



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
«Дальневосточный федеральный университет»
(ДВФУ)

ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ (ШКОЛА)

СОГЛАСОВАНО
Руководитель ОП

(подпись) Величко А.С.
(ФИО)

УТВЕРЖДАЮ
И.о. директора департамента

(подпись) Заболоцкий В.С.
(ФИО)
«_28_» декабря 2021 г.



РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ
Инструментальные средства вычислений и моделирования
Направление подготовки 01.04.04 Прикладная математика
(Аналитические, социальные и экономические сети)
Форма подготовки: очная

курс 1 семестр 1
лекции 0 час.
практические занятия 0 час.
лабораторные работы 72 час.
в том числе с использованием МАО лек. 0 час. / пр. 0 час. / лаб. 72 час.
всего часов аудиторной нагрузки 72 час.
в том числе с использованием МАО 72 час.
самостоятельная работа 144 час.
в том числе на подготовку к экзамену 36 час.
контрольные работы (количество) 3
курсовая работа / курсовой проект не предусмотрен
зачет не предусмотрен
экзамен 1 семестр

Рабочая программа составлена в соответствии с требованиями Федерального государственного образовательного стандарта по направлению подготовки 01.04.04 Прикладная математика, утвержденного приказом Минобрнауки России от 10.01.2018 г. № 15.

Рабочая программа обсуждена на заседании департамента математики, протокол № 6 от 28 декабря 2021 г.

И.о. директора департамента математики Заболоцкий В.С.

Составитель: старший преподаватель Ромашко Д.А.

Владивосток
2021

Оборотная сторона титульного листа РПД

I. Рабочая программа пересмотрена на заседании департамента:

Протокол от « ____ » _____ 20__ г. № ____

Директор департамента _____
(подпись) (И.О. Фамилия)

II. Рабочая программа пересмотрена на заседании департамента:

Протокол от « ____ » _____ 20__ г. № ____

Директор департамента _____
(подпись) (И.О. Фамилия)

III. Рабочая программа пересмотрена на заседании департамента:

Протокол от « ____ » _____ 20__ г. № ____

Директор департамента _____
(подпись) (И.О. Фамилия)

IV. Рабочая программа пересмотрена на заседании департамента:

Протокол от « ____ » _____ 20__ г. № ____

Директор департамента _____
(подпись) (И.О. Фамилия)

АННОТАЦИЯ

Дисциплина «Инструментальные средства вычислений и моделирования» предназначена для студентов направления подготовки 01.04.04 «Прикладная математика», магистерская программа «Аналитические, социальные и экономические сети».

Общая трудоемкость освоения дисциплины составляет 6 зачетных единиц (216 часов). Дисциплина реализуется на 1 курсе в 1-м семестре. Дисциплина входит в обязательные дисциплины базовой части блока «Дисциплины (модули)».

Особенности построения курса: лабораторные работы (72 часа), самостоятельная работа (108 часов), подготовка к экзамену (36 часов).

Цели освоения дисциплины: знакомство студентов с основами обработки и анализа данных; изучение существующих инструментальных средств, предназначенных для обработки и анализа данных; получение практических навыков по численной обработке данных, в том числе при работе в операционной системе Linux; получение практических навыков по разработке схем баз данных; изучение технологии разработки базы данных в одной из современных систем управления базами данных (от формальной постановки задачи и разработки схемы данных до создания пользовательских форм); изучение основ реляционной алгебры; изучение языка запросов SQL

Содержание дисциплины «Инструментальные средства вычислений и моделирования» охватывает следующие разделы: пакеты прикладных программ, системы управления базами данных.

Задачи:

- дальнейшее развитие способности осваивать современные технологии программирования;
- развитие способности отлаживать и тестировать прикладное программное обеспечение, предназначенное для анализа данных;
- развитие способности демонстрировать знания современных языков программирования — Octave и/или Matlab

- развитие способности оперировать основными понятиями теории и практики баз данных, а также основными понятиями реляционной алгебры;
- развитие способности уметь использовать различные модели данных;
- развитие способности уметь создавать запросы различной сложности на языке SQL;
- развитие способности уметь проводить декомпозицию реляционных отношений;
- развитие у студентов способности использовать в профессиональной деятельности одну из современных систем управления базами данных.

Для успешного изучения дисциплины «Инструментальные средства вычислений и моделирования» у обучающихся желательны следующие предварительные компетенции:

- способность выявить сущность проблем, возникающих в ходе профессиональной деятельности, готовность использовать для их решения соответствующий аппарат;
- знание основных разделов линейной алгебры;
- способность создавать необходимое программное обеспечение на одном из современных языков программирования.

