясняющих 95% дисперсии);

– кластеризацию методом KMeans (определите оптимальное k через силуэт и инерцию);

– визуализацию результатов в 2D (используя первые две главные компоненты или t-SNE) с цветовой маркировкой по истинным меткам и по кластерам.

Сравните качество кластеризации (adjusted Rand index, NMI) при исходных признаках и после PCA.

Требования к реализации:

Использование sklearn.decomposition.PCA, sklearn.manifold.TSNE.

Экономия памяти: для t-SNE применить метод Барнса-Хата (method='barnes_hut').

Документирование обоснования выбора числа кластеров.

Наличие функции для автоматического подбора параметров кластеризации.

Задание 7. Работа с временными рядами: скользящее окно и прогнозирование

Цель: освоить специфику подготовки данных для прогнозирования временных рядов (лаггинг, скользящие статистики) и валидацию без перемешивания.

Техническое задание:

Дан набор данных о суточном потреблении электроэнергии (файл energy.csv). Постройте модель (например, Random Forest или LSTM) для прогноза значения на следующий день, используя признаки:

- лаги (t-1, t-2, t-7);
- скользящее среднее за 7 дней;
- день недели, месяц, праздник (бинарный признак).

Разделите данные на train/validation/test без перемешивания (по времени). Оцените MAPE и SMAPE. Визуализируйте реальные и предсказанные значения на тесте.

Требования к реализации:

Собственная реализация функции генерации признаков (без утечки будущего).

Валидация с помощью TimeSeriesSplit (не KFold).

Обработка выбросов и пропусков с учётом временной структуры.

Возможность масштабирования на большие ряды (итеративная загрузка).

Задание 8. Обработка текстов: Bag-of-Words и тематическое моделирование

Цель: приобрести навыки работы с неструктурированными текстовыми данными, построения TF-IDF матриц и выделения тем.

Техническое задание:

Корпус новостных текстов (20_newsgroups). Выполните:

предобработку (приведение к нижнему регистру, удаление пунктуации и стоп-слов, лемматизацию/стемминг);

векторизацию с помощью TfidfVectorizer (ограничение max_features=5000);

кластеризацию текстов (KMeans) и интерпретацию тем через наиболее частотные слова в кластере;

тематическое моделирование методом LDA (sklearn или gensim) и визуализацию с помощью pyLDAvis.

Требования к реализации:

Обработка памяти: использование разреженных матриц (scipy.sparse).

Наличие pipeline для воспроизводимой обработки новых текстов.

Безопасность: проверка входных строк на инъекции (например, экранирование для веб-приложения не требуется, но очистка от управляющих символов — да).

Документация шагов предобработки.

Задание 9. Нейронная сеть на PyTorch/TensorFlow для классификации изображений

Цель: научиться проектировать, обучать и валидировать простые свёрточные сети, работать с дата-лоадерами и GPU-памятью.

Техническое задание:

На датасете CIFAR-10 постройте свёрточную нейронную сеть (например, 3 свёрточных слоя + 2 полносвязных) с аугментацией (RandomHorizontalFlip, RandomCrop). Обучите с оптимизатором Adam, функцией потерь CrossEntropyLoss. Реализуйте:

разделение train/val (80/20);
раннюю остановку (early stopping) по accuracy на валидации;
логирование кривых обучения (loss, accuracy);
сохранение лучшей модели.
Добейтесь accuracy на тесте >75%.

Требования к реализации:

Использование DataLoader с num_workers для эффективной загрузки.

Управление памятью GPU: очистка кэша (torch.cuda.empty_cache()), использование pin_memory.

Код должен работать как на CPU, так и на GPU (проверка device).

Type hints и docstrings для всех пользовательских классов/функций.

Задание 10. Развёртывание модели в виде REST API (MLOps)

Цель: развить навык инкапсуляции обученной модели в веб-сервис с обработкой запросов, валидацией и логированием.

