

# БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

## Механико-математический факультет

### Кафедра дифференциальных уравнений и системного анализа

СОГЛАСОВАНО

Председатель методической комиссии  
механико-математического факультета

Кротов В. Г.




17 мая 2012 г.

СОГЛАСОВАНО

Декан  
механико-математического факультета

Медведев Д. Г.



17 мая 2012 г.

Регистрационный № УД 8898

## УЧЕБНО-МЕТОДИЧЕСКИЙ КОМПЛЕКС ПО УЧЕБНОЙ ДИСЦИПЛИНЕ

### Компьютерная математика

Для специальностей:

1-31 03 01 «Математика (по направлениям)»

Составители: Атрохов К.Г., канд. физ.-мат. наук, доцент Голубева Л. Л., доктор физ.-мат. наук, профессор Липницкий В.А., канд. физ.-мат. наук, доцент Малевич А. Э., доктор физ.-мат. наук, профессор Садовский А. П., канд. физ.-мат. наук, доцент Щеглова Н. Л., канд. физ.-мат. наук Чергинцев Д.Н.

Рассмотрена и утверждена

на заседании Научно-методического совета БГУ 15 мая 2012 г.

Протокол № 7

# ОГЛАВЛЕНИЕ

<b>Пояснительная записка .....</b>	<b>4</b>
<b>Раздел I. Компьютерное моделирование в математических средах .....</b>	<b>7</b>
Конспект лекций.....	7
План лабораторных занятий.....	13
Список контрольных вопросов .....	37
Пример контрольного теста .....	42
Список рекомендуемой литературы.....	46
<b>Раздел II. Символьный математический пакет Mathematica .....</b>	<b>47</b>
Конспект лекций.....	47
План лабораторных занятий.....	49
Список контрольных вопросов .....	52
Список рекомендуемой литературы.....	56
<b>Раздел III. Числовой математический пакет MATLAB .....</b>	<b>57</b>
Конспект лекций.....	57
План лабораторных занятий.....	87
Список контрольных вопросов .....	106
Пример контрольного теста .....	114
Список рекомендуемой литературы.....	124
<b>Раздел IV. Основы компьютерного моделирования .....</b>	<b>125</b>
Конспект лекций.....	125
План лабораторных занятий.....	127
Список контрольных вопросов .....	131
Список рекомендуемой литературы.....	135
<b>Раздел V. Математические основы компьютерной алгебры .....</b>	<b>136</b>
Конспект лекций.....	136
План лабораторных занятий.....	137
Список контрольных вопросов .....	139
Примеры контрольных заданий.....	140
Список рекомендуемой литературы.....	141
<b>Раздел VI. Математические основы защиты информации .....</b>	<b>142</b>

Конспект лекций.....	142
План лабораторных занятий.....	145
Список контрольных вопросов .....	156
Примеры контрольных заданий.....	157
Список рекомендуемой литературы.....	158
<b>Раздел VII. Методы моделирования сложных систем и процессов .....</b>	<b>159</b>
Конспект лекций.....	159
План лабораторных занятий.....	175
Список контрольных вопросов .....	180
Список рекомендуемой литературы.....	182
<b>Типовая учебная программа.....</b>	<b>183</b>
<b>Воспитательно-идеологическая функция .....</b>	<b>200</b>

## ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

В настоящее время в обществе наблюдается рост интереса и внимания к проблемам информатизации общества. Это обусловлено рядом объективных и субъективных факторов.

Научно-технический прогресс, информатизация всех сфер общественной жизни, современные глобальные процессы и проблемы человечества предъявляют новые требования к уровню образованности личности, личностному и профессиональному развитию. В современном обществе, важнейшими характеристиками которого выступают непрерывность и динамизм развития, компьютерная грамотность приобретает особую значимость как общечеловеческая ценность, социальный механизм развития личности, общественного сознания, общества в целом.

В последние годы компьютерная математика становится эффективным значимым инструментом исследования не только в прикладных, но и в фундаментальных областях науки. Крупномасштабные исследования, связанные с расширением технических и углублением интеллектуальных возможностей, инновациями в образовательной практике позволили выделить наиболее эффективные модели, алгоритмы, целостные технологии познания, развития и реализации творческого потенциала личности.

Важным представляется освоение выпускниками высших учебных заведений современных основ информационных технологий, позволяющих сознательно и планомерно организовывать свою профессиональную деятельность, развивать собственный творческий потенциал, добиваться успехов в жизни и труде.

Типовая учебная программа «Компьютерная математика» предназначена для студентов специальности 1-31 03 01 Математика (по направлениям).

### Цели и задачи дисциплины

**Целью** преподавания дисциплины «Компьютерная математика» является подготовка специалистов, обладающих знаниями и умениями эффективного использования компьютерной техники, современных информационных систем, а также имеющих навыки работы в различных математических средах.

Преподавание дисциплины решает следующие **задачи**:

- приобретение студентами знаний в области теории и истории развития компьютерной техники,
- приобретение навыков работы в средах современных общеупотребительных операционных систем,

- овладение современными методами и приёмами поиска и использования информации посредством возможностей мировой компьютерной сети,
- выработка умения организации учебной и исследовательской работы с использованием современных числовых и символьных математических пакетов,
- приобретение способностей самостоятельно расширять компьютерные математические знания с дальнейшим их использованием при анализе математических моделей широкого круга прикладных задач.

### **Специалист должен**

#### ***Знать:***

- идеологию и основные принципы работы с пакетами компьютерной математики: Mathematica, Matlab, Mathcad, Maple;
- компьютерные пакеты подготовки электронных научных документов: LaTeX, Mathematica, MS Word;
- математические основы и алгоритмы компьютерной алгебры;
- математические основы кодирования и защиты информации;
- принципы построения математических и компьютерных моделей;
- методологии моделирования процессов, взаимосвязи данных, систем, объектов;
- основы системного анализа;

#### ***Уметь:***

- применять специализированные программные средства для построения моделей процессов, данных, объектов;
- владеть современными приёмами и методами поиска и использования научно-технической информации, редактировать, реферировать, математические и компьютерные разработки, статьи, готовить рукописи к печати;
- проводить научные исследования в области математики, математических методов системной интеграции, проектирования и создания информационных систем, моделирования и проектирования компьютерных комплексов для решения научных, народнохозяйственных и др. задач;
- анализировать, проектировать, разрабатывать и реализовывать программные компоненты этих комплексов.

Изучение дисциплины «Компьютерная математика» планируется за счет часов вузовского компонента цикла естественнонаучных или общепрофессиональных и специальных дисциплин в объеме до 534 часов, в т. ч. 272 часа аудиторных.

Рекомендуется проведение не менее двух контрольных работ в каждом семестре. В качестве итогового контроля предусматривается проведение зачетов и экзамена.

Учебно-методический комплекс (УМК) преследует цель оказать посильную помощь студентам в усвоении учебного и нормативного материала, сориентировать в подборе специальной литературы для подготовки к практическим занятиям, направить на развитие навыков самостоятельного решения практических задач.

# РАЗДЕЛ I. КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ В МАТЕМАТИЧЕСКИХ СРЕДАХ

## КОНСПЕКТ ЛЕКЦИЙ

### Лекция № 1. Компьютерное моделирование как метод научного познания

**Основные понятия технологий моделирования.** Исследование проблемы путем построения модели. Роль моделей в процессе познания мира. Цели и основные задачи моделирования. Объект моделирования как система. Определение системы, элемента системы. Роль связей, влияние свойств элементов системы на свойства системы в целом и обратно. Свойство эмерджентности системы.

Модель как система. Основные свойства модели. Субъект моделирования. Зависимость свойств модели от точки зрения исследователя. Классификация моделей по различным признакам. Математические модели, их классификация.

**Особенности компьютерного моделирования.** Этапы процесса познания посредством моделирования. Компьютерное моделирование как информационная технология познания окружающего мира. Особенности построения компьютерной модели. Описание свойств элементов системы, связей между ними. Данные для представления состояния системы. Процесс как смена состояний системы. Дискретность параметров, описывающих состояния системы, при компьютерном моделировании процессов.

**Этапы компьютерного моделирования.** Общая схема этапов компьютерного моделирования. Идентификация Объекта познания, представление его в качестве системы. Постановка Задачи, формулирование законов, управляющих исследованием. Отделение Информации от Объекта познания в контексте поставленной Задачи. Параллельное выполнение процессов целенаправленного изучения Объекта, отделения Информации и уточнения постановки Задачи.

Построение Формальной Модели Объекта познания. Выделение существенных элементов и связей между ними. Определение статистических и управляющих параметров Модели. Описание каждого из состояний моделируемой системы, а также процесса смены состояний.

Реализация Формальной Модели на компьютере. Выбор программных средств. Представление элементов системы в выбранной среде моделирования. Описание связей между элементами. Разработка и реализация алгоритмов с учетом выбора среды моделирования.

Разработка тестовых примеров, описывающих всевозможные регулярные и вырожденные случаи функционирования Модели. Тестирование и отладка Модели.

Вычислительный эксперимент. Исследование поведения Модели при различных значениях динамических параметров, прогнозирование, оптимизация чего-либо в контексте поставленной Задачи.

Представление результатов компьютерного эксперимента в виде Информационной Модели явления: графики зависимостей одних параметров от других, диаграммы, таблицы, демонстрация явления в реальном или виртуальном времени и т.п.

**Обзор современных математических сред компьютерного моделирования.** Системы компьютерной математики. Краткая история развития. Структура универсальных систем компьютерной математики. Пользовательский интерфейс под Windows. Средства моделирования: структуры данных, языки программирования, встроенные константы, операторы и функции. Сравнительные характеристики систем. Различие идеологий числовых и символьных компьютерных математических сред.

## **Лекция № 2. Методика ознакомления с компьютерными математическими средами на примере Mathcad**

**Интерфейс пользователя.** Окно — базовый объект интерфейса. Основные элементы окна Приложения. Основные элементы окна Документа. Основные элементы Диалогового окна.

Триада инструментальных средств Приложения: горизонтальное меню — панели инструментов — контекстное меню. Основные логические компоненты инструментальных средств: рабочая область документа, рабочие инструменты — средства для выполнения работы, элементы управления — средства настройки инструментов, режима работы Приложения.

Рабочая область (Client Area) Документа. Область (Region) — основной элемент Документа. Порядок размещения областей. Типы областей: текстовая, математическая, графическая область. Область смешанного типа. Режим отображения областей.

Панели инструментов. Standard (Стандартная) — выполнение операций с файлами, редакторской правки, вставки объектов; Formatting (Форматирование) — управление свойствами символа и абзаца; Math (Математика) — отображение математических палитр; Resources (Ресурсы) — вызов ресурсов справочной системы; Controls (Элементы управления) — вставка



в Документ стандартных элементов управления; Debug (Отладка) — управление процессом отладки.

Операции над панелями и палитрами: расположение в любом месте экрана (стандартно под строкой меню приложения, вдоль внешней границы рабочей области приложения, открепление от границ рабочей области приложения), изменение формы панели, определение набора инструментов на некоторых панелях, скрытие панели.

Меню. Основные пункты меню Приложения. Общие свойства. Типы команд в пунктах меню. Контекстное меню объекта.

Основные технологические механизмы. Указать и щелкнуть (Point-and-Click). Выполнить (Double Click). Выделить (Select). Перетащить и бросить (Drag-and-Drop). Буфер обмена (Clipboard). Технологии помещения данных в Буфер обмена и извлечение их из Буфера.

Клавиши-ускорители: клавиши быстрого вызова команд (Shortcut Keys) являются служебными. «Горячие клавиши» (Hot Keys) для вызова команд пунктов меню. Клавиши быстрого ввода символов для ввода специальных символов и операторов.

Типовой порядок работы с Документом. Создание нового Документа на основе предложенного системой шаблона, на основе шаблона, разработанного пользователем.

Установки документа Mathcad: режим вычислений, система исчисления, формат результатов вычислений, значения встроенных переменных, основные размерности переменных, стили текста, формул, графиков; колонтитулы, параметры страницы. Изменение пользователем установок документа, заданных системой по умолчанию.

Сохранение документа, основные форматы: на основе текстовой XML-разметки \*.xmcd, с дополнительной компрессией файла \*.xmcdz, полный текстовый формат \*.rtf, Web-страницы \*.htm, шаблона на основе текстовой XML-разметки \*.xmct, шаблона \*.mct, прежней версии Mathcad \*.mcd.

**Центр информационных ресурсов.** Справочная информация, доступная через пункт Help меню Приложения. Система технической поддержки. Руководство для разработчика собственных приложений. Библиотека электронных книг в форме обучающих курсов. Быстрые шпаргалки, используемые в качестве шаблонов. Встроенные электронные книги, посвященные расширениям Mathcad. Поиск встроенных математических функций. Контекстно-зависимая интерактивная справка.

**Правила оформления рабочего документа.** Виды документов: электронный документ Mathcad, печатный документ, Web-документ, электронная книга, фрагмент документа, экспортированный и оформленный в других приложениях.

Элементы документа: текстовые области, математические области, графические области, компоненты других приложений, внедренные объекты. Дополнительные элементы оформления документа: закрытые и выделенные области, колонтитулы, разрывы страниц, поля, ссылки, гиперссылки, рисунки.

Работа с областями: вставка, перемещение по документу, изменение размера, выделение области цветом или обрамлением, разделение, удаление. Понятие зоны. Создание, скрытие, раскрытие, запираение и разблокировка зоны.

Установка параметров страницы. Вставка колонтитулов. Вставка гиперссылок. Установка ссылок.

**Ввод, редактирование и форматирование данных.** Элементы интерфейса для ввода данных: указатель мыши — стандартного вида и назначения; визир — красный крестик для указания места области; курсор ввода текста — красная вертикальная линия, указывает позицию ввода символа в текстовую область; курсор ввода объекта в выражение — синие вертикальная и горизонтальная линии, для ввода и редактирования математических выражений; заполнитель места символа — черный прямоугольник, для ввода операнда; заполнитель места оператора — черная прямоугольная рамка, для ввода оператора.

Редактирование математических областей. Панели инструментов для ввода математических символов и операторов, соответствующие кнопкам сочетания клавиш. Двумерный вид курсора для ввода математического выражения. Роль нижней черты — следа курсора, управление его длиной.

Форматирование. Управление свойствами абзаца. Свойства символа. Форматирование символа в текстовых и математических областях.

### Лекция № 3. Средства вычислений в Mathcad

**Особенности вычислений.** Процессор числовых расчетов, ядро символьных вычислений. Режим немедленного вычисления областей, принцип WYSIWYG (What You See Is What You Get). Точки привязки областей. Порядок вычисления областей.

Оператор локального определения переменных и функций. Оператор глобального определения переменных и функций. Отличия, практические выводы. Операторы вывода числового и символьного результатов вычислений.

**Числа, математические константы, системные переменные.** Ввод целых, десятичных, комплексных чисел. Представление в десятичной, двоичной, восьмеричной, шестнадцатеричной системах исчисления. Свободное сочетание

форм представления чисел в различных операциях. Встроенные функции для выполнения операций с комплексными числами.

Математические константы, их интерпретация, возможность переопределения. Встроенные системные переменные, значение каждой, заданное по умолчанию. Способ вывода значений встроенных переменных на экран, возможности изменения значения встроенной переменной.

**Встроенные функции. Переменные. Операторы. Функции пользователя.** Подробнее об этих и приведенных выше темах лекции см. [1<sup>1</sup>, стр. 77–89].

## Лекция № 4. Модели геометрических объектов

**Массивы.** Два логически различных типа массивов. Ранжированная переменная: определение, особенности использования. Векторы, матрицы, тензоры. Способы построения. Доступ к строке, столбцу, строке, элементу, различным множествам элементов. Операции над векторами и матрицами. Встроенные функции для работы с массивами. Подробнее см. [1, стр. 90–93].

**Компьютерные модели геометрических объектов.** Методологическая цепочка: Объект → информация об Объекте + структура данных в среде моделирования → компьютерная модель Объекта.

Геометрические объекты. Модели прямой на плоскости и в пространстве, пучка прямых. Различные модели окружности. Модель полигона.

**Функции пользователя для описания свойств объектов.** Булевы функции и операторы. Особенности их использования при компьютерном моделировании.

Свойства геометрических объектов. Описание свойств объектов с использованием функций пользователя. Коллинеарность точек. Свойство сепарабельности прямой. Взаимное расположение точки и прямой, точки и отрезка, двух прямых на плоскости. Тесты свойств плоского полигона. Взаимное расположение двух плоскостей, прямой и плоскости.

**Модели абстрактных математических объектов.** Множества и логические операции над ними. Модель множества для построения диаграмм Эйлера-Венна, представление множества с помощью матрицы логических значений. Функции пользователя для определения операций над множествами.

## Лекция № 5. Графические образы моделей

**Построение графических объектов. Форматирование графических объектов.** См. [1, стр. 100].

---

<sup>1</sup> Голубева Л. Л. Компьютерная математика. Автоматизированное рабочее место математика: курс лекций / Л. Л. Голубева, А. Э. Малевич, Н. Л. Щеглова. Минск: БГУ, 2008. 139 с.

**Отображение статических геометрических объектов.** Графические образы точки, прямой, окружности, полигона на плоскости и в пространстве. Управляемое построение графиков функций на плоскости, декартова и полярная системы координат. Построение линий уровня. Построение поверхностей и кривых в пространстве.

**Отображение абстрактных математических объектов.** Построение диаграмм Эйлера-Венна. Функция, которая готовит данные для отображения множества. Геометрическая интерпретация комплексного числа  $z$ , корня степени  $n$  из комплексного числа  $z$ .

**Технологии анимации.** См. [1, стр. 101].

**Динамические объекты сцены.** Барицентрические координаты на плоскости. Движение точки вдоль кривой. Касательная как предельное положение секущей.

## Лекция № 6. Методы и приемы исследования функций

**Элементы программирования.** См. [1, стр. 98–100]. Построение функций. Два разных способа описания функции: рекурсия и стек вызванных подпрограмм.

**Выявление свойств заданной функции с использованием графики.** Методы и приемы изучения новых законов с использованием возможностей графики. Сравнение поведения Гамма-функции с поведением известных элементарных функций. Поиск закономерностей. Выводы свойств Гамма-функции.

**Восстановление функции по заданным свойствам.** Решение задачи определения значения Гамма-функции в произвольной точке на основании ее заданных свойств. Задачи продолжения функции, заданной на интервале, периодически на всю числовую ось, четным образом, нечетным образом.

## Лекция № 7. Фрактальные модели геометрических объектов

**Основные понятия фрактальной геометрии.** Геометрические фракталы. Линейные алгоритмы построения геометрических фракталов.

**Ввод/вывод данных.** Ввод/вывод данных для получения/хранения информации объекте. Разработка и реализация алгоритма построения геометрических фракталов. Генерация фрактальных изображений.

## ПЛАН ЛАБОРАТОРНЫХ ЗАНЯТИЙ

### Лабораторная работа № 1. Полярная система координат на плоскости

#### Постановка задачи:

Создайте мультфильм, демонстрирующий построение графика заданной функции в полярной системе координат по точкам. Исследуйте зависимость вида полученных кривых от различных значений вещественных параметров  $a$ ,  $m$ ,  $n$ ,  $l$ .

#### Порядок выполнения работы:

Этапы построения графика:

1. Подготовка данных: построение матрицы значений координат точек. Координаты точек, принадлежащих графику функции, будем размещать в матрице. В первую строку матрицы (по умолчанию индекс 0) введем значения аргумента  $\phi$ , во вторую строку (индекс 1) — значения функции  $r(\phi)$ . Таким образом, столбец матрицы будет содержать координаты точки  $M_i(\phi_i, r(\phi_i))$ . Указанное количество точек равномерно распределим в диапазоне изменения аргумента  $0 \leq \phi \leq 2\pi$

2. Построение графической области и ввод данных

Далее создаем графическую область, позволяющую построить графический объект. Используем команду View → Toolbars → Graph, возвращающую панель инструментов построения графиков. Изучите все кнопки этой панели.

Используя кнопку Polar Plot, введите графическую область.

Черные прямоугольники, называемые placeholders (держатель места), заполняются следующим образом:

- внизу и слева указываются соответственно идентификаторы-имена объектов, содержащих значения аргумента и функции соответственно;
- в правой части области можно указать пределы изменения значений функции.

3. Создание «живой» картинки: в кадре появляется очередная точка, соединенная линией с предыдущей точкой

Определим количество кадров посредством переменной Frames. Придем к соглашению, что количество точек совпадает с числом кадров, и в каждом кадре появляется еще одна новая точка графика FRAME — системная переменная, ее значение изменяется после запуска механизма анимации.

4. Построение полярной оси (выполняется на оценку 8-10)

## Лабораторная работа № 2. Касательная как предельное положение секущей

### Постановка задачи:

Создайте мультфильм, изображающий касательную к графику функции как предельное положение секущей.

### Алгоритм решения:

1. В графической области построим следующие объекты:
  - График дифференцируемой на интервале  $(\alpha, \beta)$  функции  $f(x)$
  - «Неподвижную» точку А с координатами  $(x_0, f(x_0))$ ,  $x_0 \in (\alpha, \beta)$
  - «Подвижную» точку В,  $(x_1, f(x_1))$ ,  $x_1 \in (\alpha, \beta)$ , которая с увеличением счетчика кадров скользит по графику функции, приближаясь бесконечно близко к точке А
  - «Подвижную» секущую  $y_2(x)$ , проходящую через точки А и В
  - Касательную  $y_1(x)$  к графику функции  $f(x)$  в точке А.
2. Форматируем полученные графические объекты, используя возможности цвета и начертания, толщины и т. п.
3. Используя механизм анимации, создадим мультфильм.

### Порядок выполнения работы:

1. Запустите Mathcad, создайте и сохраните новый Документ “КМ01ФамилияИмя.mcd”.
2. Создайте в Документе текстовые области, запишите название работы, атрибуты исполнителя, дату выполнения. Форматируйте текст.
3. Определите функцию  $f(x)$ , дифференцируемую на интервале  $(\alpha, \beta)$ . Используйте оператор локального определения  $:=$ , который вводится нажатием клавиши  $⋮$ .
4. Определите абсциссы точек  $x_0, x_1 \in (\alpha, \beta)$  и переменную Frames, указывающую количество кадров при анимации.
5. С помощью оператора вывода  $=$  и оператора символьного вывода  $\rightarrow$  проверьте правильность определений п.п. 3, 4.
6. Определите количество кадров мультфильма. Для этого введите переменную Frames. При включении механизма анимации пакета встроенная переменная FRAME изменяет свое значение от 0 до Frames.
7. Введите безразмерную нормированную величину  $\mu := \frac{\text{FRAME}}{\text{Frames}}$ , позволяющую управлять изображением в кадре. Используя ее, определите абсциссу «подвижной» точки.

8. Определите неподвижную точку А и «подвижную» точку В, записав их координаты в столбцы матрицы (или двух матриц).
9. Убедитесь в правильности определения при помощи оператора вывода. Научитесь извлекать любой элемент матриц, используя понятие индекса.
10. Определите касательную к графику функции в точке А. Для этого постройте функцию  $g(x)$ , являющуюся производной функции  $f(x)$ . Используйте панель Calculus для ввода оператора дифференцирования. Напомним, что уравнение касательной к графику функции  $f(x)$  определяется формулой  $y_1(x) = f(x_0) + f'(x_0)(x - x_0)$ .
11. Проверьте результаты при помощи оператора символьного вывода.
12. Используя тангенс угла наклона  $\frac{f(x_1) - f(x_0)}{x_1 - x_0}$ , запишите уравнение секущей, проходящей через точки А и В  $y_2(x) = f(x_0) + \frac{f(x_B) - f(x_0)}{x_B - x_0}(x - x_0)$ .
13. Постройте графическую область с заданными объектами.
14. Форматируйте каждый графический объект.
15. Изменяя значения встроенной константы FRAME, отладьте стоп-кадр. Значения переменной указывайте при помощи глобального оператора присваивания  $\equiv$ .
16. Выполните анимацию.

### Лабораторная работа № 3. Создание 2D-анимационных объектов

#### Постановка задачи:

Создайте мультфильм, изображающий траекторию движения тушки жука, сидящего на ободе катящегося по дороге колеса.

#### Порядок выполнения работы:

- 1) Ознакомьтесь с предложенным вам вариантом реализации проекта в Mathcad и внимательно проработайте предложенное ниже описание проекта.
- 2) Дайте математическое обоснование всех используемых в Документе формул, выполнив его в MS Word.
- 3) Добавьте в кадр лично вами созданный объект. Опишите в Документе Mathcad методику его создания. Создайте мультфильм.

Определим объекты, необходимые для реализации «живой картинки»:

- Дорога Road.
- Колесо Wheel.

- Жук Bug.
- Траектория тушки жука Track.
- Спицы Spokes.
- Объект Object, предложенный преподавателем, и лично вами созданный.

Какую последовательность действий при построении объектов мы предлагаем? Сначала для каждого объекта определим данные, позволяющие его изображать графически. Далее создадим графическую область. В предложенные системой поля (placeholder) графической области введем необходимые данные. Полученный объект отформатируем, для чего используем окно форматирования графиков.

Данные разместим в таблице, называемой матрицей. Координаты каждой точки помещаем в столбец матрицы. Так как изображение двумерно, количество строк матрицы равно двум, количество столбцов – числу точек, изображающих объект. Индексация элементов матрицы по умолчанию начинается с нуля. Нулевая строка матрицы будет содержать все абсциссы, а первая – все ординаты точек, которые представляют информацию об объекте.

Будем именовать матрицы по имени соответствующего объекта. Таким образом, в матрицах Road, Wheel, Bug, Track, Spokes, Object будем размещать координаты точек соответствующего объекта в стоп-кадре.

Следует обратить особое внимание на именование используемых переменных — каждое имя должно нести смысловую нагрузку. Рекомендуется использование составных имен для именования составных частей объекта. Структура составного имени такова: ИмяОсновное.ИмяИндекс. Просьба не путать ИмяИндекс, вводимое посредством точки, с понятием индекса переменной, вводимого с помощью [.

Картинка становится «живой», если изображение на ней меняется с течением времени. Такую возможность предоставляет механизм анимации. Mathcad имеет встроенную переменную FRAME — счетчик кадров при анимации. Чтобы изображение изменялось с течением времени, координаты точек, изображающих объект, должны зависеть от номера кадра. Таким образом, изображение объекта в кадре должно зависеть от встроенной переменной FRAME.

**Основные параметры.** Для создания движущейся картинки придем к некоторым соглашениям. Эти соглашения определяют нашу дальнейшую стратегию разработки, поэтому переменные, реализующие соглашения, объявим глобальными константами. Константа Periods отвечает за количество полных оборотов. Укажем диаметр колеса, он же - высота максимального поднятия жука над землей Height, перемещение Length. Рассчитаем наш будущий фильм на 400 кадров. Для этого определим константу Frames. (Кроме того, позже, при просчете кадров в окне “Animate”, вызываемом из пункта меню View, мы должны будем



указать пределы изменения счетчика кадров FRAME от 0 до 400). Напомним особенности глобальных констант: Mathcad присваивает им значения в первую очередь, при первом просчете документа. Отсюда следует, что математические области, в которых определяются глобальные константы, можно располагать где угодно, в любой части документа, т.к. значения констант перед основным вычислением документа уже будут определены. Значения глобальным константам присваиваются посредством знака тождественного равенства, который вводится с клавиатуры комбинацией клавиш <Ctrl+~>.

Для создания «живой» картинке в Mathcad используется встроенная переменная FRAME, предопределенная 0 и являющаяся счетчиком кадров при анимации. Далее необходимо задать безразмерную нормированную величину абстрактного времени  $t := \frac{\text{FRAME}}{\text{Frames}}$ . Величина  $t$ , изменяющаяся в пределах от 0 до 1, позволит нам вычислять значение в текущем кадре любой изменяющейся во времени нужной нам переменной. (Это важно, спросите подробнее у преподавателя).

**Создание Дороги (Road).** Дорогу будем изображать в виде отрезка, располагающегося на оси OX, длины чуть большей, чем перемещение жука. Для этого достаточно двух точек. Введем индекс точек дороги  $ir := 0..1$ . Знак  $..$  в Mathcad означает диапазон значений от начального до конечного по умолчанию с шагом 1 и вводится посредством клавиши <;>. Координаты точек создаваемого объекта будем располагать по столбцам в соответствующей матрице. Таким образом, первая строка матрицы содержит координаты X точек объекта, вторая — координаты Y. Матрицу в Mathcad можно определить следующим образом. Ввести имя матрицы, например,  $\text{Road} :=$ , далее, используя панель математических инструментов, открыть окно Insert Matrix, в котором указать нужное число строк и столбцов. Нам необходимо поместить в матрицу Road координаты двух точек  $(-0.5 \cdot \text{Height}, 0)$ ,  $(\text{Length} + 0.5 \cdot \text{Height}, 0)$ , располагая их по столбцам. Выполните это самостоятельно.

Чтобы создать графическую область, можно использовать клавишу <@>, или панель инструментов, или панель горизонтального меню. В поля ввода (placeholder) под осями введем необходимую информацию: пределы изменения переменной и имя массива

- вдоль оси X     $-0.5 \cdot \text{Height}$        $\text{Road}[0,ir$      $\text{Length} + 0.5 \cdot \text{Height}$
- вдоль оси Y      0                       $\text{Road}[1,ir$       0

Клавиша <[> означает переход к нижнему индексу. Следует обратить внимание, что индексация строк и столбцов матриц по умолчанию начинается с нуля. Начальный индекс можно изменить, присвоив нужное значение встроенной переменной ORIGIN.

Построенный графический объект форматируется, если по нему дважды щелкнуть левой кнопкой мыши. В появившемся окне форматирования измените свойства объекта Road: цвет и толщину линии.

**Изображение Колеса (Wheel) в текущем кадре.** Колесо изобразим точками, рассредоточенными по окружности. Координату радиус-вектора точки М окружности в текущем кадре вычислим как сумму радиус-вектора центра С колеса и вектора CM, соединяющего точки центра колеса С и текущую точку края колеса М. Сделайте рисунок и дайте обоснование математических формул.

Последующие переменные будем называть составными именами.

Определим число точек-краев колеса  $\text{Wheel.Edges}:=30$ , индекс-счетчик точек на окружности  $i_w:=1..\text{Wheel.Edges}$ , радиус колеса  $\text{Wheel.Radius}:=0.5 \cdot \text{Height}$ , а также координаты радиус-вектора центра С в текущем кадре в виде матрицы

$$\text{Wheel.Center} := \begin{pmatrix} \text{Length} \cdot \frac{\text{FRAME}}{\text{Frames}} \\ \text{Wheel.Radius} \end{pmatrix}.$$

В ней указаны координаты центра в текущем кадре. Координата Y центра колеса постоянна и не зависит от номера кадра, она равна значению  $\text{Wheel.Radius}$ . Координата X центра колеса в текущем кадре вычисляется с использованием безразмерной нормированной величины абстрактного времени  $\text{Length} \cdot t$ . (Объясните подробнее). Для определения координат вектора CM, соединяющего центр С колеса и точку М окружности, используем угол поворота  $\text{Wheel}.\alpha[i_w]$  текущей точки М окружности относительно положительного направления оси OX

$$\text{Wheel}.\alpha[i_w] := \frac{2 \cdot \pi \cdot i_w}{\text{Wheel.Edges}}.$$

Обратите внимание, чтобы получить угол поворота текущей точки, мы делим на максимальное количество точек и умножаем на текущий номер точки — таким образом, снова используем безразмерную нормированную величину  $\frac{i_w}{\text{Wheel.Edges}}$ , пределы изменения которой, естественно, от 0 до 1.

Рассчитываем координаты точек краев колеса в текущем кадре. Каждый столбец матрицы **Wheel** представляет собой координаты текущей точки на окружности.

$$\text{Wheel}^{<i_w>} := \text{Wheel.Center} + \text{Wheel.Radius} \cdot \begin{pmatrix} \cos(\text{Wheel}.\alpha[i_w]) \\ \sin(\text{Wheel}.\alpha[i_w]) \end{pmatrix}.$$

Верхний индекс  $<i_w>$  означает номер столбца, вводится с клавиатуры  $<\text{Ctrl}+6>$ .

После создания графической области по осям OX и OY указываются соответствующие строки массивов Road, Wheel. Отформатируйте изображения созданных объектов.

**Положение Жука (Bug) в текущем кадре.** При вращении колеса Жук, лежащий на дороге, начинает путешествие на ободе, изменяя в текущем кадре свое положение на колесе по часовой стрелке. Выбираем в качестве начала отсчета угла поворота отрицательное направление оси OY и вводим угол поворота жука в текущем кадре относительно центра колеса  $Bug.\alpha := 2 \cdot \pi \cdot \text{Period} \cdot \frac{\text{FRAME}}{\text{Frames}}$ . Тогда координаты Жука в текущем кадре рассчитываются по формуле

$$Bug := Wheel.Center - Wheel.Radius \cdot \begin{pmatrix} \sin(Bug.\alpha) \\ \cos(Bug.\alpha) \end{pmatrix}.$$

Сделайте рисунок и дайте математическое обоснование введенных переменных.

Далее следует дополнить графическую область новым объектом и отформатировать его.

Следующие объекты опишите самостоятельно, внимательно изучив Документ с предложенной реализацией.

**Изображение траектории тушки Жука(Track).** Число точек, изображающих траекторию в текущем кадре, сделаем зависимым от номера кадра, т. е. от встроенной переменной FRAME.

**Спицы колеса (Spokes).** При создании Спиц следует помнить, что Mathcad соединяет точки, находящиеся в подготовленных массивах, последовательно друг за другом, как бы не отрывая «пера» от бумаги. Поэтому, формируя массив Spokes, рекомендуется столбцы с четными номерами заполнять координатами точек, лежащих на ободе колеса, а в столбцы с нечетными номерами внести координаты центра колеса в текущем кадре. Кроме того, для создания эффекта вращения колеса угол, в зависимости от которого располагаются концы спиц на ободе, необходимо связать с номером текущего кадра, т. е. со значением переменной FRAME.

Для отладки программы рекомендуется на черновике документа встроенной переменной FRAME глобально присваивать поочередно текущие значения от 0 до 400.

## Лабораторная работа № 4. Моделирование операций над множествами

### Постановка задачи:

Моделируйте операции над подмножествами некоторого универсального множества, иллюстрируя их диаграммами Эйлера. Убедитесь в справедливости свойств операций над множествами.

### Алгоритм выполнения работы:

Выполнение поставленной задачи разобьем на основные этапы:

- выбор модели универсального множества и его подмножеств;
- построение модели универсального множества и его подмножеств;
- построение графических образов универсального множества и его подмножеств;
- определение операций над множествами;
- создание новых множеств путем определения операций над множествами;
- проверка свойств операций над множествами.

Пусть каждое из множеств, над которыми будем в дальнейшем выполнять операции, является подмножеством некоторого универсального множества *Universum*. При этом графический образ универсума представляет собой множество равномерно распределенных точек плоскости, образующих квадрат. Произвольное непустое подмножество *Subset* множества *Universum* будем изображать точками, находящимися внутри некоторого круга, который лежит внутри квадрата. Эти соглашения соответствуют общепринятому понятию диаграмм Эйлера.

В качестве модели-представления универсального множества *Universum* используем квадратную матрицу *U*, элементы которой — логические значения *true* (числовые значения 1).

Подмножество *SubsetA* универсального множества *Universum* моделируем посредством квадратной матрицы *A* той же размерности, что и матрица *U*. Матрицу *A* построим следующим образом: если элемент принадлежит подмножеству *SubsetA*, то соответствующий элемент квадратной матрицы *A* имеет значение 1 (*true*), в противном случае элемент матрицы *A* имеет значение 0 (*false*).

### Порядок выполнения работы:

1. Создайте новый документ, сохранив его на диске компьютера, на котором Вы работаете, в папке Вашего курса, Вашей группы. Имя документа должно содержать идентификатор работы и Вашу фамилию.

2. Введите основные параметры: размерность квадратной матрицы, ассоциируемой с универсальным множеством  $U$ , масштабирующие множители  $a$ ,  $b$ ,  $c$  (производные величины от размерности матрицы), позволяющие строить подмножества  $A$ ,  $B$ ,  $C$  данного универсума.
3. Задайте универсальное множество  $U$  в виде логической матрицы с единичными элементами. Убедитесь в правильности определения, используя оператор вывода.
4. Задайте непустое множество  $A$  в виде логической матрицы той же размерности. Элементы этой матрицы имеют значение 1, если соответствующий элемент универсального множества принадлежит множеству  $A$ , в противном случае элемент матрицы имеет значение 0. Множество  $A$  при этом будем ассоциировать с внутренностью некоторого круга определенного радиуса  $r$ .
5. Аналогичным образом задайте не совпадающие с  $A$  и не совпадающие друг с другом непустые множества  $B$  и  $C$ . Используйте оператор вывода, чтобы увидеть построенные множества  $A$ ,  $B$ ,  $C$ .
6. Создайте функцию `ShowSet`, строящую по заданной логической матрице некоторого множества матрицу координат точек, принадлежащих множеству этому множеству.
7. Используя функцию `ShowSet`, получите образы множеств  $U$ ,  $A$ ,  $B$ ,  $C$  для их графического отображения.
8. Постройте графическую область, введите информацию, необходимую для отображения образов множеств  $U$ ,  $A$ ,  $B$ ,  $C$ .
9. Форматируйте графические объекты. Подпишите легенду для каждого из них.
10. Напишите функции, определяющие операции над множествами. Используйте [Н, стр. 23]. Имена функций должны соответствовать названиям операций, которые функции определяют.
11. Убедитесь в правильности определенных вами операций над множествами при помощи диаграмм Эйлера.
12. Ознакомьтесь со свойствами операций над множествами, изложенными в [Н, стр. 25–26].
13. Убедитесь в справедливости изученных в п. 12 законов. Постройте диаграммы Эйлера для левой и правой частей каждого равенства и убедитесь, что результаты совпадают.

### Литература:

[Н] Новиков Ф. А. Дискретная математика для программистов. СПб: Питер, 2000.