В результате изучения данной дисциплины у обучающихся формируются следующие универсальные, общепрофессиональные, профессиональные компетенции (элементы компетенций).

Наименование категории (группы) общепрофессиональных компетенций (при наличии)	Код и наименование общепрофессиональной компетенции	Индикаторы достижения компетенции
Информационно-коммуникационные	ОПК-3. Способен разрабатывать	ОПК-3.1. Использует языки программирования и программные

технологии для профессиональной деятельности	научное программное обеспечение для автоматизации систем и процессов, а также развить информационно-коммуникационные технологии	модули и платформы для автоматизации систем и процессов ОПК-3.2. Применяет информационно-коммуникационные технологии с использованием устройств вычислительной техники, систем телекоммуникации и средств обработки информации
--	---	---

Код и наименование индикатора достижения компетенции	Наименование показателя оценивания (результата обучения по дисциплине)
ОПК-3.1. Использует языки программирования и программные модули и платформы для автоматизации систем и процессов	Знает современные технологии программирования на языке запросов SQL, на языке численной математики Matlab/Octave, на языке обработки текстов AWK
	Умеет использовать программные модули
	Владеет навыками автоматизации процессов в профессиональной деятельности
ОПК-3.2. Применяет информационно-коммуникационные технологии с использованием устройств вычислительной техники, систем телекоммуникации и средств обработки информации	Знает вычислительную технику и ее возможности для обработки и анализа информации
	Умеет разрабатывать программные продукты с помощью информационно-коммуникационных технологий
	Владеет навыками настройки вычислительной техники для работы в интегрированной среде разработки, навыками тестирования и проверки программных продуктов

Для формирования вышеуказанных компетенций в рамках дисциплины «Инструментальные средства вычислений и моделирования» применяются следующие методы активного/интерактивного обучения: занятие-дискуссия, обсуждение и разрешение проблем при создании программного обеспечения с использованием компьютерных технологий.

I. СТРУКТУРА И СОДЕРЖАНИЕ ТЕОРЕТИЧЕСКОЙ ЧАСТИ КУРСА

Не предусмотрена

II. СТРУКТУРА И СОДЕРЖАНИЕ ПРАКТИЧЕСКОЙ ЧАСТИ КУРСА

Лабораторные работы

Тема 1. Работа в операционной системе Linux

- Тема 2. Работа в издательской системе LATEX
- Тема 3. Создание презентации в пакете beamer
- Тема 4. Программирование на языке численной математики GNU Octave
- Тема 5. Построение научной графики в GNUPLOT
- Тема 6. Обработка текстов с помощью скриптового языка AWK
- Тема 7. Знакомство с СУБД
- Тема 8. Выполнение практических заданий с учебной базой данных
- Тема 9. Проектирование собственной базы данных
- Тема 10. Написание запросов на формальном языке реляционной алгебры
- Тема 11. Язык запросов SQL

III. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ

Учебно-методическое обеспечение самостоятельной работы обучающихся по дисциплине «Инструментальные средства вычислений и моделирования» представлено в Приложении 1 и включает в себя:

план-график выполнения самостоятельной работы по дисциплине, в том числе примерные нормы времени на выполнение по каждому заданию;

характеристика заданий для самостоятельной работы обучающихся и методические рекомендации по их выполнению;

требования к представлению и оформлению результатов самостоятельной работы;

критерии оценки выполнения самостоятельной работы.

IV. КОНТРОЛЬ ДОСТИЖЕНИЯ ЦЕЛЕЙ КУРСА

Контролируемые разделы дисциплины, этапы формирования компетенций, виды оценочных средств, зачетно-экзаменационные материалы, комплекты оценочных средств для текущей аттестации, описание показателей и критериев оценивания компетенций на различных этапах их формирования, методические материалы, определяющие процедуры оценивания знаний, умений и навыков и

(или) опыта деятельности, а также критерии и показатели, необходимые для оценки знаний, умений, навыков и характеризующие этапы формирования компетенций в процессе освоения образовательной программы, представлены в Приложении 2.

V. СПИСОК УЧЕБНОЙ ЛИТЕРАТУРЫ И ИНФОРМАЦИОННО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Основная литература

1. Трошина, Г. В. Численные расчеты в среде MatLab : учебное пособие / Г. В. Трошина. — Новосибирск : Новосибирский государственный технический университет, 2020. — 72 с. — ISBN 978-5-7782-4092-6. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/99243.html>.