Техническое задание:

Обученную модель из Задания 3 (Ridge регрессия) заверните в веб-приложение на FastAPI или Flask. Эндпоинты:

POST /predict — принимает JSON с признаками, возвращает предсказание;

GET /health — проверка доступности модели;

GET /metadata — возвращает информацию о модели (версия, метрики, дата обучения).

Реализуйте:

валидацию входных данных (Pydantic);

асинхронную обработку (при возможности);

логирование каждого запроса с временной меткой и IP;

unit-тесты для эндпоинтов.

Требования к реализации:

Загрузка модели из файла при старте сервера.

Обработка ошибок: 400 при неверном формате, 500 при внутренней ошибке.

Работа с памятью: модель выгружается при завершении приложения, повторная инициализация не требуется.

Код должен быть готов к контейнеризации (наличие Dockerfile).

Задания, для развития профессиональных навыков по дисциплине «Функциональное и логические программирование»

Задание 1. Сжатие данных методом RLE (функциональное программирование)

1. Название и цель

Название: Реализация RLE-сжатия и распаковки с помощью рекурсии и функций высшего порядка.

Цель (профессиональный навык):

Освоение рекурсивных стратегий обработки списков, применение функций map, fold/reduce, работа с неизменяемыми структурами данных и сопоставлением с образцом.

2. Техническое задание (ТЗ)

Разработать модуль RLE, предоставляющий две чистые функции:

encode :: Eq a => [a] -> [(Int, a)]

Сжимает входной список, группируя одинаковые подряд идущие элементы в пары (количество, элемент).

Пример: encode "aaabbca" → [(3,'a'), (2,'b'), (1,'c'), (1,'a')].

decode :: Eq a => [(Int, a)] -> [a]

Восстанавливает исходный список по сжатому представлению.

Пример: decode [(3,'a'), (2,'b'), (1,'c'), (1,'a')] → "aaabbca".

Дополнительное требование:

Для пустого списка обе функции должны возвращать корректный пустой результат.

Обработка ошибок: если в decode встретился ноль или отрицательное количество – выбросить исключение с понятным сообщением (или вернуть Maybe).

3. Требования к реализации

Кодстайл:

Имена функций в camelCase или snake_case (согласно стандартам выбранного языка – Haskell: camelCase).

Разделение типов и реализации, явные аннотации типов для всех экспортируемых функций.

Комментарии к сложным рекурсивным вызовам, пояснение базовых случаев.

Длина строки ≤ 80 символов.

Безопасность:

Использовать только чистые функции (без IO в encode/decode).

Для обработки некорректных данных в decode – возвращать Maybe [(Int,a)] или Either String [a], чтобы избежать runtime-ошибок.

Предусмотреть защиту от бесконечной рекурсии (убедиться, что рекурсивный шаг уменьшает размер входных данных).

Проверить, что функция encode работает с бесконечными списками только при ленивых вычислениях (в Haskell – допустимо, но без принудительной оценки всего списка).

Работа с памятью:

Избегать лишних копий списков: использовать хвостовую рекурсию с аккумулятором или свёртки (foldr/foldl').

В decode использовать concatMap или concat с повторением элементов, но без порождения промежуточных списков большого размера (для длинных последовательностей – replicate).

Для языка Scheme/Common Lisp – избегать конструирования больших списков через append, применять cons с последующим обращением.

Задание 2. Ленивое бинарное дерево поиска (функциональное программирование)

1. Название и цель

Название: Построение бесконечного дерева поиска и ленивый обход.

Цель (профессиональный навык):

Освоение механизма отложенных вычислений (ленивость), создание и обработка потенциально бесконечных структур данных, оптимизация за счёт разделения генерации и потребления.