## Лабораторная работа № 5. Исследование свойств Гамма-функции

### Постановка задачи:

Получите наиболее полное представление о свойствах Гамма-функции, сравнив ее поведение с поведением элементарных функций.

### Порядок выполнения работы

1. Создайте новый документ на диске компьютера, на котором Вы работаете, в папке Вашего курса, Вашей группы. Имя документа должно содержать идентификатор работы, ее номер и Вашу фамилию.
2. Постройте в декартовой системе координат графики встроенной в систему Гамма-функции и известных нам элементарных функций: степенной и экспоненты. Чтобы узнать, как система именует Гамма-функцию, зайдите в систему справки и, выбрав в левой части окна MathcadHelp одну из закладок «Указатель», «Поиск», наберите в поле ввода ключевое слово Gamma. Позаботьтесь о возможности изменять значения  $[a, b]$  границ отрезка, на котором производится построение графиков, а также степень роста степенной функции.
3. Форматируйте оси полученной графической области, используя закладку X-Y Axes окна форматирования графиков.
4. Изучите (визуально) свойства непрерывности и монотонности каждой из функций, сравните их поведение. Для этого варьируйте значения границ отрезка  $[a, b]$ , на котором построены графики.
5. Сопоставьте скорости роста функций. Для наглядности постройте графики исходных функций, введя по оси ОУ логарифмический масштаб. Сравните, как ведет себя  $\Gamma(x)$  при  $x < 0$ ,  $x > 0$ , сделайте выводы о поведении функций.
6. Попробуйте определить, как ведет себя  $\Gamma(x)$  при увеличении аргумента на единицу. Нам известно, что для показательной функции справедливо правило  $e^{x+1} = e \cdot e^x$ , то есть с увеличением аргумента на 1 значение функции увеличивается в одно и то же число раз. Как это можно было увидеть при помощи графического инструментария? Постройте график функции 
$$g(x) = \frac{e^{x+1}}{e^x}.$$
7. Как изменяется при изменении значение исследуемой нами функции  $\Gamma(x)$  при изменении аргумента на 1? Выясните это при помощи графика функции  $h(x) = \frac{\Gamma(x+1)}{\Gamma(x)}$ . Убедитесь в том, что полученный график совпадает с графиком линейной функции  $y = x$ . Это можно увидеть, построив график

- линейной функции, а также график функции  $x - \frac{\Gamma(x+1)}{\Gamma(x)}$ . Запишите найденную рекуррентную формулу для функции  $\Gamma(x)$ .
8. Запишите еще одно полезное соотношение  $e^{1-x} e^x = e$ . Постройте графики каждого из множителей левой части  $e^{1-x}$ ,  $e^x$  и график их произведения на отрезке  $[-0.1, 1.1]$ . Каким свойством обладают эти графики?
9. А как будут вести себя аналогичные пункту 8 множители нашей функции? Следуя предыдущему пункту, постройте график функции  $\Gamma(1-x)\Gamma(x)$ . Заметьте, что для данной функции характерна периодичность.
10. Исходя из внешнего вида последнего графика, попробуйте, какова зависимость  $f(x) = \frac{1}{\Gamma(1-x)\Gamma(x)}$ .
11. Последний график очень напоминает синусоиду, общее уравнение которой  $A \sin(\omega x + \phi)$ . Постройте график этой синусоиды с произвольными параметрами в той же графической области, что и  $f(x) = \frac{1}{\Gamma(1-x)\Gamma(x)}$ . Подберите значения параметров  $A, \omega, \phi$  так, чтобы колебания совпадали с исследуемой функцией.
12. Полученная функциональная зависимость значений  $\Gamma(1-x)$  и  $\Gamma(x)$  позволяет при заданном множестве значений  $\Gamma(x), x \in [1/2, 1]$  построить «зеркальное отображение» функции  $\Gamma(x)$  на отрезке  $[0, 1]$  относительно точки  $1/2$ . Какова эта полученная зависимость?
13. Как построить  $\Gamma(x), x \in (0, 1/2)$ , если нам известно множество значений  $\Gamma(x), x \in [1/2, 1]$  и функциональная зависимость значений  $\Gamma(1-x)$  и  $\Gamma(x)$ ?
14. Используя знания пакета Mathcad о Гамма-функции, получите таблицу значений  $\Gamma(x), x \in [1/2, 1]$ . Полученные точки поместите в матрицу, расположив их по строкам. Как из полученной матрицы получить новую матрицу, с расположением тех же точек по столбцам?
15. Сделайте выводы относительно свойств исследуемой Гамма-функции. Сравните ваши выводы со справочной информацией, полученной в окне MathcadHelp по ключевому слову Gamma function.
16. Подумайте и запишите (спроектируйте), как на основании заданных значений  $\Gamma(x), x \in [1/2, 1]$  и полученных вами свойств Гамма-функции моделировать поведение этой функции в любой заданной точке.

## Лабораторная работа № 6. Моделирование Гамма-функции

### Постановка задачи:

Пусть заданы таблица значений Гамма-функции в узлах на отрезке  $[1/2, 1]$  и свойства Гамма-функции:

$$1) \frac{1}{\pi} \sin(\pi x) = \frac{1}{\Gamma(1-x)\Gamma(x)}$$

$$2) \Gamma(x+1) = x \Gamma(x)$$

$$3) \Gamma(0) = \infty$$

Постройте функцию, возвращающую приближенное значение Гамма-функции в любой заданной точке. При моделировании используйте возможности программирования в Mathcad.

### Алгоритм выполнения работы:

Выполнение поставленной задачи разобьем на основные этапы:

- 1) Построение таблицы  $N$  значений Гамма-функции на отрезке  $[1/2, 1]$ .
- 2) Создание функции, вычисляющей значение Гамма-функции в произвольной заданной точке.
- 3) Сравнение полученной закономерности и встроенной в Mathcad Гамма-функции путем построения графиков.

Для вычисления значения Гамма-функции в произвольной точке построим три функции:  $\text{Gamma}(x)$ ,  $\text{Gamma01}(x)$ ,  $\text{GammaBase}(x)$ .

Функция  $\text{Gamma}(x)$  при  $x < 0$  ( $x > 1$ ) приближается к отрезку  $[0; 1]$  на единицу слева (справа), рекуррентно вызывая саму себя до тех пор, пока значение, при котором функция вызывается рекуррентно, не достигнет одного из значений отрезка  $[0; 1]$ . Для приближения аргумента  $x$  на единицу используется свойство  $\Gamma(x+1) = x \Gamma(x)$ . При  $x = 0$  функция  $\text{Gamma}(x)$  определяет значение  $\text{Gamma}(0) = \infty$ , в случае же  $x \in (0; 1]$  она вызывает функцию  $\text{Gamma01}(x)$ .

Функция  $\text{Gamma01}(x)$  вызывает функцию  $\text{GammaBase}(x)$ , если  $x \in [1/2; 1]$ , в противном случае значение функции вычисляется при помощи свойства «зеркального отображения» значений функции на отрезке  $[0, 1]$  относительно точки  $1/2$ , а именно  $\frac{1}{\pi} \sin(\pi x) = \frac{1}{\Gamma(1-x)\Gamma(x)}$ .

Функция  $\text{GammaBase}(x)$  использует таблицу  $N$  значений Гамма-функции в точках  $(X_i, Y_i), i = \overline{0, N-1}$  на отрезке  $[1/2, 1]$ , вычисляя значение функции в точке  $(x,$



$\text{GammaBase}(x)$ ),  $x \in [1/2; 1]$  по формуле  $\text{GammaBase}(x) := \frac{(X_{\text{right}} - x)Y_{\text{left}} + (x - X_{\text{left}})Y_{\text{right}}}{X_{\text{right}} - X_{\text{left}}}$ ,

где  $x \in [X_{\text{left}}, X_{\text{right}}]$ ,  $Y_{\text{left}} = \Gamma(X_{\text{left}})$ ,  $Y_{\text{right}} = \Gamma(X_{\text{right}})$ .

### Порядок выполнения работы

1. Создайте новый документ на диске компьютера, на котором Вы работаете, в папке Вашего курса, Вашей группы. Имя документа должно содержать идентификатор работы и Вашу фамилию.
2. Постройте таблицу GammaData значений Гамма-функции на отрезке  $[1/2, 1]$  с шагом 0.1 в виде матрицы, располагая координаты точек по строкам.
3. Убедитесь визуально в правильности значений GammaData, построив полученные точки и график встроенной Гамма-функции.
4. Создайте функцию Gamma(x), при отладке возвращая координату x в точке вызова функции Gamma01(x). Проверьте правильность работы функции Gamma(x).
5. Создайте функцию GammaBase(x), для удобства предварительно определив два вектора: вектор X, содержащий абсциссы точек таблицы GammaData, и вектор Y, содержащий их ординаты.
6. Создайте функцию Gamma01(x).
7. Уберите отладочные выражения из тела функции Gamma(x), указав при этом вызов функции Gamma01(x). Расположите все три функции в Документе правильно, с учетом порядка вычисления областей в Mathcad.
8. Постройте на отрезке  $[-10, 10]$  графики функций Gamma(x) и встроенной Гамма-функции, определив предварительно порядка  $10^3$  узлов для построения графиков.

## Лабораторная работа № 7. Элементы компьютерной геометрии

### Постановка задачи:

Решите задачи I–III, вариант задания предлагает преподаватель.

I. Постройте требуемые в задачах 1–12 функции и тестируйте их работу на конкретных примерах. Изобразите двумерные аналоги объектов (точка, вектор), которые указаны в задачах, в декартовой системе координат.

1. Даны точки  $A = (x_A, y_A, z_A)$ ,  $B = (x_B, y_B, z_B)$ . Напишите функцию VecCoordinates, возвращающую координаты вектора  $\overrightarrow{AB}$ .
2. Дан вектор  $\overrightarrow{AB} = \vec{a} = (a_x, a_y, a_z)$ . Напишите функцию VecCoordTerminus, возвращающую координаты конца вектора  $\overrightarrow{AB}$ , если его начало совпадает с точкой  $A = (x_A, y_A, z_A)$ .

3. Дан вектор  $\overrightarrow{AB} = \vec{a} = (a_x, a_y, a_z)$ . Напишите функцию `VecCoordOrigin`, возвращающую координаты начала вектора  $\overrightarrow{AB}$ , если его конец совпадает с точкой  $B = (x_B, y_B, z_B)$ .
4. Даны точки  $A = (x_A, y_A, z_A)$ ,  $B = (x_B, y_B, z_B)$ . Напишите функцию `VecModulus`, возвращающую длину вектора  $\overrightarrow{AB}$ .
5. Напишите функцию `IsVecCollinear`, тестирующую, являются ли два заданных вектора  $\vec{a} = (a_x, a_y, a_z)$  и  $\vec{b} = (b_x, b_y, b_z)$  коллинеарными.
6. Даны точки  $A = (x_A, y_A, z_A)$ ,  $B = (x_B, y_B, z_B)$ . Напишите функцию `VecUnit`, возвращающую координаты его орта: единичного вектора, коллинеарного вектору  $\overrightarrow{AB}$  и одинаково с ним направленного.
7. Дан вектор  $\vec{a} = (a_x, a_y, a_z)$ . Напишите функцию `VecCodirectional`, возвращающую вектор  $\vec{b}$  заданной длины  $L$ , одинаково направленный с вектором  $\vec{a}$  и образующий тупой угол с осью  $OX$ .
8. Два неколлинеарных вектора  $\vec{a} = (a_x, a_y, a_z)$  и  $\vec{b} = (b_x, b_y, b_z)$  приложены к одной точке. Напишите функцию `VecBisectorUnit`, возвращающую координаты единичного вектора  $\vec{p}$ , направленного по биссектрисе угла между векторами  $\vec{a}$  и  $\vec{b}$ .
9. Даны длины векторов  $l = |\vec{a}|$ ,  $m = |\vec{b}|$ ,  $n = |\vec{a} + \vec{b}|$ . Напишите функцию `VecDifferenceModulus`, возвращающую длину вектора  $|\vec{a} - \vec{b}|$ .
10. Даны точки  $A = (x_A, y_A, z_A)$ ,  $B = (x_B, y_B, z_B)$  и величина  $\lambda \in (0; +\infty)$ . Напишите функцию `VecBipartition`, возвращающую координаты точки  $M$ , принадлежащей интервалу  $AB$  и делящей отрезок  $AB$  в отношении  $AM : MB = \lambda$ .
11. Даны точки  $A = (x_A, y_A, z_A)$ ,  $B = (x_B, y_B, z_B)$ . Напишите функцию `VecDirectionalCosines`, возвращающую направляющие косинусы вектора  $\overrightarrow{AB}$ .
12. Даны длины векторов  $l = |\vec{a}|$ ,  $m = |\vec{b}|$ ,  $n = |\vec{a} - \vec{b}|$ . Напишите функцию `VecSumModulus`, возвращающую длину вектора  $|\vec{a} + \vec{b}|$ .

II. При решении каждой из задач 1–12 создайте функцию пользователя. Имя функции пользователя должно отражать назначение функции. Обдумайте количество, типы, порядок следования и имена формальных аргументов функции. В теле функции пользователя используйте функции, построенные при решении задач пункта I.

1. Вектор  $\vec{a}$ , длина которого  $|\vec{a}| = l$ , составляет с осями ОХ и ОУ углы  $\alpha$  и  $\beta$  соответственно. Напишите функцию, возвращающую координаты вектора  $\vec{a}$ . Решите задачу, если  $l = 2$ ,  $\alpha = 60^\circ$ ,  $\beta = 120^\circ$ .
2. Напишите функцию, вычисляющую координаты точки М, лежащей на оси ОУ и равноудаленной от точек  $A = (x_A, y_A, z_A)$  и  $B = (x_B, y_B, z_B)$ . Решите задачу, если  $A = (1, -4, 7)$ ,  $B = (5, 6, -5)$ .
3. В треугольнике ABC заданы вершины  $A = (x_A, y_A, z_A)$ ,  $B = (x_B, y_B, z_B)$  и  $C = (x_C, y_C, z_C)$ . Напишите функцию, вычисляющую длину медианы, проведенной из вершины А. Решите задачу, если  $A = (3, -1, 5)$ ,  $B = (4, 2, -5)$ ,  $C = (-4, 0, 3)$ .
4. Могут ли три компланарных вектора  $\vec{a} = (a_x, a_y, a_z)$ ,  $\vec{b} = (b_x, b_y, b_z)$ ,  $\vec{c} = (c_x, c_y, c_z)$  являться сторонами треугольника? Напишите булеву функцию, тестирующую эту возможность. Решите задачу, если  $\vec{a} = (-2, 1, -2)$ ,  $\vec{b} = (-2, -4, 4)$ ,  $\vec{c} = (4, 3, -2)$ .
5. Могут ли векторы  $\vec{a} = (a_x, a_y, a_z)$ ,  $\vec{b} = (b_x, b_y, b_z)$  быть ребрами куба? Напишите булеву функцию, тестирующую эту возможность. Решите задачу, если  $\vec{a} = (7, 6, -6)$ ,  $\vec{b} = (6, 2, 9)$ .
6. Напишите функцию, вычисляющую проекцию вектора  $\vec{a} = (a_x, a_y, a_z)$  на ось  $\vec{b} = (b_x, b_y, b_z)$ . Решите задачу, если  $\vec{a} = (3, 1, -4)$ ,  $\vec{b} = (1, 3, -3)$ .
7. Напишите функцию, вычисляющую вектор  $\vec{b} = (b_x, b_y, b_z)$ , коллинеарный вектору  $\vec{a} = (a_x, a_y, a_z)$  и удовлетворяющий условию  $(\vec{b}, \vec{a}) = c$ . Решите задачу, если  $\vec{a} = (1, 2, -3)$ ,  $c = 28$ .
8. В треугольнике ABC заданы вершины  $A = (x_A, y_A, z_A)$ ,  $B = (x_B, y_B, z_B)$  и  $C = (x_C, y_C, z_C)$ . Напишите функцию, вычисляющую острый угол между медианой BD и стороной AC. Решите задачу, если  $\vec{a} = (1, 1, -1)$ ,  $\vec{b} = (2, 3, 1)$ ,  $\vec{c} = (3, 2, 1)$ .
9. Напишите функцию, вычисляющую ортогональную составляющую вектора  $\vec{a} = (a_x, a_y, a_z)$  вдоль вектора  $\vec{b} = (b_x, b_y, b_z)$ . Решите задачу, если  $\vec{a} = (-1, 2, 1)$ ,  $\vec{b} = (1, 1, 1)$ .

10. Отрезок АВ разделен точками С, D, E, F на 5 равных частей. Даны точки  $C = (x_C, y_C, z_C)$ ,  $F = (x_F, y_F, z_F)$ . Найдите координаты точек А, В, D, E. Решите задачу, если  $C = (3, -5, 7)$ ,  $F = (-2, 4, -8)$ .
11. Напишите функцию тестирующую, взаимное расположение заданных векторов  $\vec{a} = (a_x, a_y, a_z)$  и  $\vec{b} = (b_x, b_y, b_z)$ . Варианты ответов: «ортогональны», «противонаправлены», «сонаправлены», «неколлинеарны и неортогональны».
12. Напишите функцию, возвращающую координаты концов отрезка, который точками  $C = (x_C, y_C, z_C)$  и  $D = (x_D, y_D, z_D)$ . Решите задачу, если  $C = (2, 0, 2)$ ,  $D = (5, -2, 0)$ .

III. Решите задачи 1–12, сделав чертеж в приложении Paint. В среде числового пакета Mathcad постройте требуемые функции, тестируйте их работу. Средствами трехмерной графики изобразите объекты, указанные в задачах.

1. В параллелограмме ABCD заданы векторы  $\vec{CB} = \vec{a} = (a_x, a_y, a_z)$ ,  $\vec{CD} = \vec{b} = (b_x, b_y, b_z)$  и точка  $A = (x_A, y_A, z_A)$ . Напишите функцию, возвращающую координаты А, В, С, D вершин параллелограмма. Изобразите параллелограмм ABCD, если  $\vec{CB} = (2, -1, 4)$ ,  $\vec{CD} = (-3, 2, 1)$ ,  $A = (5, -3, 2)$ .
2. В треугольнике ABC, где точка М – середина стороны АВ, точка N – середина стороны ВС, заданы векторы  $\vec{AB} = \vec{a} = (a_x, a_y, a_z)$ ,  $\vec{MN} = \vec{b} = (b_x, b_y, b_z)$  и точка  $C = (x_C, y_C, z_C)$ . Напишите функцию, возвращающую координаты А, В, С вершин треугольника. Изобразите треугольник ABC, если  $\vec{AB} = (6, -1, 4)$ ,  $\vec{MN} = (-4, 3, 5)$ ,  $C = (1, 1, 1)$ .
3. В параллелограмме ABCD заданы векторы  $\vec{AB} = \vec{a} = (a_x, a_y, a_z)$ ,  $\vec{CB} = \vec{b} = (b_x, b_y, b_z)$  и точка  $A = (x_A, y_A, z_A)$ . Напишите функцию, возвращающую координаты точки пересечения диагоналей параллелограмма. Изобразите параллелограмм ABCD, если  $\vec{AB} = (-5, -1, 2)$ ,  $\vec{CD} = (-3, -3, 4)$ ,  $A = (2, 8, -2)$ .
4. В трапеции ABCD с основаниями ВС и AD заданы векторы  $\vec{AB} = \vec{a} = (a_x, a_y, a_z)$ ,  $\vec{AC} = \vec{b} = (b_x, b_y, b_z)$ ,  $\vec{AD} = \vec{d} = (d_x, d_y, d_z)$ . Напишите функцию, возвращающую координаты вектора  $\vec{MN}$ , где М и N – середины сторон АВ и CD. Изобразите трапецию ABCD, если  $\vec{AB} = (-2, 2, 5)$ ,  $\vec{AC} = (3, 6, -2)$ ,  $\vec{AD} = (10, 8, -14)$ ,  $B = (1, 1, 1)$ .

5. В параллелограмме ABCD заданы координаты трех вершин  $A = (x_A, y_A, z_A)$ ,  $B = (x_B, y_B, z_B)$ ,  $C = (x_C, y_C, z_C)$ . Напишите функцию, возвращающую координаты точки пересечения диагоналей параллелограмма. Изобразите параллелограмм ABCD, если  $A = (6, -5, 4)$ ,  $B = (2, 5, 1)$ ,  $C = (-3, 4, -2)$ .
6. В четырехугольнике ABCD заданы векторы  $\overrightarrow{AB} = \vec{a} = (a_x, a_y, a_z)$ ,  $\overrightarrow{BC} = \vec{b} = (b_x, b_y, b_z)$ ,  $\overrightarrow{AD} = \vec{d} = (d_x, d_y, d_z)$ , а векторы  $\vec{m}$  и  $\vec{n}$  — его диагонали. Напишите функцию, возвращающую модуль скалярного произведения векторов  $\vec{m}$  и  $\vec{n}$ . Изобразите четырехугольник ABCD, если  $\overrightarrow{AB} = (4, -1, 5)$ ,  $\overrightarrow{BC} = (-2, 8, 1)$ ,  $\overrightarrow{AD} = (-3, 4, 2)$ ,  $C = (1, 1, 1)$ .
7. В параллелограмме ABCD заданы векторы  $\overrightarrow{AB} = \vec{a} = (a_x, a_y, a_z)$ ,  $\overrightarrow{BD} = \vec{b} = (b_x, b_y, b_z)$  и точка  $A = (x_A, y_A, z_A)$ . Напишите функцию, возвращающую координаты A, B, C, D вершин параллелограмма. Изобразите параллелограмм ABCD, если  $\overrightarrow{AB} = (-3, 4, 1)$ ,  $\overrightarrow{BD} = (-2, 4, 1)$ ,  $A = (-5, 2, 8)$ .
8. В параллелограмме ABCD заданы координаты двух вершин  $A = (x_A, y_A, z_A)$ ,  $B = (x_B, y_B, z_B)$ , и координаты точки пересечения диагоналей  $Q = (x_Q, y_Q, z_Q)$ . Напишите функцию, возвращающую координаты A, B, C, D вершин параллелограмма. Изобразите параллелограмм ABCD, если  $A = (4, -2, 4)$ ,  $B = (5, -1, -8)$ ,  $Q = (7, -1, 4)$ .
9. Векторы  $\vec{a} = (a_x, a_y, a_z)$  и  $\vec{b} = (b_x, b_y, b_z)$ , проведенные из точки  $C = (x_C, y_C, z_C)$ , являются боковыми сторонами равнобедренного треугольника ABC. Напишите функцию, возвращающую координаты основания высоты треугольника, проведенной из вершины C. Изобразите треугольник ABC, если  $\vec{a} = (-3, 2, -4)$ ,  $\vec{b} = (4, 3, -2)$ ,  $C = (-6, 4, 3)$ .
10. В параллелограмме ABCD заданы векторы  $\overrightarrow{DC} = \vec{a} = (a_x, a_y, a_z)$ ,  $\overrightarrow{DB} = \vec{b} = (b_x, b_y, b_z)$  и точка  $D = (x_D, y_D, z_D)$ . Напишите функцию, возвращающую координаты точки пересечения диагоналей параллелограмма. Изобразите параллелограмм ABCD, если  $\overrightarrow{DC} = (-2, -4, 4)$ ,  $\overrightarrow{DB} = (6, -3, 2)$ ,  $D = (2, -6, 5)$ .
11. В трапеции ABCD заданы координаты ее вершин  $A = (x_A, y_A, z_A)$ ,  $B = (x_B, y_B, z_B)$ ,  $C = (x_C, y_C, z_C)$ ,  $D = (x_D, y_D, z_D)$ , BC и AD — основания

трапеции. Напишите функцию, возвращающую длину средней линии трапеции ABCD. Изобразите трапецию ABCD, если  $A = (3, -1, 2)$ ,  $B = (1, 2, -1)$ ,  $C = (-1, 1, -3)$ ,  $D = (3, -5, 3)$ .

12. Заданы неколлинеарные векторы  $\overrightarrow{AC} = \vec{a} = (a_x, a_y, a_z)$ ,  $\overrightarrow{AB} = \vec{b} = (b_x, b_y, b_z)$ , выходящие из некоторой точки A. Напишите функцию, возвращающую координаты вектора  $\overrightarrow{AO}$ , если O — точка пересечения медиан треугольника ABC. Изобразите треугольник ABC и точку O, если  $A = (1, 1, 1)$ ,  $\overrightarrow{AB} = (-3, 4, 1)$ ,  $\overrightarrow{AC} = (3, 6, -2)$ .

## Лабораторная работа № 8. Самостоятельная работа

### Постановка задачи:

Создайте мультфильм, демонстрирующий движение вдоль заданной линии следующих объектов: а) точки  $p_0 = (x_0, y_0)$ , б) направляющего вектора касательной прямой к этой линии в точке  $p_0$  (если он существует), в) вектора нормали в точке  $p_0$ , право ориентированного к направляющему вектору.

1. циклоида 
$$\begin{cases} x = a(t - \sin t), \\ y = a(1 - \cos t). \end{cases}$$
2. астроида 
$$\begin{cases} x = R \cos^3 t, \\ y = R \sin^3 t. \end{cases}$$
3. кардиоида 
$$\begin{cases} x = R \cos t (\cos t + 1), \\ y = R \sin t (\cos t + 1). \end{cases}$$
4. Декартов лист 
$$\begin{cases} x = \frac{3at}{t^3 + 1}, \\ y = \frac{3at^2}{t^3 + 1}. \end{cases}$$
5. циссоида 
$$\begin{cases} x = \frac{a}{t^2 + 1}, \\ y = \frac{a}{t(t^2 + 1)}. \end{cases}$$
6. верзьера Анъези 
$$\begin{cases} x = t, \\ y = \frac{a^3}{t^2 + a^2}. \end{cases}$$

$$\begin{aligned}
& \left\{ \begin{aligned} x &= (r_0 + r) \sin t - r \sin\left(\left(\frac{r_0}{r} + 1\right)t\right), \\ y &= (r_0 + r) \cos t - r \cos\left(\left(\frac{r_0}{r} + 1\right)t\right). \end{aligned} \right. & 0 < r < r_0 \\
7. \text{ эпициклоида} & \\
& \left\{ \begin{aligned} x &= (r_0 - r) \sin t - r \sin\left(\left(\frac{r_0}{r} - 1\right)t\right), \\ y &= (r_0 - r) \cos t + r \cos\left(\left(\frac{r_0}{r} - 1\right)t\right). \end{aligned} \right. & 0 < r < r_0 \\
8. \text{ гипоциклоида} &
\end{aligned}$$

## Лабораторная работа № 9. Фракталы. Задание фрактального преобразования

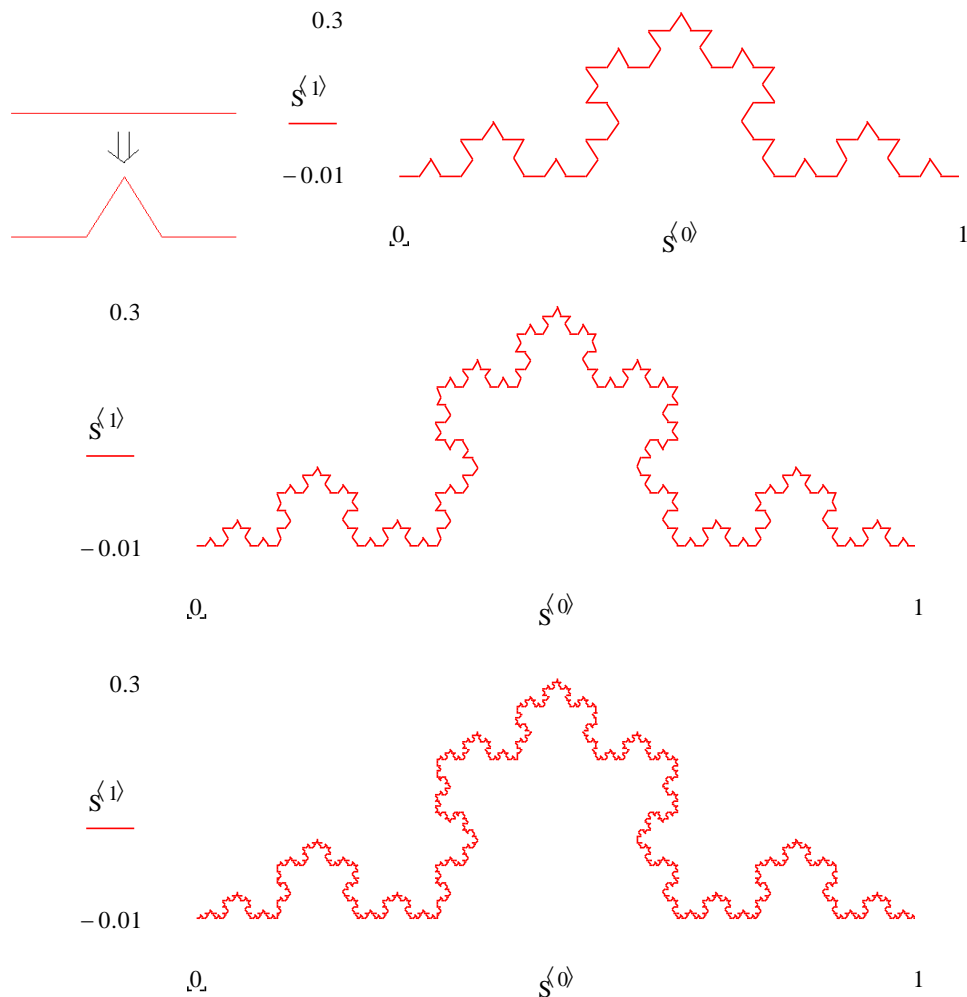
### Постановка задачи:

Построить аффинные преобразования, задающие построение элементарного фрактального объекта.

**Понятие о фрактальной графике.** Фрактал — фигура, элементы которой подобны самой фигуре. Элементы фрактального объекта повторяют свойства всего объекта. Фрактальными свойствами обладают многие объекты живой и неживой природы. Обычная снежинка, многократно увеличенная, оказывается фрактальным объектом. Взгляните на ветку папоротникового растения, и вы увидите, что каждая дочерняя ветка во многом повторяет свойства ветки более высокого уровня. Фрактальные алгоритмы лежат в основе роста кристаллов и растений.

В компьютерных технологиях фрактальные графические объекты создаются автоматической генерацией изображений. Создание фрактальной художественной композиции состоит не в рисовании и оформлении, а в программировании. Создается простейший, элементарный фрактальный объект, образ которого должен быть многократно повторен в фигуре. Далее задается отображение, закон, с помощью которого будут генерироваться этот образ — элемент, подобный элементарному объекту. При этом элементы последующих поколений должны полностью наследовать свойства своих родительских структур. Так рождается фрактальная фигура.

Один из примеров фрактального изображения — Кривая Кох. Из отрезка формируется ломаная линия, являющаяся исходным, базовым элементом для создания фигуры. Далее программным методом генерируется фрактальное изображение. Указывая различные параметры генерации, можно получать различные фигуры.



Вернемся к нашей задаче. В отдельных ветках деревьев можно проследить свойства всего дерева. А если ветку поставить в воду, то вскоре можно получить саженец, который со временем разовьется в полноценное дерево (это легко удастся сделать с веткой тополя).

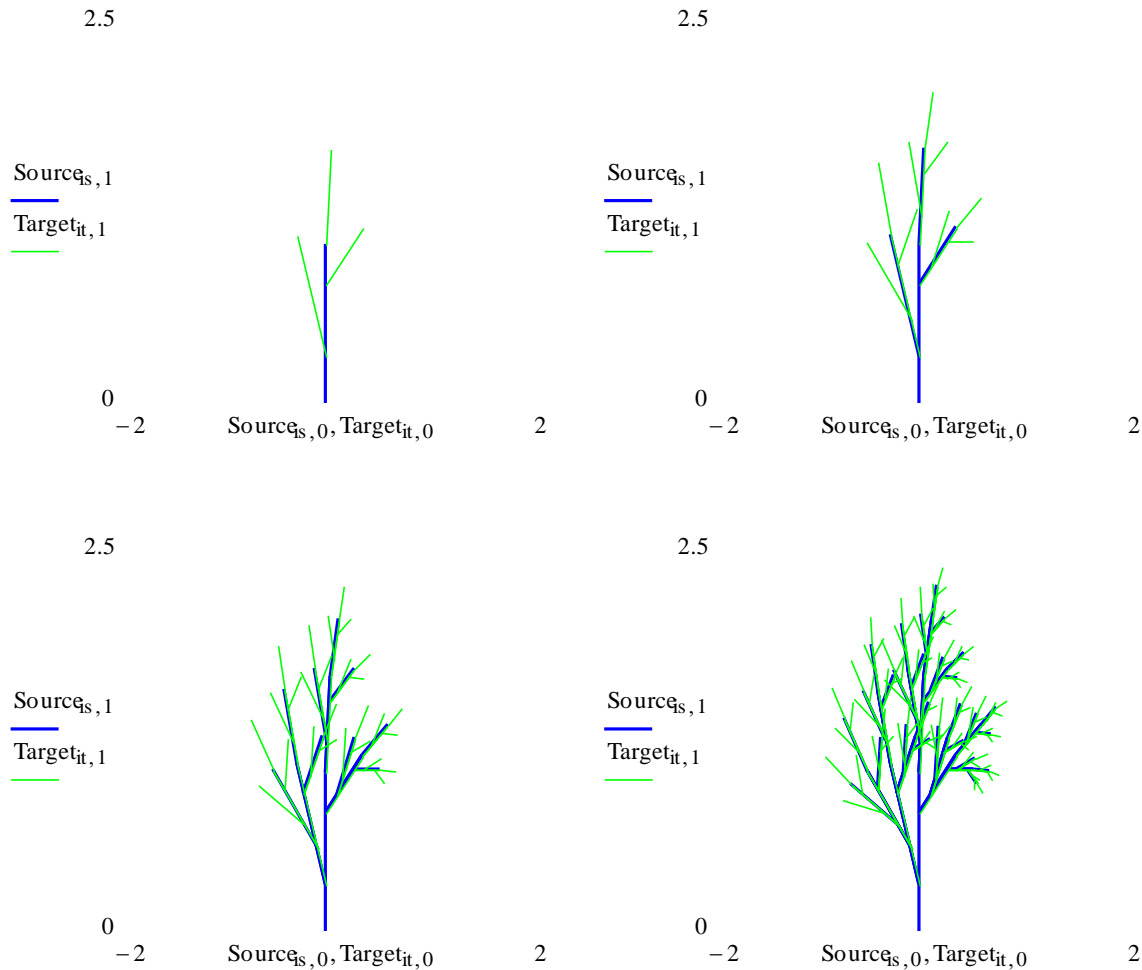
Таким образом, наша задача создания дерева посредством фрактальной графики разбивается на два этапа:

Построение ростка, который будет элементарным фрактальным объектом,

Создание фрактального преобразования, неоднократное применение которого позволит вырастить фрактальное дерево.

Этапы создания фрактала представлены ниже.





На первом этапе работы мы создадим росток, элементарный фрактальный объект, свойства которого определяют общий вид будущей фигуры. Для этого используем

**Элементарные аффинные преобразования плоскости.** Пусть на плоскости  $\alpha$  в аффинной системе координат дана точка  $P(x, y)$ . Преобразование вида

$$P' = A \cdot P + b, \quad (1)$$

где  $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ ,  $b = \begin{bmatrix} m \\ n \end{bmatrix}$ , ставящее в соответствие произвольной точке  $P(x, y)$  плоскости  $\alpha$  некоторую точку  $P'(x', y')$ , будем называть аффинным преобразованием плоскости  $\alpha$ .

Из (1) видно, что координаты точек  $P(x, y)$  и  $P'(x', y')$  связаны соотношениями  $\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} m \\ n \end{bmatrix}$ , которые можно представить в виде

$$\begin{cases} x' = ax + by + m \\ y' = cx + dy + n \end{cases} \quad (2)$$

Чтобы записать преобразование (1) с помощью некоторой матрицы  $C$

$$P' = C \cdot P, \quad (3)$$

используем понятие проективных координат, то есть для точек введем третью координату  $P' = [x', y, 1]^T$ ,  $P = [x, y, 1]^T$ .

Тогда  $C = \begin{bmatrix} a & b & m \\ c & d & n \\ 0 & 0 & 1 \end{bmatrix}$  и выражение (3) запишется в матричной форме

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & m \\ c & d & n \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}. \quad (4)$$

Выражение (4) задает преобразование плоскости. Нас интересуют преобразования сжатия, поворота и параллельного переноса.

Например, преобразование поворота радиус-вектора  $\overline{OP}$ ,  $O(0,0)$ ,  $P(x, y)$  на угол  $\phi$  задается матрицей  $C$  вида

$$RotateCS(\phi) \equiv \begin{bmatrix} \cos(\phi) & -\sin(\phi) & 0 \\ \sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

### Порядок выполнения работы:

1. Определите отрезок  $AB$ ,  $A(0,0)$ ,  $B(1,0)$  в качестве матрицы  $Source$ , располагая координаты точек по строкам. Чтобы Mathcad рисовал, отрывая перо от бумаги, добавьте третью точку с координатами  $(\infty, \infty)$ :

$$\begin{array}{ll} \text{координаты точки A} & \begin{pmatrix} 0 & 0 \end{pmatrix} \\ \text{координаты точки B} & \begin{pmatrix} 0 & 1 \end{pmatrix} \\ \text{отрываем перо от бумаги} & \begin{pmatrix} \infty & \infty \end{pmatrix} \end{array}$$

2. Составьте функцию  $Complete(Source)$ , добавляющую третью, проективную координату для каждой точки. Координаты точек в новой матрице должны располагаться по столбцам

$$\begin{pmatrix} 0 & 0 & \infty \\ 0 & 1 & \infty \\ 1 & 1 & 1 \end{pmatrix}.$$

3. Нарисуйте и форматируйте полученный объект  $Source$ , используя возможности графики Mathcad.