2. Кузнецов, С. Д. Введение в реляционные базы данных : учебное пособие / С. Д. Кузнецов. — 3-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. — 247 с. — ISBN 978-5-4497-0902-8. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/102002.html>.

3. Назаров, С. В. Современные операционные системы : учебное пособие / С. В. Назаров, А. И. Широков. — 3-е изд. — Москва, Саратов : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. — 351 с. — ISBN 978-5-4497-0385-9. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/89474.html>.

Дополнительная литература

1. Красавин, А. В. Компьютерный практикум в среде matlab : учебное пособие для вузов / А. В. Красавин, Я. В. Жумагулов. — 2-е изд. — Москва : Издательство Юрайт, 2020. — 277 с. — (Высшее образование). — ISBN 978-5-

534-08509-9. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/455824>.

2. Шустова, Л. И. Базы данных : учебник / Л.И. Шустова, О.В. Тараканов. — Москва : ИНФРА-М, 2021. — 304 с. + Доп. материалы [Электронный ресурс]. — (Среднее профессиональное образование). - ISBN 978-5-16-014161-9. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1189322>.

3. Сафонов, В. О. Основы современных операционных систем : учебное пособие / В. О. Сафонов. — 3-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. — 826 с. — ISBN 978-5-4497-0552-5. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/94855.html>.

Перечень ресурсов информационно-телекоммуникационной сети «Интернет»

1. GNU Octave. URL: <http://www.gnu.org/software/octave/>
2. Linux. URL: <http://www.linux.org/>
3. Руководство по интерфейсу пользователя Microsoft Office Access 2010. URL: <http://office.microsoft.com/ru-ru/access-help/HA010341735.aspx>
4. Introduction to Databases | Stanford Online. URL: https://class.stanford.edu/courses/Engineering/db/2014_1/about

Перечень дополнительных информационно-методических материалов

1. Бендел Д., Нейпир Р. Использование Linux. 6-е издание. М.: Вильямс, 2003.
2. Дейт К. Дж. Введение в системы баз данных, 8-е изд. М.: Издательский дом «Вильямс», 2005.
3. Илюшечкин В.М. Основы использования и проектирования баз данных. М.: Издательство Юрайт, 2011.

4. Одиночкина С.В. Разработка баз данных в Microsoft Access 2010. СПб.: НИУ ИТМО, 2012.
5. Дейт К. Дж. SQL и реляционная теория. Как грамотно писать код на SQL. СПб.: Символ-Плюс, 2010.
6. Мишра С., Бьюли А. Секреты Oracle SQL. СПб.: Символ-Плюс, 2006.
7. Сурядный А.С. Microsoft Access 2010. Лучший самоучитель. М.: Астрель, 2012.
8. Кренке Д. Теория и практика построения баз данных, 8-е изд. СПб.: Питер, 2003.

Перечень информационных технологий и программного обеспечения

При осуществлении образовательного процесса по дисциплине в разные моменты времени используется система управления базами данных Microsoft Access одной из последних версий или СУБД MySQL, пакет численной математики GNU Octave (или Matlab).

VI. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Рекомендации по планированию и организации времени, отведенного на изучение дисциплины, описание последовательности действий обучающихся

Освоение дисциплины следует начинать с изучения рабочей учебной программы, которая содержит основные требования к знаниям, умениям и навыкам. Обязательно следует учитывать рекомендации преподавателя, данные в ходе установочных занятий. Затем – приступать к изучению отдельных разделов и тем в порядке, предусмотренном программой.

Получив представление об основном содержании раздела, темы, необходимо изучить материал с помощью рекомендуемой основной литературы. Целесообразно составить краткий конспект или схему, отображающую смысл и связи основных понятий данного раздела и

включенных в него тем. Обязательно следует записывать возникшие вопросы, на которые не удалось ответить самостоятельно.

Подготовку к началу обучения включает несколько необходимых пунктов:

1) Необходимо создать для себя рациональный и эмоционально достаточный уровень мотивации к последовательному и планомерному изучению дисциплины.

2) Необходимо изучить список рекомендованной основной и дополнительной литературы и убедиться в её наличии у себя дома или в библиотеке в бумажном или электронном виде.

3) Необходимо иметь «под рукой» специальные и универсальные словари, справочники и энциклопедии, для того, чтобы постоянно уточнять значения используемых терминов и понятий. Пользование словарями и справочниками необходимо сделать привычкой. Опыт показывает, что неудовлетворительное усвоение предмета зачастую коренится в неточном, смутном или неправильном понимании и употреблении понятийного аппарата учебной дисциплины.