2. Техническое задание (ТЗ)

Реализовать модуль для работы с ленивым бинарным деревом поиска, которое хранит элементы в отсортированном порядке. Дерево определяется как:

text

data LazyBST a = Empty | Node a (LazyBST a) (LazyBST a)

Функции:

fromList :: Ord a => [a] -> LazyBST a – строит сбалансированное (насколько возможно) дерево из списка, но без полного вычисления структуры (только корень и необходимые ветви по требованию).

contains :: Ord a => a -> LazyBST a -> Bool – проверяет наличие элемента, не обходя всё дерево (ранний выход).

takeWhileTree :: (a -> Bool) -> LazyBST a -> [a] – возвращает список элементов из дерева в порядке возрастания, пока выполняется предикат. Работает лениво: при обращении к следующему элементу вычисляется только необходимый фрагмент дерева.

Пример использования (Haskell):

haskell

let bigTree = fromList [1..1000000] -- не занимает память мгновенно

takeWhileTree (< 10) bigTree -- [1,2,3,4,5,6,7,8,9]

3. Требования к реализации

Кодстайл:

Использовать экспорт только необходимых функций, скрыть конструкторы (LazyBST экспортировать как абстрактный тип).

Определить Foldable и Traversable (если язык позволяет) для демонстрации идиоматичности.

Документировать ленивые свойства функций (где и почему вычисление откладывается).

Безопасность:

fromList не должен зависать или вызывать переполнение стека для больших списков – использовать итеративный алгоритм построения с ленивым разделением (например, построение из отсортированного списка с помощью индексов).

contains должен корректно работать с бесконечными деревьями (если элемент отсутствует в левой бесконечной ветви – не уходить в правую).

Все функции обязаны быть тотальными (обрабатывать Empty).

Работа с памятью:

Избегать удержания ссылок на неиспользуемые части дерева (thunk leaks). В Haskell – использовать seq или BangPatterns там, где необходима строгость (например, при подсчёте элементов).

Для takeWhileTree организовать обход так, чтобы после получения результата не сохранялись корни неиспользованных ветвей (позволить GC собирать их).

При создании дерева из списка не хранить исходный список после построения.

Задание 3. Задача о восьми ферзях (логическое программирование)

1. Название и цель

Название: Решение задачи расстановки ферзей на шахматной доске с использованием механизма поиска с возвратом.

Цель (профессиональный навык):

Применение декларативного подхода, использование автоматического перебора (backtracking), отсечений (cut) для повышения эффективности, работа со списками и генерацией решений.

2. Техническое задание (ТЗ)

На языке Prolog реализовать предикат queens(N, Positions), который:

N – размер доски (N×N) и количество ферзей.

Positions – список координат [X1/Y1, X2/Y2, ...] (или список столбцов, где индекс – номер строки).

Предикат должен быть многоразовым (через ; выдавать все возможные решения).

Дополнительно: предикат queens_all(N, ListOfSolutions) собирающий все решения в список.

Требования к решению:

Ферзи не должны бить друг друга (нет двух на одной горизонтали, вертикали, диагонали).

Использовать стандартное представление: позиция задаётся столбцом для каждой строки от 1 до N.

Обеспечить отсечение заведомо неверных ветвей (при добавлении очередного ферзя сразу проверять конфликты).

Пример запроса:

prolog

?- queens(4, Pos).

Pos = [2,4,1,3] ; % или [1,3,2,4] в другом представлении

false.

3. Требования к реализации

Кодстайл:

Имена предикатов строчными буквами, логические переменные начинаются с заглавной.

Разбиение на вспомогательные предикаты (safe/3, member/2, delete/3 – не использовать встроенные, если они не оптимизированы).

Комментарии для каждого правила: что оно выражает.

Использовать :- module для инкапсуляции.

Безопасность:

Избегать бесконечного зацикливания – гарантировать, что рекурсия уменьшает количество оставшихся ферзей.

Обрабатывать случай N = 0 (пустое решение) и N = 1.

Не использовать ! без необходимости, но если используется – документировать, какие ветки отсекаются и почему это корректно.

Работа с памятью:

В Prolog память управляется автоматически, но следует избегать хранения всех решений одновременно (если не требуется queens_all).

Для queens_all использовать findall/3 или bagof/3, но понимать, что при N > 10 список может быть огромным.