4. Используя глобальный оператор определения, создайте функции, формирующие матрицы сжатия  $Scale(Sx, Sy)$ , поворота  $RotateCS(a)$  и сдвига  $Shift(m, n)$ :

$$Scale(\alpha, \beta) \equiv \begin{bmatrix} \alpha & 0 & 0 \\ 0 & \beta & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad RotateCS(\phi) \equiv \begin{bmatrix} \cos(\phi) & -\sin(\phi) & 0 \\ \sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad Shift(m, n) \equiv \begin{bmatrix} 1 & 0 & m \\ 0 & 1 & n \\ 0 & 0 & 1 \end{bmatrix}$$

5. Постройте три образа объекта  $Complete(Source)$ , действуя на него последовательно некоторыми конкретными операторами сжатия, поворота и сдвига  $A_i$ , где  $A_i \equiv Shift(m, n) \cdot RotateCS(t) \cdot Scale(a, b)$ ,  $i = 1..3$ .
6. Создайте новый объект  $Target$ , содержащий, помимо ростка  $AB$ , все три его образа  $A_i \cdot Complete(Source)$ . Используйте функцию `augment`.
7. Нарисуйте росток  $AB$  и новый объект  $Target$  в одной графической области. Подберите значения параметров матриц линейных операторов экспериментальным путем.
8. Сохраните построенные линейные операторы в текущей директории, используя встроенную функцию `Mathcad WRITEPRN("filename.Dat")`. Убедитесь в том, что такой файл с необходимой информацией существует.
9. Запишите полученный в результате преобразования объект  $Target$  на носитель, предварительно убрав проективные координаты и транспонируя матрицу, ему соответствующую.

## Лабораторная работа № 10. Фракталы. Генерация изображения

### Постановка задачи

Используя аффинные преобразования, полученные в предыдущей работе, из ростка (отрезка  $AB$ ,  $A(0,0)$ ,  $B(0,1)$ ) вырастить фрактальное дерево.

### Алгоритм выполнения работы:

Теперь, на предыдущем занятии определены аффинные преобразования плоскости и получен фрактальный элемент — объект  $Target$ , решение задачи сводится к выполнению следующей последовательности действий:

- 1) Программное задание фрактального преобразования
- 2) Описание одного шага генерации фрактала
- 3) Создание графического образа фрактального дерева

### Порядок выполнения работы:

1. Создайте новый документ на диске компьютера, на котором Вы работаете, в папке Вашего курса, Вашей группы. Имя документа должно содержать идентификатор работы и Вашу фамилию.

2. Откройте Ваш документ KM04xxxxxxx.mcd, чтобы Mathcad имел возможность, просчитав открытый документ, записать на носитель определенные вами аффинные преобразования плоскости и полученный вами фрактальный элемент.
3. Откройте каждый из файлов данных и убедитесь в адекватности информации, в них содержащейся.
4. Скопируйте из рабочего документа работы № 9 функцию Complete(Source), добавляющую третью, проективную координату для каждой точки нашего прообраза.
5. Напишите функцию Fractal(A, Source), выполняющую фрактальное преобразование, которое было реализовано пошагово в рабочем документе работы № 9. Оформите тело функции в виде программного блока. Результатом выполнения функции должен быть новый фрактальный объект в том виде, в котором мы храним данные на носителе: нулевой столбец хранит координаты  $x$  нового объекта, первый — соответствующие координаты  $y$ .
6. При помощи функции READPRN получите с носителя аффинные преобразования  $A_i, i=0,1,2$ , и объект Target, который на новом шаге генерации фрактала должен сыграть роль прообраза Source.
7. Определите новый объект Target посредством вызова функции Fractal(A, Source).
8. Полученный результат при помощи функции WRITEPRN запишите в файл, который содержит образ текущего шага фрактального преобразования, или, что тоже самое — прообраз следующего шага фрактального преобразования.
9. Создайте графическую область, отображающую прообраз и образ текущего шага фрактального преобразования.
10. Пересчитайте документ, реализуя таким образом следующий шаг генерации фрактала.

## СПИСОК КОНТРОЛЬНЫХ ВОПРОСОВ

### Тема № 1. Азбука компьютерных технологий

1. Аппаратные и программные средства как единство двух составляющих информационной технологии.
2. Функции аппаратных и программных средств, образующих компьютерную технологию.
3. Что такое кодовая таблица. Наиболее широко используемые кодировки.
4. Какова основная конфигурация персонального компьютера?
5. Какие аппаратные компоненты компьютера располагаются в системном блоке?
6. Перечислите носители информации и укажите их основные характеристики.
7. Монитор. Назовите режимы отображения данных и укажите их характеристики.
8. Клавиатура. На какие функциональные группы условно разделяются клавиши?
9. Назначение памяти компьютера, ее виды, характеристики.
10. Представление и организация данных в компьютере.
11. Программное обеспечение. Его основные классы.
12. Понятие операционной системы, ее основные функции.
13. Что такое файловая система, ее организация.
14. Объясните фразу «Объектно-ориентированная среда Windows».
15. Объясните метафору Windows «Рабочий стол». Как провести его настройку?
16. Что такое графический интерфейс пользователя. Основная идея интерфейса Windows.
17. Три основных вида окон. Перечислите не менее семи элементов окна Приложения.
18. Три основных вида окон. Перечислите не менее семи элементов Диалогового окна.
19. Окно Документа. Где найти и как использовать инструменты для работы с Документом?
20. Главное меню Windows: назначение, основные пункты.
21. Контекстное меню объекта: назначение, способ вызова, основные пункты?
22. Меню Приложения (меню активного документа). Его основные пункты.
23. Понятия «служебных» и «горячих» клавиш. Разъясните их отличие. Приведите примеры.
24. Какова методика ознакомления с новыми Приложениями?

- 25. Каков типовой порядок работы с электронным Документом?
- 26. Перечислите основные технологические механизмы Windows.
- 27. Объясните фразу «инструментальная триада средств Приложения».
- 28. Что такое Буфер обмена и какие операции выполняются с его помощью?

## **Тема № 2. Офисные технологии. Работа с текстом**

- 29. Иерархия объектов Ms Word, подробнее.
- 30. Ms Word. Свойства символа. Форматирование.
- 31. Ms Word. Свойства абзаца. Форматирование.
- 32. Ms Word. Объясните понятия «мягкого» и «жесткого» пробела.
- 33. Дайте определение операциям редактирования и форматирования текста.
- 34. Укажите основные операции форматирования текста, как они выполняются.
- 35. Укажите основные операции редактирования текста, методику их выполнения.
- 36. Ms Word. Как осуществить перемещение по тексту, поиск и замену фрагментов текста.
- 37. Ms Word. Как работать с Редактором формул.
- 38. Таблицы в Ms Word. Их создание, редактирование, форматирование.
- 39. Ms Word. Как создать колонтитулы; какую информацию они, как правило, содержат.
- 40. Ms Word. Как создать Предметный Указатель.
- 41. Ms Word. Как создать Оглавление.
- 42. Ms Word. Оформление страницы: ее параметры, вставка номеров, сносок, разрыв страницы.
- 43. Ms Word. Настройка проверки правописания.
- 44. Объясните различие двух способов представления текстовой информации (.txt и .rtf).

## **Тема № 3. Числовой математический пакет Mathcad**

- 45. Mathcad. Назначение и структура.
- 46. Mathcad. Окно приложения.
- 47. Mathcad. Окно документа.
- 48. Mathcad. Элементы диалогового окна, примеры.
- 49. Mathcad. Меню приложения (меню активного документа).
- 50. Mathcad. Назовите 5 основных панелей инструментов, объясните их назначение.
- 51. Mathcad. Панель Math: ее математические палитры, назначение каждой из них.

52. Mathcad. Настройка панели инструментов: скрытие и отображение, плавающие панели.
53. Mathcad. Типы областей Regions. Создание областей, управление их расположением и отображением.
54. Mathcad. Элементы интерфейса редактора формул.
55. Mathcad. Редактирование формул с использованием двумерного курсора (линий ввода).
56. Mathcad. Математические области внутри текстовой: вставка, вычисление.
57. Mathcad. Редактирование областей как части документа.
58. Mathcad. Два способа импорта текста в документ.
59. Mathcad. Операции поиска и замены, их режимы выполнения.
60. Mathcad. Операторы локального и глобального определения.
61. Mathcad. Операторы вывода результата.
62. Mathcad. Основные типы данных.
63. Mathcad. Действительные и комплексные числа, системы счисления.
64. Mathcad. Встроенные константы, системные переменные, тип данных NaN.
65. Mathcad. Строковые выражения, встроенные функции для работы с ними.
66. Mathcad. Допустимые имена переменных и функций пользователя.
67. Mathcad. Особенности определения (построения) и использования функций.
68. Mathcad. Что такое дискретный интервал (аргумент), где и как он используется.
69. Mathcad. Укажите пять способов создания вектора или матрицы.
70. Mathcad. Как извлечь столбец матрицы? Строку матрицы?
71. Mathcad. Как узнать размеры матрицы, вектора? Как извлечь элемент матрицы, вектора?
72. Mathcad. Способы создания тензоров.
73. Mathcad. Способы отображения векторов и матриц.
74. Mathcad. Перечислите арифметические операции над матрицами, их особенности.
75. Mathcad. Решение систем линейных и нелинейных уравнений.
76. Mathcad. Три основных этапа построения графиков, подробнее.
77. Mathcad. Назовите семь видов графиков, представленных в пакете.
78. Mathcad. Как построить и форматировать декартов график функции, нескольких функций.
79. Mathcad. Как построить и форматировать график в полярной системе координат.
80. Mathcad. Укажите последовательные шаги построения графика поверхности.

81. Mathcad. Как построить линии уровня функции? Что это такое? Где используются?
82. Mathcad. Этапы построения трехмерного точечного графика. Его форматирование.
83. Mathcad. Каковы этапы создания и воспроизведения анимационного изображения.
84. Mathcad. Перечислите инструменты символьной математики.
85. Mathcad. Команды символьной математики, техника их выполнения.
86. Mathcad. Операторы символьных преобразований, техника их использования.
87. Mathcad. Команды и операторы алгебраических преобразований.
88. Mathcad. Решение линейных и нелинейных алгебраических уравнений и систем.
89. Mathcad. Назовите десять операторов программирования.
90. Mathcad. «Условный» оператор, как организовать выбор в программе.
91. Mathcad. Оператор цикла, как организовать повторение действий.
92. Mathcad. Создание подпрограмм. Стекло вызванных подпрограмм.

#### **Тема № 4. Символьный математический пакет Maple**

93. MapleV. Назначение пакета. Интерфейс пользователя пакета.
94. MapleV. Определение типа выражения. Изменение типа выражения.
95. MapleV. Типы атомарных объектов.
96. MapleV. Функции, используемые для анализа структуры выражения.
97. Числа в MapleV и встроенные функции для работы с ними.
98. MapleV. Строковые константы и встроенные функции для работы с ними.
99. MapleV. Переменные и идентификаторы. Придание переменным статуса предполагаемых.
100. MapleV. Списки и наборы. Их использование в аргументах встроенных функций.
101. MapleV. Последовательности: создание, использование.
102. MapleV. Перечислите типы операторов, приведите примеры.
103. MapleV. Встроенная функция `map`, ее назначение и использование.
104. MapleV. Создание матрицы, определение ее размеров.
105. MapleV. Арифметические операции над матрицами, особенности.
106. MapleV. Решение алгебраических уравнений, неравенств и систем.
107. MapleV. Функции преобразования алгебраических выражений.
108. MapleV. Операции с многочленами.
109. MapleV. Операции над множествами (наборами).
110. MapleV. Конструкции управления вычислением.



- 111.** MapleV. «Условный» оператор, как организовать выбор в программе.
- 112.** MapleV. Операторы цикла, как организовать повторение действий.
- 113.** MapleV. Как создать простейшую процедуру.
- 114.** MapleV. Пакеты приложений, не менее пяти. Их функции.
- 115.** MapleV. Двумерная графика.
- 116.** MapleV. Трёхмерная графика.

## ПРИМЕР КОНТРОЛЬНОГО ТЕСТА

### Офисные технологии

1. Внешний вид текста задают посредством операций
  - a. ввода
  - b. редактирования
  - c. форматирования
2. Файлы, сохраняющие всю информацию о внешнем виде текста, имеют в системе Windows расширение
  - a. \*.doc
  - b. \*.rtf
  - c. \*.txt
  - d. \*.mcd
3. Что означает символ ¶ в тексте документа?
  - a. разрыв страницы
  - b. конец абзаца
  - c. принудительный переход на новую строку в том же абзаце
  - d. разрыв таблицы
4. Нажатие каких клавиш реализует переход на новую строку в этом же абзаце?
  - a. Shift + Enter
  - b. Enter
  - c. Ctrl + Enter
  - d. Tab
5. «Мягкий пробел» — это символ,
  - a. появляющийся в строке при нажатии клавиши Space
  - b. автоматически появляющийся при выравнивании текста по ширине
  - c. появляющийся при нажатии клавиши табуляции Tab
6. Что такое «колонтитул»?
  - a. титульный лист документа
  - b. заголовок документа
  - c. текст, помещаемый вверху или внизу каждой страницы документа
  - d. заголовок колонки документа
7. Что такое «гlossарий»?
  - a. список имен собственных, встречающихся в документе, с указанием номеров соответствующих страниц
  - b. список терминов, наиболее часто упоминаемых в документе, и их определение

- с. список изображений описываемых в документе предметов с указанием места их расположения
  - d. список терминов, наиболее часто упоминаемых в документе, с указанием номеров страниц, на которых они встречаются
8. Чтобы удалить выделенный фрагмент текста и поместить его в буфер обмена, необходимо
- a. нажать клавишу Del
  - b. использовать комбинацию клавиш Ctrl + V
  - c. использовать комбинацию клавиш Ctrl + C
  - d. использовать комбинацию клавиш Ctrl + X
9. Для отображения выделенного фрагмента текста курсивом используют комбинацию клавиш
- a. Ctrl + I
  - b. Ctrl + U
  - c. Ctrl + B
10. Чтобы вставить фрагмент текста из буфера обмена, необходимо
- a. нажать клавишу Ins
  - b. использовать комбинацию клавиш Ctrl + V
  - c. использовать комбинацию клавиш Ctrl + C
  - d. использовать комбинацию клавиш Ctrl + X

### **Числовой математический пакет Mathcad**

11. Строка меню числового пакета Mathcad не содержит пункт
- a. Numeric
  - b. Tools
  - c. Symbolic
  - d. Insert
12. Выражения в рабочей области Mathcad располагаются
- a. в ячейках Cell ввода и вывода последовательно одно за другим
  - b. в областях Regions в любом месте документа
  - c. в областях Regions в строго определенных местах
13. Оператор глобального определения обозначается символом
- a.  $:=$
  - b.  $=$
  - c.  $=$
  - d.  $\equiv$
  - e.  $\rightarrow$
14. Оператор символьного вывода выражения имеет вид
- a.  $:=$

- b. =
- c. =
- d.  $\equiv$
- e.  $\rightarrow$

**15.** Какие символы недопустимы при задании имени переменной или функции

- a. символ бесконечности
- b. символ процента
- c. символ подчеркивания
- d. символ тождественного равенства

**16.** Ранжированная переменная, или дискретный интервал, — выражение, относящееся к типу данных

- a. числа
- b. строки
- c. массивы
- d. NaN

**17.** Для ввода числа в восьмеричной системе исчисления необходимо

- a. после ввода последней цифры числа ввести символ b
- b. перед вводом первой цифры числа ввести символ b
- c. перед вводом первой цифры числа ввести символ o
- d. после ввода последней цифры числа ввести символ o

**18.** Встроенная системная переменная, определяющая номер начального индекса, носит имя

- a. TOL
- b. ORIGIN
- c. FRAME
- d. PRNPRECISION

**19.** Оператор диапазона, используемый при определении дискретного интервала, имеет вид

- a. ..
- b. \$
- c. #
- d. `

**20.** При выводе числовых данных результат представляется с порядком, если задан следующий формат результата

- a. десятичный
- b. научный
- c. инженерный
- d. дробный

**21.** Для построения следующих типов трехмерных графиков удобно использование встроенной функции CreateMesh

- a. поверхность, заданная матрицей аппликат ее точек
  - b. поверхность, заданная в сферической системе координат
  - c. поверхность вращения графика функции одной переменной
  - d. точечный график, заданный матрицей аппликат его точек
- 22.** Доступ к элементу вектора или матрицы осуществляется при помощи
- a. имени вектора или матрицы с указанием нижнего индекса
  - b. имени вектора или матрицы с указанием верхнего индекса
  - c. имени вектора или матрицы с указанием позиции элемента в круглых скобках
  - d. имени вектора или матрицы с указанием позиции элемента в квадратных скобках
- 23.** Операция векторизации матрицы предполагает
- a. разбиение матрицы на векторы
  - b. вычисление векторного произведения столбцов матрицы
  - c. вычисление векторного произведения строк матрицы
  - d. одновременное проведение некоторой скалярной операции над элементами вектора или матрицы
- 24.** Число элементов вектора возвращает встроенная функция
- a. max
  - b. last
  - c. length
  - d. min
- 25.** Инструкция Break, используемая с инструкциями циклов while или for
- a. обеспечивает переход в конец тела цикла
  - b. обеспечивает переход в начало тела цикла
  - c. прерывает выполнение программы
  - d. передает управление вычислением в точку, следующую за телом цикла

## СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ

### Основная:

1. Голубева Л. Л. Компьютерная математика. Автоматизированное рабочее место математика: курс лекций / Л. Л. Голубева, А. Э. Малевич, Н. Л. Щеглова. Минск: БГУ, 2008. 139 с.
2. Шафрин Ю. Информационные технологии. В 2 частях. Часть 1. Основы информатики и информационных технологий. М.: Бином, Лаборатория знаний, 2004.
3. Дьяконов В., Новичков Ю., Рычков В. Компьютер для студента. СПб: Питер, 2001.
4. Кирьянов Д. Самоучитель Mathcad 13. СПб: БХВ-Петербург, 2006.
5. Heck A. Introduction to Maple. Third Edition. Springer, 2003.

### Дополнительная:

6. Петрушина Т. С., Рабцевич Т. И. Основы операционной системы Windows. Текстовый редактор Word. Практикум. Минск: БГУ, 2002.
7. Рассолько Г. А., Кремень Е. В., Кремень Ю. А. Методы программирования. Использование Excel для выполнения математических расчетов. Учеб.-метод. пособие. Минск: БГУ, 2002.
8. Рассолько Г. А., Кремень Е. В. Система тестов по математике и информатике на базе пакета Mathcad 2000. В 3 частях. Ч.1, Ч.2, Ч.3. Учеб.-метод. пособие. Минск: БГУ, 2002.
9. Вирт Н. Алгоритмы и структуры данных. М.: Мир, 1989.

## РАЗДЕЛ II. СИМВОЛЬНЫЙ МАТЕМАТИЧЕСКИЙ ПАКЕТ MATHEMATICA

### КОНСПЕКТ ЛЕКЦИЙ

Электронные версии лекций размещены в электронной библиотеке БГУ по адресу <http://elib.bsu.by/handle/123456789/13484>. Ниже приведен тематический план лекций.

#### Лекция № 1. Знакомство с пакетом

Назначение и структура Mathematica. Система справки. Создание документа и выбор стиля. Создание ячеек. Типы и свойства ячеек. Группировка ячеек. Форматы выражений. Особенности ввода данных. Символ операции вместо имени функции. Формы записи функции при вызове. Режимы выполнения функции. Функция пользователя как глобальные правила.

#### Лекция № 2. Выражение как базовый объект

Рекуррентное определение выражения. Полная форма выражения. Выражение как дерево. Уровни выражения. Позиция подвыражения. Извлечение части составного выражения. Действие функцией на часть выражения. Замена головы выражения. Функции для анализа структуры выражения. Чистая функция. Атомарные выражения. Извлечение атома.

#### Лекция № 3. Образцы

Понятие образца. Сравнение выражения с образцом. Особенности именования образца. Образцы в функциях пользователя. Образцы с условием. Образцы, содержащие альтернативу. Повторяющиеся образцы. Задание значения по умолчанию. Встроенное значение части выражения. Образцы в аргументах встроенных функций.

#### Лекция № 4. Символ — основной объект вычислений

Встроенные числовые константы. Системные переменные. Атрибут символа как свойство функции. Свойство коммутативности. Свойство ассоциативности. Дистрибутивность операции относительно списка. Атрибут OneIdentity. Атрибут Constant. Атрибуты Protected, Locked. Изменение порядка вычисления выражения. Списки как базовое выражение.

## **Лекция № 5. Знакомство с графикой**

Режимы выполнения функций графики. Функция Show. Базовые функции двумерной графики. Базовые функции трехмерной графики. Графические примитивы.

## **Лекция № 6. Локальные правила преобразований**

Понятие правила преобразований. Определение локальных правил преобразований. Выполнение локальных правил преобразований. Программирование в стиле локальных преобразований.

## **Лекция № 7. Глобальные правила преобразований**

Определение глобального правила преобразований. Собственное значение символа. Нижнее значение символа. Верхнее значение символа. Список дальних значений символа. Программирование в стиле глобальных правил преобразований.

## **Лекция № 8. Вычисление выражения**

Семейство Map-функций. Операторы повторного действия. Порядок вычисления выражения. Управление вычислительным процессом.



## ПЛАН ЛАБОРАТОРНЫХ ЗАНЯТИЙ

Электронные версии лабораторных работ размещены в электронной библиотеке БГУ по адресу <http://elib.bsu.by/handle/123456789/13484>. Ниже приведено краткое описание каждой работы.

### Лабораторная работа № 1. Знакомство с пакетом

#### Задание:

В среде Mathematica приобрести навыки работы:

- с системой справки;
- электронным документом;
- данными (освоить операции ввода, редактирования и форматирования данных);
- функциями (научиться находить и использовать встроенные функции, определять и применять простейшие функции пользователя).

### Лабораторная работа № 2. Структура выражения

#### Задание:

В среде Mathematica:

- изучите встроенные функции, позволяющие определить структуру выражения;
- научитесь представлять и читать выражение в каноническом виде;
- освоите операции извлечения частей выражения;
- создайте функцию, возвращающую для заданного выражения список его подвыражений каждого уровня и голову каждого из них.

### Лабораторная работа № 3. Функция дифференцирования

#### Задание:

Построить функцию пользователя  $\text{Dif}[\text{expr}, x]$ , вычисляющую производную функции  $\text{expr}$  одной переменной.

### Лабораторная работа № 4. Дифференцирование с ведением протокола

#### Задание:

Построить функцию, вычисляющую производную функции одной переменной и ведущую в процессе вычисления протокол выполнения операций дифференцирования.

## Лабораторная работа № 5. Знакомство с графикой

### Задание 1:

Требуется построить графики:

- функции  $y = f(x)$ , заданной на интервале;
- нескольких многочленов Тейлора функции  $y = f(x)$  в окрестности точки  $x_0 \in (x_1, x_2)$ .

### Задание 2:

Создайте мультфильм, демонстрирующий касательную к графику функции  $y = f(x)$ ,  $x_1 < x < x_2$  в точке  $(x_0, f(x_0))$ ,  $x \in (x_1, x_2)$  как предельное положение секущей, проходящей через эту точку.

### Задание 3:

Создайте мультфильм, изображающий замкнутую ломаную последовательно поступающих точек плоскости. В кадре, помимо графического образа объекта, отобразите его свойства: количество точек, их координаты, периметр замкнутой ломаной, диаметр множества ее вершин.

## Лабораторная работа № 6. Локальные преобразования и графика

### Задание 1:

Пусть задана геометрическая фигура `Figure`, состоящая только из линий, расположенных в трехмерном пространстве (каркасная фигура). Напишите функцию `Shift[Figure, Vector]`, клонирующую данную фигуру `Figure` в указанном направлении `Vector`.

### Задание 2:

Напишите процедуру `HyperCube(n)`, строящую с помощью функции `Shift` изображение псевдо- $n$ -мерного гиперкуба.

## Лабораторная работа № 7. Аппроксимация функций

### Задание 1:

Изучите функцию  $y = f(x)$ , заданную на интервале  $x_1 < x < x_2$ , непрерывно-дифференцируемую  $n$  раз в окрестности точки  $x_0 \in (x_1, x_2)$ , приближая ее в окрестности точки  $x_0$  многочленом Тейлора и соответствующей аппроксимацией Паде.

**Задание 2:**

Изучите функцию  $x(t)$ , непрерывную в окрестности точки  $t=0$  и удовлетворяющую соотношению

$$\begin{cases} x(t) = x^2(t) + 1 \\ x(0) = 0 \end{cases}, \quad -1 < t < 10$$

приближая ее в окрестности точки  $t=0$  многочленом Тейлора и соответствующей аппроксимацией Паде.

**Задание 3:**

Изучите функцию  $y = f(x)$ , заданную в неявном виде  $F(x, y) = 0$ , приближая ее в окрестности заданной точки  $x_0$  многочленом Тейлора и соответствующей многочлену Тейлора аппроксимацией Паде.

**Лабораторная работа № 8. Отображения точек плоскости****Задание:**

Осуществите указанное преобразование множества точек комплексной плоскости. Сделайте анализ результатов, используя понятие комплексного числа.

**Лабораторная работа № 9. Головоломка Ханойская башня****Задание:**

Напишите объектно-ориентированную программу решения головоломки «Ханойская башня»:

- разработайте и реализуйте алгоритм решения;
- реализуйте визуальное представление Ханойской башни;
- создайте мультфильм, демонстрирующий процесс решения.

## СПИСОК КОНТРОЛЬНЫХ ВОПРОСОВ

### Тема № 1. Знакомство с пакетом

1. Назначение пакета Mathematica, его сравнение с другими математическими пакетами.
2. Основные составные части Mathematica.
3. Интерфейс пользователя Mathematica.
4. Пункты меню приложения Mathematica.
5. Help | Documentation Center | Notebooks and Documents | Notebook Basics | Menu Items.
6. Назначение и возможности фасада, или оболочки Front End.
7. Приобретение знаний Mathematica во время сессии Session.
8. Сообщения и предупреждения Mathematica.
9. Help | Virtual book | Introduction | Getting Information from Mathematica | Warnings and Messages.
10. Help | Documentation Center, ввести в окно поиска Mathematica Sessions, по гиперссылке Mathematica Sessions (Mathematica Tutorial) перейти к разделу The Main Loop.
11. Структура справочной системы.
12. Различные способы получения справки по встроенным функциям Mathematica.
13. Создание различных типов ячеек ввода.
14. Объединение ячеек в группы, разъединение групп ячеек, управление отображением групп ячеек посредством пунктов меню и горячих клавиш.
15. Рабочий документ Mathematica: его секционирование посредством пунктов меню и горячих клавиш.
16. Help | Virtual book | Introduction | Working with the Notebook Interface | Notebooks as Documents.
17. Входные текстовые ячейки, их форматирование.
18. Help | Documentation Center | Notebooks and Documents | Formatting & Styling.
19. Ввод специальных символов с использованием клавиши <Esc>.
20. Help | Documentation Center, ввести в окно поиска Operators, по гиперссылке перейти к разделам Basic Mathematical Operators, Operators in Calculus and Algebra, Logical and Other Connectives.
21. Двумерный ввод математических выражений с клавиатуры.
22. Help | Virtual book | Introduction | Working with the Notebook Interface | Two-Dimensional Expression Input, разделы Typing Fractions, Powers, Square Roots, Radicals, Vectors, Matrices, Subscripts, Superscripts.

- 23. Три основных вида входных форматов выражения, преобразование выражения из одного формата в другой.
- 24. Формы записи функции при ее вызове.
- 25. Построение простейших функций пользователя.

## **Тема № 2. Выражение как базовый объект**

- 26. Рекуррентное определение выражения.
- 27. Полная, или каноническая форма записи выражения.
- 28. Функция, вычисляющая каноническую форму выражения.
- 29. Представление выражения в виде дерева.
- 30. Уровни выражения.
- 31. Спецификатор уровней выражения, его использование в аргументах встроенных функций.
- 32. Глубина выражения.
- 33. Понятие позиции подвыражения.
- 34. Вычисление позиции подвыражения.
- 35. Функции для извлечения части составного выражения.
- 36. Понятие атомарного выражения.
- 37. Типы атомарных выражений.
- 38. Полная форма атома каждого типа.
- 39. Извлечение части атомарного выражения.
- 40. Назначение и использование функции Map.
- 41. Назначение и использование функции Apply.
- 42. Построение чистой функции.

## **Тема № 3. Образцы**

- 43. Понятие образца.
- 44. Булева функция для сравнения выражения с образцом.
- 45. Различные способы именования образца.
- 46. Образцы с условием.
- 47. Образцы, содержащие альтернативу.
- 48. Повторяющиеся образцы.
- 49. Возможность задания значения части выражения по умолчанию.
- 50. Встроенное значение части выражения, соответствующей образцу.
- 51. Использование образцов в аргументах встроенных функций.
- 52. Использование образцов при построении функций пользователя.

## **Тема № 4. Символ — основной объект вычислений**

- 53. Атрибут символа как свойство функции.
- 54. Свойство коммутативности.
- 55. Свойство ассоциативности.
- 56. Дистрибутивность операции относительно списка.
- 57. Атрибут OneIdentity.
- 58. Атрибут Constant.
- 59. Атрибуты Protected, Locked.
- 60. Атрибуты семейства Holdxxx.
- 61. Примеры изменения порядка вычисления выражения посредством установки атрибута символа.

## **Тема № 5. Знакомство с графикой**

- 62. Режимы выполнения функций графики.
- 63. Функция Show.
- 64. Базовые функции двумерной графики.
- 65. Базовые функции трехмерной графики.
- 66. Двумерные графические примитивы.
- 67. Трехмерные графические примитивы.
- 68. Директивы функций графики.

## **Тема № 6. Локальные правила преобразований**

- 69. Правила преобразований, сфера видимости и момент вычисления.
- 70. Понятие локальных правил преобразований.
- 71. Функции, определяющие локальные правила преобразований.
- 72. Функции для применения локальных правил преобразований.
- 73. Отличия функций ReplaceAll, ReplaceRepeated, Replace.

## **Тема № 7. Глобальные правила преобразований**

- 74. Понятие глобальных правил преобразований.
- 75. Понятие глобальных правил преобразований.
- 76. Функции, определяющие глобальные правила преобразований.
- 77. Список собственных значений символа.
- 78. Список нижних значений символа.
- 79. Список верхних значений символа.
- 80. Список дальних значений символа.

## **Тема № 8. Вычисление выражения**

- 81.** Обзор функций семейства Map.
- 82.** Функция MapAt.
- 83.** Функция MapAll.
- 84.** Функция MapThread.
- 85.** Функция MapIndexed.
- 86.** Итерационные процессы с использованием функций семейства Nest, Fold.
- 87.** Функции Nest, NestList, FixedPoint.
- 88.** Функции Fold, FoldList.

## СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ

1. Голубева Л. Л. Компьютерная математика. Символьный пакет Mathematica: курс лекций / Л. Л. Голубева, А. Э. Малевич, Н. Л. Щеглова. Минск: БГУ, 2005. 103 с.
2. Воробьев Е. М. Введение в систему «Математика»: Учеб. пособие. М: Финансы и статистика, 1998. 262 с.
3. Wolfram S. The Mathematica Book. Fourth Edition. Cambridge, University Press, 1999.



## РАЗДЕЛ III. ЧИСЛОВЫЙ МАТЕМАТИЧЕСКИЙ ПАКЕТ MATLAB

### КОНСПЕКТ ЛЕКЦИЙ

MATLAB® (MATrix LABoratory) — это высокоэффективная система инженерных и научных расчетов фирмы The MathWorks, Inc (USA). В следующих темах излагаются основы работы в среде MATLAB:

- ✓ Тема 1 — назначение и структура пакета MATLAB, графический интерфейс пользователя.
- ✓ Тема 2 — представление данных в виде массивов, матриц и векторов, вычисления арифметических выражений.
- ✓ Тема 3 — основные конструкции языка программирования MATLAB.
- ✓ Тема 4 — сценарии и функции, локальные и глобальные переменные.
- ✓ Тема 5 — основы объектно ориентированного программирования.
- ✓ Тема 6 — высокоуровневая графика.
- ✓ Тема 7 — графические объекты как элементы дескрипторной графики.
- ✓ Тема 8 — импорт и экспорт данных, организация работы с файлами.
- ✓ Тема 9 — создание приложений, оснащенных графическим интерфейсом пользователя.
- ✓ Тема 10 — вычисления в MATLAB.

## Тема № 1. Назначение и структура MATLAB

MATLAB поддерживает диалоговый интерактивный режим для выполнения простых вычислений и операций, а также, являясь операционной средой и обладая языком программирования, позволяет писать программы, оснащенные графическим интерфейсом.

Представление данных в виде массивов (матриц и векторов) обеспечивает пользователю широкие возможности решения задач, связанных с матрицами и матричными вычислениями.

### Структура системы MATLAB

Система MATLAB состоит из пяти основных частей:

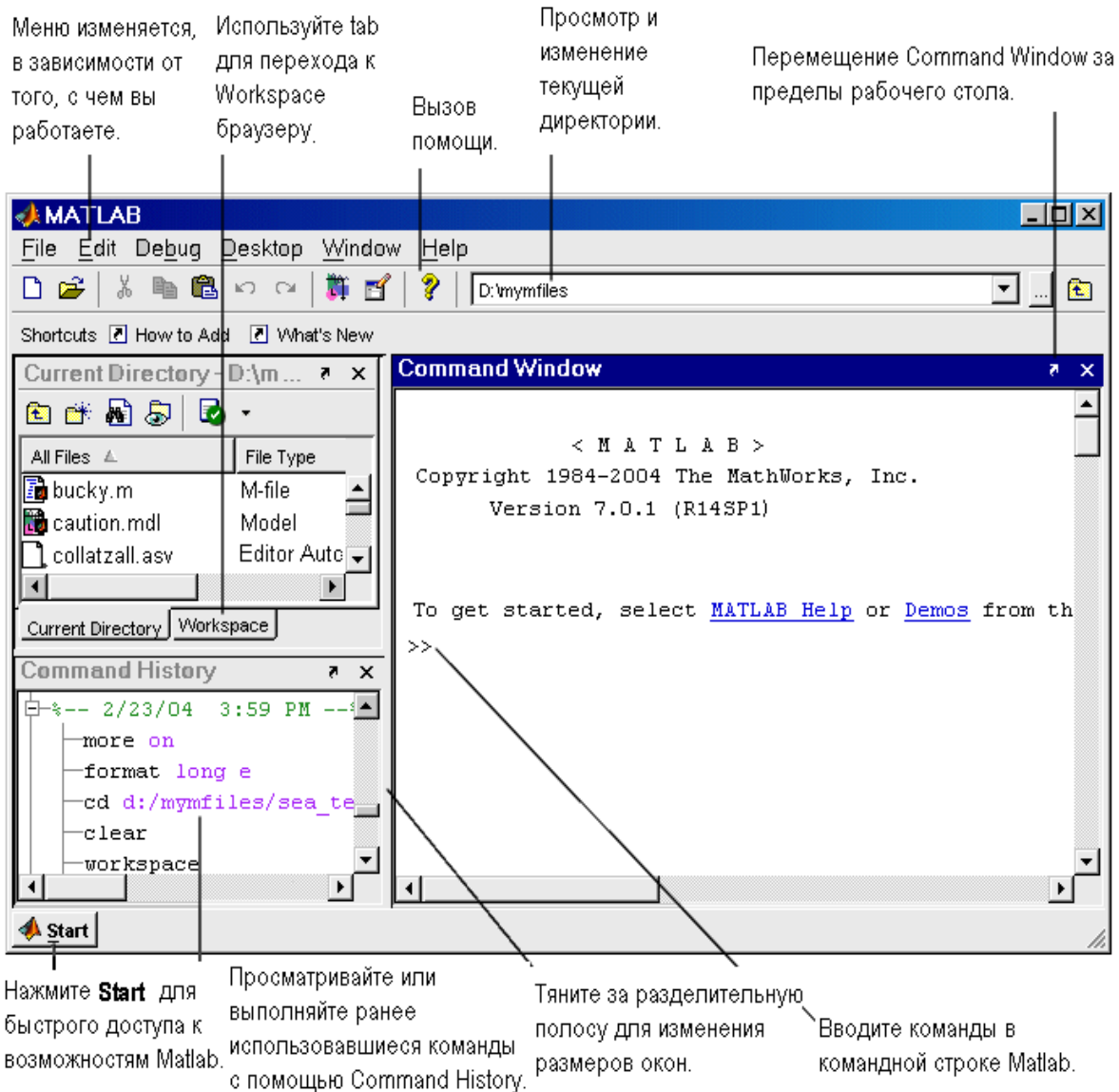
- среда разработки;
- библиотека математических функций;
- язык программирования;
- высокоуровневая и дескрипторная графика;
- интерфейс прикладного программирования.

### Интерфейс пользователя

**Среда разработки** — это набор инструментов и средств, которые помогают использовать функции и файлы MATLAB, а также поддерживают связь MATLAB с пользователем и другими приложениями. Среда разработки включает в себя рабочий стол **MATLAB Desktop**, браузеры справочной системы **Help**, файлов и поискового пути **Search Path**.

Интерфейс рабочего стола **MATLAB Desktop** представляет собой графический интерфейс, содержащий следующие элементы:

- несколько окон;
- меню, пункты которого автоматически перенастраиваются в зависимости от того, какое окно является в данный момент активным;
- кнопку **Start** для получения доступа к содержимому рабочей среды, модулям **Toolbox**, справочной системе и демонстрационным программам;
- панель инструментов с кнопками и раскрывающимся списком;
- строку состояния.



К окнам **MATLAB Desktop** относятся:

- командное окно **Command Window**;
- окно просмотра рабочего пространства **Workspace Browser**;
- окно текущего каталога **Current Directory**;
- окно истории команд **Command History**;
- окно редактора-отладчика М-файлов **M-file Editor/Debugger**;
- окна для работы с графикой **Figures**.

Вспомогательные окна:

- окно редактора массивов **Array Editor**;
- диалоговое окно установки путей **Set Path**;

- окно **Launch Pad** доступа к различным модулям **Toolbox**;
- окно графического редактора **Layout Editor** для создания GUI;
- окно просмотра справочной системы **Help Browser**.