4) Желательно в самом начале периода обучения возможно тщательнее спланировать время, отводимое на работу с источниками и литературой по дисциплине, представить этот план в наглядной форме (график работы с датами) и в дальнейшем его придерживаться, не допуская срывов графика индивидуальной работы и «аврала» в предсессионный период. Пренебрежение этим пунктом приводит к переутомлению и резкому снижению качества усвоения учебного материала.

Рекомендации по работе с литературой

1) Всю учебную литературу желательно изучать «под конспект». Чтение литературы, не сопровождаемое конспектированием, даже пусть самым кратким – бесполезная работа. Цель написания конспекта по дисциплине – сформировать навыки по поиску, отбору, анализу и формулированию учебного

материала. Эти навыки обязательны для любого специалиста с высшим образованием независимо от выбранной специальности.

2) Написание конспекта должно быть творческим – нужно не переписывать текст из источников, но пытаться кратко излагать своими словами содержание ответа, при этом максимально структурируя конспект, используя символы и условные обозначения. Копирование и «заучивание» неосмысленного текста трудоемко и по большому счету не имеет большой познавательной и практической ценности.

3) При написании конспекта используется тетрадь, поля в которой обязательны. Страницы нумеруются, каждый новый вопрос начинается с нового листа, для каждого экзаменационного вопроса отводится 1-2 страницы конспекта. На полях размещается вся вспомогательная информация – ссылки, вопросы, условные обозначения и т.д.

4) В итоге данной работы «идеальным» является полный конспект по программе дисциплины, с выделенными определениями, узловыми пунктами, примерами, неясными моментами, проставленными на полях вопросами.

5) При работе над конспектом обязательно выявляются и отмечаются трудные для самостоятельного изучения вопросы, с которыми уместно обратиться к преподавателю при посещении установочных лекций и консультаций, либо в индивидуальном порядке.

6) При чтении учебной и научной литературы всегда следить за точным и полным пониманием значения терминов и содержания понятий, используемых в тексте. Всегда следует уточнять значения по словарям или энциклопедиям, при необходимости записывать.

7) При написании учебного конспекта обязательно указывать все прорабатываемые источники, автор, название, дата и место издания, с указанием использованных страниц.

**Подготовка к промежуточной аттестации по дисциплине: экзамену
(зачету)**

К аттестации допускаются студенты, которые систематически в течение всего семестра посещали и работали на занятиях и показали уверенные знания в ходе выполнении практических заданий и лабораторных работ.

Непосредственная подготовка к аттестации осуществляется по вопросам, представленным в рабочей учебной программе. Тщательно изучите формулировку каждого вопроса, вникните в его суть, составьте план ответа. Обычно план включает в себя:

- определение сущности рассматриваемого вопроса, основных положений, утверждений, определение необходимости их доказательства;
- запись обозначений, формул, необходимых для полного раскрытия вопроса;
- графический материал (таблицы, рисунки, графики), необходимые для раскрытия сущности вопроса;
- роль и значение рассматриваемого материала для практической деятельности, примеры использования в практической деятельности.

VII. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Для осуществления образовательного процесса по дисциплине желательна учебная аудитория для проведения занятий лекционного типа и практических занятий: компьютерный класс.



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«Дальневосточный федеральный университет»
(ДВФУ)

ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ (ШКОЛА)

**УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ
РАБОТЫ ОБУЧАЮЩИХСЯ**

**по дисциплине «Инструментальные средства вычислений и
моделирования»**

**Направление подготовки 01.04.04 Прикладная математика
магистерская программа «Аналитические, социальные и экономические сети»
Форма подготовки очная**

**Владивосток
2021**

План-график выполнения самостоятельной работы по дисциплине

№ п/п	Дата/сроки выполнения	Вид самостоятельной работы	Примерные нормы времени на выполнение	Форма контроля
1	4 неделя	Повторение практического материала дисциплины, изученного в ходе аудиторных занятий; изучение основной и дополнительной литературы, указанной в рабочей учебной программе дисциплины,	24 часа	Собеседование, вопросы 1-2, проект 1
2	6 неделя	Самостоятельный разбор заданий и задач, решаемых на практических занятиях; самостоятельный повтор действий, осуществляемых в ходе выполнения лабораторных работ, в том числе при работе со специальным программным обеспечением	12 часов	Собеседование, вопросы 1-2, проект 1
3	8 неделя	Повторение практического материала дисциплины, изученного в ходе аудиторных занятий; изучение основной и дополнительной литературы, указанной в рабочей учебной программе дисциплины,	24 часа	Собеседование, вопросы 3-4, проект 2
4	12 неделя	Самостоятельный разбор заданий и задач, решаемых на практических занятиях; самостоятельный повтор действий, осуществляемых в ходе выполнения лабораторных работ, в том числе при работе со специальным программным обеспечением	12 часов	Собеседование, вопросы 3-4, проект 2
5	14 неделя	Самостоятельный разбор заданий и задач, решаемых на практических занятиях; самостоятельный повтор действий,	24 часа	Собеседование, вопросы 5-6, проект 3