Оптимизировать хвостовую рекурсию: в предикате генерации перестановок использовать аккумулятор и проверять безопасность на лету, чтобы не хранить все перестановки.

Задание 4. Экспертная система для диагностики ПК (логическое программирование)

1. Название и цель

Название: Разработка экспертной системы с прямым и обратным цепочками рассуждений.

Цель (профессиональный навык):

Построение базы знаний на основе правил, реализация механизма логического вывода, работа с недетерминизмом и вводом/выводом (консольный интерфейс).

2. Техническое задание (ТЗ)

Создать экспертную систему, которая помогает пользователю диагностировать неисправности компьютера. Система задаёт вопросы пользователю (Yes/No/Unknown) и на основе правил делает заключение.

База знаний (минимум 5 правил и 8 фактов):

Пример правил:

компьютер_не_включается → проверить питание_подаётся?, если нет → проблема_с_блоком_питания.

экран_не_работаетΛиндикатор_монитора_горит→неисправен_кабель_монитора. **и** т.д.

Функциональность:

Система задаёт вопросы, пока не достигнет одного или нескольких заключений.

Поддержка обратного цепочки: начиная с гипотезы, ищет подтверждающие факты.

Возможность объяснения (why): почему задаётся вопрос?

Вывод списка возможных неисправностей с оценкой уверенности (просто / возможно / маловероятно).

3. Требования к реализации

Кодстайл:

Модульная организация: база правил, механизм вывода, пользовательский интерфейс.

Предикаты для вопросов: ask(Question, Answer) с кэшированием ответов.

Использовать динамические предикаты (dynamic) для хранения полученных фактов.

Безопасность:

Обрабатывать ввод пользователя, не допускать зацикливания при противоречивых правилах.

Задавать каждый вопрос не более одного раза за сессию.

Предусмотреть возможность сброса сессии (retract всех фактов).

Работа с памятью:

Не накапливать бесконечное количество фактов – после вывода заключения очищать динамическую базу (или использовать assert временных фактов).

Для длинных цепочек рассуждений – ограничить глубину рекурсии (например, счётчиком шагов).

Если используется SWI-Prolog – избегать утечек через recorda/recorded, использовать nb_setval только при необходимости.

Задания, для развития профессиональных навыков по дисциплине «Инженерная графика»

Задание 1. Алгоритм Брезенхема для отрезка

Цель: освоить растровый алгоритм построения отрезка, закрепить навыки работы с целочисленной арифметикой и пиксельной графикой.

Техническое задание:

Реализовать функцию drawLine(x1, y1, x2, y2, color), которая строит отрезок на растровом изображении (или в окне) с использованием алгоритма Брезенхема. Отрезок должен корректно обрабатывать все октанты.

Требования к реализации:

Кодстайл: именование функций и переменных в едином стиле (CamelCase или snake_case), комментарии к сложным участкам.

Безопасность: проверка выхода координат за границы изображения, отказ от рекурсии.

Работа с памятью: если используется динамический буфер пикселей — явное выделение/освобождение памяти или std::vector.

Задание 2. Заливка многоугольника (алгоритм построчного сканирования)

Цель: научиться заполнять замкнутые области, работать со списками рёбер и сортировкой.

Техническое задание:

Дан многоугольник, заданный массивом вершин (координаты целые).

Реализовать заливку цветом с помощью активного списка рёбер. Для каждого скана (y) определить пересечения с рёбрами и заполнить пиксели между парами.

Требования к реализации:

Кодстайл: разбиение на логические функции (добавление ребра, обновление X, сортировка).

Безопасность: обработка случая вырожденных рёбер (вертикальных или нулевой длины).

Работа с памятью: динамическое создание списка рёбер, удаление после заливки, избегание утечек.

Задание 3. Трёхмерное вращение куба (матричные преобразования)

Цель: получить навык работы с однородными координатами и матрицами поворота в 3D.

Техническое задание:

Построить каркасную модель куба с центром в начале координат. Реализовать поворот куба вокруг оси Y на заданный угол (градусы). Отобразить проекцию на плоскость XY (ортографическую). Управление углом с клавиатуры или пошагово.