### Справочная система

- Открытость кода встроенных функций MATLAB.
- Полная справочная система с подробным описанием компонентов MATLAB, его структуры, встроенных функций и примерами.
- Доступ из командной строки к кратким сведениям о встроенных функциях:
  - **doc** — запуск справочной системы по всем разделам MATLAB в отдельном окне **Help**;
  - **help** — вывод разделов встроенной справки в командное окно;
  - **help group** — обзор указанного раздела в командном окне. Например, **>>help elfun** — обзор элементарных функций;
  - **help function** — описание указанной функции или команды в командном окне;
  - **helpwin** — запуск краткой справочной системы по разделам MATLAB в окне **Help**;
  - **helpdesk** — запуск справочной системы по всем разделам MATLAB в отдельном окне **Help**;
  - **lookfor Key** — поиск М-файлов по ключу;
  - **demo** — демонстрационные примеры в окне **Help**;
  - **type <function>** — просмотр текста М-файла.

## Тема № 2. Данные и вычисления

Все данные в MATLAB представляются в виде массивов, использование которых позволяет эффективно решать многие вычислительные задачи, связанные с векторно-матричными формулировками.

### Арифметические вычисления

Встроенные функции MATLAB позволяют находить значения различных выражений. Команды для вычисления выражений имеют вид, свойственный всем языкам программирования высокого уровня.

- При записи оператора с неявным присваиванием вида **x + y** результат присваивается автоматически создаваемой системой переменной с именем **ans** (ANSwer).

- Переменные с результатом выполнения любого арифметического действия размещаются в рабочем пространстве.

### Рациональные, комплексные и специальные числа

- Ввод в десятичной системе счисления.
- Формат: [Знак] Число [Экспонента]:
  - знак: **+** или **–**;
  - число: минимально одна цифра; целочисленное или вещественное представление;
  - экспонента: **e** или **E** перед целым числом.
- Представление комплексного числа : два рациональных числа через знак **+** или **–** .
- Мнимая часть числа: символ **i** либо **j**.
- Если мнимая часть числа равна нулю, то выводится только вещественное число.
- Комплексные числа при умножении, делении и возведении в степень заключаются в круглые скобки.
- Для вычисления комплексно-сопряженного числа применяется апостроф, который следует набирать сразу за числом, без пробела.
- **pi**: число  $\pi$ ;
- **Inf**: бесконечность ( $\infty$ ) :
  - возникает при переполнении;
  - имеет знак.
- **NaN**: нечисловое значение (Not a Number):
  - возникает при вычислениях с неопределенным результатом;
  - операции с **NaN** возвращают **NaN**.

### Простейшие вычисления

- Операторы: **+**, **.\***, **—**, **.-**, **\***, **.\***, **/**, **./**, **\**, **.\**, **^**, **.^**, **'**, **.'**
- Порядок операций:
  - возведение в степень **^**;
  - умножение и деление **\***, **/**, **\**;
  - сложение и вычитание **+**, **–**;
  - слева направо.
- Изменение порядка: **()** .

### Форматы представления данных

- Вид, в котором выводится результат вычислений, зависит от формата вывода, установленного в MATLAB.
- Переустановка формата: команда **>> format**.

### Встроенные математические функции

- Имена функций набираются строчными буквами.
- Аргументы функций заключаются в круглые скобки и разделяются запятыми.
- Выходные параметры функций заключаются в квадратные скобки и разделяются запятыми или пробелами.
- Список всех встроенных элементарных функций с их кратким описанием: команда **>> help elfun**.
- Список всех встроенных специальных математических функций: команда **>> help specfun**.

### Типы данных

- Все данные MATLAB являются массивами.
- Размерность массивов не надо указывать явно.
- Минимальная размерность массива  $0 \times 0$  и может увеличиваться до требуемого значения динамически.
- Типы данных: **logical, char, int8, uint8, int16, uint16, int32, uint32, int64, uint64, single, double, cell, struct, function handle, user classes, java classes**.
- Индексы массивов начинаются с 1.
- Тип **sparse** предназначен для работы с разреженными матрицами.
- Перечень встроенных типов данных: команда **>> help datatypes**.

### Переменные

- Имя переменной начинается с буквы, содержит буквы, цифры и символ подчеркивания.
- Переменные в MATLAB не нужно предварительно описывать, указывая их тип и размерность. Все данные хранятся в виде массивов.
- Зарезервированные имена: **ans, pi, NaN, Inf, end**.
- Команда **>> clear** очищает рабочее пространство **Workspace**.

- Команда **>> who** возвращает в окне **Command Window** список всех переменных, содержащихся в **Workspace**.
- Команда **>> whos** возвращает список всех переменных, содержащихся в **Workspace**, с указанием их размерности, количества байт в памяти и класса (типа).

## Типы массивов

- Массивы — основные объекты системы MATLAB, над которыми выполняются действия. Типы массивов:
  - одномерные массивы — векторы;
  - двумерные массивы — матрицы;
  - многомерные массивы — тензоры.
- Создание матрицы — операция конкатенации **[]**.
- Ввод элементов одной строки — через запятую **,** или пробел.
- Ввод строк — через точку с запятой **;**.
- Скаляр — матрица  $(1 \times 1)$ .
- Вектор может быть представлен вектор-столбцом (матрица  $(n \times 1)$ ) или вектор-строкой (матрица  $(1 \times n)$ ).
- Доступ к элементу — при помощи операции индексации **()**:
  - один индекс (порядковый номер элемента);
  - группа индексов, разделяемых запятыми (первый индекс — номер строки, второй — номер столбца, все остальные — номера страниц).
- Индексация начинается с 1, последнее значение индекса — **end**.
- Элементы любого массива упорядочены по столбцам.
- Справка об элементарных матрицах и матричных манипуляциях: команда **>> help elmat**.

## Векторы и матрицы

- Создание вектора:
  - операция конкатенации;
  - последовательным вводом элементов;
- операцией диапазона значений: **a:s:b**: операция формирования диапазона значений. Вектор  $[a, a+s, a+2*s, \dots, a+n*s]$ , где  $n = \text{floor}((a-b)/s)$ . Аналог: функция **colon(a,s,b)**.

## Специальные матрицы

- **[ ], diag, eye, gallery, eye, hilb(n), magic, ones, rand, randn, tril, triu, zeros.**

## Доступ к элементам векторов и матриц

- Доступ к элементу — при помощи операции индексации **( )**:
  - один индекс (порядковый номер элемента);
  - группа индексов, разделяемых запятыми (первый индекс — номер строки, второй — номер столбца).
- В качестве индекса может задаваться вектор.
- Чтение несуществующего элемента: сообщение об ошибке.
- Запись несуществующего элемента: недостающие элементы заполняются нулями.

## Преобразования векторов и матриц

- **A.' , A'**: транспонирование, комплексно-сопряженное транспонирование. Аналог функций **transpose(A)**, **ctranspose(A)**.
- **A(:)**: преобразование матрицы в вектор-столбец.
- **diag(A,k)**: создание диагональной матрицы или выделение диагонали.
- **fliplr(A)**, **flipud(A)**, **rot90(A)**: отражение, поворот.
- **reshape(A,rn,cn)**:  $(rn \times cn)$  матрица из **A**. Элементы выбираются последовательно по столбцам.
- **repmat(A,rb,cb)**:  $(rb \times cb)$  блок-матрица из матрицы **A**.
- **tril(A,k)**, **triu(A,k)**: выделение нижней или верхней треугольной части матрицы **A**.

## Размеры векторов и матриц

- **size(A)**: вектор с количеством строк и столбцов матрицы **A**.
- **size(A,k)**: **k=1**: количество строк, **k=2**: количество столбцов.
- **[r, c]=size(A)**: количество строк в **r**, количество столбцов в **c**.
- **length(A)**: возвращает **max(size(A))** или 0 для пустой матрицы.

## Арифметические операции над векторами и матрицами

- **., .-, .\*, ./, .\, .^**: поэлементные операторы сложения, вычитания, умножения, деления и возведения в степень. Если один из операндов — скаляр, то он воздействует на каждый элемент другого операнда;
- **A+B**: матричное сложение, аналог **A.+B**;



- **A-B**: матричное вычитание, аналог **A. -B**;
- **A\*B**: матричное умножение (по правилу «строка на столбец»);
- **A^n**: n-кратное матричное умножение;
- **X=B/A**: левостороннее деление, решение системы линейных уравнений **XA=B**;
- **X=A\B**: правостороннее деление, решение системы линейных уравнений **AX=B**;
- **A.'**: транспонирование матрицы;
- **A'**: комплексно-сопряженное транспонирование матрицы.

### Операции отношения и логические операции

- **>, >=, <, <=, ==, ~=**: операторы отношения, аналоги функций **gt, ge, lt, le, eq, ne**. Поэлементные сравнения матриц одинакового размера. Скаляр сравнивается с каждым элементом массива. Результат — логическая матрица.
- Операции **==, ~=** проводят сравнения вещественных и мнимых частей комплексных чисел, а операции **>, <, >=, <=** — только вещественных частей.
- **&, |, ~**: логические операторы, аналоги функций **and, or, not, xor**.
- **find**: выделение части массива, элементы которого удовлетворяют определенному условию.
- **isempty, isfinite, isinf, isnan, isprime, isreal**: построение логических матриц в зависимости от выполнения условия.

### Матричные функции

- **sum, prod**: сумма (произведение) всех элементов вектора; сумма (произведение) элементов каждого столбца/строки матрицы;
- **max, min**: максимальный (минимальный) элемент вектора; максимальный (минимальный) элемент в каждом столбце/строке матрицы;
- **sort**: сортировка, действие аналогично предыдущим функциям;
- **all**: проверка на наличие нулевого элемента в массиве;
- **any**: проверка на наличие ненулевого элемента в массиве;
- **find**: нахождение индексов и значений ненулевых элементов;
- **spy**: визуализация ненулевых элементов матрицы;
- **norm(M)**: норма матрицы;

- скаляр: **rank**, **det**, **trace**, **cond**;
- справка: команда **>> help matfun**.

## Текстовые строки

- Тип данных **char** («символ»);
- текстовые данные, в том числе и одиночный символ, заключаются с обеих сторон в апострофы (одинарные кавычки):
- символьные переменные в виде массивов (в системе MATLAB все есть массив) типа **char**;
- ввод текстовых строк:
  - групповое присваивание;
  - поэлементное присваивание с помощью операции индексации;
  - операция конкатенации;
- справка: **help strfun**.

## Структуры

- Структуры — объекты данных (тип **struct**), содержащие разную поименованную информацию (поля структуры).
- Форма: **VariableName.FieldName**.
- Создание структуры:
  - присваивание значений всем полям при помощи оператора присваивания;
  - функция-конструктор **struct**, позволяющая внести значения сразу в несколько полей структуры.
- Набор полей структуры (массива структур) можно изменять динамически.
- Поля структуры могут содержать любой тип данных.
- Отдельная структура — это структурный массив размерности  $1 \times 1$ .
- Вложенность структур: в качестве имени вложенной структуры выступает имя поля внешней структуры.
- Доступ к структурам, входящим в массив, осуществляется при помощи индексации: **VariableName(1)**.
- Доступ к элементу матрицы, являющейся значением поля структуры, происходит при помощи двух операций индексации и одной операции доступа к полю структуры: **VariableName(1).FieldName(1,2)**.
- Вывод содержимого отдельных полей массива структур: фигурные скобки **{VariableName.FieldName}**, если с полем связаны символьные

данные, и квадратные скобки **[VariableName.FieldName]**, если с полем связаны числовые данные.

- Если два массива структур имеют одинаковый набор полей, то допускается групповое присваивание.
- Функции для работы со структурами:
  - **fieldnames**: получение названий полей структуры;
  - **getfield**: получение содержимого определенных полей;
  - **rmfield**: удаление полей структуры;
  - **setfield**: присвоение значения полю структуры;
  - **struct**: создание структур и массива структур;
  - **struct2cell**: преобразование массива структур в ячейки;
  - **deal**: организация доступа к полям массива структур;
  - **isstruct**: проверка, является ли указанный объект структурой;
  - **isfield**: проверка, содержит ли структура указанное поле.

### Массивы ячеек

- Массив ячеек (тип **cell**): хранение разнородных объектов (массивов разных размерностей, разнотипных данных).
- Массив ячеек состоит из ячеек, или контейнеров, каждый из которых может содержать данные разных типов.
- Массив ячеек может быть полем структуры.
- Создание массива ячеек:
  - присваивание значений всем ячейкам при помощи оператора присваивания;
  - функция-конструктор **cell**, позволяющая внести значения сразу в несколько ячеек массива.
- Ячейка — это содержимое плюс некоторая оболочка (служебная структура данных) вокруг этого содержимого, позволяющая хранить в ячейке произвольные типы данных любого размера.
- Выражение любого типа данных можно превратить в ячейку, заключив его в фигурные скобки.
- **()**: доступ к ячейке, **{}**: доступ к содержимому ячейки.
- **cell**: создание пустого массива ячеек заданного размера;
- **cellfun**: применение функции к содержимому массива ячеек;
- **cellstr**: преобразование массива символов в массив ячеек;

- **cell2struct/cell2matrix**: преобразование массива ячеек в массив структуры / в матрицу;
- **struct2cell num2cell**: преобразование массива структур / числового массива в ячейки;
- **deal**: извлечение содержимого ячеек;
- **cellplot**: отображение содержимого массива ячеек в графическом окне;
- **celldisp**: вывод содержимого массива ячеек в командное окно;
- **iscellstr**: проверка, является ли объект массивом ячеек из строк;
- **iscell**: проверка, является ли указанный объект массивом ячеек.

### Тема № 3. Элементы программирования

Система MATLAB включает в себя **язык программирования** — высокоуровневый язык с операторами управления, функциями, структурами данных, вводом-выводом и возможностями объектно-ориентированного программирования.

#### Управление вычислением

- Операторы управления: операторы ветвления и операторы цикла:
  - **if**: оператор условия: вычисляет логическое выражение и выполняет группу операторов, если выражение истинно;
  - **switch**: оператор переключения: выполняет группу операторов в зависимости от значения переменной или выражения;
  - **while**: оператор цикла: повторяет группу операторов до тех пор, пока выполняется логическое условие;
  - **for**: оператор цикла: выполняет группу операторов заданное число раз;
  - **break**: оператор управления: выполняет досрочный выход из циклов **for** или **while**;
  - **continue**: оператор управления: передает управление следующей итерации циклов **for** или **while**.
- Логические выражения операторов управления задаются операторами отношения или логическими операторами:
  - **>, >=, <, <=, ==, ~=**: операторы отношения, аналоги функций **gt, ge, lt, le, eq, ne**;
  - **&, |, ~**: логические операторы, аналоги функций **and, or, not, xor**;

- **find**: выделение части массива, элементы которого удовлетворяют определенному условию;
  - **all**: проверка на наличие нулевого элемента в массиве;
  - **any**: проверка на наличие ненулевого элемента в массиве. В случае применения к матрицам функции **all** и **any** возвращают значения по каждому столбцу;
  - **isempty**, **isfinite**, **isinf**, **isnan**, **isprime**, **isreal**, **isstr**, **ischar**: индикаторные функции.
- Обработка исключительных ситуаций:
    - назначение: операторы, выполнение которых может привести к ошибке, заключаются в конструкцию между **try** и **catch**;
    - если эти операторы выполняются без ошибок, то управление передается первой команде, стоящей после ключевого слова **end**;
    - если возникает ошибка во время выполнения любого из этих операторов, то управление передается блоку, заключенному между ключевыми словами **catch** и **end**. Если и здесь встречается ошибка, то выполнение программы прекращается;
    - ключевые слова **try**, **catch**, **end**.

## Тема № 4. Сценарии, функции и переменные

Режимы работы:

- Работа в интерактивном режиме: ввод команд с клавиатуры в командной строке командного окна **Command Window**.
- Работа с М-файлами: сохранение последовательности команд в текстовых файлах, дополнение их комментариями и выполнение из редактора М-файлов или запуск из командной строки в командном окне **Command Window**.

Для работы с М-файлами предназначен встроенный текстовый редактор М-файлов, который вызывается командой меню **File | New ► M-file** и работает в своем собственном окне, обеспечивая создание, редактирование, запуск и отладку кода.

### Типы М-файлов

**М-файлы**: содержат команды и операторы MATLAB.

Типы М-файлов:

- **файл-сценарии (Script M-files)**, содержащие последовательность команд;
- **файл-функции (Function M-files)**, в которых описываются функции.

## Сценарии

**Сценарий (script)** — это текстовый файл, в котором записаны команды, подлежащие последовательному выполнению. Свойства сценария:

- нет специального оформления;
- доступны все переменные, определенные в рабочем пространстве;
- не допускает входных и выходных параметров;
- все переменные, которые создаются и используются в сценарии, сохраняются в рабочем пространстве **Workspace** и доступны по окончании работы сценария;
- предназначен для упрощения многократного вызова группы команд;
- не может использоваться в качестве операндов в выражениях или в качестве входных аргументов функций;
- сохранение в файл на диске: текстовый файл с расширением **m** (например, `file1.m`);
- для выполнения сценария содержащий его М-файл должен находиться в текущем каталоге **Current Directory** либо путь к нему должен быть указан в пути поиска **Search Path**;

## Функции

- Функции оформляются в виде текстовых файлов с расширением **m**.
- Определение М-функции:
- **function [список\_вых\_параметров] = имя\_функции (список\_вход\_параметров)**
- М-функция включает следующие компоненты:
  - **строка заголовка функции** задает имя функции, количество и порядок следования выходных и входных аргументов;
  - **первая строка подсказки** определяет исчерпывающее краткое назначение функции; выводится на экран командой **lookfor** или **help**;

- **текст подсказки** представляет собой более детальное описание назначения и способов вызова функции, выводится на экран командой **help**;
  - **тело функции** представляет собой непосредственно программный код, который реализует вычисления и присваивает значения выходным аргументам;
  - **комментарий** представляет собой текст внутри программного кода, описывающий алгоритм выполнения функции.
- По умолчанию внутренние переменные являются локальными и хранятся во внутреннем рабочем пространстве, относящемся к данной функции.
  - Функция используется для расширения возможностей языка MATLAB.
  - Функция может использоваться как составная часть выражения или в качестве аргумента другой функции.
  - Входные аргументы функции заключаются в круглые скобки и следуют после имени функции.
  - Если функция имеет несколько выходных аргументов, то они заключаются в квадратные скобки.
  - Для разделения входных аргументов используется запятая, для разделения выходных аргументов используется запятая или пробел.
  - Однострочным комментарием является любая последовательность символов, располагающаяся за знаком **%** и до конца текущей строки.
  - М-файлы могут содержать коды нескольких функций. Первая функция в файле — это первичная (основная) функция, вызываемая по имени М-файла. Другие функции внутри файла — это подфункции, которые являются видимыми только в пределах данного файла.
  - Если имя файла и имя функции не совпадают, то внутреннее имя функции игнорируется. Вызов функции происходит по имени файла.
  - Частные функции размещаются в подкаталогах со специальным именем **private**, видимы только для ограниченной группы функций, находящихся в каталоге предыдущего уровня.

### Переменные и их область действия

Область действия (**scope rules**) переменной — это правила, которые устанавливают, какие данные доступны из данного места программы. С точки зрения области действия переменных различают три типа переменных.

- Локальные переменные — это переменные, объявленные внутри функции. Локальная переменная доступна только внутри функции, в которой она

объявлена, и существует до тех пор, пока выполняется функция. При выходе из функции эта переменная и ее значение теряются.

- Формальные параметры — это переменные, объявленные при описании функции как ее аргументы. Область действия формальных параметров — блок, являющийся телом функции.
- Глобальные переменные — это переменные, объявленные при помощи ключевого слова **global**. Обращение к глобальной переменной возможно из любой функции, где эта переменная объявлена глобальной.
- Устойчивые переменные (**persistent variables**) — это локальные переменные, сохраняющие значение между вызовами функции.

### Отладка М-функций

- Окно редактирования М-файлов является одновременно и окном встроенного отладчика `medit.exe`.
- Пункты меню и кнопки панели инструментов, позволяющие выполнять отладочные действия:
  - **Set/Clear BreakPoint F12**: поставить/убрать точку останова;
  - **Clear All BreakPoints**: убрать все точки останова;
  - **Step In F11**: выполнять построчно с заходом в вызываемые функции;
  - **Single Step F10**: выполнять построчно;
  - **Continue**: выполнять до следующей точки останова;
  - **Stop if Errors/Warnings...**: установить режимы отладки при возникновении определенных ситуаций;
  - **Quit Debugging**: завершить отладку.

## Тема № 5. Объектно-ориентированное программирование

Современное объектно-ориентированное программирование основано на возможности введения новых типов данных и определения операций для них. В MATLAB классы пользователя **User Classes** наследуют родительскому классу **struct**, т. е. все новые типы (классы) данных базируются на структурах.

### Правила создания класса

Последовательность действий при создании класса в среде MATLAB на примере класса **myclass**:

- Класс — это папка-контейнер — каталог с именем `...\@myclass`.



- Для работы с классом **myclass** текущий каталог **Current Directory** должен быть открыт на папке, содержащей папку-класс **@myclass**.
- Имя функции-конструктора класса должно совпадать с названием класса (функция **myclass**, описанная в М-файле **myclass.m**).
- Объект класса хранит свои данные в структуре, все поля которой являются скрытыми (**private**).
- Функции-методы класса реализуются в М-файлах. Функция-метод — это функция, один из аргументов которой — объект данного класса. Имя М-файла — имя действия, реализованного функцией.
- Скрытые частные (**private**) функции класса **myclass** размещаются в папке **@myclass\private**, не имеют в качестве входного аргумента объект класса, используются как вспомогательные функции и не оперируют непосредственно с объектом класса.
- Базовые методы классов MATLAB:
  - **class constructor**: создание объекта класса;
  - **double, char**: конверторы;
  - **display**: вывод содержимого объекта в командное окно, если выражение не завершается разделителем точкой с запятой;
  - **set** и **get**: доступ к свойствам объекта (функция **get** — чтение значений свойств, а функция **set** — запись новых значений);
  - **subsref**: индексная ссылка (переопределение методов для операторов **a(i)**, **a{i}**, **a.field**);
  - **subsasgn**: индексное присваивание (переопределение методов для операторов **a(i)=b**, **a{i}=b**, **a.field=b**);
  - **subindex**: индексный дескриптор, т. е. использование объекта класса в качестве целочисленного индекса другого объекта (переопределение методов для **x(a)**);
  - **end**: последний индекс по указанной размерности.
- Переопределение арифметических функций и операторов: функции **plus**, **minus**, **mtimes**, ...
- Команда **>> clear classes** вызывается после каждого переопределения класса для удаления объектов класса.
- Встроенные функции MATLAB для идентификации объектов классов:
  - **class(p)**: определение класса объекта;
  - **isa(p, 'myclass')**: проверка принадлежности объекта данному классу;

- **isobject(p)**: выявление принадлежности объекта к какому-либо классу MATLAB;
- **methods('myclass')**: вывод списка методов данного класса;
- **whos p**: вывод подробной информации об объекте.

## Тема № 6. Высокоуровневая графика

**Высокоуровневая графика (High-Level Graphics)** включает команды и функции, предназначенные для построения графиков в прямоугольных и полярных системах координат, трехмерных поверхностей и линий уровня, гистограмм, диаграмм, анимации и т. д. Графические команды высокого уровня автоматически контролируют масштаб, выбор цветов, маркировку осей.

- Все графические изображения строятся в специальном графическом окне **Figure**.
- Создание нового окна: пункт меню **File | New ► New Figure** или команда **figure**.
- Активизация окна с номером **N** или создание окна с указанным номером: **figure(N)**.
- Переключение между окнами: **Alt+Tab**.
- Построение нескольких графиков:
  - в одной системе координат: **hold on**;
  - одном окне в разных системах координат: **subplot**;
  - разных графических окнах: **figure**.
- Удаление графика: **clf**, **clf reset**, **cla**, **cla reset**.
- Закрытие графического окна: **close(FigureN)**, **close all**.

### Двумерная графика

- График функции одной переменной, векторных и матричных данных в линейном масштабе в декартовой системе координат: **plot**.
  - **plot(x,y)** — график с узлами  $(x_k, y_k)$ ;
  - **plot(y)** — график с узлами  $(k, y_k)$ ;
    - дополнительный строковый аргумент задает:
    - тип маркеров: **.**, **+**, **\***, **o**, **x**, **s**, **d**, **v**, **^**, **<**, **>**, **p**, **h**;
    - стиль линии: **—**, **:**, **—.**, **--**;
    - цвет линии: **y**, **m**, **c**, **r**, **g**, **b**, **w**, **k**;

- **plot(x1,y1,x2,y2,...)** или **plot(x,[y1;y2;y3,...])** — построение нескольких графиков в одной системе координат;
- График в полулогарифмическом или логарифмическом масштабах: **semilogx** (логарифмический масштаб по оси **x**), **semilogy** (логарифмический масштаб по оси **y**) и **loglog** (логарифмический масштаб по обеим осям).
- Справка: **help graph2d, help specgraph**.

### Трехмерная графика

- График функции двух переменных в линейном масштабе в декартовой системе координат: **plot3**.
  - **plot3(x,y,z)** — график с узлами (**x<sub>k</sub>**, **y<sub>k</sub>**, **z<sub>k</sub>**);
  - дополнительный строковый аргумент задает:
    - тип маркеров: **.**, **+**, **\***, **o**, **x**, **s**, **d**, **v**, **^**, **<**, **>**, **p**, **h**;
    - стиль линии: **—**, **:**, **—.**, **--**;
    - цвет линии: **y**, **m**, **c**, **r**, **g**, **b**, **w**, **k**;
  - **plot3(x1,y1,z1,x2,y2,z2,...)** — построение нескольких графиков в одной системе координат;
  - справка: **>> help plot3**.
- График, заданный параметрически: **ezplot3**.
- Поверхность, заданная параметрически: **ezmesh, ezsurf**.
- Поверхность: **mesh, surf, surf1, cylinder, sphere**; вспомогательные функции: **meshgrid, ndgrid**.
- Каркасные поверхности: **mesh, meshc, meshz, peaks, surf, surfc, waterfall**; вспомогательные функции: **hidden**.
- Формирование, построение и маркировка линий уровня: **contour, contour3, contourc, contourf, clabel**.
- Триангуляционная поверхность: **trisurf, trimesh**.
- Диаграммы и гистограммы: **bar3, bar3h, hist3, pie3, stem3**.
- Закрашенный многоугольник в трехмерном пространстве: **fill3**.
- График функции в указанном интервале: **fplot, ezplot**.
- Анимированный график трехмерной линии: **comet3, ezplot3**.
- Закрашивание поверхности: **shading, brighten, colormap, colorbar**.
- Справка: **help graph3d, help specgraph**.

## Оформление графиков функций

Для максимального восприятия построенные графики функций оформляются: цветом, толщиной и типом линий, маркерами в дискретных вычисляемых точках, а также различными надписями в пределах графического окна и настройками осей системы координат графического окна.

## Управление видом графика

- **axis([xmin xmax ymin ymax zmin zmax]), axis([xmin xmax ymin ymax]):** установка границ осей координат;
- **axis on, axis off:** восстановление/отключение изображения осей и насечек;
- **axis('ij'), axis('xy'):** матричная/декартова система координат;
- **axis square, axis normal, axis equal, axis tight, axis auto, axis manual:** режимы масштабирования осей;
- **daspect:** изменение или получение масштаба осей;
- **pbaspect:** установка или определение соотношения длин осей;
- **grid on, grid off:** восстановление/отключение изображения координатной сетки;
- **box on, box off:** восстановление/отключение изображения системы координат в box-формате;
- **view([az,el]), view([x,y,z]), view(2), view(3):** установка или определение точки обзора (положения камеры);
- **camdolly, camlookat, camorbit, campan, campos, camproj, camroll, camtarget, camup, camzoom:** управление камерой.

## Изменение свойств линии

- Дополнительный строковый аргумент функций **plot, plot3** задает цвет и стиль линии и тип маркеров:
  - тип маркеров: **.**, **+**, **\***, **o**, **x**, **s**, **d**, **v**, **^**, **<**, **>**, **p**, **h**;
  - стиль линии: **—**, **:**, **—.**, **--**;
  - цвет линии: **y**, **m**, **c**, **r**, **g**, **b**, **w**, **k**;
  - в строковом аргументе можно указывать один, два или три символа;
  - порядок, в котором указываются символы, не существен.
- Изменение толщины линии: изменение свойства **'LineWidth'** графического объекта (дескрипторная графика).

## Надписи на графиках

- **clabel**: подписи к линиям уровня на контурных графиках;
- **gtext**: вывод текста в интерактивно определяемое место на графике (указать место мышкой и щелкнуть);
- **legend**: идентификация графиков (легенда);
- **title**: вывод заголовка;
- **text**: вывод текста в заданную позицию графика;
- **xlabel, ylabel, zlabel**: маркировка осей;
- **colorbar**: цветовая легенда;
- **grid on/off**: координатная сетка.

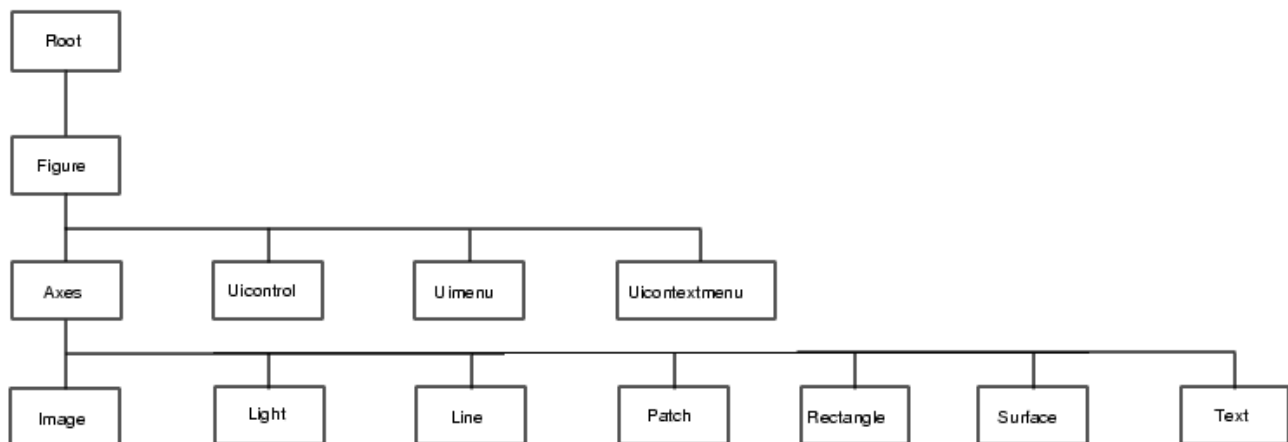
## Тема № 7. Дескрипторная графика

**Дескрипторная графика (Handle Graphics)** — это объектно ориентированная графическая система, основанная на низкоуровневых графических функциях. Объектами дескрипторной графики являются базовые графические элементы, используемые для визуализации данных. Каждый графический объект пронумерован, ему присваивается уникальный идентификатор — дескриптор (*handle*), используя который можно управлять свойствами графических объектов при помощи соответствующих функций. Дескрипторная графика позволяет разрабатывать приложения, оснащенные графическим интерфейсом пользователя.

## Иерархия графических объектов

- Древовидная иерархическая структура графических объектов.
- Каждый тип объекта обладает своими собственными свойствами.
- Каждому объекту ставится в соответствие уникальное число-дескриптор (*Handle*).
- Типы графических объектов:
  - **root** — вершина иерархии, соответствующая экрану компьютера (*Handle 0*);
  - **figure** — окно для вывода графики и пользовательских интерфейсов (*Handle* — номер графического окна, натуральное число);
  - **axes** — оси (система координат) в графическом окне;
  - **uicontrol** — управляющий элемент GUI;
  - **uimenu** — оконное меню пользователя;

- **uicontextmenu** — контекстное меню пользователя, появляющееся при щелчке правой кнопки мыши на объекте;
- **image** — двумерное растровое изображение (2D-объект);
- **light** — источник освещения, воздействующий на графические объекты типа **patch** и **surface** (3D-объект);
- **line** — плоская или пространственная линия (3D-объект);
- **patch** — замкнутый покрашенный полигон с ребрами (3D-объект);
- **rectangle** — двумерная фигура, изменяющаяся от прямоугольника до эллипса (2D-объект);
- **surface** — трехмерная поверхность (3D-объект);
- **text** — символьная строка (3D-объект).



### Дескрипторы графических объектов

- При помощи дескриптора **Handle** можно изменять свойства графического объекта;
- **gcf**: дескриптор активного графического окна **Figure**;
- **gca**: дескриптор активной системы координат **Axes**;
- **allchild(H)**: дескрипторы всех графических объектов, подчиненных объекту с дескриптором **H**;
- **get**: показать свойства объекта:
  - **get(H)**: вывод списка текущих значений свойств графического объекта с дескриптором **H**;
  - **get(H, 'PropertyName')**: вывод текущего значения указанного свойства объекта;

- **set**: установить свойства объекта или показать возможные значения отдельного свойства:
  - **set(H)**: вывод списка возможных значений всех свойств объекта;
  - **set(H, 'PropertyName')**: вывод списка возможных значений указанного свойства объекта;
  - **set(H, 'PropertyName', 'PropertyValue')**: установка нового значения указанного свойства объекта;
- **set(gcf)**: вывод списка возможных значений всех свойств текущего графического окна **Figure**;
- **set(gca)**: вывод списка возможных значений всех свойств текущей системы координат **Axes**;
- **propedit(H)**: вызов Редактора Свойств **Property Editor** для редактирования свойств графического объекта с дескриптором **H**;
- **reset(H)**: восстановление стандартных значений свойств по умолчанию графического объекта с дескриптором **H**.

### Свойства по умолчанию графических объектов

- Возможно изменение значения свойств, установленных по умолчанию, на уровне родительских объектов;
- **set(0, 'DefaultObjectProperty', PropertyValue)**: изменение заданного по умолчанию свойства объекта на уровне родительского объекта **Root** (все вновь создаваемые объекты выбранного типа будут иметь новое значение по умолчанию указанного свойства);
- **set(gcf, 'DefaultObjectProperty', PropertyValue)**: изменение заданного по умолчанию свойства объекта на уровне родительского объекта **Figure** (для всех вновь создаваемых в текущем графическом окне **Figure** объектов выбранного типа);
- **set(gca, 'DefaultObjectProperty', PropertyValue)**: изменение заданного по умолчанию свойства объекта на уровне родительского объекта **Axes** (для всех вновь создаваемых в текущей системе координат **Axes** объектов выбранного типа);
- **get(0, 'default')**: возвращает все измененные значения свойств по умолчанию;
- **get(gcf, 'default')**: возвращает все измененные значения свойств по умолчанию для текущего графического окна;
- **get(gca, 'default')**: возвращает все измененные значения свойств по умолчанию для текущей системы координат;

- **set(0, 'DefaultObjectProperty', 'remove')**: восстанавливает значение свойства по умолчанию на уровне объекта **Root**;
- **set(gcf, 'DefaultObjectProperty', 'remove')**: восстанавливает системное значение свойства по умолчанию на уровне текущего графического окна;
- **set(gca, 'DefaultObjectProperty', 'remove')**: восстанавливает системное значение свойства по умолчанию на уровне текущей системы координат.

## Тема № 8. Импорт и экспорт данных

### Чтение/запись файлов разных форматов

- Протокол-файл: **diary**.
- Операции над именем файлов: **fileparts, filesep, fullfile, tempdir, tempname**.
- Перемещение данных между файлами и рабочим пространством **WorkSpace**: **importdata, load, save, open, winopen**.
- Низкоуровневый файловый ввод/вывод (Low-Level File I/O): **fopen, fclose, fwrite, fprintf, fread, fscanf**.
- Форматированный или с разделителями ввод/вывод в текстовый файл: **csvread, csvwrite, dlmread, dlmwrite, textread, textscan**.
- Документы XML: **xmlread, xmlwrite, xslt**.
- Электронные таблицы (Excel и Lotus 123 файлы): **xlsinfo, xlsread, xlswrite, wk1info, wk1read, wk1write**.
- Научные данные CDF, FITS, HDF форматов: **cdfinfo, cdfepoch, cdfread, cdfwrite, fitsinfo, fitsread, hdf, hdfinfo, hdfread, hdf5write**.
- Аудио-, видеофайлы: **audioplayer, audiorecorder, beep, lin2mu, mu2lin, sound, soundsc, auread, auwrite, wavplay, wavread, wavrecorder, wavwrite, addframe, aviinfo, aviread, movie2avi**.
- Графические файлы: **im2java, imfinfo, imread, imwrite**.
- Справка: **help iofun**.



## Сохранение данных рабочего пространства

- Специальный формат MAT-файла, позволяющий сохранять данные из рабочего пространства **Workspace** в файлах на диске и считывать из файла в рабочее пространство.
- Сохранение данных из рабочего пространства **Workspace** в файле: **save** **<FileName>** **<VariablesList>**:
  - файл получает расширение **.mat**;
  - **save** сохраняет значения всех переменных рабочего пространства в MAT-файл **MATLAB.mat**;
  - **save file** сохраняет значения всех переменных рабочего пространства в файл **file**. Если у файла **file** не указано расширение, то данные запишутся в двоичный файл **file.mat**;
  - **save file X Y Z...** сохраняет указанные переменные.
- **load** — загрузка данных из файла в рабочее пространство:
  - **load** загружает все переменные из MAT-файла **MATLAB.mat**;
  - **load file X Y Z** загружает указанные переменные;
  - **load file** загружает все переменные из файла **file**. Если у **file** нет расширения, то функция ищет файл с именем **file** или **file.mat** и рассматривает его как двоичный MAT-файл;
  - **S = load(...)** возвращает содержимое MAT-файла в переменной **S**. Если файл является MAT-файлом, то **S** — структура, содержащая поля-переменные.
- **importdata** — загрузка данных из файла в рабочее пространство **Workspace**:
  - **importdata('file')** загружает данные из файла **file** в рабочее пространство;
  - **A = importdata('file')** загружает данные из файла **file** в переменную **A**;
  - **A = importdata('file','delimiter')** загружает данные из файла **file**, используя **delimiter** как разделитель столбцов (в случае текстового файла).