		осуществляемых в ходе выполнения лабораторных работ, в том числе при работе со специальным программным обеспечением		
6	16 неделя	Самостоятельный разбор заданий и задач, решаемых на практических занятиях; самостоятельный повтор действий, осуществляемых в ходе выполнения лабораторных работ, в том числе при работе со специальным программным обеспечением	12 часов	Собеседование, вопросы 5-6, проект 3
		Экзамен	36 часов	Экзамен

Характеристика заданий для самостоятельной работы обучающихся и методические рекомендации по их выполнению

Задания и методические рекомендации по темам раздела I

1. Продемонстрировать умение работать удаленно с операционной системой Linux – подключиться по ssh к серверу, создать и отредактировать текстовые файлы.

Справка командной строки

Программа Linux предоставляет несколько команд, с помощью которых вы можете получить справку по многим командам.

- ✓ **help**: показывает список команд оболочки – подмножество команд Linux, включая `cd`, `alias`, `bg`, `jobs` и другие. Введите `help`, а затем имя команды, чтобы получить ее описание.
- ✓ **man**: Выводит документацию обо всех программах, установленных в системе Linux. Чтобы увидеть описание какой-либо команды, введите `man + ИМЯ КОМАНДЫ`. Если вы не знаете имени команды, которую ищете, вы можете искать в описаниях команд, используя элемент управления `-k`. Например, если вы введете `man -k сору`, будет выведен список всех команд, описание которых включает слово `сору`.
- ✓ **info**: Программа, предоставляющая документацию в виде иерархического меню тем. Вы можете выбрать необходимую тему. К тому же справочная информация содержит гиперссылки. Программа `info` часто может предоставить более полную информацию, чем программа `man`. Когда вы вводите `info` без аргументов, будет показан верхний уровень меню. Если вы знаете название раздела справки, вы можете ввести `info topicname`, где `topicname` – название раздела.
- ✓ **help option**: Многие программы предлагают параметр, с помощью которого можно получить информацию об использовании программы. Необходимо ввести имя программы, за которым следуют комбинации `-h` или `--help`. Например, если ввести `info --help`, можно получить краткий файл справки.

2. Набрать в Latex страницу текста из какой-либо математической статьи (с формулами).

Источник: Котельников И.А., Чеботаев П.З. LATEX по-русски. 3-е изд., перераб. и доп. Новосибирск: Сибирский хронограф, 2004.

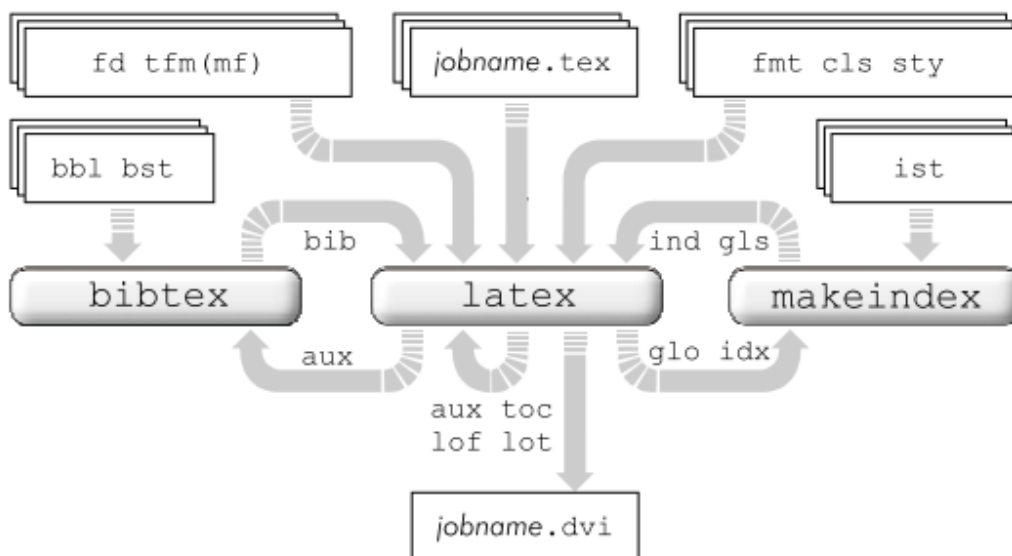


Рис. 1.2. Поток информации при работе компилятора latex

3. Изучение языка программирования Octave.