Требования к реализации:

Кодстайл: отдельный класс Matrix4x4, методы для умножения матриц и векторов.

Безопасность: проверка выхода за диапазон при индексации, использование assert для отладки.

Работа с памятью: массив вершин – фиксированный (стек) или std::array, никаких new[] без delete[].

Задание 4. Чтение и отображение простого DXF-файла (только LINE)

Цель: приобрести практику парсинга текстовых графических форматов и извлечения геометрических примитивов.

Техническое задание:

Написать парсер для DXF файла (ASCII), который находит секцию ENTITIES, вычитывает все сущности типа LINE и извлекает координаты начал и концов. Отобразить все отрезки в окне (масштабирование по габаритам).

Требования к реализации:

Кодстайл: обработка файла потоковым методом, именование констант для кодов групп DXF.

Безопасность: проверка существования файла, корректное закрытие потока, отказ от gets() и буферных переполнений.

Работа с памятью: хранение отрезков в std::vector, динамическое расширение без ручного управления памятью.

Задание 5. Проекция 3D-точки на плоскость

Цель: понять математику перспективного и ортографического проецирования.

Техническое задание:

Задана точка в пространстве (x,y,z) и параметры камеры (позиция, направление взгляда, угол обзора). Реализовать две функции: orthoProject (ортографическая) и perspProject (перспективная с делением на z). Вывести координаты проекции на экран.

Требования к реализации:

Кодстайл: структура Point3D, отдельные функции для преобразований.

Безопасность: проверка деления на ноль (z близкое к нулю), обработка точек за спиной камеры.

Работа с памятью: все данные передаются по значению или константной ссылке, никакого динамического выделения для одной точки.

Задание 6. Алгоритм Midpoint для окружности

Цель: закрепить целочисленные растровые алгоритмы, симметрию окружности.

Техническое задание:

Реализовать функцию drawCircle(xc, yc, radius, color), строящую окружность по алгоритму средней точки. Использовать симметрию восьми октантов.

Требования к реализации:

Кодстайл: вспомогательная функция setPixel с проверкой границ, чёткое разделение логики вычисления и отрисовки.

Безопасность: радиус не может быть отрицательным, целочисленные переполнения в промежуточных вычислениях – использовать long при необходимости.

Работа с памятью: без динамической памяти (вся работа с пикселями через прямой доступ к буферу или вызовы графической библиотеки).

Задание 7. Классы «Точка» и «Вектор» с операциями

Цель: научиться проектировать геометрические типы данных, перегрузку операторов.

Техническое задание: Создать классы Point2D и Vector2D. Реализовать операции: сложение векторов, вычитание точек → вектор, умножение на скаляр, скалярное произведение, длина вектора, нормализация. Написать юнит-тесты (или демонстрационную программу).

Требования к реализации:

Кодстайл: инкапсуляция (приватные поля x,y), константные методы, перегрузка операторов +, -, *, /.

Безопасность: проверка нормализации нулевого вектора (выброс исключения или возврат нулевого).

Работа с памятью: классы не используют динамическую память (правило трёх/пяти не требуется).

Задание 8. Отсечение отрезка прямоугольным окном (Коэн–Сазерленд)

Цель: получить навык реализации классического алгоритма отсечения, кодирования областей.

Техническое задание:

Заданы прямоугольное окно отсечения (x_min, y_min, x_max, y_max) и отрезок. Реализовать функцию, возвращающую видимую часть отрезка (или флаг, что отрезок невидим). Отобразить исходный и отсечённый отрезки.

Требования к реализации:

Кодстайл: таблица кодов (константы), функции вычисления кода точки, обработка всех случаев пересечений.

Безопасность: избегать бесконечного цикла (максимум 4 итерации), корректно обрабатывать вертикальные/горизонтальные отрезки.

Работа с памятью: только локальные переменные, без динамического выделения.

Задание 9. Перспективная проекция с выбором точки схода

Цель: понять влияние параметров перспективы на визуальное восприятие.