## Низкоуровневый файловый ввод-вывод

- Открыть файл и получить указатель файла (**file identifier**) — функция **fopen**.
- Выполнить операции с данными файла:
  - чтение, запись бинарных данных — функции **fread**, **fwrite**;

- чтение текстовых строк из файла — функции **fgets/fgetl**;
- чтение форматированных ASCII данных — функция **fscanf**;
- запись форматированных ASCII данных — функция **fprintf**.
- Закрывать файл — функция **fclose**.
- **fopen** — открытие файла и получение информации о файлах (выходной параметр — файловый идентификатор, который используют все остальные функции низкоуровневого файлового ввода-вывода):
  - **fid = fopen (filename)** открывает файл **filename** для чтения и возвращает файловый идентификатор **fid**;
  - **[fid,message] = fopen (filename, permission)** открывает файл **filename** в режиме, указанном в разрешении **permission**. В **message** содержится сообщение о системной ошибке, если файл не открыт.
- Режимы разрешения **permission**:
  - **'r'**: открыть файл для чтения (по умолчанию);
  - **'w'**: открыть файл или создать новый файл для записи, уничтожая старые данные;
  - **'a'**: открыть файл или создать новый файл для записи, добавляя данные в конец файла;
  - **'r+'**: открыть файл для чтения и записи;
  - **'w+'**: открыть файл или создать новый файл для чтения и записи, уничтожая старые данные;
  - **'a+'**: открыть файл или создать новый файл для чтения и записи, добавляя данные в конец файла;
  - **'t'**: открыть текстовый файл;
  - **'b'**: открыть бинарный файл.
- **fclose** — закрытие файлов:
  - **status = fclose(fid)** закрывает указанный файл, если он открыт, возвращая 0 в случае успеха и —1 в противном случае;
  - **status = fclose('all')** закрывает все открытые файлы, кроме файлов с идентификатором 0, 1 или 2.
- **ftell** — получение текущей позиции в файле. Вызывается:
- **position = ftell(fid)** возвращает текущую позицию.
- **fseek** — переустановка текущей позиции в файле. Вызывается:
- **status = fseek(fid, offset, origin)** перемещает текущую позицию в файле на **offset** байтов относительно **origin**. Допустимые значения аргумента **origin**:

- **'bof'** или **—1**: от начала файла;
  - **'cof'** или **0**: от текущей позиции в файле;
  - **'eof'** или **1**: от конца файла.
- Допустимые значения аргумента **offset**:
  - **offset > 0** — передвижение к концу файла;
  - **offset = 0** — текущая позиция не изменяется;
  - **offset < 0** — передвижение к началу файла.
- **feof** — проверка достижения конца файла. Вызывается:
- **eofstat = feof (fid)** возвращает 1, если указатель указывает на конец файла, и 0 в других случаях.
- **frewind** — переход на начало файла.
- **frewind (fid)** устанавливает текущую позицию файла с идентификатором **fid** на начало файла.
- **fread** — чтение двоичных данных из файла.
- **fwrite** — запись двоичных данных в файл.
- Функции для чтения и записи информации в текстовый файл:
  - **A = fgetl (fid)** — чтение следующей строки текстового файла, включая символ перевода строки;
  - **A = fgets (fid)** — чтение следующей строки текстового файла без символа перевода строки;
  - **[A, count] = fscanf (fid, format, size)** — чтение данных из текстового файла, записанных в определенном формате;
  - **count = fprintf (fid, format, A, ...)** — форматный вывод в текстовый файл. Если идентификатор файла равен 1 или отсутствует, то вывод идет на экран;
  - **A = sscanf(s,format,size)** — чтение форматированных данных из строки (а не из текстового файла);
  - **[s,errmsg] = sprintf(format,A,...)** — запись форматированных данных в строку (не в текстовый файл).
- Форматные спецификации для чтения и записи текстовых данных:
  - **%<+,-,0><Number>.<Number>ConversionCharacter**.
- **%**: признак начала форматной спецификации.
- Значения управляющего символа формата данных:
  - **-**: выравнивание по левому краю;
  - **+**: обязательное написание знака числа (+ или -);
  - **0**: заполнение недостающих значений поля нулями.
- Значения управляющего символа форматной спецификации:

- **%c**: одиночный символ;
- **%d**: десятичное представление (знаковое);
- **%f**: представление с фиксированной точкой (вещественное);
- **%e %E**: экспоненциальное представление;
- **%g %G**: более компактное представление **%f** или **%e**;
- **%i**: десятичное представление (знаковое);
- **%o**: восьмеричное представление (беззнаковое);
- **%s**: строка символов;
- **%u**: десятичное представление (беззнаковое);
- **%x %X**: шестнадцатеричное представление.

### Работа с графическими файлами

- **image**: отображение графического изображения;
- **imread**: чтение графических данных из файла в массив;
- **imwrite**: запись графических данных из матрицы в файл;
- **imfinfo**: получение информации о графическом файле и содержащемся в нем изображении.

## Тема № 9. Графический интерфейс пользователя

Под графическим интерфейсом пользователя (**Graphical User Interface – GUI**) подразумевается тип экранного представления, при котором пользователь может выбирать команды, запускать задачи, вводить и просматривать исходные данные, обрабатывать полученные результаты, используя элементы управления, представленные на экране.

### Управляющие элементы

- Окно графического интерфейса пользователя приложения, созданного в среде MATLAB, – одно или несколько графических окон **Figures**.
- Объекты **uicontrol**, **uimenu**, **uicontextmenu** являются потомками объекта **Figure** и, следовательно, не зависят от **Axes**.
- Выпадающее меню: **uimenu**.
- Контекстное меню: **uicontextmenu**.
- Управляющие элементы: **uicontrol**. Типы, имеющиеся в распоряжении (определяются свойством **Style**).

### Основные свойства управляющих элементов

- Управляющий элемент **uicontrol**:

- **String**: надпись на элементе, а также список строк компонентов **ListBox** и **PopupMenu**;
- **Visible**: видимость элемента управления;
- **Enable**: доступность элемента управления;
- **Style**: тип элемента управления;
- **Position**: положение и размеры управляющего элемента;
- **Units**: единицы измерения;
- **FontAngle**, **FontName**, **FontSize**, **FontUnits**, **FontWeight**, **HorizontalAlignment**: настройки шрифта;
- **Tag**: имя элемента;
- **Callback**: управляющее действие;
- **CreateFcn**: действия, выполняемые при создании элемента;
- **DeleteFcn**: действия, выполняемые при удалении элемента.
- Управляющий элемент **uimenu**:
  - **Label**: метка меню;
  - **Separator**: разделительная линия;
  - **Visible**: видимость;
  - **Enable**: доступность;
  - **Callback**: управляющее действие.
- Управляющий элемент **uicontextmenu**:
  - **Callback**: управляющее действие.
- Доступ к свойствам управляющих элементов: **get** и **set**.

## Среда GUIDE

- Визуальная реализация графического интерфейса пользователя GUI: среда **GUIDE**.
- Команда **>> guide**: запуск графического редактора **Layout Editor**, содержащий палитру графических элементов управления и форму для создания GUI.
- Файлы: файл графического макета (**.fig**), файл интерфейса (**.m**).

## Тема № 10. Вычисления в MATLAB

Пакет MATLAB обладает целым рядом встроенных функций, реализующих различные численные методы для нахождения корней уравнений, дифференцирования и интегрирования, приближения табличных функций, решения задач линейной алгебры, решения систем дифференциальных уравнений и т. д.

## Исследование функций

### Решение уравнений

- **fzero** — поиск корней произвольных уравнений;
- **roots** — поиск корней полиномов;
- **fplot** — построение графика исследуемой функции в определенном диапазоне.

### Нахождение экстремумов функции

- **fminbnd** — поиск локального минимума функции одной переменной;
- **fminsearch** — поиск локального минимума функции нескольких переменных.

### Интерполирование

- **polyfit** — интерполяция полиномами по методу наименьших квадратов функции одной переменной;
- **interp1** — интерполяция сплайнами функции одной переменной;
- **interp2** — интерполяция сплайнами функции нескольких переменных.

### Решение систем линейных уравнений

- **x=A\b** — решение системы ЛУ, где A — матрицы коэффициентов при неизвестных, b — столбец свободных членов;
- **linsolve** — решение системы ЛУ;
- **solve** — символьное решение алгебраических уравнений и систем;
- **det** — вычисление определителя матрицы;
- **inv** — обращение матрицы;
- **eig** — вычисление собственных значений и собственных векторов матрицы.

## ПЛАН ЛАБОРАТОРНЫХ ЗАНЯТИЙ

### Лабораторная работа № 1

#### Задание 1:

Запустите **MATLAB**. Модифицируйте расположение всех открытых окон интерфейса пользователя **MATLAB**. Последовательно закройте все окна **MATLAB Desktop**, кроме командного окна **Command Window**. Восстановите настройки по умолчанию интерфейса **MATLAB Desktop**, выбрав пункт меню **Desktop | Desktop Layout ► Default**.

#### Задание 2:

В командной строке введите команду **>> help plot**. Изучите способы построения графиков функций. Постройте график функции:

$$\text{a) } y = \frac{\sqrt{3}}{4} \sin(3x) \cos(2x); \quad \text{b) } y = \frac{\sin(3x - 5\pi/11) - \sqrt{8} - 3\pi}{e^{-2} + 4};$$

где переменная **x** изменяется в пределах от  $-3\pi$  до  $3\pi$  с шагом 0.01.

#### Задание 3:

Постройте графики функций **2sin(-5x)**, **3cos(0.2x)** в одной системе координат. Переменная **x** изменяется от  $-6\pi$  до  $6\pi$  с шагом 0.01.

#### Задание 4:

Напишите анонимную функцию, вычисляющую:

- a) объем цилиндра;
- b) площадь поверхности цилиндра;
- c) площадь поверхности правильного тетраэдра;
- d) площадь треугольника по формуле Герона.

#### Задание 5 (полиномы Чебышева):

Постройте графики шести функций, соответствующих полиномам Чебышева, в одной системе координат. Значение аргумента изменяется в пределах от  $-1$  до  $1$  с шагом 0.01. Полиномы генерируйте в цикле. Определите,

как будут храниться значения полиномов. Рекурсивная формула определения полиномов имеет вид

$$T_n(x) = 2x \cdot T_{n-1}(x) - T_{n-2}(x); \quad T_1(x) = x; \quad T_0(x) = 1;$$

## Лабораторная работа № 2

### Задание 1:

Запустите MATLAB, введите команду `>> help datatypes` и изучите встроенные типы данных, представленные в MATLAB.

### Задание 2:

Формат представления данных определяется командой `format`. Команда `>> help format` выводит список возможных форматов представления данных.

Создайте матрицу `A = [20*rand(2,3)-7; 0 Inf NaN]` и выведите ее в командное окно в разных форматах.

### Задание 3:

Вычислите значения:

$$\text{a) } \frac{\sqrt{15}-5\pi}{e^2+6}; \quad \text{b) } \frac{\sin(5\pi/11)-\sqrt{8}-3\pi}{e^{-2}+4}; \quad \text{c) } \frac{(\sqrt{7}-5\pi)^2}{e^4+e^3};$$

### Задание 4:

Вычислите значения  $e^{1+2i}$ ,  $(1+2i)^e$ ,  $(1+2i)^{(1+2i)}$ .

### Задание 5:

Создайте вектор-строку  $1 \times 4$  `x=(1+2i 2-j 3 4)`. Введите команды и определите, в чем состоит их отличие?

```
>> y=x'
>> z=x.'
```

Переведите вектор-строку `x` в вектор-строку `X=(1-2i 2+i 3 4)`.



**Задание 6:**

Создайте матрицу **A** как матрицу 4×3  $A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & -5 & 6 \\ 7 & 18 & -9 \\ -10 & 11 & 12 \end{pmatrix}$ .

Выполните над матрицей **A** следующие действия (каждое действие выполняется над исходной матрицей **A**):

- задайте два способа доступа к элементу 7;
- задайте три способа доступа к последнему элементу матрицы;
- выделите третью строку матрицы;
- переставьте в матрице первый и третий столбцы;
- измените порядок следования строк матрицы;
- найдите минимальное значение в каждом столбце;
- найдите максимальное значение каждой строки;
- найдите минимальное и максимальное значение матрицы;
- транспонируйте матрицу в вектор-столбец;
- транспонируйте матрицу в вектор-строку;
- выберите из матрицы все элементы с нечетными порядковыми номерами;
- все отрицательные элементы матрицы замените на 0;
- все элементы матрицы с порядковыми номерами, кратными 3, замените на 33;
- удалите второй столбец;
- удалите третью строку;
- удалите число 18.

**Задание 7:**

Последовательно введите команды, выполняющие арифметические действия над матрицами, и изучите результаты их выполнения.

```
>> A = [1; 2; 3; 4; 5; 6; 7; 8; 9; 10; 11; 12]
>> B = A'
>> C = A.'
>> G = C^2
>> D = -A
>> B = A + A;
>> C = A' * A;
```

```
>> D = B - A;  
>> a = [1:4];  
>> E = a' * a;  
>> f = a * a';  
>> b = 2;  
>> d = A/b;  
>> B = A.^2;  
>> B = A./A;  
>> C = A.*A;  
>> b=diag(A)  
>> B=diag(b)
```

### Задание 8 (игра «Жизнь» — Conway's Game of Life):

Задан торический мир  $X$  размером  $M \times M$  клеток ( $M = 101$ ), первоначально заселенный живыми организмами с вероятностью 10 %. В каждой клетке может находиться ровно один живой организм. В каждом следующем поколении: организм зарождается в пустой клетке, вокруг которой собрались ровно 3 организма, или организм погибает:

- от ТОСКИ, если у него меньше двух соседей;
- ГОЛОДА, если у него больше трех соседей.

Смена поколений равна 100. Каждое поколение отображать графически. Прервать надоевшую эпоху можно, нажав клавиши **Ctrl-C**.

## Лабораторная работа № 3

### Задание 1:

Чему равно наибольшее число, для которого можно вычислить факториал в пакете MATLAB? (Помощь: **Inf**, **isinf(n)**)

### Задание 2:

Организуйте цикл, в котором вводите с клавиатуры название дня недели, и при помощи оператора **switch** определите номер дня недели по названию дня недели. Выход из цикла — по вводу текстовой строки «exit». Обработайте некорректный ввод названия дня недели.

**Задание 3:**

Последовательность чисел Фибоначчи имеет следующий вид:

$$n_k = n_{k-1} + n_{k-2}, \quad n_1 = n_2 = 1.$$

Вычислите следующие 10 элементов этой последовательности при помощи оператора цикла. Начните с вектора **[1 1]**. На каждом шаге цикла добавляйте к этому вектору один элемент. Какой элемент будет превышать значение  $10^{20}$ ? Для этого организуйте цикл с подходящим условием. Последовательность Фибоначчи может быть представлена формулой

$$n_k = \frac{F^k - (1-F)^k}{\sqrt{5}}, \text{ где } F = \frac{1+\sqrt{5}}{2}.$$

Вычислите значения с 12-го по 20-й элемент этой последовательности, не используя для организации вычисления циклы.

**Задание 4:**

3D-графика. Введите команды

```
>> a=linspace(1,10,100);
>> [x y]=meshgrid(a);
>> z=sin(x).*cos(y);
>> surf(x,y,z)
```

Теперь введите команды

```
>> colormap winter % default, hot, cool, spring,
    % summer, autumn, gray, pink...
>> shading flat % interp, faceted
```

Что изменилось в графическом окне **Figure**?

**Задание 5:**

Оформление графиков функций.

```
>> X=linspace(0,2*pi); Y=[sin(X);cos(X)];
plot(X,Y)
title('Trigonometrische Funktionen')
xlabel('Winkel \alpha');
text(pi/4,sin(pi/4),' sin \pi/4 = cos \pi/4')
```

```
axis equal
axis tight
grid on
legend(['sin \alpha'; 'cos \alpha'], -1)
```

**Задание 6:**

Построение графиков в разных системах координат в одном графическом окне. Введите команды

```
>> [X,Y]=meshgrid(-1:.05:1);
Z=X.^3-3*X.*Y.^2;
surf(X,Y,Z)
axis equal
axis tight
box on
```

Теперь введите следующие команды:

```
>> ind=Z>-0.1 & Z<0.1;
Z(ind)=NaN;
subplot(1,2,1)
surf(X,Y,Z)
```

Что получится в результате выполнения команды

```
>> subplot(1,2,1) ?
```

Теперь введите команды

```
>> subplot(1,2,2)
contour(X,Y,Z,20);
axis equal
axis tight
colorbar('horiz');
```

Изучите работу функции **subplot**.

**Задание 7 («Томограмма». 3D-графика и анимация):**

Построить график заданной функции двух переменных и просканировать (снизу вверх) ее горизонтальные уровни. Для этого разбить графическое окно **Figure** на две системы координат:

- в левой системе координат изобразить саму поверхность (3D-поверхность);
- в правой системе координат нарисовать карту ее произвольного горизонтального сечения (2D-изображение).

В цикле для любого уровня  $z=z_0$ :

- из поверхности ВЫРЕЗАТЬ данный уровень;
- справа изобразить карту данного уровня, то есть местоположение вырезанных ячеек.

## Лабораторная работа № 4

### Задание 1:

Напишите рекурсивную функцию, высчитывающую факториал заданного числа.

### Задание 2:

Напишите функцию, высчитывающую первые  $n$  полиномов Чебышева. Первая строка М-функции должна иметь вид

**function T = tscheby(x,n)**

Полиномы рассчитываются по формуле

$$T_0(x) = 1; \quad T_1(x) = x; \quad T_n(x) = 2x * T_{n-1}(x) - T_{n-2}(x);$$

Постройте графики функций в интервале  $[-1,1]$ .

### Задание 3:

Напишите М-функцию вида:

$$\text{a) } f(x) = \begin{cases} x, & x < 0 \\ x^2, & 0 \leq x < 2; \\ 4, & x \geq 2 \end{cases} \quad \text{b) } f(x) = \begin{cases} 1, & |x| \leq 1 \\ x^2, & 1 < |x| < 2; \\ 0, & |x| \geq 2 \end{cases} \quad \text{c) } f(x) = \begin{cases} 0, & x < 0 \\ \sin\left(\frac{\pi * x}{2}\right), & 0 \leq x \leq 1. \\ 1, & x > 1 \end{cases}$$

Протестируйте функцию для значений  $x = [-2, 1/3, 3/2, 2, 6]$ . Постройте график функции в интервале  $[-3,3]$ .

**Задание 4:**

Напишите рекурсивную функцию, высчитывающую последовательность чисел Фибоначчи:

$$n_k = n_{k-1} + n_{k-2}, \quad n_1 = n_2 = 1.$$

**Задание 5 («Скользящий график»):**

Напишите функцию, отображающую скользящий график заданной функции. Функция должна иметь следующий заголовок и подсказку:

```
function KM4(dx,y)
% KM4 -- Floating Plot
% Graphics Initialization:
% km4('reset',[xmin xmax]) - reset plot area
% km4('reset')-reset plot area ([xmin xmax]=[0 100])
% Input the next point of graphics:
% km4(dx,y) - add new point to current plot
% km4(y) - add new point to current plot (step dx=1)
% EXAMPLE:
% >> km4('reset',[0 80])
% >> for x=0:0.01:10*pi;...
% km4(0.1,sin(20*x)+sin(23*x)); pause(0.01); end
```

Функция должна выводить график указанной функции. Количество точек вдоль оси **X** фиксировано. Если номер текущей отображаемой точки превышает заданное количество точек, то график сдвигается влево вдоль оси **X** («скользит»), так чтобы количество отображаемых точек вдоль оси **X** оставалось постоянным. Диапазон значений вдоль оси **Y** корректируется программно таким образом, чтобы весь график попадал в область вывода.

**Лабораторная работа № 5**

Создайте класс **polynom**. В классе должны быть реализованы методы, определяющие функциональное назначение данного класса.

**Задание 1:**

Для создания объектов класса **polynom** используется функция-конструктор. Конструктор должен находиться в М-файле с именем **@polynom/polynom.m**.

Данный конструктор создает полином из заданного вектора коэффициентов полинома при убывающих степенях переменной **x**. В примере применяются три типа конструктора.

- Конструктор по умолчанию. Создается шаблон объекта, обычно с пустыми полями. В отдельных случаях поля инициализируются некоторыми начальными значениями.
- Конструктор копирования. Конструктор возвращает копию объекта. При этом в конструкторе необходимо использовать функцию **isa(p, 'polynom')**, чтобы определить: является ли входной аргумент объектом данного класса?
- Конструктор с параметрами. Конструктор создает структуру и инициализирует ее поля, используя входные данные. Затем из этой структуры конструируется сам объект при помощи встроенной функции **class()**. У этой функции есть два обязательных параметра. Первый — структура, которая будет представлять данные объекта, а второй — текстовая строка, содержащая имя создаваемого класса.

```
function p = polynom(a)
%POLYNOM Polynomial class constructor.
% p = POLYNOM(v)
if nargin == 0
    p.c = [];          % default constructor
    p = class(p, 'polynom');
elseif isa(a, 'polynom') % copy-constructor
    p = a;
else
    p.c = a(:).';      % constructor with parameters
    p = class(p, 'polynom');
end
```

### Задание 2:

Для преобразования объекта одного класса к другому классу используется функция-конвертор. Имя этой функции и имя ее М-файла должно совпадать с именем класса, к которому она будет преобразовывать исходный объект. Для преобразования объекта класса **polynom** обратно в вектор его коэффициентов используется метод **double()**:

```
>> p = polynom([1 2 -4 5]);
```

```
>> double(p)
```

```
ans =  
1 2 -4 5
```

### Задание 3:

Для преобразования объекта к текстовому виду используется метод **char()**. В примере класса **polynom** этот метод определяется в М-файле **@polynom/char.m** и используется следующим образом:

```
>> p = polynom([1 2 -4 5]);
```

```
>> char(p)
```

```
ans =  
x^3 +2*x^2 - 4*x + 5
```

Преобразование к текстовому виду используется в методе **display()**, который относится к базовым методам классов MATLAB. Этот метод вызывается всякий раз, когда в среде MATLAB введено выражение, не завершенное точкой с запятой.

### Задание 4:

Для переопределения основных математических операторов необходимо в папке-контейнере, содержащей методы класса, создать М-файл с именем, соответствующим имени переопределяемого оператора, и в этом файле определить функцию с этим же именем. Так функция, определенная в файле **@polynom/plus.m**, складывает полиномы.

```
>> p = polynom([1 2 -1 3]);
```

```
q = polynom([4 -5]);
```

```
p1 = p + 1;
```

```
s = p + q;
```

### Задание 5:

Создайте класс-папку **@polynom**, определите структуру для хранения данных, реализуйте для класса **polynom** следующие методы:

- **char** — преобразование полинома к текстовому виду;
- **diff** — вычисление производной полинома;
- **display** — вывод в командное окно;



- **double** — функция-конвертор полинома в вектор его коэффициентов;
- **int** — вычисление неопределенного интеграла полинома **p**. **int(p)** возвращает интеграл с нулевым свободным членом, **int(p,c)** возвращает интеграл со свободным членом **c**;
- **minus** — вычитание полиномов;
- **mtimes** — умножение полиномов;
- **plot** — построение графика полинома **p**. Команда **plot(p)** выводит график полинома в диапазоне, содержащем все его корни. Команда **plot(p,x)** выводит график полинома в заданной области определения;
- **plus** — сложение полиномов;
- **polynom** — конструктор;
- **polyval** — вычисление значения полинома **p** в точке **x**;
- **roots** — вычисление всех корней полинома;
- **subsref** — операция индексной ссылки, реализующая вычисление значения полинома в указанных точках.

### Лабораторная работа № 6

#### Задание 1:

Построить график функции в полярных координатах:

a)  $y = \sin(2t) \cos(2t), t \in [0, 2\pi]$

b)  $r = 2 \cos(\varphi) \sin(20t), \varphi \in [-\pi/2, \pi/2]$

#### Задание 2:

Построить график функции:

a)  $f(x) = e^{\sin 0.1x}, x \in [-2\pi, 2\pi];$       b)  $f(x) = \sqrt{5} + 3^{|\cos x|}, x \in [-2\pi, 2\pi];$

c)  $f(x) = (1/3)^{|\sin x|} - 2, x \in [-5\pi, 5\pi];$       d)  $f(x) = 4^{\cos|x|}, x \in [-2\pi, 2\pi];$

#### Задание 3:

Решить графически уравнения:

a)  $3^x = 4 - x;$       b)  $(1/2)^x = x + 3;$       c)  $(1/3)^x = x + 1;$

d)  $4^x = 5 - x$ ;      e)  $3^{1-x} = 2x - 1$ ;      f)  $3^{-x} = -3/x$ .

**Задание 4:**

Построить график показательной функции  $f(x) = 10^{2x}$  на отрезке  $[-0.5, 5]$  в линейном, логарифмическом и полулогарифмическом масштабах в одном графическом окне в разных системах координат.

**Задание 5:**

Построить график функции `sin(2*t) .* cos(2*t)` в полярных координатах,  $t \in [0, 2\pi]$ .

**Задание 6:**

Построить график функции, заданной параметрически:

- a)  $x(t) = \sin(2t)$ ,  $y(t) = (1 - t^2) \cos(20t)$ ,  $z = t$ ,  $t \in [-5, 5]$
- b)  $x(t) = (1 + t^2) \sin(20t)$ ,  $y(t) = (1 - t^2) \cos(20t)$ ,  $t \in [-5, 5]$
- c)  $x(t) = \cos(3t) \cos(t)$ ,  $y(t) = \cos(3t) \sin(t)$ ,  $t \in [0, 2\pi]$  (график трохойды)
- d)  $x(u, v) = 0.3u \cos(v)$ ,  $y(u, v) = 0.3u \sin(v)$ ,  $z(u, v) = 0.6u$ ,  $u, v \in [-2\pi, 2\pi]$  (конус)

**Задание 7:**

Построить прозрачную каркасную поверхность фигуры, заданной соотношениями:

- a)  $x(u, v) = 0.3u \cos(v)$ ,  $y(u, v) = 0.3u \sin(v)$ ,  $z(u, v) = 0.6u$ ,  $u, v \in [-2\pi, 2\pi]$  (конус)
- b)  $x(u, v) = \cos(u)$ ,  $\cos(v)$ ,  $y(u, v) = 0.7 \cos(u) \sin(v)$ ,  $z(u, v) = 0.8 \sin(u)$ ,  $u, v \in [-2\pi, 2\pi]$  (эллипсоид)

на квадратной области определения.

**Задание 8:**

Построить график функции двух переменных на прямоугольной области определения:

- a)  $z(x, y) = 4 \sin(2\pi x) \cos(1.5\pi y)(y - 1)$ ,  $x \in [-1, 1]$ ,  $y \in [0, 1]$
- b)  $z(x, y) = -e^{-y^2} \cos(3\pi x)(1 - x)xy$ ,  $x \in [-1, 1]$ ,  $y \in [-2, 2]$
- c)  $z(x, y) = \sin(x) \sin(y)$ ,  $x \in [0, 100]$ ,  $y \in [0, 100]$

**Задание 9:**

Построить график баллистической траектории, заданной параметрически:  $\mathbf{x}(t) = \mathbf{v} t \cos(\varphi)$ ,  $y(t) = \mathbf{v} t \sin(\varphi) - g t^2/2$ . Начальная скорость  $\mathbf{v} = 100$  км/ч, угол вылета снаряда:  $\varphi = 60^\circ$ .

**Задание 10:**

Построить траекторию движения точки на плоскости или в пространстве в течение времени  $t$ :

a)  $x(t) = \sin(t) / (t+1)$ ,  $y(t) = \cos(t) / (t+1)$ ,  $t \in [0, 10]$

b)  $x(t) = e^{-|t-50|/50} \sin(t)$ ,  $y(t) = e^{-|t-50|/50} \cos(t)$ ,  $t \in [0, 100]$

**Задание 11:**

Создать два графических окна. Построить в одном окне графики функций  $\mathbf{f}(\mathbf{x}) = \sin \mathbf{x}$ ,  $\mathbf{g}(\mathbf{x}) = 0.5 \sin \mathbf{x}$ , в другом окне — график функции  $\mathbf{q}(\mathbf{x}) = \ln \mathbf{x}$ . Оформить графические окна: нанести заголовки, легенды, подписи к осям. Нанести координатную сетку на второй график. Аргументы изменяются в пределах  $[0, 10]$ .

## Лабораторная работа № 7

**Задание 1:**

Изучите свойства графических объектов и приведите примеры их использования.

**Задание 2:**

Графический объект **Rectangle** представляет двумерную закрашенную область, которая может принимать определенную форму от прямоугольника до эллипса. Получите дескриптор графического объекта **Rectangle**, выведите текущие и возможные значения его свойств. Постройте различные прямоугольники, изменяя значения их свойств.

**Задание 3:**

Создайте два графических окна **Figure** и циклически выведите графики  $\mathbf{f}_k(\mathbf{x}) = k \sin(\mathbf{x})$  и  $\mathbf{g}_k(\mathbf{x}) = k \cos(\mathbf{x})$ ,  $k=1, \dots, 6$ ,  $\mathbf{x} \in [-3\pi, 3\pi]$  поочередно в одно из окон.

**Задание 4:**

Введите команды, рисующие двадцать линий разных цветов сплошной линией толщиной 0.5 (свойства по умолчанию).

```
>> t = 0:pi/20:2*pi;  
a = ones(length(t),9);  
for i = 1:20 a(:,i) = sin(t-i/5)'; end  
plot(t,a)
```

Затем найдите все линии красного цвета и удалите их. После этого найдите все линии синего цвета и измените их толщину на 2. Найдите все линии какого-либо третьего цвета и измените их стиль на штрихпунктирный.

**Лабораторная работа № 8****Задание 1:**

Удалите из рабочего пространства **Workspace** все переменные

```
>> clear
```

Теперь введите команды

```
>> a = 5;  
>> b = [1 2 3; 4 5 6];  
>> c = magic(3);
```

Сохраните созданные переменные и их значения в файл **MATLAB.mat**

```
>> save
```

Вновь очистите рабочее пространство и введите команду

```
>> s = load
```

Что получится в результате?

**Задание 2:**

Создайте в любом текстовом редакторе текстовый файл `Weather.txt`. Для каждого из четырех городов: Минск, Мельбурн, Берлин, Токио введите в него следующую информацию: 1-я строка содержит название города, 2-я строка содержит среднемесячную температуру в течение года (двенадцать значений формата 17.56), третья строка содержит среднемесячное давление в течение года (двенадцать значений формата 767). В MATLAB создайте структуру `weather` подходящего формата и загрузите в нее данные из файла `Weather.txt`. В одном графическом окне в разных системах координат, расположенных одна под

одной, постройте: в первой системе координат — графики температур, в другой — диаграммы давлений. Снабдите графики всей необходимой информацией (заголовок, подписи к осям координат, легенды).

### Задание 3:

Для предыдущего задания создайте в MATLAB структуру `weather` подходящего формата и заполните ее необходимыми данными. После этого сохраните содержимое структуры `weather` в текстовом файле `Weather.txt`.

### Задание 4:

По отсканированным в интерактивном режиме трехмерным точкам построить многогранник. Для этого написать функцию, имеющую следующую структуру:

```
function KML08( act, filename )
% KML08 -- Input/Output
% KML08('Points','File') - enter 3D-points
% and save into the 'File'
% KML08('Patch','File') - plot surface from
% Pnt/Fct files
% 0. Диспетчер
% 1. Оцифровка 3D-точек на графике
% 1.1 Подготовка осей к вводу точек
% 1.2 Вывод на 3D-график номеров введенных точек
%     в отсканированные позиции
% 1.3. Сохранение точек в файл File.pnt
% 2. Инициализация осей для вывода графика
% 2.1. Чтение файла координат точек File.pnt
% 2.2. Чтение файла описания граней File.fct
% 2.3. Построение поверхности
```

Указания по заданию:

1. Сканирование точек выполните при помощи функции **ginput**. Для указания координаты **Z** (высоты) точки надо нажимать цифровую клавишу '0'–'9'. При этом координаты **X**, **Y** указываются перекрестием мыши непосредственно в осях.

2. Для вывода на 3D-график номеров введенных точек в отсканированные позиции изучите следующий код:  

```
x=[1 2 3]'; y=[1 8 4]'; z=[5 9 7]';  
c=sprintf('%d|',1:size(x,1)); c(end)=[];  
clf; plot3(x,y,z,'o'); hold on  
text(x,y,z,c,'Color','b','FontWeight','bold');  
grid on; view(3);
```
3. Отсканированные координаты сохраните в текстовый файл `File.pnt`. В файле `File.pnt` указываются координаты **X**, **Y**, **Z** вершин. Одна строка файла — одна вершина 3D-поверхности.
4. В файле `File.fct` описываются грани поверхности:
  - 1-я строка: количество вершин на грани;
  - 2-я строка: номера вершин 1-й грани; ... ;
  - N-я строка: номера вершин (N-1)-й грани.
5. Файлы `File.pnt` и `File.fct` записываются в текущий каталог.
6. Для построения 3D-поверхности примените функцию **patch**.

## Лабораторная работа № 9

### Задание 1:

Постройте графики функций  **$Y=\sin(X)$**  и  **$Z=\cos(X)$**  с шагом 0.01 на отрезке  $[-3\pi; 3\pi]$  в одном графическом окне в разных системах координат. Свяжите с графиком функции  **$Y=\sin(X)$**  контекстное меню с тремя пунктами, изменяющими тип линии: **dashed**, **dotted**, **solid**. Свяжите с графиком функции  **$Z=\cos(X)$**  контекстное меню с тремя пунктами, изменяющими цвет линии: **red**, **green**, **blue**.

### Задание 2:

Постройте графики функций  $z=\sqrt{x}\cos(x)$  на отрезке  $[0; 6\pi]$  и  $y=x^2$  на отрезке  $[-3\pi; 3\pi]$  с шагом 0.01 в одном графическом окне в разных системах координат. Свяжите с графиком функции  $y=x^2$  контекстное меню с двумя пунктами, устанавливающими толщину линии. Выбор первого пункта меню увеличивает толщину линии в два раза, выбор второго пункта меню — восстанавливает значение толщины линии по умолчанию. Свяжите с графиком функции  $z=\sqrt{x}\cos(x)$  контекстное меню с тремя пунктами, изменяющими цвет линии: **red**, **green**, **blue**.

**Задание 3:**

Постройте стандартную трехмерную поверхность — сферу. Свяжите с поверхностью контекстное меню. Первые три пункта контекстного меню изменяют стиль ребер поверхности: **dashed**, **dotted**, **solid**. Следующие четыре пункта меню изменяют цвет поверхности: **red**, **green**, **blue**, **flat**. Группы пунктов меню разделите разделительной линией.

**Задание 4:**

Постройте график функции, заданной параметрически. Свяжите с построенной фигурой контекстное меню с пунктами, изменяющими маркеры ее узлов: кружок, звездочка, треугольник, квадрат. Последний пункт меню удаляет маркеры.

- a)  $x(t) = \cos(t)$ ,  $y(t) = \sin(t)$ ,  $z(t) = t$ ,  $t \in [0, 2\pi]$  (винтовая линия)  
b)  $x(t) = e^{-|t-50|/50} \sin(t)$ ,  $y(t) = e^{-|t-50|/50} \cos(t)$ ,  $z(t) = t$ ,  $t \in [0, 100]$

**Задание 5:**

Постройте гладкую поверхность фигуры, заданной соотношениями:

- a)  $z(x, y) = \sin(x) \sin(y)$ ,  $x \in [0, 100]$ ,  $y \in [0, 100]$   
b)  $z(x, y) = \sin(x) \sin(y) + 0.3 \cos(2x - 3y) - 0.2 \sin(x + 2y)$ ,  $x, y \in [0, 100]$

Создайте пять пунктов главного меню, изменяющих цветовую палитру закраски построенной поверхности.

**Задание 6:**

В середину графического окна поместите кнопку. При нажатии на кнопку ее местоположение изменяется случайным образом в пределах всего графического окна.

**Задание 7:**

Дан графический (JPG-)файл, содержащий изображение некоторого аналогового сигнала. Следует оцифровать данный сигнал, то есть представить его в виде таблицы значений (**X**, **Y**). Графический интерфейс пользователя должен содержать кнопки, позволяющие выполнить следующие действия:

- **Image**: загрузка и визуализация графического изображения.
- **Digitization**: оцифровка сигнала. В данном режиме при помощи функции **ginput** оцифровываются координаты (**X**, **Y**) точек аналогового

сигнала. После оцифровки каждой точки значения ее координат сохраняются в массиве данных. Все оцифрованные точки сортируются по координате **X** и отображаются поверх графического изображения ломаной красного цвета. Узлы ломаной выделяются меткой 'o'.

- **Delete**: удаление точки оцифрованного сигнала. В данном режиме можно удалить ранее оцифрованную точку. После попадания курсора мыши в метку выбранной точки и нажатия кнопки мыши координаты данной точки удаляются из массива данных. Затем происходит перерисовка ломаной линии.
- **Load**: сохранение массива координат в бинарном файле.
- **Save**: загрузка массива координат из бинарного файла.
- **Zoom**: увеличение/уменьшение фрагментов изображения.

### Лабораторная работа № 10

#### Задание 1:

Решить уравнение  $\sin x - x^2 \cos x = 0$  на отрезке  $[-5;5]$ . Построить график функции левой части уравнения на указанном промежутке.

#### Задание 2:

Найти все корни полинома  $p = x^3 + 2x^2 - 3x + 1$ . Постройте график полиномиальной функции.

#### Задание 3:

Построить график и найти локальные минимумы и максимумы функции  $y = \sin(2x)e^{x/10}$ ,  $x \in [0,8]$ .

#### Задание 4:

Построить гладкую поверхность и найти локальные минимумы и максимумы функции  $f(x, y) = \sin(\pi x) \sin(\pi y)$ ,  $x, y \in [0, 2]$ .