4.1 Запуск пакета GNU Octave

Все управление процессом вычислений в пакете GNU Octave осуществляется через командную строку. Также можно заранее подготовить на языке Octave скрипт-файл для последующего его исполнения.

Для того, чтобы запустить пакет GNU Octave необходимо в командной строке исполнить команду `octave`. Программа запустится в текущей консоли, при этом сперва будет выдано длинное сообщение о версии, авторах и прочая информация о пакете, а затем приглашение командной строки к работе с Octave

```
octave:1>
```

Если при запуске Octave дать команду `octave -q`, то начальное сообщение выводиться не будет.

Выйти из Octave можно либо нажав комбинацию клавиш `<Ctrl>+<D>`, либо набрав в командной строке Octave одну из команд `exit` или `quit`

```
octave:1> exit
```

Командой `exit` также можно закончить выполнение любой программы, написанной на языке Octave.

Работая из командной строки Octave каждая введенная команда исполняется сразу после нажатия клавиши `<Enter>`. Рассмотрим простой пример по созданию матрицы

$$A = \begin{pmatrix} 1 & 3 & 5 \\ 2 & 4 & 7 \end{pmatrix}.$$

Любую матрицу или вектор с заданными элементами в Octave можно создать путем перечисления этих элементов в квадратных скобках `[]`, разделяя столбцы пробелом или запятой, строки — знаком "точка с запятой"

```
octave:1> A=[1 3 5; 2 4 6]
```

После нажатия клавиши `<Enter>` на экране появится результат выполнения команды

```
A =  
  1   3   5  
  2   4   6
```

Теперь в памяти Octave есть матрица A и далее с ней можно производить любые доступные операции, например, транспонировать: операция "штрих" (`'`) после имени матрицы

```
octave:2> A'
```

Результат выполнения автоматически будет сохранен в переменную `ans` и выведен на экран

```
ans =  
  
    1    2  
    3    4  
    5    6
```

Везде далее после введения команды Octave будем печатать результат ее исполнения.

Можно создать новую матрицу $B = A^T$ при помощи операции присвоения (=)

```
octave:3> B=A'  
B =  
  
    1    2  
    3    4  
    5    6
```

Далее можно оперировать уже двумя матрицами A и B .

Как уже было отмечено, работа через командную строку Octave — это один из способов проводить процесс вычислений. Альтернативой является создание скрипт-файла и последующая его обработка пакетом GNU Octave.

Опишем процесс создания матрицы A и B в отдельном файле, например, `matrix.oc`², который должен содержать следующие строки:

```
A=[1 3 5; 2 4 6]  
B=A'
```

После сохранения файла `matrix.oc` из командной строки консоли можно запустить его на исполнение командой

```
octave matrix.oc
```

Результат работы будет выведен на экран.

В ходе реализации более сложных вычислений, как правило, существуют промежуточные операции, результат выполнения которых выводить нет необходимости. Чтобы отменить вывод следует ставить знак "точка с запятой" после соответствующей команды. Например, если поставить "точка с запятой" после команды создания матрицы A

```
A=[1 3 5; 2 4 6];  
B=A'
```

то по мере обработки скрипт-файла `matrix.oc` будет выведена только матрица B .

Код программы может сопровождаться комментариями. Octave игнорирует строки, помеченные знаками "решетка" (#) или "процент" (%).

```
A=[1 3 5; 2 4 6];
# Транспонирование матрицы A
B=A'
```

Далее возможности языка покажем на примерах, реализованных через командную строку Octave, однако все описанные ниже команды можно записать в отдельный файл *.os и обработать пакетом как самостоятельную программу.

4.2 Простые вычисления

Средствами языка Octave можно выполнить любые арифметические и алгебраические операции, доступные самому продвинутому калькулятору. Операндами могут быть как вещественные, так и комплексные числа.

Вещественные числа определяются в виде целого значения или конечной десятичной дроби, при этом дробная часть числа отделяется точкой, например 3.65. Допустима экспоненциальная форма записи вещественных чисел, например 1e-4, что эквивалентно значению 0.0001, или 1e4, что эквивалентно значению 10000. В вычислениях также можно использовать иррациональные константы $\pi = 3.146\dots$ команда pi и $e = 2.718\dots$ команда e.