Техническое задание:

Создать программу, которая отображает простой параллелепипед (8 вершин) в перспективной проекции. Пользователь интерактивно задаёт координаты точки схода (x_s, y_s, z_s). Реализовать пересчёт проекций и перерисовку.

Требования к реализации:

Кодстайл: отдельный модуль для перспективного преобразования, структура Vertex.

Безопасность: проверка совпадения точки схода с положением камеры, избегание деления на ноль.

Работа с памятью: динамический массив вершин (использовать std::vector), безопасное изменение размера.

Задание 10. Построение кривой Безье (кубической)

Цель: освоить параметрическое задание кривых, вычисление полиномов Бернштейна.

Техническое задание:

Заданы 4 опорные точки (P0, P1, P2, P3). Реализовать отрисовку кривой Безье путём вычисления 100 точек для $t \in [0,1]$ и соединения их отрезками. Отобразить контрольный многоугольник (линии между P0-P1, P1-P2, P2-P3).

Требования к реализации:

Кодстайл: функция bezierPoint(t, points) возвращает точку, основной цикл с шагом dt.

Безопасность: проверка количества точек (ровно 4), диапазон t от 0 до 1 включительно.

Работа с памятью: опорные точки хранятся в std::array или std::vector фиксированного размера, промежуточные точки вычисляются на лету или в локальном массиве.

Общие рекомендации для всех заданий:

Использовать систему контроля версий (Git) для каждого задания.

Для графического вывода допускается любая простая библиотека (SFML, Qt, WinAPI GDI, OpenGL в режиме immediate).

В отчёте по практике обязательно приводить диаграммы классов/алгоритмов и скриншоты результатов.

2.2. 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

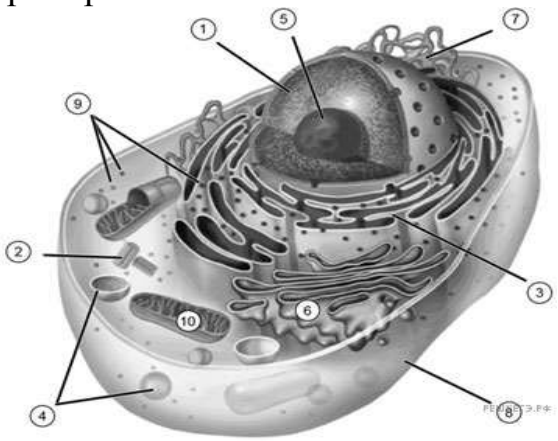
экспортная выручка

№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**.

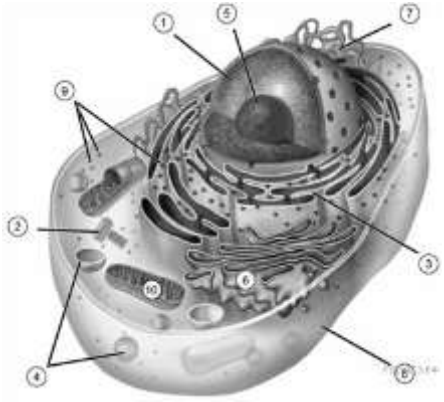
Количество ответов должно быть от 4 до 6, также обязателен хотя бы один неправильный ответ. Каждый правильный ответ предваряется ключевым словом **№да**, все они последовательно располагаются после текста задания, каждый неправильный ответ предваряется ключевым словом **№нет**, все неправильные ответы 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>

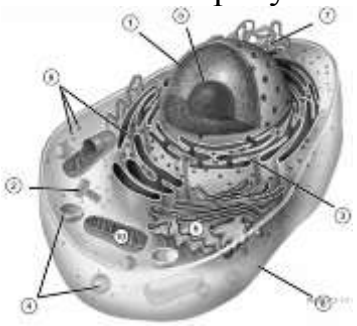
Примеры задания формы 4:

№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

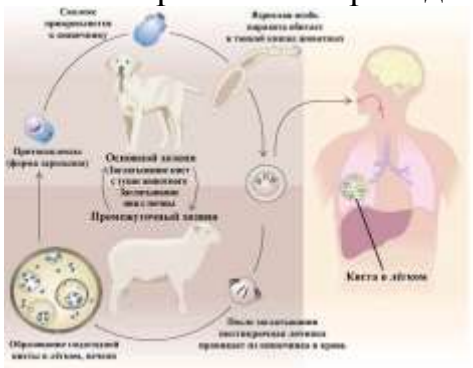
№ключ

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.