#### Задание 5:

Для заданных массивов выполнить интерполирование несколькими полиномами различных степеней по методу наименьших квадратов:  
**x = 0:0.1:5; y = erf(x).**



**Задание 6:**

Для заданных массивов выполнить интерполирование сплайнами:

```
t = 1900:10:1990;  
p = [75.995 91.972 105.711 123.203 131.669...  
     150.697 179.323 203.212 226.505 249.633];
```

**Задание 7:**

По точкам заданной поверхности выполнить интерполяцию сплайнами:

```
[X,Y] = meshgrid(-3:.25:3);  
Z = peaks(X,Y);
```

**Задание 8:**

Решить систему уравнений: 
$$\begin{cases} x - y = 1, \\ y + z = 7, \\ z - x = -2. \end{cases}$$

## СПИСОК КОНТРОЛЬНЫХ ВОПРОСОВ

### Тема № 1. Назначение и структура MATLAB

В результате изучения темы студент должен:

#### Знать:

1. Назначение пакета MATLAB, версии.
2. Сравнение MATLAB с другими математическими пакетами.
3. Интерфейс пользователя пакета MATLAB.
4. Шесть основных компонент интерфейса пользователя MATLAB.
5. Назначение окна **Command Window**.
6. Назначение окна **Workspace Browser**.
7. Назначение окна **Current Directory**.
8. Назначение окна **Command History**.
9. Назначение окна **Figure**.
10. Назначение окна **M-file Editor**.
11. Назначение **Help & Demo**.
12. Интерактивный режим работы.
13. Составные части командного окна.
14. Приглашающий символ командной строки.

#### Уметь:

1. Использовать графический интерфейс пользователя (GUI) пакета MATLAB.
2. Отключать и добавлять окна GUI MATLAB.
3. Перенастраивать расположение окон GUI MATLAB.
4. Выполнять переход к справочной системе.
5. Выполнять переход к демонстрационным примерам.
6. Получать интерактивную справку по встроенным функциям и операторам.
7. Вводить в командном окне простейшие команды.
8. Создавать элементарные векторы и матрицы.
9. Создавать анонимные функции.
10. Выводить графики элементарных встроенных функций одной переменной в одной системе координат.
11. Разделять логическую команду на несколько физических строк.
12. Подавлять вывод на экран результата выполнения команды.
13. Очищать командное окно и рабочее пространство.
14. Выводить список всех переменных рабочего пространства.
15. Получать доступ к ранее введенным командам.

## Тема № 2. Данные и вычисления

В результате изучения темы студент должен:

### Знать:

1. Разделители операторов, комментарии и «длинные» операторы.
2. Основной тип данных в MATLAB.
3. Как определить тип переменной.
4. Способы создания векторов.
5. Способы создания матриц.
6. Что является разделителем элементов строки матрицы.
7. Что является разделителем строк матрицы.
8. Что такое многомерный массив, как его создать.
9. Что такое размерность и валентность массива.
10. Индексация массива.
11. Расположение элементов массива в памяти.
12. Функции для создания «стандартных матриц».
13. Как узнать размеры (другие параметры) массива.
14. Функции для работы с текстовыми строками.
15. Структуры, массивы структур.
16. Функции для работы со структурами.
17. Массивы ячеек.
18. Функции для работы с массивами ячеек.

### Уметь:

1. Выполнять основные операции над вещественными и комплексными числами.
2. Создавать векторы, матрицы и тензоры.
3. Определять размеры (другие параметры) массива.
4. Извлекать элементы, группы элементов из массива.
5. Удалять элементы, группы элементов из массива.
6. Находить минимальный (максимальный) элемент массива.
7. Создавать векторы при помощи операции формирования диапазона значений.
8. Выполнять арифметические операции над массивами.
9. Выполнять логические операции над массивами.
10. Изобразить «портрет» матрицы.
11. Создавать текстовые строки.
12. Применять функции для работы с текстовыми строками.
13. Создавать структуры, массивы структур.

14. Применять функции для работы со структурами.
15. Создавать массивы ячеек.
16. Осуществлять доступ к ячейке и ее содержимому.
17. Применять функции для работы с массивами ячеек.

### Тема № 3. Элементы программирования

В результате изучения темы студент должен:

#### Знать:

1. Операторы управления: операторы ветвления и операторы цикла.
2. Условный оператор **if**.
3. Оператор переключения **switch**.
4. Оператор цикла **for**.
5. Оператор цикла **while**.
6. Операторы **break**, **continue**.
7. Понятие векторизации алгоритма.

#### Уметь:

1. Использовать условный оператор **if**.
2. Использовать оператор переключения **switch**.
3. Использовать оператор цикла **for**.
4. Использовать оператор цикла **while**.
5. Применять операторы **break**, **continue**.
6. Выполнять векторизацию алгоритма.
7. Обрабатывать исключительные ситуации.
8. Создавать несколько систем координат в одном графическом окне.
9. Строить графики в разных системах координат.
10. Строить поверхности как функции двух переменных.
11. Вырезать из поверхности ее часть.
12. Оформлять графики.
13. Управлять свойствами построенной поверхности.
14. Вводить данные с клавиатуры и выводить результаты в командное окно.

### Тема № 4. Сценарии, функции и переменные

В результате изучения темы студент должен:

**Знать:**

1. Режимы работы в MATLAB.
2. Типы М-файлов.
3. Различия между М-сценарием и М-функцией.
4. Как создать функцию в MATLAB.
5. Типовая структура М-файла.
6. Оператор заголовка функции.
7. Входные/выходные параметры функции.
8. Что является признаком конца функции.
9. Как оформить комментарий.
10. Типы функций.
11. Первичные функции.
12. Подфункции.
13. Анонимные функции.
14. Частные функции.
15. Видимость имен функций.
16. Проверка параметров функций.
17. Переменные и их область действия.
18. Локальные, глобальные переменные.
19. Устойчивые переменные.
20. Перегрузка функций.
21. Отладка функций.

**Уметь:**

1. Создавать сценарии и функции в MATLAB.
2. Оформлять однострочный и многострочный комментарий.
3. Создавать анонимные функции.
4. Выполнять проверку параметров функций.
5. Объявлять и использовать локальные и глобальные переменные.
6. Объявлять и использовать устойчивые переменные.
7. Выполнять перегрузку функций.
8. Выполнять отладку функций.

**Тема № 5. Объектно-ориентированное программирование**

В результате изучения темы студент должен:

**Знать:**

1. Справочные функции: **class**, **isa**.
2. Основные принципы ООП, реализованные в MATLAB.

3. Правила создания нового класса.
4. Функции для идентификации класса.
5. Конструктор класса, типы конструкторов.
6. Деструктор класса.
7. Методы класса.
8. Работа с индексами: **subsref**, **subsagn**, **subindex**.
9. Правила переопределения арифметических операций, имена файлов для реализации базовых операций MATLAB.
10. Изменение свойств объекта: **set**, **get**.
11. Конверторы **double** и **char**.

**Уметь:**

1. Применять правила создания нового класса.
2. Использовать функции для идентификации класса.
3. Создавать различные типы конструкторов класса.
4. Создавать и удалять объекты класса пользователя.
5. Оформлять методы класса.
6. Работать с индексами: **subsref**, **subsagn**, **subindex**.
7. Переопределять встроенные арифметические операции
8. Изменять свойства объекта при помощи функций **set**, **get**.
9. Переопределять функции-конверторы **double** и **char**.
10. Переопределять функцию вывода на печать **display**.
11. Создавать специализированные функции класса пользователя.

## **Тема № 6. Высокоуровневая графика**

В результате изучения темы студент должен:

**Знать:**

1. Особенности высокоуровневой графики.
2. Особенности построения двумерной и трехмерной графики.
3. Декартовы и полярные координаты.
4. Функции, заданные параметрически.
5. Оформление графиков функций: оси координат.
6. Оформление графиков функций: надписи на графике.
7. Изменение свойств линий.
8. Поверхности, линии уровня.
9. Камера, управление камерой.
10. Сохранение графических изображений.
11. Форматы графических файлов.

**Уметь:**

1. Использовать высокоуровневые графические функции.
2. Строить графики функций одной и двух переменных.
3. Строить графики функций в декартовых координатах.
4. Строить графики функций в полярных координатах.
5. Строить графики функций, заданных параметрически.
6. Создавать анимированный двумерный график.
7. Строить диаграммы и гистограммы.
8. Оформлять оси координат.
9. Оформлять надписи на графике.
10. Изменять свойства линий.
11. Строить поверхности и их линии уровня.
12. Управление камерой и положением точки обзора.
13. Сохранять графические изображения.

**Тема № 7. Дескрипторная графика**

В результате изучения темы студент должен:

**Знать:**

1. Основы дескрипторной графика.
2. Взаимосвязь между высокоуровневой и дескрипторной графикой.
3. Дескриптор графического объекта.
4. Иерархия графических объектов.
5. Графический объект **root**.
6. Графический объект **figure**.
7. Графический объект **axes**.
8. Объекты **uimenu**, **uicontextmenu**, **uicontrol**.
9. Графический объект **image**.
10. Графический объект **light**.
11. Графический объект **line**.
12. Графический объект **patch**.
13. Графический объект **rectangle**.
14. Графический объект **surface**.
15. Графический объект **text**.
16. Управление свойствами графических объектов.
17. Свойства по умолчанию графических объектов.

**Уметь:**

1. Создавать объекты из иерархии графических объектов.

2. Получать дескриптор графического объекта.
3. Получать доступ к свойствам графического объекта.
4. Изменять свойства графических объектов.
5. Устанавливать свойства по умолчанию для графических объектов на разных уровнях.
6. Находить группу графических объектов, обладающих одинаковым свойством.
7. Удалять группу графических объектов, обладающих одинаковым свойством.
8. Изменять свойство группы графических объектов, обладающих одинаковым свойством.

### **Тема № 8. Импорт и экспорт данных**

В результате изучения темы студент должен:

#### **Знать:**

1. Импорт и экспорт данных.
2. Типы форматов файлов данных, поддерживаемые MATLAB.
3. Работа с именем файла.
4. Перемещение данных рабочего пространства.
5. Низкоуровневый файловый ввод/вывод.
6. Ввод/вывод бинарных файлов.
7. Ввод/вывод текстовых файлов.
8. Форматная спецификация текстовых данных.

#### **Уметь:**

1. Сохранять в mat-файл и загружать из mat-файла все или выборочные данные рабочего пространства.
2. Работать с именем файла.
3. Выполнять низкоуровневый ввод/вывод бинарных файлов.
4. Выполнять низкоуровневый ввод/вывод текстовых файлов.
5. Выполнять форматированный ввод-вывод данных.
6. Сохранять и загружать графические файлы.

### **Тема № 9. Графический интерфейс пользователя**

В результате изучения темы студент должен:

#### **Знать:**

1. Графический интерфейс пользователя (GUI).



2. Способы создания GUI, поддерживаемые MATLAB.
3. Интерактивный интерфейс.
4. Элементы дескрипторной графики для создания GUI.
5. Объект **uicontrol**.
6. Объект **uimenu**.
7. Объект **uicontextmenu**.
8. Свойства **UI**-объектов.
9. Доступ к свойствам **UI**-объектов.
10. **Callback**-функции.
11. Редактор **Layout Editor**.
12. Типы файлов для создания GUI.

**Уметь:**

1. Создавать графический интерфейс пользователя (GUI) программно и при помощи мастера создания GUI.
2. Уметь вводить-выводить данные в интерактивном режиме.
3. Создавать объекты управления, меню и контекстное меню.
4. Управлять свойствами объектов GUI.
5. Определять **Callback**-функции и соотносить их с объектами GUI.

### Тема № 10. Вычисления в MATLAB

В результате изучения темы студент должен:

**Знать:**

1. Способы решения алгебраических и полиномиальных уравнений.
2. Способы поиска локальных экстремумов функций одной или нескольких переменных.
3. Способы интерполирования функций одной или нескольких переменных.
4. Методы решения систем ЛУ.

**Уметь:**

1. Определять **m**-функции, анонимные и **inline**-функции.
2. Находить нули функций на заданном интервале.
3. Находить корни полиномиального уравнения.
4. Находить локальные экстремумы функций одной или нескольких переменных.
5. Выполнять интерполирование функций одной или нескольких переменных полиномами или сплайнами.
6. Решать системы ЛУ различными методами.

## ПРИМЕР КОНТРОЛЬНОГО ТЕСТА

### Общие сведения о программе

1. Пакет MATLAB
  - a. Числовой
  - b. Символьный
2. Точность вычислений в пакете MATLAB
  - a. Не ограничена
  - b. Ограничена разрядностью процессора
3. Тип вычислений в пакете MATLAB
  - a. Целочисленная арифметика
  - b. Арифметика с плавающей точкой
4. Простейшими объектами в MATLAB являются:
  - a. Числа
  - b. Константы
  - c. Выражения
  - d. Матрицы
  - e. Строки
  - f. Имена
  - g. Структуры

### Интерфейс программы

5. Командное окно предназначено:
  - a. Для просмотра и редактирования значений и изменения размерности данных рабочего пространства
  - b. Для работы в интерактивном режиме
  - c. Для просмотра и повторного вызова ранее введенных команд
  - d. Для просмотра и редактирования имен и значений данных рабочего пространства
  - e. Для работы с графикой и визуализации 2D- и 3D-данных
  - f. Для поиска, просмотра и изменения каталогов и файлов
  - g. Для создания, редактирования, отладки и сохранения М-файлов
6. Окно текущего директория предназначено:
  - a. Для просмотра и редактирования значений и изменения размерности данных рабочего пространства
  - b. Для работы в интерактивном режиме
  - c. Для просмотра и повторного вызова ранее введенных команд
  - d. Для просмотра и редактирования имен и значений данных рабочего пространства

- e. Для работы с графикой и визуализации 2D- и 3D-данных
  - f. Для поиска, просмотра и изменения каталогов и файлов
  - g. Для создания, редактирования, отладки и сохранения М-файлов
7. Графическое окно предназначено:
- a. Для просмотра и редактирования значений и изменения размерности данных рабочего пространства
  - b. Для работы в интерактивном режиме
  - c. Для просмотра и повторного вызова ранее введенных команд
  - d. Для просмотра и редактирования имен и значений данных рабочего пространства
  - e. Для работы с графикой и визуализации 2D- и 3D-данных
  - f. Для поиска, просмотра и изменения каталогов и файлов
  - g. Для создания, редактирования, отладки и сохранения М-файлов
8. Строка ввода команды Input Line помечена символом:
- a. >
  - b. ?
  - c. }
  - d. ]]
  - e. >>
  - f. \*\*
9. Нажатие клавиш <↑> и <↓> в командной строке осуществляют:
- a. Прокрутку всего экрана
  - b. Прокрутку зоны редактирования
  - c. Прокрутку ранее введенных команд

### Массивы, векторы и матрицы

10. Элементы любого массива хранятся в памяти компьютера:
- a. упорядоченными по столбцам
  - b. упорядоченными по строкам
11. Для разделения элементов строки матрицы используется:
- a. Пробел или точка с запятой ;
  - b. Пробел или запятая ,
  - c. Точка .
  - d. Точка с запятой ;
  - e. Слеш /
  - f. Слеш \
  - g. Подчеркивание \_
  - h. Апостроф `
  - i. Кавычки ``

j. Нет верного ответа

12. Для матрицы  $A = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \end{pmatrix}$  результатом выполнения операции **A(2)**

будет:

a. 2

b. 5

c. 5 6 7 8

2

d. 6

10

e. Сообщение об ошибке

13. Для матрицы  $A = \begin{pmatrix} 1 & 3 & 5 \\ 7 & 9 & 11 \end{pmatrix}$  результатом выполнения операции **A(2,3) = []**

будет:

a. 2

b. 5

c. 5 6 7 8

2

d. 6

10

e. 1 3 5 7 9

1

3

f. 5

7

9

g. Сообщение об ошибке

14. Результатом выполнения операции **A=[1 2 3; 4 5]** будет:

a.  $A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 0 \end{pmatrix}$

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

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

d. Сообщение об ошибке

15. Строки  $\begin{matrix} 1 & 2 & 3 & 4 \\ 9 & 10 & 11 & 12 \end{matrix}$  из матрицы  $A = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \end{pmatrix}$  нельзя выделить в

результате выполнения операции:

- a.  $A(1:2:3,:)$
- b.  $A(2,:)$
- c.  $A(2,:)=[]$
- d.  $A(:,2)=[]$
- e.  $A([1\ 3],:)$
- f.  $A([1,3],:)$

16. Из матриц  $A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$  и  $B = \begin{pmatrix} 7 & 8 & 9 \\ 10 & 11 & 12 \end{pmatrix}$  получится матрица

$C = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ 10 & 11 & 12 \end{pmatrix}$  в результате выполнения операции:

- a.  $C=[A;B]$
- b.  $C=[A,B]$
- c.  $C=[A\ B]$
- d.  $C= \text{cat}(2,A,B)$
- e.  $C=(A,B)$
- f. Нет верного ответа

17. Результат применения к матрице  $A = \begin{pmatrix} 1 & -2 & 3 & 4 \\ 5 & 6 & -7 & 8 \\ 9 & 10 & -11 & -12 \end{pmatrix}$  операции **max (A)** :

- a. 10
- b. 9 10 3 8
- c. 8
- d. Нет верного ответа

18. Из матриц  $A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$  и  $B = \begin{pmatrix} 5 & 6 \\ 7 & 8 \end{pmatrix}$  получится матрица  $C = \begin{pmatrix} 3.0000 & -2.0000 \\ 2.000 & -1.0000 \end{pmatrix}$  в

результате выполнения операции:

- a.  $C=A./B$
- b.  $C=A/B$
- c.  $C=B./A$
- d.  $C=B/A$

е. Нет верного ответа

**19.**Результат применения к матрице  $A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$  операции  $C=2 ./A$ :

a.  $C = \begin{pmatrix} 0.5000 & 1.0000 \\ 1.5000 & 2.0000 \end{pmatrix}$

b.  $C = \begin{pmatrix} 2.0000 & 1.0000 \\ 0.6667 & 0.5000 \end{pmatrix}$

с. Нет верного ответа

### Вычисление арифметических выражений

**20.**Что получится в результате выполнения операции  $1/0$ :

- a. -Inf
- b. NaN
- c. Inf
- d. end
- е. Сообщение об ошибке

**21.**Что получится в результате выполнения операции  $0/0$ :

- a. -Inf
- b. NaN
- c. Inf
- d. end
- е. Сообщение об ошибке

**22.**Что получится в результате выполнения операции  $1+5i + 3+2i'$ :

- a. - 7.0000i
- b. + 7.0000i
- c. + 3.0000i
- d. -4.0000 - 7.0000i
- е. -2.0000 + 3.0000i
- f. Сообщение об ошибке

**23.**Результат выполнения операции  $\text{sqrt}(-1.0) * (2.+3i) + (1+5i + 3+2i')'$ :

- a. 1.000
- b. - 1.0000i
- c. + 9.0000i
- d. + 1.0000i
- е. Сообщение об ошибке

**24.**Результат выполнения операции  $[1+2i \ 3-5j] . '$ :

- a.  $1-2i \ 3+5i$

- b.  $-1+2i -3-5i$
- c.  $1+2i$   
 $3-5i$
- d.  $1-2i$   
 $3+5i$
- e. Сообщение об ошибке

## Язык программирования MATLAB

25. Команда **who** возвращает в командное окно:

- a. список имен всех переменных, содержащихся в рабочем пространстве, с указанием их размерности, количества байт в памяти и класса (типа)
- b. список имен всех переменных, содержащихся в рабочем пространстве
- c. список имен и значений всех переменных, содержащихся в рабочем пространстве
- d. список имен и типов всех переменных, содержащихся в рабочем пространстве

26. Специальный символ **[]** не применяется в следующем случае:

- a. Формирование векторов и матриц
- b. Удаление элемента матрицы
- c. Указание последовательности выходных параметров функции
- d. Указание индексов элементов массива

27. Специальный символ **:** применяется :

- a. Указание последовательности выходных параметров функции
- b. Указание последовательности входных параметров функции
- c. Формирование векторов с помощью операции диапазона значений
- d. Подавление вывода в командное окно результатов вычислений
- e. Указание индексов элементов вектора, матрицы

28. Для создания и хранения кодов сценариев и функций в пакете MATLAB используются файлы с расширением:

- a. \*.c
- b. \*.cpp
- c. \*.dfm
- d. \*.fig
- e. \*.mat
- f. \*.m

29. Специальная переменная, содержащая результат последней выполненной операции, если во введенной команде отсутствует операция присваивания результата `=`, называется:
- a. `ans`
  - b. `NaN`
  - c. `end`
  - d. `Inf`
30. Команда `load a b c`:
- a. Загружает в рабочее пространство все переменные из файлов `a.mat`, `b.mat`, `c.mat`
  - b. Загружает в рабочее пространство переменные `b`, `c` из файла `a.mat`
  - c. Загружает в рабочее пространство переменные `a`, `b`, `c` из файла `matlab.mat`
  - d. Загружает в рабочее пространство переменные `a`, `b` из файла `c.mat`
31. Для вывода в командное окно подсказки о назначении и способах вызова функции `plot` необходимо ввести команду:
- a. `>> show plot`
  - b. `>> help plot`
  - c. `>> demo plot`
  - d. `>> look plot`
32. MATLAB:
- a. Различает верхний и нижний регистр
  - b. Не различает верхний и нижний регистр
33. Индексация массива начинается с:
- a. 1
  - b. 0
34. Текстовые данные, в том числе и одиночный символ, должны заключаться с обеих сторон:
- a. В скобки `{ }`
  - b. В скобки `( )`
  - c. В кавычки `" "`
  - d. В апострофы `' '`
  - e. Нет правильного ответа
35. Какая команда не является функцией низкоуровневого файлового ввода-вывода:
- a. `fopen`
  - b. `fclose`



- c. fread
- d. fwrite
- e. feval
- f. fseek
- g. frewind
- h. feof
- i. fscanf
- j. fprintf
- k. fgetl
- l. fgets

36. Что не является определением функции:

- a. `function y = funcName(a,b,c)`
- b. `function [] = funcName(x)`
- c. `function funcName(x)`
- d. `function (x,y,z) funcName[a,b,c]`
- e. `function [x, y, z] = funcName()`
- f. `function funcName`

37. Команда **`f = @(arglist)expression`** определяет:

- a. Первичную функцию
- b. Подфункцию
- c. Сценарий
- d. Анонимную функцию
- e. Вложенную функцию
- f. Частную функцию
- g. Другой ответ

38. Объявить устойчивую переменную можно при помощи команды

- a. `>> static`
- b. `>> constant`
- c. `>> persistent`
- d. Другой ответ

39. Объект класса хранят свои данные в виде:

- a. Матрицы
- b. Отдельных переменных
- c. Структуры
- d. Массива ячеек
- e. Тензора
- f. Другой ответ

## Дескрипторная графика

40. В результате выполнения команд

```
>> x = -2*pi : 0.01 : 2*pi;  
>> y = sin(x);  
>> z = 0.5*cos(x);  
>> plot(x, y, x, z);
```

- a. В одном графическом окне в разных системах координат строятся графики функций
- b. В разных графических окнах строятся графики функций
- c. В одном графическом окне в одной системе координат строятся графики функций разным цветом
- d. В одном графическом окне в одной системе координат строятся графики функций одним цветом

41. Графический объект, не являющийся графическим объектом пакета MATLAB:

- a. root
- b. figure
- c. axes
- d. uicontrol
- e. uimenu
- f. uisubmenu
- g. circle
- h. image
- i. line
- j. patch
- k. rectangle
- l. surface
- m. text

42. Функции, которые выполняют действия, связанные с реакцией графических элементов управления на события, называются:

- a. Anonymous-функциями
- b. Callback-функциями
- c. Nested-функциями
- d. Primary-функциями
- e. Private-функциями
- f. Subfunction-функциями

43. Команда `>> gcf`

- a. Создает графическое окно

- b. Создает систему координат
- c. Возвращает дескриптор текущего графического окна
- d. Возвращает дескриптор текущей системы координат
- e. Очищает текущее графическое окно
- f. Очищает текущую систему координат
- g. Закрывает текущее графическое окно
- h. Обновляет текущее графическое окно

44. Команда **>> set(hobj)** осуществляет:

- a. Вывод списка текущих значений свойств графического объекта с дескриптором hobj
- b. Вывод списка возможных значений всех свойств объекта с дескриптором hobj
- c. Вывод текущего значения указанного свойства объекта с дескриптором hobj
- d. Вывод списка возможных значений указанного свойства объекта с дескриптором hobj
- e. Установку нового значения указанного свойства объекта с дескриптором hobj
- f. Восстановление стандартных значений свойств по умолчанию объекта с дескриптором hobj

## СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ

1. Голубева Л. Л. Компьютерная математика. Числовой пакет MATLAB: курс лекций / Л. Л. Голубева, А. Э. Малевич, Н. Л. Щеглова. Минск: БГУ, 2007. 164 с.
2. Голубева Л. Л. Компьютерная математика. Числовой пакет MATLAB: лабораторный практикум / Л. Л. Голубева, А. Э. Малевич, Н. Л. Щеглова. Минск: БГУ, 2008. 171 с.
3. Ануфриев И. Е. Самоучитель MATLAB 5.3/6.x. СПб: БХВ-Петербург, 2004. 736 с.
4. Говорухин В., Цибулин. Компьютер в математическом исследовании: учеб. курс. СПб: Питер, 2001. 624 с.
5. Потемкин В. Г. Система MATLAB 5 для студентов. М.: ДИАЛОГ-МИФИ, 1998. 314 с.
6. Потемкин В. Г. Система инженерных и научных расчетов MATLAB 5.x: в 2 т. М.: ДИАЛОГ-МИФИ, 1999. Т. 1. 366 с.
7. Потемкин В. Г. Система инженерных и научных расчетов MATLAB 5.x: в 2 т. М.: ДИАЛОГ-МИФИ, 1999. Т. 2. 304 с.
8. Angermann A. MATLAB — Simulink — Stateflow. Grundlagen, Toolboxes, Beispiele / A. Angermann, M. Beuschel, M. Rau, U. Wohlfarth. Oldenburg Verlag München Wien, 2007. 495 p.
9. Adam S. MATLAB und Mathematik kompetent einsetzen. WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim, 2006. 462 p.
10. MathWorks MATLAB Documentation.  
URL: <http://www.mathworks.com/help/matlab/index.html>
11. MathWorks Getting Started with MATLAB.  
URL: <http://www.mathworks.com/help/matlab/getting-started-with-matlab.html>
12. MathWorks MATLAB Examples.  
URL: <http://www.mathworks.com/help/matlab/examples/index.html>

*Все ссылки проверены на работоспособность 3 февраля 2013 г.*

## РАЗДЕЛ IV. ОСНОВЫ КОМПЬЮТЕРНОГО МОДЕЛИРОВАНИЯ

### КОНСПЕКТ ЛЕКЦИЙ

Электронные версии лекций размещены в электронной библиотеке БГУ по адресу <http://elib.bsu.by/handle/123456789/4171>. Ниже приведен тематический план лекций.

#### Лекция № 1. Введение в Simulink

**Назначение и структура Simulink.** Модель и моделирование. Предметная область. Виды моделирования: концептуальное, физическое, структурно-функциональное, математическое, имитационное. Компьютерное моделирование.

Назначение Simulink. Среда разработки. Запуск Simulink. Создание и открытие S-модели. Интерфейс пользователя для работы в Simulink. Браузер библиотек блочных компонентов. Поиск блоков по названию. Окно S-модели. Построение S-модели. Управление работой S-модели. Завершение работы в Simulink. Справочная система Simulink.

**Основные приемы редактирования S-модели.** Действия с объектами: выделение объектов, изменение размеров блоков, соединение блоков, создание отвода соединительной линии, редактирование соединительных линий, удаление блоков и соединительных линий, вставка блоков в соединение, перемещение блоков. Размещение текстовых надписей на модели, именование объектов. Изменение масштаба изображения. Форматирование объектов.

#### Лекция № 2. Блок интегратор и параметры конфигурации модели

**Универсальная библиотека Simulink.** Обзор универсальной библиотеки Simulink. Обзор блоков раздела Sources. Обзор блоков раздела Sinks. Обзор разделов Math Operations и Logic and Bit Operations. Блок Integrator. Исследование блоков других разделов.

**Параметры конфигурации S-модели.** Параметры конфигурации S-модели. Установка параметров расчета S-модели. Время моделирования. Параметры расчета. Обмен данными с рабочей областью. Загрузка данных из рабочей области. Запись результатов в рабочую область. Варианты записи.

#### Лекция № 3. Обмен данными между MATLAB и Simulink

Взаимодействие MATLAB и Simulink. Обмен данными между MATLAB и Simulink. Задание параметров блоков формулами. Передача данных в Simulink. Блок входного порта. Блок считывания данных из Workspace. Блок считывания

данных из файла. Передача данных из Simulink. Передача данных через блок выходного порта. Блок записи данных в Workspace. Блок записи данных в файл. Запуск m-файла из S-модели. Ручной запуск m-файла из S-модели. Автоматический запуск m-файла из S-модели. Запуск S-модели из m-файла.

### **Лекция № 4. S-функции**

Определение и назначение S-функций. Описание S-функций. Языки написания S-функций. Использование S-функций в моделях. Базовая структура S-функции. Блок S-Function, его параметры. Математическое описание S-функции. Этапы выполнения расчетов Simulink-модели.

Создание S-функций на языке MATLAB. Заголовок S-функции. Входные и выходные параметры S-функции. Определение блока характеристик S-функции. Callback-методы S-функции. Применение S-функции для создания анимации.

### **Лекция № 5. Управляемые подсистемы**

Назначение и описание подсистем. Виды подсистем. Виртуальные и атомарные подсистемы. Управляемые и неуправляемые подсистемы.

Типы управляемых подсистем. Управляемая уровнем сигнала подсистема. Управляемая фронтом сигнала подсистема. Управляемая уровнем и фронтом сигнала ET-подсистема.

Подсистемы управляющей логики. Управляемая по условию If Action подсистема. Управляемая по условию For Iterator подсистема. Управляемая по условию подсистема Switch Case Action Subsystem. Управляемая подсистема While Iterator Subsystem.

### **Лекция № 6. Stateflow, или системы, управляемые событиями**

Виды сложных систем. Гибридные системы. Трансформационные и реактивные системы.

Последовательность состояний Stateflow. Stateflow и Simulink. Программная среда StateFlow. Компоненты Stateflow. Графический редактор StateFlow. Обозреватель StateFlow.

### **Лекция № 7. Диаграммы состояний и переходов StateChart**

Конечный автомат. Конечный автомат как ориентированный граф. Основные компоненты диаграммы состояний и переходов: состояние, событие, переход, действие. Построение диаграммы состояний и переходов. Основные графические и неграфические компоненты и нотации диаграммы состояний и переходов.

## ПЛАН ЛАБОРАТОРНЫХ ЗАНЯТИЙ

Электронные версии лабораторных работ размещены в электронной библиотеке БГУ по адресу <http://elib.bsu.by/handle/123456789/4171>. Ниже приведено краткое описание каждой работы.

### Лабораторная работа № 1. Назначение и структура Simulink

См. стр. 14–15 электронного документа.

### Лабораторная работа № 2. Модель прыгающего мячика

#### Задание:

Создать математическую и S-модель движения мячика, брошенного со скоростью 15 м/с вертикально вверх с высоты 10 м над твердой горизонтальной поверхностью. При моделировании учесть сопротивление воздуха и упругость мячика.

См. стр. 23–32 электронного документа.

### Лабораторная работа № 3. Модель затухающих колебаний маятника

#### Задание:

К неподвижной опоре на невесомом нерастяжимом стержне длины  $L$  м подвешен груз массой  $m$  кг. Построить математическую и S-модель колебаний маятника, если ускорение свободного падения равно  $g$  м/с<sup>2</sup>, а сопротивление среды прямо пропорционально скорости движения маятника с коэффициентом  $k$  кг/с. Вывести графики зависимости угла и угловой скорости маятника от времени; фазовую диаграмму колебаний; анимацию движения маятника. Учесть в модели сопротивление среды.

См. стр. 45–55 электронного документа.

### Лабораторная работа № 4. Модель колебаний маятника с ограничением движения

#### Задания:

1. К неподвижной опоре на невесомом нерастяжимом негнущемся (металлическом) стержне длины  $L$  м подвешен груз массой  $m$  кг. На расстоянии  $b$  м от точки опоры находится вертикальная стена. Построить модель колебаний маятника, если ускорение свободного падения  $g = 9.81$  м/с<sup>2</sup>, сопротивление среды прямо пропорционально скорости движения маятника с коэффициентом  $k$  кг/с, а в результате удара о стену

скорость маятника  $\dot{\varphi}$  меняет знак и становится равной  $-\alpha\dot{\varphi}$ . Начальное положение маятника  $\varphi(0)=0$ , начальная угловая скорость маятника  $\dot{\varphi}(0)=\dot{\varphi}_0$ . Требуется построить: математическую модель колебаний маятника; S-модель колебаний маятника; графики зависимости угла и угловой скорости маятника от времени; анимацию движения маятника.

2. Исследуйте зависимость колебаний маятника от начальной угловой скорости. Найдите значение начальной угловой скорости  $\dot{\varphi}_0$ , при котором маятник: не ударяется о стену; ударяется о стену только в нижнем положении; ударяется о стену как в нижнем, так и в верхнем положениях.
3. Учесть в модели не абсолютно упругий удар. Отредактируйте S-модель, чтобы после удара угловая скорость маятника  $\dot{\varphi}$  стала равной  $-\alpha\dot{\varphi}$ . Подберите время моделирования, чтобы в конце моделирования маятник не ударялся о стену.
4. Учесть в модели сопротивление среды. Как зависят колебания от массы  $m$  маятника? Подберите такую начальную угловую скорость  $\dot{\varphi}_0$ , при которой маятник, в конце концов, остановился, упершись в стену.
5. Учтите в модели наличие второй стены, расположенной: симметрично первой стены относительно точки опоры (на расстоянии  $b$ ); не симметрично первой стены относительно точки опоры (на расстояниях  $b$  и  $b_1$  от точки опоры).

См. стр. 62–70 электронного документа.

### **Лабораторная работа № 5. Пружинно-амортизирующая система автомобиля**

#### **Задания:**

1. Проанализировать поведение подвесной системы автомобиля при помощи моделирования упрощенной пружинно-амортизирующей системы. Записать дифференциальное уравнение, описывающее реакцию подвесной системы на входную функцию, моделирующую различные типы дорожных условий. Исследовать зависимость вертикального перемещения  $y$  и вибрации кузова автомобиля от коэффициента амортизации  $c$  и коэффициента жесткости рессоры  $k$ . Построить S-модель подвесной системы автомобиля. Для задания параметров блоков использовать переменные MATLAB. Результаты моделирования выводить в графическое окно Figure. Инициализацию переменных, вызов S-модели и вывод результатов в графическое окно Figure выполнить с использованием m-файла.



2. Исследуйте поведение системы при различных значениях коэффициента демпфирования  $s$  и фиксированном значении коэффициента жесткости  $k$ .
3. Исследуйте поведение системы при различных значениях коэффициента жесткости  $k$  и фиксированном значении коэффициента демпфирования  $s$ .
4. Найдите оптимальное решение, при котором достигается наименьшее перемещение кузова автомобиля и скорейшее затухание его колебаний.
5. Реализуйте автоматический вызов  $m$ -файлов, выполняющих задание и расчет необходимых параметров, а также загрузку  $mat$ -файлов и вывод результатов в графическое окно из  $S$ -модели перед началом и по окончании выполнения моделирования.

См. стр. 94–108 электронного документа.

### **Лабораторная работа № 6. Анимация свободных колебаний маятника**

#### **Задания:**

1. При помощи  $S$ -функции реализуйте анимацию затухающих колебаний маятника (Лабораторная работа № 3).
2. При помощи  $S$ -функции реализуйте анимацию колебаний маятника с ограничением движения (Лабораторная работа № 4).

См. стр. 117–123 электронного документа.

### **Лабораторная работа № 7. Полет тела, брошенного под углом к горизонту**

#### **Задания:**

1. К неподвижной опоре на невесомой нерастяжимой нити подвешен груз. Нить препятствует удалению груза от центра на расстояние, большее длины нити, но никак не мешает перемещению груза внутри круга. Движение груза происходит с учетом силы тяжести и силы сопротивления среды, прямо пропорциональной скорости движения груза. В начальный момент времени груз находится в нижней точке подвеса и движется строго горизонтально слева направо со скоростью  $V_0$  м/с. Требуется построить: математическую модель колебаний маятника;  $S$ -модель колебаний маятника; анимацию движения маятника.
2. Выделить основные виды движения груза (движение по окружности, свободное падение, удар) и реализовать  $S$ -модель с использованием соответствующих подсистем.

См. стр. 133–146 электронного документа.

## Лабораторная работа № 8. Конечный автомат

### Задания:

1. Реализовать в виде Simulink-Stateflow-модели алгоритм по нахождению наибольшего общего делителя двух натуральных чисел
2. Построить модель Термостатической Установки (Thermostat), поддерживающей температуру в заданных пределах  $T_{min} \leq T \leq T_{max}$ . Для этого используются Нагреватель (Heater: Device=1) и Холодильник (Cooler: Device=-1). В нормальных условиях оба устройства выключены (Idle: Device=0).

Нагреватель Heater можно включать не раньше чем через 5 минут после его выключения.

Холодильник Cooler состоит из компрессора и вентилятора. Сначала запускается компрессор, на что требуется определенное время, затем можно включать вентилятор, который и является охлаждающим элементом.

Любое из устройств (Devices) может отказать (Fail) с вероятностью 2%. В последнем случае должен включиться сигнал тревоги (Alarm) для вызова мастера, который устраняет неисправность, о чем делает запись LogItem в специальном журнале.

Дневное изменение температуры смоделировать синусоидой Sources\Sine Wave: днем теплеет до 20 °C, а ночью холодает до 0 °C, на которую наложить случайный шум  $\pm 5$  °C Sources\Random Number. Желаемую температуру ( $T_{min}, T_{max}$ ) задавать ползунками Math Operations\Slider Gain. Сбои в работе моделировать случайными числами Sources\Uniform Random Number.