Любое комплексное число вводится как формальная сумма $x + iy$, где x и y — вещественные числа, i — мнимая единица, то есть число, удовлетворяющее уравнению $i^2 = -1$.

```
octave:16> c = 1-3e-2i
c = 1.000000 - 0.030000i
```

Операции над комплексными числами осуществляются согласно общих правил и отдельно в настоящем руководстве рассматриваться не будут.

Применяя операции сложения (+), вычитания (-), умножения (*), деления (/), возведения в степень (^) можно вычислять самые разнообразные математические выражения

```
octave:1> (1+2^3-4)/5
ans = 1
```

В Octave легко вычислить квадратный корень, логарифмические и тригонометрические функции, имеются функции округления, взятия минимума и максимума и т.д. Список некоторых(!) функций приведен в таблице 5. Аргумент x для тригонометрических функций задается в радианах.

Напомним, что

$$\text{sign}(x) = \begin{cases} 1, & \text{если } x > 0; \\ 0, & \text{если } x = 0; \\ -1, & \text{если } x < 0. \end{cases}$$

Приведем некоторые примеры вычислений:

- 1) вычислить синус угла в 30 градусов

```
octave:2> sin(30*pi/180)
ans = 0.50000
```

- 2) найти наибольший общий делитель чисел 24, 86, 144

```
octave:4> gcd(24,84,144)
ans = 12
```

3) вычислить функцию $\ln 10 + 2^5 \cdot 13.7$

```
octave:5> log(10)+pow2(13.7,5)
ans = 440.70
```

4.3 Функции и переменные

Как и в любом языке программирования в Octave есть понятия переменной и функции, создавать которые намного проще, чем, к примеру, в таких языках как C или Pascal.

Переменные в Octave начинают существовать, как только им было присвоено значение. Тип переменной также определяется присвоенным значением. Одномерные и двумерные массивы переменных определяются как вектора и матрицы через квадратные скобки. Требования к именам переменных стандартные, как для многих языков: имя может состоять из букв латинского алфавита, цифр (везде, кроме первого символа имени) и символа нижнего подчеркивания.

```
octave:1> x1=2
x1 = 2
octave:2> x2=3.7
x2 = 3.7000
octave:3> v=[2 4.6 1]
v =

    2.0000    4.6000    1.0000
octave:4> A = [1 3 4; 2 4 6]
A =

    1    3    4
    2    4    6
```

В приведенном примере показано, как создаются целочисленная переменная $x1$, вещественная переменная $x2$, одномерный массив чисел v и двумерный массив чисел A .

Задания и методические рекомендации по темам раздела II

1. Провести проектирование базы данных «Аэропорт». Продумать схему данных (таблиц должно быть не менее 5), связи между ними. Создать данную базу данных в СУБД Microsoft Access или в MySQL. В приложении должно быть не менее 3 форм, 2 отчетов, 1 запроса и одна главная кнопочная форма для пользователя (для MS Access).
2. Приведено описание реляционной базы данных. Требуется написать формулу на формальном языке реляционной алгебры для извлечения определенных данных из базы.
3. Приведено описание реляционной базы данных. Требуется написать запрос на языке SQL для извлечения определенных данных из базы.
4. Провести нормализацию предложенной схемы данных. Выписать все функциональные зависимости.

- 1.2. Каковы преимущества использования системы баз данных? Каковы недостатки использования системы баз данных?
- 1.3. Как вы понимаете термин *реляционная система*? Укажите различия между реляционной и нереляционной системами.
- 1.4. Как вы понимаете термин *модель данных*? Объясните различие между моделью данных и ее реализацией. Почему так важно это различие?
- 1.5. Приведите результат выполнения следующих операторов SQL выборки информации из базы данных винного погреба, представленной в табл. 11.
- a) `SELECT WINE, PRODUCER
FROM CELLAR
WHERE BIN# = 72 ;`
- б) `SELECT WINE, PRODUCER
FROM CELLAR
WHERE YEAR > 2 000 ;`
- в) `SELECT BIN#, WINE, YEAR
FROM CELLAR WHERE READY <
2003 ;`
- г) `SELECT WINE, BIN#, YEAR FROM
CELLAR WHERE PRODUCER = 'Robt.
Mondavi' AND BOTTLES > 6 ;`
- 1.6. Дайте собственную словесную интерпретацию типичной строки в одном из ответов на упр. 1.5, представив ее в виде истинного высказывания.
- 1.7. Приведите результат выполнения следующих операторов SQL внесения изменений в базу данных винного погреба, представленную в табл. 11.
- a) `INSERT
INTO CELLAR (BIN#, WINE, PRODUCER, YEAR, BOTTLES, READY) f?
VALUES (80, 'Syrah', 'Meridian', 1998, 12, 2003) ;`
- б) `DELETE
FROM CELLAR
WHERE READY > 2004 ;`
- в) `UPDATE CELLAR
SET BOTTLES = 5 WHERE
BIN# = 50 ;"`
- г) `UPDATE CELLAR
SET BOTTLES = BOTTLES + 2
WHERE BIN# = 50 ;`
- 1.8. Напишите оператор SQL для выполнения приведенных ниже операций в базе данных винного погреба.
- a) Выбрать номер ячейки, наименование вина и количество бутылок для всех вин производства Geyser Peak.
- б) Выбрать номер ячейки и наименование вина для всех вин, запас которых составляет больше пяти бутылок.
- в) Выбрать номер ячейки для всех красных вин.
- г) Добавить три бутылки в ячейку с номером 3 0.