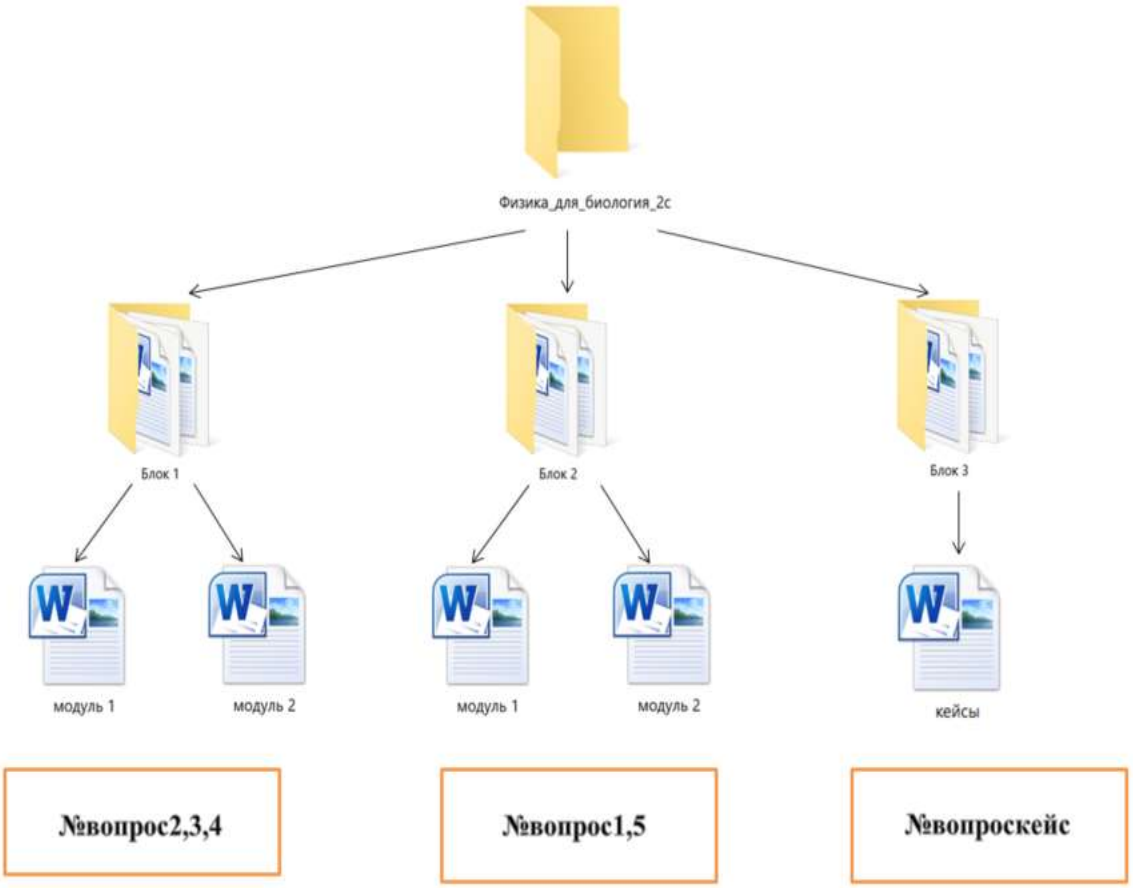
Рассмотрите схему жизненного цикла эхинококка и ответьте, каким из хозяев является человек. Ответ поясните.