## СПИСОК КОНТРОЛЬНЫХ ВОПРОСОВ

### Тема № 1. Компьютерное моделирование и Simulink

1. Определение понятия и назначение моделирования.
2. Виды моделирования
3. Компьютерное моделирование и компьютерная модель. Предмет и цель компьютерного моделирования
4. Математическое моделирование, математическая модель. Виды математического моделирования
5. Имитационное моделирование
6. Simulink как программа имитационного моделирования динамических и событийно-управляемых систем
7. Этапы моделирования (этапы построения компьютерной модели)
8. Понятие динамической системы, фазового пространства
9. MATLAB как средство реализации компьютерного моделирования
10. Преимущества Simulink
11. Основные составляющие модели Simulink
12. Общие сведения по работе с Simulink: запуск, создание модели, файл модели, библиотека компонентов, блоки, параметры блоков, соединение блоков, выполнение расчета модели

### Тема № 2. Блоки Solver и Integrator

13. Аналитическое и численное моделирование
14. Параметры конфигурации Simulink-модели Configuration Parameters
15. Задание интервала моделирования Simulation Time и шага моделирования Step Size
16. Параметры расчета Solver Options: способ моделирования (Type) и метод расчета нового состояния системы (Solver)
17. Дискретные и непрерывные системы
18. Численные методы решения дифференциальных уравнений
19. Блок Integrator. Назначение, параметры
20. Входные и выходные порты блока Integrator
21. Формула блока Integrator
22. Начальное значение и источник начального значения выходного сигнала блока Integrator
23. Внешний сброс блока Integrator
24. Порт состояния блока Integrator. Алгебраические циклы.

### **Тема № 3. Взаимодействие и обмен данными между MATLAB и Simulink**

- 25. Основные составляющие Simulink-модели
- 26. Способы взаимодействия между MATLAB и Simulink
- 27. Уровни обмена данными между MATLAB и Simulink
- 28. Использование переменных Workspace для задания параметров блоков
- 29. Установка параметров обмена с рабочей областью Workspace: Data Import/Export
- 30. Поле загрузки из рабочей области Load from workspace
- 31. Поле записи в рабочую область Save to workspace
- 32. Параметры записи Save options
- 33. Блоки передачи информации в модель Simulink и блоки для передачи результатов из модели Simulink
- 34. Блок входного порта Sources\Inport
- 35. Блок считывания данных из рабочей области Sources\From Workspace
- 36. Блок считывания данных из файла Sources\From File
- 37. Блок выходного порта Sinks\Outport
- 38. Блок сохранения данных в рабочей области Sinks\To Workspace
- 39. Блок сохранения данных в файле Sinks\To File
- 40. Ручной запуск m-файлов из Simulink-модели
- 41. Автоматическое выполнение команд m-файлов до и после выполнения модели
- 42. Запуск Simulink-модели из m-файла

### **Тема № 4. S-функции**

- 43. Назначение S-функций, цель их использования
- 44. Когда используются S-функции
- 45. Описание S-функций
- 46. Языки написания S-функций
- 47. Блок S-Function, его параметры
- 48. Математическое описание S-функции
- 49. Этапы Simulink-моделирования
- 50. Создание S-функций на языке MATLAB. Определение S-функции.
- 51. Входные и выходные параметры. Определение блока характеристик S-функции
- 52. Callback-методы S-функции
- 53. Применение S-функции для создания анимации

## **Тема № 5. Управляемые подсистемы**

- 54.** Назначение подсистем
- 55.** Виртуальные и монолитные подсистемы
- 56.** Управляемые и неуправляемые подсистемы
- 57.** Типы управляемых подсистем
- 58.** Связь подсистемы с моделью
- 59.** Способы создания подсистем
- 60.** Управляемая уровнем сигнала подсистема (E-Subsystem)
- 61.** Управляемая фронтом сигнала подсистема (T-Subsystem)
- 62.** Управляемая уровнем и фронтом сигнала подсистема (ET-Subsystem)
- 63.** If Action Subsystem
- 64.** Switch Case Action Subsystem
- 65.** For Iterator Subsystem
- 66.** While Iterator Subsystem

## **Тема № 6. Системы, управляемые событиями**

- 67.** Виды сложных систем. Гибридные системы. Трансформационные системы. Реактивные системы.
- 68.** Основные компоненты диаграммы состояний.
- 69.** Определение конечного автомата
- 70.** Запуск Stateflow
- 71.** Графический редактор Stateflow
- 72.** Виды объектов Stateflow
- 73.** Графические объекты
- 74.** Неграфические объекты

## **Тема № 7. Диаграммы состояний**

- 75.** Что такое Stateflow и Statechart
- 76.** Stateflow-машина
- 77.** Simulink-модель и Stateflow-машина
- 78.** Состояния, их характеристики
- 79.** Типы действий, связанные с состояниями
- 80.** Нотации меток состояний
- 81.** Переходы
- 82.** Действия, связанные с переходами
- 83.** Условия, связанные с переходами
- 84.** Нотации меток переходов
- 85.** Безусловный переход

- 86.**Соединительные переходы
- 87.**Хронологические соединения
- 88.**Действия
- 89.**События
- 90.**Данные

## СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ

1. Голубева Л. Л. Компьютерная математика. Пакет имитационного моделирования Simulink: лаб. практикум / Л. Л. Голубева, А. Э. Малевич, Н. Л. Щеглова. Минск: БГУ, 2010. 151 с.
2. Голубева Л. Л. Компьютерная математика. Числовой пакет MATLAB: курс лекций / Л. Л. Голубева, А. Э. Малевич, Н. Л. Щеглова. Минск: БГУ, 2007. 164 с.
3. Голубева Л. Л. Компьютерная математика. Числовой пакет MATLAB: лаб. практикум / Л. Л. Голубева, А. Э. Малевич, Н. Л. Щеглова. Минск: БГУ, 2008. 171 с.
4. Амелькин В. В. Дифференциальные уравнения в приложениях. 2-е изд., доп. М.: Едиториал УРСС, 2003. 208 с.
5. Бенькович Е. С. Практическое моделирование динамических систем / Е. С. Бенькович, Ю. Б. Колесов, Ю. Б. Сениченков. СПб: БХВ-Петербург, 2002. 464 с.
6. Черных И. В. SIMULINK: среда создания инженерных приложений. М.: Диалог-МИФИ, 2003. 496 с.
7. Черных И. В. Simulink: Инструмент моделирования динамических систем. URL: <http://matlab.exponenta.ru/simulink/book1/index.php>
8. Angermann A. MATLAB — Simulink — Stateflow. Grundlagen, Toolboxen, Beispiele / A. Angermann, M. Beuschel, M. Rau, U. Wohlfarth. Oldenburg Verlag München Wien, 2007. 495 p.
9. MathWorks MATLAB Documentation.  
URL: <http://www.mathworks.com/help/matlab/index.html>
10. MathWorks Getting Started with MATLAB.  
URL: <http://www.mathworks.com/help/matlab/getting-started-with-matlab.html>
11. MathWorks MATLAB Examples.  
URL: <http://www.mathworks.com/help/matlab/examples/index.html>
12. MathWorks Simulink Documentation.  
URL: <http://www.mathworks.com/help/simulink/index.html>
13. MathWorks Getting Started with Simulink.  
URL: <http://www.mathworks.com/help/simulink/getting-started-with-simulink.html>
14. MathWorks Simulink Examples.  
URL: <http://www.mathworks.com/help/simulink/examples/index.html>

*Все ссылки проверены на работоспособность 3 февраля 2013 г.*

## РАЗДЕЛ V. МАТЕМАТИЧЕСКИЕ ОСНОВЫ КОМПЬЮТЕРНОЙ АЛГЕБРЫ

### КОНСПЕКТ ЛЕКЦИЙ

Электронные версии лекций размещены в электронной библиотеке БГУ по адресу <http://elib.bsu.by/handle/123456789/9213>. Ниже приведен тематический план лекций.

#### Тема № 1. Аффинные многообразия и идеалы. Алгоритмы деления

Определение аффинного многообразия и полиномиального идеала. Базис идеала. Параметризация аффинных многообразий. Задача описания идеала. Задача о принадлежности к идеалу. Задача решения полиномиальных уравнений. Задача неявного представления. Типы упорядочений мономов в  $k[x_1, \dots, x_n]$ . Алгоритм деления в  $k[x_1, \dots, x_n]$ .

#### Тема № 2. Базисы Грёбнера

Мономиальные идеалы и лемма Диксона. Теорема Гильберта о базисе. Базисы Грёбнера, свойства базисов Грёбнера. Алгоритм Бухбергера. Минимальный базис Грёбнера, редуцированный базис Грёбнера. Усовершенствования алгоритма Бухбергера.

#### Тема № 3. Теория исключения

Теоремы об исключении и продолжении,  $k$ -й исключающий идеал. Единственность разложения на множители и результаты. Результаты и теорема о продолжении. Обобщенные результаты.



## ПЛАН ЛАБОРАТОРНЫХ ЗАНЯТИЙ

Электронные версии лабораторных работ размещены в электронной библиотеке БГУ по адресу <http://elib.bsu.by/handle/123456789/9213>. Ниже приведено краткое описание каждой работы.

### Лабораторная работа № 1. Аффинные многообразия и их рациональная параметризация

#### Задания:

1. Найти все решения системы (над полем  $R$ ) и представить их в виде многообразий.
2. Проверить, что многообразие допускает рациональную параметризацию, заданную уравнениями.
3. Решить систему (над полем  $R$ ) и параметризовать многообразие.

### Лабораторная работа № 2. Мономиальные упорядочения. Алгоритм деления в $k[x_1, \dots, x_n]$

#### Задания:

1. Упорядочить мономы в списках и в полиномах, используя для каждого из них следующие упорядочения: lex, grlex, grevlex.
2. С помощью алгоритма деления найти частные и остаток от деления полинома  $f$  на набор полиномов  $(f_1, \dots, f_n)$ . Алгоритм деления здесь необходимо запрограммировать, а результат деления проверить с помощью PolynomialReduce.

### Лабораторная работа № 3. Базисы Грёбнера и их свойства

#### Задания:

1. Найти базисы Грёбнера для заданных идеалов. Проверить результат с помощью встроенной функции GroebnerBasis.
2. Показать, что при любом порядке переменных базис Грёбнера состоит из одних и тех же элементов.
3. Найти базисы Грёбнера наибольшей и наименьшей длины, используя различный порядок переменных.

### Лабораторная работа № 4. Усовершенствованный алгоритм Бухбергера

**Задания:**

1. Найти базисы Грёбнера для заданных идеалов, используя усовершенствованный алгоритм Бухбергера. Написать процедуру, выводящую на печать базис Грёбнера, его длину и количество итераций в цикле.
2. Найти порядок переменных, при котором базис Грёбнера совпадает с заданным идеалом.

### Лабораторная работа № 5. Решение систем уравнений с помощью базисов Грёбнера

**Задание:**

Решить систему нелинейных уравнений с помощью базиса Грёбнера. Выписать все решения в виде идеалов, затем выбрать из них те, которые не повторяются и не являются подмножествами других (это можно сделать, написав алгоритм проверки на подмножества, базирующийся на встроенной функции PolynomialReduce).

### Лабораторная работа № 6. Теория исключения. Результаты

**Задание:**

Для системы нелинейных колебаний  $\dot{x} = y + hx^2 + dxy$ ,  $\dot{y} = -x + ax^2 + cy^2 + 3lx^2y$  найти условия, при которых особая точка  $O(0,0)$  является центром.

### Лабораторная работа № 7. Обобщенные результаты

**Задание:**

Для системы нелинейных колебаний  $\dot{x} = y(1 + dx + px^2)$ ,  $\dot{y} = -x + ax^2 + kx^3 + mxy^2$  найти условия, при которых центр  $O(0,0)$  является изохронным.

## СПИСОК КОНТРОЛЬНЫХ ВОПРОСОВ

1. Аффинное многообразие.
2. Полиномиальный идеал.
3. Параметризация аффинных многообразий.
4. Задача описания идеала.
5. Задача о принадлежности к идеалу.
6. Задача неявного представления.
7. Типы упорядочений мономов в  $k[x_1, \dots, x_n]$ .
8. Алгоритм деления в  $k[x_1, \dots, x_n]$ .
9. Мономиальные идеалы.
10. Лемма Диксона.
11. Теорема Гильберта о базисе.
12. Базис Гребнера и его свойства.
13. Алгоритм Бухбергера.
14. Минимальный базис Гребнера.
15. Редуцированный базис Гребнера.
16. Теоремы об исключении и продолжении.
17. Единственность разложения на множители и результаты.
18. Результаты и теорема о продолжении.
19. Обобщенные результаты.

## ПРИМЕРЫ КОНТРОЛЬНЫХ ЗАДАНИЙ

1. Найти параметризацию сферы  $x^2 + y^2 + z^2 = 1$ . При построении параметризации можно взять плоскость  $z=0$  и точку  $(0,0,1)$  и сопоставлять паре  $(u,v)$  точку на окружности  $(x,y,z)$ , которая является пересечением прямой, проведенной через точки  $(0,0,1)$  и  $(u,v,0)$ , и окружности.
2. Для упорядоченного набора  $F$  подобрать такую функцию  $f$ , чтобы при изменении порядка функций в  $F$  менялся бы остаток от деления  $f$  на  $F$  ( $x > y > z$ , grlex).
3. Докажите, что  $f \in \langle f_1, f_2 \rangle$ .
4. Используя grlex-упорядочение, найдите элемент  $g \in \langle f_1, f_2 \rangle$ , остаток от деления которого на  $(f_1, f_2)$  не равен нулю (можно взять такой элемент  $g$ , что остаток равен самому  $g$ ).
5. Пусть  $I = \langle g_1, g_2, g_3 \rangle$  является идеалом кольца  $k[x, y, z]$ . Используя lex-упорядочение, приведите пример полинома  $g \in I$ , такого, что  $LT(g) \notin \langle LT(g_1), LT(g_2), LT(g_3) \rangle$ .
6. Принадлежат ли полиномы  $f_1$  и  $f_2$  идеалу  $I = \langle g_1, g_2, g_3 \rangle$  кольца  $k[x, y, z]$ ?
7. Получить для многообразия  $V(S)$  минимальное разложение. Построить данное многообразие при помощи полученных неприводимых компонент.
8. Получить для многообразия  $V(S)$  минимальное разложение. Написать тест-функцию  $SubIdealQ[I_1, I_2, vars]$ , которая выдает True, если  $I_1 \subset I_2$  и False в противном случае. При помощи данной функции проверить, что ни одно из многообразий не является подмногообразием другого многообразия ( $V_i \not\subset V_j$  для  $i \neq j$ ).
9. Для данных систем установить тип особой точки  $(0,0)$  линеаризованной системы и сделать рисунок фазовых кривых линеаризованной системы. При помощи функции численного интегрирования  $NDSolve$  изобразить фазовый портрет нелинейной системы в окрестности особой точки и сравнить его с фазовым портретом линеаризованной системы.
10. Для заданных систем представить многообразие  $V(g_1, g_2, \dots, g_n)$  в виде объединения неприводимых многообразий ( $n$  — количество входящих в систему параметров).

## СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ

1. Садовский А. П. Полиномиальные идеалы и многообразия: пособие для студентов. Минск, БГУ, 2008. 199 с.
2. Кокс Д., Литтл Дж., О'Ши Д. Идеалы, многообразия и алгоритмы. Введение в вычислительные аспекты алгебраической геометрии и коммутативной алгебры. М.: Мир, 2000. 687 с.
3. Adams W., Loustau P. An Introduction to Gröbner Bases. Graduate Studies in Mathematics. Amer. Math. Soc. Providence, 1994. 289 p.
4. Becker T., Weispfenning V. Gröbner Bases: A Computational Approach to Commutative Algebra. Springer Verlag, Berlin and New York, 1993. 512 p.
5. Просолов В. В. Многочлены. М.: МЦНМО, 2000. 336 с.
6. Атья М., Макдональд И. Введение в коммутативную алгебру. М.: Мир. 1972. 160 с.
7. Рид М. Алгебраическая геометрия для всех. М.: Мир. 1991. 152 с.
8. Быков В. И., Кытманов А. М., Лазман М. З. Методы исключения в компьютерной алгебре многочленов. Новосибирск, Наука, 1991. 232 с.
9. Ван дер Верден Б. Л. Алгебра. М.: Наука, 1976. 648 с.
10. Курош А. Г. Курс высшей алгебры. М.: Физматгиз, 1962. 432 с.
11. Ленг С. Алгебра. М.: Мир, 1968. 564 с.

## РАЗДЕЛ VI. МАТЕМАТИЧЕСКИЕ ОСНОВЫ ЗАЩИТЫ ИНФОРМАЦИИ

### КОНСПЕКТ ЛЕКЦИЙ

Электронные версии лекций размещены в электронной библиотеке БГУ по адресу <http://elib.bsu.by/handle/123456789/9181>. Ниже приведен тематический план лекций.

#### **Лекция № 1. Введение в криптографию**

Основные определения криптографии. Классические криптосистемы: шифр Сципиона, криптосистема Цезаря, шифр Виженера.

#### **Лекция № 2. Введение в теорию сложности вычислений и основные алгоритмы теории чисел**

Полиномиальный, субэкспоненциальный и экспоненциальный алгоритмы. Алгоритм Евклида, расширенный алгоритм Евклида, возведение в степень в кольце классов вычетов. Вычисление обратных элементов в мультипликативной группе  $Z_n^*$ .

#### **Лекция № 3. Генерирование больших простых чисел**

Решето Эратосфена. Малая теорема Ферма. Псевдопростые числа. Числа Кармайкла. Свидетели простоты. Вероятностный тест на простоту Миллера-Рабина. Гипотеза Римана. Детерминированный полиномиальный алгоритм проверки простоты чисел. Детерминированный и недетерминированный алгоритмы. Алгоритмы с нулевой, односторонней и двусторонней ошибками.

#### **Лекция № 4. Факторизация чисел. Экспоненциальные алгоритмы**

Китайская теорема об остатках. Метод пробных делений. Ро-метод Полларда. Классы P и NP.

#### **Лекция № 5. Факторизация чисел. Субэкспоненциальные алгоритмы**

Факторизация Ферма и факторные базы. Метод Диксона. Метод квадратичного решета.

## **Лекция № 6. Криптосистемы с открытым ключом. Криптосистема RSA**

Криптосистемы с открытым ключом. Односторонние функции. Криптосистема RSA.

## **Лекция № 7. Рюкзачные криптосистемы**

Задача о рюкзаке. Задача о рюкзаке с быстрорастущим вектором. Рюкзачная криптосистема Меркля-Хеллмана. Алгоритм Шамира.

## **Лекция № 8. Функции хеширования**

Хеш-функции. Коллизии. Алгоритм вычисления контрольной суммы. Криптографические хеш-функции. СТБ 34.101.31-2007.

## **Лекция № 9. Дискретное логарифмирование**

Определение дискретного логарифма. Первообразный корень. Алгоритм больших и малых шагов. Алгоритм Полига-Хеллмана.

## **Лекция № 10. Криптосистема Эль-Гамала**

Криптосистема Эль-Гамала. Обмен ключами Диффи-Хеллмана.

## **Лекция № 11. Электронная цифровая подпись**

Электронная цифровая подпись. Подпись Эль-Гамала. Аутентификация Шнорра. Электронная цифровая подпись Шнорра. Электронная цифровая подпись СТБ 1176.2-99.

## **Лекция № 12. Криптосистема Рабина**

Криптосистема Рабина. Квадратичный вычет. Критерий Эйлера. Символ Лежандра. Алгоритм извлечения квадратного корня в кольце классов вычетов.

## **Лекция № 13. Эллиптические кривые**

Основные понятия и определения. Теорема Хассе. Алгоритм вычисления произведения натуральных чисел с точками эллиптической кривой. Вычисление порядка группы точек эллиптической кривой над конечным полем.

## **Лекция № 14. Криптография на эллиптических кривых**

Криптосистемы на эллиптических кривых. Электронная цифровая подпись ГОСТ Р 34.10-2001. Аналог ключевого обмена Диффи-Хеллмана.





## ПЛАН ЛАБОРАТОРНЫХ ЗАНЯТИЙ

Электронные версии лабораторных работ размещены в электронной библиотеке БГУ по адресу <http://elib.bsu.by/handle/123456789/9181>. В связи с тем, что работы набраны в пакете Mathematica, они не интегрируются в Word и поэтому ниже приведено лишь их краткое описание. Численные значения переменных смотри в электронной версии.

### Лабораторная работа № 1. Введение в криптографию

#### Задание 1:

Реализовать функции шифрования  $En[M\_String, k\_]$  и дешифрования  $De[c\_String, k\_]$  криптосистемы Сциналь. Здесь  $M$  — открытый текст (message),  $c$  — шифртекст (ciphertext),  $k$  — ключ (key), указывающий на количество строк в матрице при шифровании.

#### Задание 2:

Криптоаналитик осуществляет вскрытие шифра, основанного на операции перестановки. Известно, что шифрующая функция шифрует сообщения длиной  $L = 15 + \text{Mod}[4V, 15]$ , где  $V$  — номер варианта студента (выдается преподавателем, ведущим лабораторные занятия). Также криптоаналитику доступна функция шифрования  $En(M)$  (но считается, что не доступно тело функции) и он может шифровать любое количество сообщений. От криптоаналитика требуется получить ключ (ключом является перестановка), написать дешифрующую функцию и найти порядок перестановки.

#### Задание 3:

Криптоаналитик перехватил два шифртекста  $C1$  и  $C2$ , зашифрованных при помощи криптосистемы Цезаря. Необходимо вскрыть два шифртекста  $C1$  и  $C2$  (ключи не известны). Определить ключи.

#### Задание 4:

Криптоаналитик перехватил зашифрованный текст  $C$ , полученный при помощи криптосистемы Виженера с длиной ключевого слова  $N$  символов. Найти ключевое слово  $k$  и открытый текст  $M$ .

### Лабораторная работа № 2. Кольцо классов вычетов

#### Задание 1:

Построить таблицы сложения и умножения в кольце класса вычетов  $Z_n$  и  $Z_m$ .

**Задание 2:**

Вычислить по определению  $\varphi(n_1)$  (подсчитать количество натуральных чисел, меньших  $n_1$  и взаимно простых с  $n_1$ ),  $n_1 = 10^6 + N \cdot 1$  ( $N$  — вариант), при помощи функции Timing определить время вычислений.

Вычислить функцию Эйлера для чисел  $n_1$ ,  $n_2 = 21^{100+N}$ ,  $n_3$  двумя способами: при помощи формулы

$$\varphi(n) = n \left(1 - \frac{1}{p_1}\right) \left(1 - \frac{1}{p_2}\right) \dots \left(1 - \frac{1}{p_r}\right) \quad (\text{при условии что } n = p_1^{m_1} p_2^{m_2} \dots p_r^{m_r})$$

и встроенной функции EulerPhi. Сравнить время вычислений при помощи функции Timing. Для какого из трех чисел  $n_1, n_2, n_3$  значение функции Эйлера вычисляется дольше всего? Почему?

**Задание 3:**

В кольцах  $Z_n$  и  $Z_m$  ( $n$  и  $m$  из первого задания) найти пары взаимно обратных относительно умножения элементов. Какие из данных колец являются полями?

**Задание 4:**

Реализовать алгоритм вычисления  $a^d \pmod{n}$ . При помощи функции Timing сравнить скорость вычислений со встроенной функцией PowerMod.

**Задание 5:**

Вычислить, где это возможно, обратные по умножению элементы к 5, 6 и 7 в кольцах  $Z_{n_1}$ ,  $Z_{n_2}$ ,  $Z_{n_3}$  (числа  $n_1, n_2, n_3$  из второго задания) при помощи:

- 1) Теоремы Эйлера (для возведения в степень используйте полученную в Задании 4 функцию);
- 2) Расширенного алгоритма Евклида (реализовать расширенный алгоритм Евклида);
- 3) Встроенной функции PowerMod[a,-1,n].

При помощи функции Timing определите самый быстрый способ.

**Задание 6:**

- 1) В кольцах  $Z_{n_1}$  и  $Z_{n_2}$  найти делители нуля;
- 2) Выписать все элементы мультипликативных групп  $Z_{n_1}^*$ ,  $Z_{n_2}^*$ . Сравнить их количество с  $\varphi(n_1)$ ,  $\varphi(n_2)$ ;
- 3) Найти все образующие в группах  $Z_{n_1}^*$ ,  $Z_{n_2}^*$  и сравнить их количество  $\varphi(n_1 - 1)$ ,  $\varphi(n_2 - 1)$ ;
- 4) Являются ли группы  $Z_{n_1}^*$ ,  $Z_{n_2}^*$  циклическими?

### Лабораторная работа № 3. Генерирование больших простых чисел

#### Задание 1:

При помощи метода пробных делений выяснить, является ли число  $n$  простым, найти время вычисления при помощи функции `Timing`.

#### Задание 2:

При помощи малой теоремы Ферма доказать, что число  $n$  составное. Попробуйте разложить число  $n$  на простые множители при помощи встроенной функции `TimeConstrained[FactorInteger[n],60]`.

#### Задание 3:

Вычислить первое число Кармайкла, следующее за числом  $10^6 + N \cdot 10^5$  ( $N$  — вариант). Для полученного числа  $K$  выяснить, является ли оно числом вида  $K = p_1 p_2 \dots p_r, r > 2$ , где все простые числа  $p_i$  — различны, причем  $K-1$  делится на каждую разность  $p_i - 1$ .

#### Задание 4:

Написать функцию `WitnessQ[a_Integer, n_Integer]`, которая возвращает `True`, если число  $a$  является свидетелем простоты числа  $n$  и `False` в противном случае.

#### Задание 5:

Реализовать вероятностный тест на простоту Миллера-Рабина, который для числа  $n$  возвращает `True`, если сто наугад взятых чисел  $a$  оказались свидетелями простоты числа  $n$ . Сравнить скорость вычислений со встроенной функцией `PrimeQ`.

#### Задание 6:

Используя вероятностный тест на простоту Миллера-Рабина, написать функцию `PrimeGeneration[x_Integer]`, которая возвращает случайное простое число из промежутка  $[1, x]$ .

### Лабораторная работа № 4. Факторизация целых чисел.

#### Экспоненциальные методы

#### Задание 1:

Реализовать функцию, которая методом пробных делений находит все простые множители натурального числа  $n$ . При желании, формат вывода функции `TrialDivision` можно сделать таким же, как у функции `FactorInteger`.

**Задание 2:**

Написать функцию, которая находит все нетривиальные простые делители натурального числа  $n$  методом пробных делений с обратным отсчетом от  $\sqrt{n}$  до 2.

**Задание 3:**

Разложить на множители число  $n$  при помощи функций TrialDivisionReverse и FactorInteger. Сравнить время вычислений.

**Задание 4:**

Реализовать  $\rho$ -метод Полларда. Найти делитель чисел  $n_1$  и  $n_2$  и сравнить время его вычисления, объяснить, от чего оно зависит.

**Задание 5:**

Пусть  $\lambda = 2$ ,  $p$  — наименьший простой делитель числа  $n$ . Для ста случайных пар  $(f(x), x_0)$  вида  $f(x) = x^2 + \text{Random}[\text{Integer}, \{1, n-1\}]$ ,  $x_0 = \text{Random}[\text{Integer}, \{1, n-1\}]$  подсчитать сколько из них благоприятствуют событиям  $A, B, A \cap \bar{B}$ . Сравнить полученные результаты с  $P(A), P(B), P(A \cap \bar{B})$ . Насколько пара  $(f(x), x_0)$  действительно задает случайную последовательность  $x_0, x_1, \dots, x_{4L}$ . Провести вычисления для чисел  $n_1, n_2$ . Чем между собой отличаются результаты и почему?

**Задание 6:**

Пусть  $f(x) = x^2 + 1, x_0 = 1, x_k = f(x_{k-1}) \pmod n$ . Для чисел  $n_1, n_2$  выяснить, найдутся ли такие  $i_0, j_0$ , что  $1 < \gcd(x_j - x_i, n) < n$ .

**Лабораторная работа № 5. Факторные базы. Алгоритм Диксона****Задание 1:**

Реализовать алгоритм Диксона, использующий В-гладкие числа по положительному вычету, включающий в себя следующие процедуры:

- 1) процедура, вычисляющая факторную базу исходя из  $n, a$ ;
- 2) процедура, тестирующая число  $b$  на В-гладкость для данной факторной базы  $B$  и числа  $n$ ;
- 3) процедура, случайным образом вычисляющая  $k+1$  В-гладкое число  $b_1, b_2, \dots, b_{k+1}$  и соответствующие им степени  $v_1, v_2, \dots, v_{k+1}$ ;
- 4) алгоритм Диксона, вычисляющий нетривиальный делитель числа  $n$ .

**Задание 2:**

При помощи функции Timing определите значение параметра  $a$ , при котором  
 а) для числа  $n_1$ ; б) для числа  $n_2$  тратится меньше всего времени на:

- 1) вычисление  $k+1$  В-гладкого числа;
- 2) работу алгоритма Диксона.

**Задание 3:**

Реализовать алгоритм Диксона, использующий В-гладкие числа по абсолютному вычету, при формировании факторной базы первым элементом взять  $-1 (p_1 = -1)$ . Сравнить его время вычисления с алгоритмом Диксона, использующим В-гладкие числа по положительному вычету.

**Задание 4:**

Пусть  $m := L[n]^{2/3}$ ,  $u := \frac{\text{Log}[n]}{\text{Log}[m]}$ . Подсчитать, сколько из  $10^4$  случайных чисел из промежутка  $1 \leq b \leq n-1$  являются В-гладкими по положительному вычету (обозначим это количество через  $q$ ) и сколько В-числами по абсолютному вычету ( $qA$ ). Сравнить числа  $u^{-u}, \frac{q}{10^4}, \frac{qA}{10^4}$ . Выполнить вычисления для  $n_1, n_2$ .

**Задание 5:**

Пусть факторная база  $V$  и числа  $n_1, n_2$  такие же, как в предыдущем задании. Сколько из случайно взятых  $10^4$  чисел из  $Z_n$  являются В-гладкими по вычету 1? Сколько из них не являются ни В-гладкими по положительному вычету, ни В-гладкими по абсолютному вычету?

## Лабораторная работа № 6. Криптосистема RSA

**Задание 1:**

Написать функцию генерации ключей `GenerationKey[order_Integer]`, которая возвращает ключи `PublicKey`, `PrinateKey` криптосистемы RSA, где  $n = p \cdot q$ , простые числа  $p$  и  $q$  из интервала  $1 < p, q < \text{order}$  генерируются при помощи изученного нами ранее алгоритма Миллера-Рабина.

**Задание 2:**

Задать функцию `EnRSA[m_Integer, PublicKey_]`, которая шифрует число  $m$ ,  $0 < m < n$ , при помощи открытого ключа `PublicKey = {e, n}`.

**Задание 3:**

Определить дешифрующую функцию  $\text{DeRSA}[c\_Integer, \text{PublicKey\_}, \text{PrivateKey\_}]$ , возвращающую число  $m$ .

**Задание 4:**

Числа  $p$  и  $q$  при генерации ключей необходимо выбирать не близкими друг к другу, иначе рассмотренными ранее способами их можно найти. Взломать шифр  $c$ , полученный криптосистемой RSA с открытым ключом  $\text{PublicKey} = \{e, n\}$ .

**Задание 5:**

Найти открытый текст  $m$ , если известно, что он шифровался при помощи ключей  $\{e1, n\}$  и  $\{e2, n\}$  и получились шифры  $c1$  и  $c2$  соответственно:

- 1)  $c1 = \text{EnRSA}[m, \{e1, n\}]$ ,
- 2)  $c2 = \text{EnRSA}[m, \{e2, n\}]$ .

**Лабораторная работа № 7. Рюкзачные криптосистемы****Задание 1:**

Реализовать алгоритм генерации случайного быстрорастущего вектора.

**Задание 2:**

Реализовать алгоритм решения задачи о рюкзаке для быстрорастущего вектора.

**Задание 3:**

Реализовать алгоритмы генерации ключей, шифрования и дешифрования для криптосистемы Меркля-Хелмана.

Изучается криптоанализ при помощи LLL-алгоритма.

**Лабораторная работа № 8. Функции хеширования****Задание 1:**

Написать функцию  $y = \text{CRC}(m)$ , которая вычисляет контрольную сумму  $r = (r_{n-1}, r_{n-2}, \dots, r_0)$  сообщения  $m = (a_k, a_{k-1}, \dots, a_0)$ ,  $a_i \in \{0, 1\}$ , и выдает  $y = m \parallel r = (a_k, \dots, a_0, r_{n-1}, \dots, r_0)$ . Реализовать функцию  $\text{Ver}(y)$ , которая выдает True, если многочлен, соответствующий сообщению  $y$ , делится на примитивный многочлен  $q(x)$  и False в противном случае. Сравнить хеш-коды сообщений  $m_1$  и  $m_2$ . Вычислить  $\text{Ver}(y_1)$  и  $\text{Ver}(y_2)$ . При реализации данных функций удобно

пользоваться следующими встроенными функциями Exponent, PolynomialReduce, PadLeft, Reverse, CoefficientList.

**Задание 2:**

Для сообщения  $m_1$  из Задания 1, используя тот же примитивный многочлен  $q(x)$ , найти 5 коллизий, первый бит которых равен 1.

**Задание 3:**

Изучить блочную симметричную криптосистему согласно документу СТБ 34.101.31-2011.pdf (стр. 1-10) и ее реализацию.

**Задание 4:**

Реализовать функцию хеширования, описание которой находится на стр. 18. Поразмышлять о ее криптостойкости. Попробовать найти причину того, что данная криптосистема (а с ней и хеш-функция) работает очень медленно.

### Лабораторная работа № 9. Дискретный логарифм

**Задание 1:**

Написать функцию ExhaustiveSearch[a\_,b\_,n\_], вычисляющую дискретный логарифм  $\log_a b$  методом полного перебора всех возможных решений от 1 до  $\varphi(n)$ . Подобрать такие числа  $a$ ,  $b$ ,  $n$ , чтобы дискретный логарифм не существовал. Найти такие  $a, n$ , чтобы дискретный логарифм имел единственное решение. Написать функцию ExhaustiveSearchAll[a\_,b\_,n\_], которая находит все решения уравнения  $a^x \equiv b \pmod{n}$  в кольце  $Z_{\varphi(n)}$ . Подобрать такие числа  $a$ ,  $b$ ,  $n$ , чтобы уравнение имело  $V$  ( $V$  — номер варианта) различных в кольце  $Z_{\varphi(n)}$  решений.

**Задание 2:**

Написать функцию PrimitiveQ[a\_,p\_], которая определяет, является ли  $a$  образующим группы  $Z_p^*$ . На основе данной функции написать функцию, вычисляющую случайный образующий элемент группы  $Z_p^*$ .

**Задание 3:**

Реализовать алгоритм согласования двумя способами 1) и 2), подобрать такие числа  $a, p, b_1$  и  $a, p, b_2$ , чтобы дискретный логарифм  $\log_a b_1$  быстрее вычислялся первым способом, а  $\log_a b_2$  быстрее вычислялся вторым способом.

**Задание 4:**

Реализовать алгоритм Полига-Хелмана. Эмпирически сравнить его скорость вычислений с алгоритмом больших и малых шагов.

**Задание 5:**

Для указанных ниже чисел  $a$ ,  $b$ ,  $p$  вычислить одним из способов дискретный логарифм  $\log_a b$ .

**Лабораторная работа № 10. Криптосистема и электронная цифровая подпись Эль-Гамала****Задание 1:**

Написать функцию  $\text{GenKey}[n\_]$ , случайным образом генерирующую ключ  $k := \{k_e, k_d\}$ .

**Задание 2:**

Реализовать функции  $\text{EncipherElGamal}[m\_ , k_e : \{y\_ , g\_ , p\_ \}]$ ,  $\text{DecipherElGamal}[\{a\_ , b\_ \}, k_e : \{y\_ , g\_ , p\_ \}, k_d : x\_]$ , которые соответственно шифруют сообщение  $m$  и расшифровывают шифр  $\{a, b\}$ .

**Задание 3:**

Получить текст, зашифрованный криптосистемой ElGamal с ключами  $k_e = K1, k_d = K2$  (не забудьте к полученному списку чисел применить функцию  $\text{FromBlock}$ ).

**Задание 4:**

Взломать криптошифр  $\{a, b\}$  (использовать алгоритм Полига-Хеллмана и функцию  $\text{FromBlock}$ ).

**Задание 5:**

Реализовать функции  $G[n]$ ,  $S[m, k_e, k_d]$ ,  $V[m, s, k_e]$ .

**Задание 6:**

Для сообщения  $m$  получена подпись  $s = S[m, k_e, k_d]$  ( $V[m, s, k_e] = \text{True}$ ). Пользуясь тем, что  $g$  не является образующим группы  $Z_p^*$ , заменить сообщение  $m$  сообщением  $m2$  ( $m2 < p$ ) таким образом, чтобы тест на достоверность подписи все равно выдавал  $V[m2, s, k_e] = \text{True}$  с той же самой подписью  $s$ . ( $k_e = K1$ )

**Задание 7:**

Проверить подпись  $s$  на соответствие открытым ключам  $k_{e1}, k_{e2}, k_{e3}, k_{e4}$  (вычислить  $V[m, s, k_{ei}]$ ). Объяснить противоречие, то есть почему не выполняется



требование  $V(m, S(m, k_e, k_d), k_e) = 1 \Leftrightarrow k = (k_e, k_d)$  — ключ электронной цифровой подписи.

### Задание 8:

Подписать сообщение  $m := 100v$  ( $v$  — номер варианта) за абонента с ключом  $k_e := (y, g, p)$ ,  $y := v+1$ , если известно такое  $k$ , что  $g^k \equiv \frac{p-1}{2} \pmod{p}$ ,  $(k, p-1) = 1$ . То есть найти такую пару  $s=(a,b)$ , чтобы  $V(m, s, k_e) = \text{True}$ . Проверить справедливость полученной подписи  $s$  при помощи функции  $V$ . Каким образом стоит модифицировать функцию  $V$ , чтобы она выдавала  $\text{True}$  лишь в случае, когда подпись получена при помощи секретного ключа  $k_d$ .