4.2. Даны таблицы PART и PART_STRUCTURE, определенные следующим образом.

```
CREATE TABLE PART
  ( P# P#, DESCRIPTION CHAR(100),
    PRIMARY KEY ( P# ) )
; CREATE TABLE
PART_STRUCTURE
  ( MAJOR P# P#, MINOR P# P#, QTY QTY,
    PRIMARY KEY ( MAJOR P#, MINOR P# ),
    FOREIGN KEY ( MAJOR P# ) REFERENCES
PART, FOREIGN KEY ( MINOR P# )
REFERENCES PART ) ;
```

В таблице PART_STRUCTURE показано, какие детали (MAJOR_P#) содержат другие детали (MINOR_P#) как компоненты первого уровня. Напишите на языке SQL программу для получения списка всех компонентов данной детали на все имеющихся уровнях (задача разузлования деталей). *Примечание.* Значения показанные в качестве примера на рис. 4.6, могут помочь вам более наглядно представить предложенную выше задачу. Следует отметить, что таблица PART_STRUCTURE демонстрирует, как информация о *составе изделий* (см. глав 1, раздел 1.3, подраздел "Сущности и связи") обычно представляется реляционных системах.

MAJOR_P#	MINOR_P#	QTY
P1	P2	2
P1	P3	4
P2	P3	1
P2	P4	3
P3	P5	9
P4	P5	8
P5	P6	3

Рис. 4.6. Таблица PART_STRUCTURE (значения приведены в качестве примера)

Упражнение 4.1. «Дубликаты нужны в базах данных, потому что естественным образом встречаются в реальном мире. Например, все копейки – дубликаты друг друга». Как бы вы возразили против такого аргумента?

Упражнение 4.2. Пусть r – некое отношение, и пусть bx и by – булевы выражения. Тогда существует правило (оно используется в реляционных системах для оптимизации), согласно которому $(r \text{ WHERE } bx) \text{ UNION } (r \text{ WHERE } by) \equiv r \text{ WHERE } bx \text{ OR } by$ (где символом \equiv обозначается эквиваленция). Применимо ли это правило, если r – не отношение, а таблица, содержащая дубликаты?

Требования к представлению и оформлению результатов самостоятельной работы

Самостоятельная работа включает в себя повторение теоретического и практического материала дисциплины, заслушиваемого и конспектируемого в ходе аудиторных занятий; изучение основной и дополнительной литературы, указанной в рабочей учебной программе дисциплины, самоконтроль ответов на основные проблемные вопросы по темам занятий; самостоятельный повтор действий, осуществляемых в ходе выполнения лабораторных работ, в том числе при работе со специальным программным обеспечением.

Результаты самостоятельной работы представляются и оформляются в виде ответов на основные положения теоретического и практического материала дисциплины по темам; работающего исходного кода результата решения практических заданий и задач; собственных действий, осуществляемых в ходе выполнения лабораторных работ.

Критерии оценки выполнения самостоятельной работы

Общие критерии оценки выполнения самостоятельной работы – правильность ответов на вопросы по темам теоретической части дисциплины, верность получаемых ответов в ходе решения практических заданий и задач, достижение правильного результата при осуществлении собственных действий по лабораторным работам.

Оценивание знаний в форме собеседования проводится по критериям:

- логичность изложения, знание и понимание основных теоретических и практических положений и концепций по теме;
- владение теоретическими и практическими приёмами написания, отладки и тестирования программ на языке программирования Octave, AWK
- умение написать нужный запрос на структурированном языке запросов SQL.