№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. Примерный перечень контрольных вопросов при приеме материалов учебной практики

1. Предмет теории вероятностей. Событие. Классификация событий.
2. Классическое и статистическое определение вероятности. Свойства вероятности.
3. Элементы комбинаторики.
4. Зависимые и независимые события. Произведение событий.
5. Теоремы умножения вероятностей.
6. Независимые события. Теорема умножения для независимых событий.
7. Условная вероятность. Теорема умножения вероятностей зависимых событий.
8. Сумма событий.
9. Совместные и несовместные события.
10. Теоремы сложения вероятностей.
11. Полная группа событий. Сумма вероятностей событий, образующих
12. полную группу.
13. Вероятность противоположного события; вероятность осуществления только одного события; вероятность осуществления хотя бы одного события.
14. Формула полной вероятности.
15. Вероятность гипотез. Формула Бейеса.
16. Формула Бернулли.
17. Локальная и интегральная теоремы Муавра-Лапласа.
18. Формула Пуассона для редких событий.
19. Наивероятнейшее число появления события в серии повторяющихся испытаний.
20. Дискретные и непрерывные случайные величины.
21. Закон распределения вероятностей случайной величины.
22. Математическое ожидание дискретной случайной величины.
23. Свойства математического ожидания.
24. Вероятностный смысл математического ожидания.
25. Дисперсия, среднеквадратическое отклонение, мода дискретной случайной величины.
26. Свойства дисперсии дискретной случайной величины.
27. Определение функции распределения и ее свойства.
28. График функции распределения дискретной случайной величины.
29. Плотность распределения вероятностей непрерывной случайной
30. Свойства функции плотности распределения вероятностей непрерывной случайной величины.
31. Математическое ожидание непрерывной случайной величины.
32. Дисперсия и среднеквадратическое отклонение непрерывной случайной величины.
33. Моменты случайной величины.
34. Асимметрия, эксцесс, мода, медиана случайной величины.
35. Биномиальное распределение.
36. Распределение Пуассона.

37. Равномерный закон распределения.
38. Показательный закон распределения.
39. Нормальный закон распределения. Основные параметры. Вероятностный
40. смысл параметров.
41. Вероятность попадания нормально распределенной непрерывной случайной величины в заданный интервал.
42. Правило «трех» сигм.
43. Теорема Чебышева.
44. Теорема Бернулли.
45. Формулировка центральной предельной теоремы (теорема Ляпунова).
46. Математическая статистика, основные задачи. Понятие первичной статистической совокупности.
47. Задачи математической статистики.
48. Обработка статистических данных.
49. Генеральная совокупность. Выборка. Выбор.
50. Вариационный и статистический ряды.
51. Выборочные числовые характеристики.
52. Определение и свойства статистической оценки.
53. Точечные оценки неизвестных параметров.
54. Интервальные оценки неизвестных параметров
55. Понятие статистической гипотезы.
56. Основные этапы проверки гипотезы.
57. Проверка гипотез о числовых значениях параметров.
58. Сравнение выборочной средней с гипотетической генеральной средней
59. нормальной совокупности.
60. Проверка гипотез о равенстве числовых характеристик генеральных совокупностей.
61. Проверка гипотезы о равенстве долей признака.
62. Исключение резко выделяющихся наблюдений.
63. Проверка гипотезы о виде распределения.
64. Критерий согласия Пирсона
65. Корреляционная связь и ее статистическое изучение.
66. Значимость выборочного коэффициента линейной корреляции.
67. Корреляция и причинная связь.
68. Ранговая корреляция
69. Аппроксимационные модели.
70. Определение параметров аппроксимирующих функций по методу наименьших квадратов.
71. Определение уравнений линейных регрессий.
72. Критерий Фишера проверки адекватности модели.
73. Оценка значимости параметров регрессии. Интервальный прогноз на основе линейного уравнения регрессии.

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: студент 3 курса,
очной формы обучения
направления 09.03.04.
Программная инженерия
Магомедов А.А.

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

Махачкала 2026