## Лабораторная работа № 11. Электронные цифровые подписи Шнорра и СТБ 1176.2-99

### Задание 1:

Реализовать функции генерации ключей  $G[n]$ , выработки подписи  $s = S[m, k_e, k_d]$ , проверки подписи  $V[m, s, k_e]$  электронной цифровой подписи Шнорра. В качестве функции хеширования использовать СТБ 34.101.31-2007  $h[a, m, q]$ .

### Задание 2:

Проверить функцией  $V$  подпись  $s$  сообщения  $m$  для различных значений  $y$ . Что можно сказать о криптостойкости в данном случае? Какая особенность подписи  $s$  повлияла на криптостойкость?

### Задание 3:

Подписать сообщение  $m$  подписью абонента с открытым ключом  $k_e$  (найти такое  $s$ , чтобы  $V[m, s, k_e] = \text{True}$ ).

### Задание 4:

Реализовать функции генерации ключей  $G_{STB}[n]$ , выработки подписи  $s = S_{STB}[m, k_e, d_d]$ , проверки подписи  $V_{STB}[m, s, k_e]$  электронной цифровой подписи СТБ 1176.2-99. В качестве функции хеширования использовать СТБ 34.101.31-2007  $h[a, m, q]$ .

## Лабораторная работа № 12. Квадратичные вычеты. Криптосистема Рабина

### Задание 1:

Реализовать Алгоритм Шенкса. Сколько корней может иметь уравнение

$$x^2 \equiv a \pmod{n} \quad (*)$$

для простого  $n$ ?

### Задание 2:

Сколько различных корней в  $Z_n$  может иметь уравнение (\*) при  $n=pq$ ,  $p, q$  — простые? Посмотреть эмпирически, сколько различных корней в  $Z_n$  может иметь уравнение (\*) при  $n=p^a$ ,  $p$  — простое?

Подобрать такие  $a$  и  $n$ , чтобы уравнение (\*) имело:

- 1) 2 Prime[V+10] корней;
- 2) 4 Prime[V+10] корней;
- 3) Prime[V+5] Prime[V+6] корней.

Здесь  $V$  — номер варианта.

### Задание 3:

Реализовать три алгоритма генерации ключей, шифрования и дешифрования криптосистемы Рабина.

### Задание 4:

Известны шифры  $c_1, c_2, c_3$  сообщений  $m_1, m_2, m_3$  соответственно, полученные при помощи криптосистемы Рабина с открытым ключом PublicKey. Найти хотя бы одно из сообщений  $m_1, m_2, m_3$ .

## Лабораторная работа № 13. Эллиптические кривые

### Задание 1:

При помощи функции ImplicitPlot пакета ImplicitPlot построить три эллиптические кривые для случаев  $D>0$ ,  $D=0$ ,  $D<0$ , где  $D$  — дискриминант кривой.

### Задание 2:

Реализовать функцию, которая вычисляет все точки эллиптической кривой.

### Задание 3

Реализовать функцию сложения в группе точек эллиптической кривой.

**Задание 4:**

Найти порядок эллиптических кривых  $\mathcal{E}_1, \mathcal{E}_2, \mathcal{E}_3$ . Что можно сказать о цикличности группы  $\mathcal{E}_3$  исходя из его порядка. Определить, являются ли циклическими группы  $\mathcal{E}_1, \mathcal{E}_2$ .

**Задание 5:**

Реализовать алгоритм вычисления  $dp \in \mathcal{E}(k)$ .

## СПИСОК КОНТРОЛЬНЫХ ВОПРОСОВ

1. Что такое криптография?
2. Что такое криптосистема?
3. Сформулировать определение полиномиального, экспоненциального и субэкспоненциального алгоритмов.
4. Нахождение обратного относительно умножения элемента в кольце классов вычетов.
5. Возведение в степень в кольце классов вычетов.
6. Теорема Эйлера и малая теорема Ферма.
7. Числа Кармайкла.
8. Тест на простоту Миллера-Рабина.
9. Детерминированный и вероятностный алгоритмы.
10. Классы сложности  $P$ ,  $NP$ .
11. Классы сложности  $NP$ -hard и  $NP$ -complete.
12. Парадокс дней рождения.
13.  $p$ -метод Полларда.
14. Китайская теорема об остатках.
15. Односторонние функции и односторонние функции с лазейкой.
16. Криптосистемы с открытым ключом.
17. Криптосистема  $RSA$ .
18. Задача о рюкзаке.
19. Задача о рюкзаке для быстрорастущего вектора.
20. Рюкзачная криптосистема Меркля-Хеллмана.
21.  $NP$ -полнота задачи о рюкзаке.
22.  $LLL$ -атака на рюкзачную криптосистему Меркля-Хеллмана.

## ПРИМЕРЫ КОНТРОЛЬНЫХ ЗАДАНИЙ

1. Найти сообщение, если известен шифртекст «тВ1.а наир», полученный перестановкой.
2. Сравнить остатки от деления чисел  $2011^{1200}$  и  $2012^{2014}$  на 2013.
3. Текст ШИЛОАПКОЦАМТАЛАТЕЯЯАССЮЕБЬБЬНРПИМЕТТКМАЧНУ получен из исходного сообщения перестановкой его букв.  
Текст ЖКПТГНЗЙСЭЦЫПЦЗЙПЦЗИЖЦЕЮБЯЦЫЩТЙЪИЦЕГЭКЦЗГ получен из того же исходного сообщения заменой каждой буквы на другую букву так, что разные буквы заменены разными, а одинаковые — одинаковыми. Восстановите исходное сообщение.
4. Каким образом распилить 15 кг на 4 гири, чтобы на чашечных весах можно было отмерить любой натуральный вес от 1 до 15 кг.
5. Каким образом распилить 40 кг на 4 гири, чтобы на чашечных весах можно было отмерить любой натуральный вес от 1 до 40 кг.
6. Рассмотрим преобразование цифрового текста, в котором каждая цифра заменяется остатком от деления значения многочлена  $F(x) = b(x^3 + 7x^2 + 3x + a)$  на число 10, где  $a, b$  — фиксированные натуральные числа. Выясните, при каких значениях  $a, b$  указанное преобразование может быть шифрпреобразованием, т. е. допускает однозначное расшифрование.
7. Одна фирма предложила устройство для автоматической проверки пароля. Паролем может быть любой непустой упорядоченный набор букв в алфавите  $\{a, b, c\}$ . Будем обозначать такие наборы большими латинскими буквами. Устройство перерабатывает введенный в него набор  $P$  в набор  $Q = \varphi(P)$ . Отображение  $\varphi$  держится в секрете, однако про него известно, что оно определено не для каждого набора букв и обладает следующими свойствами. Для любого набора букв  $P$ :
  - 1)  $\varphi(aP) = P$ ;
  - 2)  $\varphi(bP) = \varphi(P)a\varphi(P)$
  - 3) набор  $\varphi(cP)$  получается из набора  $\varphi(P)$  выписыванием букв в обратном порядке.
 Устройство признает предъявленный пароль верным, если  $\varphi(P) = P$ . Например, трехбуквенный набор  $bab$  является верным паролем, так как  $\varphi(bab) = \varphi(ab)a\varphi(ab) = bab$ .  
Подберите верный пароль, состоящий более чем из трех букв.

## СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ

### Основная:

1. Алферов А. П. Основы криптографии. Учебное пособие, 2-е изд., испр. и доп. / А. П. Алферов, А. Ю. Зубов, А. С. Кузьмин, А. В. Черемушкин. М.: Гелиос АРВ, 2002. 480 с.
2. Василенко О. Н. Теоретико-числовые алгоритмы в криптографии. М.: МЦНМО, 2003. 328 с.
3. Введение в криптографию / Под ред. В. В. Ященко. М.: МЦНМО-ЧеРоб, 1998. 271 с.
4. Коблиц Н. Курс теории чисел и криптографии. М.: Научное изд-во ТВП, 2001. 254 с.
5. Романец Ю. В. Защита информации в компьютерных системах и сетях / Ю. В. Романец, П. А. Тимофеев, В. Ф. Шаньгин. М.: Радио и связь, 2001. 376 с.
6. Тилборг Х. К. А. ван. Основы криптологии. М.: Мир, 2006. 471 с.
7. Харин Ю. С. Математические основы криптологии / Ю. С. Харин, В. И. Берник, Г. В. Матвеев. Минск: БГУ, 1999. 319 с.
8. Харин Ю. С., Агиевич С. В. Компьютерный практикум по математическим методам защиты информации. Минск: БГУ, 2001. 190 с.
9. Шнайер Б. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си. 2-е издание. М.: Триумф, 2002. 816 с.

### Дополнительная:

10. Menezes A. Handbook of Applied Cryptography / A. Menezes, P. van Oorschot and S. Vanstone. CRC Press, 1997. 780 p.
11. Болотов А. А. Элементарное введение в эллиптическую криптографию: Алгебраические и алгоритмические основы / А. А. Болотов, С. Б. Гашков, А. Б. Фролов, А. А. Часовских. М.: КомКнига, 2006. 328 с.
12. Дориченко С. А. 25 этюдов о шифрах / С. А. Дориченко, В. В. Ященко. М.: ТЕИС, 1994. 69 с.
13. Кнут Д. Искусство программирования. Т. 2. Получисленные алгоритмы. 3-е издание. М.-СПб.-Киев: Вильямс, 2003.
14. Нестеренко Ю. В. Теория чисел: учебник для студ. высш. учеб. заведений. М.: Издательский центр «Академия», 2008. 272 с.
15. Острик В. В., Цфасман А. М. Алгебраическая геометрия и теория чисел: рациональные и эллиптические кривые. М.: МЦНМО, 2001. 48 с.
16. Саломаа А. Криптография с открытым ключом. М.: Мир, 1996. 320 с.
17. Птицын Н. Приложение теории детерминированного хаоса в криптографии. М.: МГТУ им. Н.Э. Баумана, 2002. 80 с.

## РАЗДЕЛ VII. МЕТОДЫ МОДЕЛИРОВАНИЯ СЛОЖНЫХ СИСТЕМ И ПРОЦЕССОВ

### КОНСПЕКТ ЛЕКЦИЙ

#### Лекция № 1. Профессия Системный аналитик. Понятие и перспективы карьеры ИТ-специалиста

Как сегодняшнему студенту получить интересную работу, которая бы также приносила стабильный доход? Большинство выпускников ММФ БГУ связывают свое будущее с профессиями ИТ-сектора. Среди этого множества молодых профессий можно выделить совсем юные, зародившиеся в начале нового тысячелетия — это профессии, в названиях которых содержится ключевое слово Аналитик.

##### Развитие профессий ИТ-сектора

50-е — 80-е годы двадцатого века. Профессии, связанные с программным обеспечением ЭВМ. Революционное появление IBM PC. Активное развитие профессий компьютерного сектора. Трансформация компьютерной индустрии. Качественные изменения бизнеса в ИТ-индустрии.

Становление огромного рынка → снижение стоимости ПК → повышение доступности ПК → развитие новых направлений ИТ.

Необходимость ведения бизнеса с использованием компьютерных систем. Роль ИТ в бизнесе 90-х. Распространение Интернет-технологий. Эра стартапов. Появление новых массово потребляемых ИТ-профессий. Интерактивность технологии web2.0, возможность самовыражения пользователей.

Приобретение бизнесом удобного инструмента для своего совершенствования и защиты. ИТ как средство достижения цели бизнеса. Потребность в управлении бизнесом с привлечением ИТ-сферы. Рождение аналитиков в сфере ИТ.

##### Кто же такой системный аналитик?

Квалификационная характеристика Системного аналитика. Аналитики в сфере информационных технологий, их роли в бизнесе. Задачи, решаемые аналитиками. Ключевая роль системного аналитика в проекте автоматизации компании.

**Особенности профессии.** Типичные функциональные обязанности системного аналитика. Плюсы и минусы профессии. Место работы. Важные качества. Ключевые знания и навыки. Где готовят специалистов? Оплата труда. Ступеньки

карьеру и перспективы. Трактование отличий профессии системного аналитика и бизнес-аналитика.

## **Лекция № 2. Понятие системного анализа и его место в современном мире**

### **Предпосылки возникновения системного анализа**

Исторические корни системного анализа. Системный анализ как эффективное средство решения широкого круга практических задач. Возникновение системного анализа как самостоятельной отрасли знаний. Концепция управления посредством использования систематизированных аналитических проработок. Междисциплинарная группа, ее задачи. Появление новых научных дисциплин, в них - использование метода моделирования, применение математического аппарата, заимствование понятий и методов точных и технических наук. Системный анализ как новая научно-прикладная дисциплина, основанная на системном подходе к рассмотрению изучаемых объектов и явлений.

### **Системный анализ и системный подход в управлении**

Системный анализ как методология распознавания и решения системных проблем. Характерные черты (признаки) системных проблем. Системный подход к постановке и решению проблемы. Сущность системного подхода. Переход от частных моделей к общей концепции. Системный подход как новый тип управленческого мышления. Реализация системного подхода к решению политических, социально-экономических, технических и др. проблем в различных сферах человеческой деятельности.

Содержание понятия «системный анализ» и область его применения.

Поиск решения проблемы через определение и упорядочение целей деятельности системы, при функционировании которой возникла данная проблема. Установка соответствия между этими целями, возможными путями решения возникшей проблемы и потребными для этого ресурсами. Полная и всесторонняя проверка различных вариантов действий с точки зрения количественного и качественного сопоставления затраченных ресурсов с получаемым эффектом.

Слабоструктурированные проблемы, их характеристика. Задачи системного анализа при решении таких проблем. Содержание объекта системного анализа в теоретическом и прикладном аспекте. Место системного анализа среди других родственных ему научных направлений, других системных дисциплин.



## **Системный анализ и другие системные дисциплины**

Различие понятий «системный анализ» и «системный подход». Дисциплины Исследование операций, Системотехника, Общая теория систем и Системный анализ — общее и отличия. Области применения системного анализа. Цели применения методов системного анализа в промышленности и в коммерческой области.

### **Лекция № 3. Базовые понятия системного анализа**

#### **Системы и их классификация**

Понятие системы. Цель системы. Система целей. Системность окружающего мира. Взаимозависимость поведения элементов системы и системы в целом. Свойство эмерджентности системы.

Элементарная система. Характеристика ее основных частей. Три вида элементов входа, их характеристики. Классификация систем по различным критериям. Понятие большой системы, используемое для задач управления экономикой. Управление системами. Принцип обратной связи. Основные функции обратной связи.

Жизненный цикл системы. Жизненный цикл проекта по разработке поддержке программного обеспечения. Инициализация. Анализ. Проектирование. Реализация. Тестирование. Внедрение. Сопровождение. Завершение.

#### **Понятие модели**

Определение модели. Основные свойства модели. Точка зрения. Цели и задачи. Связь с моделируемой системой. Признаки, по которым возможна классификация моделей. Типы моделей. Нотации, их виды, основные требования к нотации модели.

#### **Общепринятые принципы моделирования**

Принцип корректности. Принцип релевантности. Принцип соизмеримости затрат и выгод. Принцип прозрачности. Принцип сравнимости. Принцип систематизированной структуры. Характеристики каждого принципа моделирования.

#### **Принципы и виды моделей жизненного цикла ПО**

Каскадная модель. Основные этапы. Определений требований. Планирование системного и программного обеспечения. Реализация и тестирование отдельных модулей. Интеграция и тестирование системы. Эксплуатация и сопровождение.

Итерационная разработка. Цикл Деминга. Планирование — Реализация — Проверка — Оценка. Методология RUP разработки программного обеспечения.

Шесть основных принципов разработки. Девять процессных областей для классификации задач. Жизненный цикл разработки RUP. Начало, уточнение, построение, внедрение. Гибкие методологии разработки ПО. Экстремальное программирование. Scrum-методология.

### **Понятие о CASE-средствах**

Определение, назначение, история развития, виды, примеры CASE-систем.

## **Лекция № 4. Методология SADT структурного анализа**

### **Понятие структурного анализа**

Структурный анализ как разновидность системного анализа. Основная идея структурного анализа. Цель структурного анализа. Задачи структурного анализа. Переход от системы к структуре. Этапы процесса структуризации системы. Множественность структур системы. Декомпозиция, или детализация. Глубина декомпозиции. Методология SADT структурного анализа, ее основоположники.

### **Основные идеи SADT-методологии**

SA-блок как универсальная единица универсальной пунктуации для неограниченного строго структурного анализа. Совокупность блоков и дуг. Декомпозиция для облегчения восприятия модели.

Роли: SA-авторы, SA-читатели, SA-библиотекари. Организация цикла автор/читатель для непрерывной проверки качества модели.

Взаимоотношения методологии SA и других форм передачи информации. Соотношение SA с имитационным моделированием и базами данных. Два основных направления в SA-моделировании: функциональные модели и модели данных. Области использования SA.

### **Синтаксис диаграмм**

Процесс создания модели на примере Экспериментального механического цеха. Постановка задачи. Краткое описание процесса. Попутный список вопросов. Формулировка цели модели. Список вопросов как детализация цели и контроль качества результата. Степень точности модели, ее достижимость и измеримость.

Связь модели с окружающей средой. Определение границ системы. Рассмотрение модели с одной и той же позиции. Точкой зрения модели. SADT-модель как описание системы, у которого есть единственный субъект, цель и одна точка зрения.

Синтаксические правила создания диаграмм. Поддержка сложности диаграмм и модели на уровне, доступном для чтения, понимания и использования.

Определенное назначение каждой стороны SA-блока. Основной принцип: входы преобразуются в выходы, управление ограничивает или предписывает условия выполнения преобразований, механизмы показывают, кем, что и как выполняет функция.

Принцип размещения блоков на диаграмме, принцип доминирования. Роль номеров блоков. Дуга как множество объектов. Правила именования блоков и дуг. Четыре отношения между объектами и функциями. Пять типов взаимосвязей между блоками для описания их отношений. Разветвление дуг. Слияние дуг.

С-номерные коды диаграмм при итерационном процессе моделирования.

### **Синтаксис SADT-моделей**

Понятие SADT-модели. Назначение синтаксиса SADT-моделей. Процесс согласования диаграмм. Родительская диаграмма, родительский блок, диаграмма-потомок. Принцип ограничения объекта моделирования. Контекстная диаграмма. Нумерация блока самого верхнего уровня модели.

### **Идентификация декомпозиции номерами узлов**

Развитие SADT-модели в процессе структурной декомпозиции сверху вниз. Согласование имени диаграммы и названия декомпозируемого блока. Идентификатор диаграммы.

### **Связывание декомпозиции с помощью С-номеров**

Назначение С-номеров. «Направленная вниз» связь от родительской диаграммы к диаграмме-потомку. Ссылка на родительскую диаграмму. Нотация ссылок. С-номера как контроль за введением новых диаграмм в иерархию модели.

### **Коды ICOM**

Гарантия правильного соединения всех диаграмм для образования согласованной модели. Внешние дуги как интерфейс между данной диаграммой и остальной частью модели. Согласование по числу и наименованию (но не обязательно по расположению) с дугами, касающимися декомпозированного блока родительской диаграммы.

Система обозначений, позволяющая аналитику точно идентифицировать и проверять связи по дугам между диаграммами. Схема кодирования дуг ICOM для идентификации и проверки связи по дугам между диаграммами. Правила введения кодов ICOM.

## **Организация процесса моделирования**

SADT как методология, ее возможности. Итеративный процесс моделирования в SADT, описанный с помощью SADT-диаграммы. Роли экспертов, авторов, библиотекаря, читателей, Комитет технического контроля модели.

## **Получение знаний в процессе опроса**

Методики сбора информации - опросы или интервью. Источники информации. Графический язык SADT как средство для заметок, которые служат основой для построения диаграмм.

## **Документирование полученных знаний**

Создание модели. Документирование полученных знаний о проблемной области, представление их в виде одной или нескольких SADT-диаграмм. Метод детализации ограниченного субъекта. Анализ объектов, входящих в систему, анализ функций системы. Создание диаграмм, объединение сходных объектов и функций.

## **Итеративное рецензирование**

Цикл автор/читатель. Рецензирование одновременно несколькими читателями, поступление замечаний к автору. Анализ и обобщение критики, содержащейся в замечаниях. Поддерживает как параллельного, так и асинхронного просмотра модели. Эффективность распределения работы в коллективе. Координация работы коллектива авторов, совместного рецензирования возникающих идей.

## **Координация процесса рецензирования**

Значение своевременной обратной связи. Специальная роль наблюдателя за процессом рецензирования. Функции библиотекаря как главного координатора процесса моделирования в SADT.

## **Использование модели после одобрения**

Проверка модели на соответствие цели. Применение модели для достижения поставленной цели. Функции Комитета технического контроля. Определение текущего направления развития проекта и выработка предложений по его корректировке. Рецензирование и публикация моделей.

## **Резюме**

SADT - это методология, потому что она интегрирует процесс моделирования, управление конфигурацией проекта, использование дополнительных языковых средств и руководство проектом со своим графическим языком. Процесс

моделирования может быть разделен на несколько этапов: опрос экспертов, создание диаграмм и моделей, распространение документации, оценка адекватности моделей и принятие их для дальнейшего использования. Этот процесс хорошо отлажен, потому что при разработке проекта специалисты выполняют конкретные обязанности, а библиотекарь обеспечивает своевременный обмен информацией. Успех SADT-проектов объясняется также и тем, что Комитет технического контроля оценивает модели с точки зрения их реального использования.

## **Лекция № 5. Основы теории процессов**

### **Экономические системы**

Классификационные признаки экономических систем. Искусственность, материальный характер экономических систем. Системы открытого типа. Экономические системы как системы динамического типа. Вероятностность структуры, целей, функций, задач, ресурсов экономических систем.

Надсистема и подсистемы экономических систем, согласованность их целей. Система Экономика. Система Природа и Общество как надсистема. Основная цель Экономики, ее согласование с целями системы Природа и Общество.

Объекты системы Экономика. Совокупности отношений между объектами для образования экономических подсистем. Укрупненная модель Производственной подсистемы системы Экономика. Роль человека по отношению к системе Экономика. Понятие ресурса. Диаграмма классов, связанных с понятием Экономические ресурсы. Атрибуты и операции класса.

### **Организация как подсистема экономической системы**

Определение организации согласно международному стандарту ИСО 9000:2000. Типы организаций. Модель технических терминов понятия «организация».

Противопоставление организации окружающей среде. Понятие «границы» системы. Множество взаимных связей между средой и системой организация.

Создание и анализ системы целей организации. Источники образования целей. Планирование на основе системы целей. Отличие плана от цели. Методика постановки SMART целей: Specific, Measurable, Achievable, Relevant, Timed. Уровни иерархии целей: видение, миссия, цели, задачи, стратегии, тактики.

Методология системного анализа и системный подход при построении модели организации.

## **Структура системы Организация**

Понятие подсистемы организации. Типовые структуры, встречающиеся наиболее часто. Организационная, производственная, функциональная, финансово-экономическая, штатная, информационная, социальная, территориальная и другие структуры. Предметные области моделирования.

## **Функционально-ориентированное управление**

Иерархическая организационная структура предприятия и функциональная направленность. Управление деятельностью предприятия по организационным элементам. Вертикальная топология, построенная в соответствии выполняемыми функциями, и строгая иерархическая подчиненность «сверху-вниз». Основные характеристики функциональной структуры. Минусы и плюсы функционально-ориентированного управления.

## **Процессно-ориентированное управление**

Понятие проекта. Его свойства: временность, уникальность продукции, последовательная разработка. Отличия проекта от операционной деятельности.

Определение понятия процесс согласно Международному стандарту ИСО 9000:2000. Модель технических терминов процесса. Контекст процесса. Процедуры действия процесса. Процедуры проверки. Непрерывное улучшение процессов: цикл Деминга. Основные и вспомогательные процессы. Этапы подготовки описания процесса.

Владелец процесса. Взаимодействие с поставщиками входных потоков процесса и с потребителями его результатов. Критерии выбора владельца процесса. Входные потоки. Первичные и вторичные входы. Выходные потоки. Первичные и вторичные выходы. Ресурсы. Границы процесса. Интерфейс процесса.

Минусы и плюсы процессно-ориентированного управления. Мероприятия по внедрению процессного подхода.

## **Классификация процессов организации**

Перечень, предложенный Международной Бенчмаркинговой Палатой (International Benchmarking Clearinghouse)

- 1) Маркетинг рынка и пожеланий заказчиков.
- 2) Разработка стратегии.
- 3) Разработка продукции (услуг).
- 4) Организация продаж.
- 5) Производство и поставка продукции.
- 6) Организация сервиса (для сервисно-ориентированных организаций).

- 7) Обслуживание заказчика и выписка счета-фактуры.
- 8) Управление человеческими ресурсами.
- 9) Управление информационными ресурсами.
- 10) Управление финансовыми и физическими ресурсами.
- 11) Управление экологией.
- 12) Управление внешними связями.
- 13) Управление улучшениями и изменениями.

## **Лекция № 6. Архитектура интегрированных информационных систем**

### **Методология АРИС**

Концепций АРИС (Архитектура Интегрированных Информационных Систем) — ARIS (Architecture of Integrated Information Systems) моделирования, разработанная профессором А.В. Шеером. ARIS как архитектура, и концепция, и методология, и инструментальная среда, и нотация. Инструментальная среда ARIS Toolset, разработанная компанией IDS Scheer AG. Возможности инструментария ARIS. Платформа АРИС как основа для разработки ИС, предназначенных для представления, анализа и публикации процессов.

### **Здание АРИС**

Представление деятельности организации в виде различных моделей и возможность сведения этих различных представлений в единую систему. Здание АРИС — пять типов представлений, связанных между собой и отражающих основные аспекты деятельности организации. Представление Организация. Представление Данные. Представление Функция. Представление Продукт/Услуга. Представление Управление. Репозиторий.

Типы представления как первая компонента архитектуры. Использование результатов структурирования и выделения частей БП. Методы моделирования, включенные в методологию АРИС.

### **Этапы описания системы. Фазовая модель АРИС**

Эффективное управление БП посредством использования ИТ. Типичные фазы проектирования ИС, их построения и поддержки. Аналогичные фазы для проектирования бизнеса, реинжиниринга БП, других процессно-ориентированных методов и концепций архитектуры предприятия. Пять фаз ARIS-модели конструктивного времени.

Описание стратегических целей с ориентацией на использование ИТ. Выявление долгосрочных корпоративных целей организации, критических факторов успеха, критериев распределения ресурсов. Фаза Анализ проблем бизнеса. Качество моделей этого уровня.

Построение целевых моделей БП. Уровень определения требований. Требования к прикладной информационной системе, создаваемой для решения рассматриваемой проблемы бизнеса. Формализованные языки описания. Включение в описание общих объектов ИТ: баз данных, программного обеспечения. Уровень определения требований как репозиторий для прикладных программных систем, используемых в течение длительного времени, и как стартовая точка при описании реализации. Жизненный цикл документов, созданных на уровне определения требований. Назначение семантической модели, ее приоритет.

Разработка спецификации проекта. Адаптация к требованиям интерфейсов инструментальных средств реализации ПС (баз данных, сетевых архитектур, языков программирования и т. д.).

Создание описания реализации, где разработанные требования реализуются в виде физических структур данных, аппаратных компонентов и реальных продуктов. Трансформация спецификации проекта в конкретные аппаратные и программные компоненты. Осуществление физической связи с информационной системой. Фаза сопровождения системы.

Фазы «конструктивного времени». Фаза «реального времени».

Общий концептуальный проект БП в виде здания ARIS и четырех фаз ARIS-модели конструктивного времени.

Разбиение общего концептуального проекта БП на модели различных типов и документация их. Разложение каждого типа представления на три уровня описания: определение требований, спецификацию проекта и описание реализации.

## Модели АРИС

- Модели стратегического планирования
  - Objective diagram. Диаграмма целей
  - BSC Cause-and-effect diagram. Причинно-следственная диаграмма
  - Key performance indicator allocation diagram. Диаграмма окружения ключевых показателей результативности
- Organizational Chart. Организационная схема
- Основные модели для описания процессов
  - Value Added Chain Diagram. Диаграмма цепочек добавленного качества
  - Process selection diagram. Диаграмма выбора процесса
  - Extended event driven process chain (eEPC). Событийная цепочка процесса
  - Function allocation diagram. Диаграмма окружения функции



- Дополнительные модели для описания процессов
  - Function Tree. Дерево функций
  - Event diagram. Диаграмма событий
  - Rule diagram. Диаграмма правил
  - Process Chain Diagram. Диаграмма цепочки процесса
- Модели для описания документов и их статусов
  - Technical Terms Model. Модель технических терминов.
  - Information carrier diagram. Диаграмма носителей информации
- Модели для описания знаний и полномочий
  - Knowledge map. Карта знаний
  - Knowledge structure diagram. Диаграмма структуры знаний
  - Authorization map. Карта полномочий
  - Authorization hierarchy. Иерархия полномочий
- Модели для описания материальных и производственных ресурсов

## **Лекция № 7. Основные модели АРИС для описания процессов**

### **Диаграмма целей (Objective diagram)**

Назначение диаграммы целей. Понятие цели как системообразующее и организующее для экономической системы. Описание целей и условий их реализации.

Декомпозиция общей цели на составляющие ее подцели, вплоть до элементарных. Дерево целей - основа современных подходов и способов управления. Виды целей.

Основные объекты диаграммы целей. Цель. Критические факторы успеха. Функция. Связи между объектами диаграммы целей.

### **Организационное моделирование (Organizational Chart)**

Назначение организационной схемы. Понятие организационной структуры. Виды связей между органами управления и работниками.

Два типа управления организациями: бюрократический (иерархический) и органический (адаптивный). Линейная организационная структура управления. Функциональная организационная структура управления. Преимущества и недостатки каждого вида.

Проектные и матричные формы организации управления. Характеристика форм проектного управления. Полномочия руководителя проекта, его умения и личные качества. Гибкость проектной оргструктуры. Проблемы координации сети проектов. Матричная структура управления.

Правила построения модели организационной структуры. Иерархический тип модели. Уровни описания подразделений. Объекты и связи диаграммы организационной структуры.

### **Диаграмма цепочки добавленного качества (Value Added Chain Diagram)**

Назначение диаграммы цепочки добавленного качества. Определение понятия качество. Объекты и связи диаграммы. Иерархия подчиненности функций в модели. Классификация процессов верхнего уровня. Порядок построения диаграммы цепочки добавленного качества.

### **Событийная цепочка процесса (Extended event driven process chain, eEPC)**

Назначение модели Событийная цепочка процесса (eEPC). Четыре типа объектов модели EPC: событие, функция, правила, ресурсы. Функция как ключевой элемент модели. Правила описания функции. Выявление для каждой функции начального и конечного событий, ответственных исполнителей, материальных и документальных потоков. Инициализация функции событиями и завершение ее событиями. Правила описания событий. Присвоение имен событиям eEPC. Виды событий: начальное, внутренние, конечное. Итоговое событие одного процесса как триггер другого процесса.

Цепочка из функций и событий — событийная цепочка процесса. Правила работы с событийной цепочкой процессов. Использование событий при проверке корректности описания функции, определении момента принятия решения.

Правила, или логические операторы. Сложные потоки, параллельные пути, решения, правила переключатели-триггеры. Три основных типа правил: логическое «И», логическое «ИЛИ», логическое исключаяющее «ИЛИ». Правила при синхронном и несинхронном наступлении событий.

Моделирование решений. Принятие решений функциями. Роль имени функции. Соединение функции с правилом, которое описывает логику результатов.

Как описать возможные пути решения? На примере приведены две модели одного и того же процесса производства продукта. На первом — модель проста, но этот подход искажает модель. Если заказ действительно направляется в разные процессы, то лучше смоделировать функцию, представляющую решение о маршрутизации, даже если она выполняется автоматически. Два правила решений: Исключающее ИЛИ и ИЛИ. Объединение путей решений. Ветвление. Параллельные пути. Циклы.

Объекты Организационная единица, Прикладная система, Документ, Кластер информации. Правила расположения графических элементов на диаграмме eEPC.

**Диаграмма выбора сценария процесса (Process selection diagram)**

Назначение диаграммы выбора сценария процесса. Структура матрицы. Отслеживание вариантов исполнения процессов организации.

**Диаграмма окружения функции (Function allocation diagram, FAD)**

Назначение диаграммы окружения функции. Объекты FAD и связи между ними. Отличия EPC и FAD. Схема использования EPC и FAD.

**Дополнительные модели для описания процессов**

Дополнительные модели для описания процессов:

- 1) Function Tree. Дерево функций
- 2) Event diagram. Диаграмма событий
- 3) Rule diagram. Диаграмма правил
- 4) Process Chain Diagram. Диаграмма цепочки процесса

**Дерево функций (Function Tree)**

Назначение дерева функций. Понятия функции, базовой функции. Именованность функций. Число уровней в дереве функций. Критерии построения функционального дерева. Декомпозиция функции верхнего уровня на подфункции. Типы подчиненности функций: объектно-ориентированная, процессно-ориентированная, операционно-ориентированная.

**Диаграмма событий (Event diagram)**

Назначение диаграммы событий. Связь событий с информационными объектами. Объекты диаграммы событий и связи между ними.

**Лекция № 8. Общие принципы моделирования в ARIS**

Требования и правила, однозначные для понимания и единые для всех участников моделирования.

**Выбор типов моделей**

Избыточность полного набора моделей для проведения конкретного проекта. Рост трудозатрат на поддержание моделей в актуальном состоянии.

Определение набора используемых типов моделей для конкретного проекта. Настройки методологических фильтров ARIS и назначения этих фильтров соответствующим пользователям и их группам, участвующим в проводимых работах. Правила и настройка графических элементов модели.

Структуризация информации при моделировании. Критерии структуризации. Необходимость разработки структуры модели организации, которая позволяет

осуществлять быстрый поиск требующейся информации. Разработка иерархии папок — аналога файловой структуры. Папки в качестве места хранения объектов и моделей.

### **Определение глубины моделирования**

Общий обзор процесса моделирования. каждый этап которого затем Итеративное уточнение каждого этапа дополнительными моделями. Определение степени обобщения и детализации разрабатываемых моделей, количества уровней моделей, представляющих переход от общих моделей к детальным.

Иерархические модели: диаграмма целей, дерево функций, организационная схема и другие. Правила их детализации. Неиерархические модели: модели процессов VAD, eEPC, PCD и другие. Управление степенью загруженности модели. Формирование карты иерархии моделей.

### **Задание правил именования моделей**

Необходимость наличия правил присвоения имен объектам: обеспечение целостности модели, удобства ее чтения, а также облегчение поиска объектов в базе. Определение правил именования для каждого типа объекта.

Два аспекта, учитываемые в процессе задания правил именования объектов, — синтаксический и семантический. Синтаксический аспект правил как определение формы используемых слов и словосочетаний. Глоссарий терминов для преодоления семантической проблемы именования объектов.

Анализ и описание терминологии предметной области. Использование моделей технических терминов. Источники информации: устоявшиеся формы рабочих документов, типовые отчеты, стандартные бланки и т.п.. Занесение документов в базу данных до начала моделирования, организация возможности их совместного использования.

### **Задание правил графического оформления моделей**

Свойство наглядности моделей. Выработка правил оформления моделей: внешний вид объектов, соответствующее (взаимное) расположение объектов в модели, отображение важных характеристик объектов в поле моделирования. Определение внешнего вида объектов (цвет, форма, элементы оформления). Важность визуальной фильтрации объектов по их предназначению для быстрого чтения и анализа модели. Повышение наглядности модели посредством дополнительной информации по каждому объекту (атрибуты объекта). Задание правил взаимного расположения объектов. Использование встроенного механизма шаблонов (Templates) ARIS всеми участниками моделирования.

## Проверка моделей

Необходимость соблюдения правил двух видов — синтаксических и семантических.

Синтаксические правила создания моделей. Метамодель ARIS, дополнительные настройки, настройка методологических фильтров. Проверка следования этим правилам системой. Семантические правила для определения степени полноты проработки и согласованности моделей и объектов. Механизм семантических проверок, реализованный в модуле ARIS Semantic Check. Проведение проверки после завершения процесса моделирования. Семь основных типов правил семантических проверок. Использование набора правил, групп правил и типов правил, создание специального набора правил и их групп.

## Анализ моделей

Анализ моделей при помощи модуля ARIS Analysis. Применение стандартных или разработанных пользователем сценариев (скриптов). Стандартные скрипты для анализа моделей и объектов: организационные изменения в процессе (Organizational Change in the Process), информационные разрывы в процессе (Media Breaks in the Process), прикладные системы в процессе (Application Systems in the Process), классификация функций (Function Classification), сравнение моделей (Model Comparison), сравнение вариантов моделей (Variant Comparison). Два сценария для анализа моделей BSC системы сбалансированных показателей.

## Виды пользователей и их права

Создание элементов ARIS «пользователи» и «их группы». Назначение этих элементов: распределенное моделирование, структурированный контролируемый доступ к моделям организации, распределение полномочий и ответственности участников. Способы осуществления распределения полномочий. Классификация персонала, задействованного в работе. Распределение задач, прав, обязанностей. Настройка стандартных паролей, которые используют при работе с системой.

Таким образом, система ARIS как инструментарий для комплексного моделирования и анализа деятельности организации заставляет всерьез задуматься о подготовке процесса моделирования с точки зрения методологии, правил проверки, распределения доступа.

Игнорирование данного этапа с большой вероятностью приведет к получению непригодного для анализа материала в виде неполноценных и несогласованных моделей, которые в этом случае не будут отличаться от других неформализованных способов отражения информации.

Только грамотная подготовка и контроль позволят получить результаты моделирования, которые можно будет эффективно использовать в дальнейшем.

### **Общие принципы документирования**

Документирование моделей как вывод представленной в моделях информации в виде текстовых описаний, содержащихся в файлах заданного формата. Набор документации по проведенным работам как необходимый сопутствующий конечный продукт моделирования. Форма, удобная для ознакомления и для дальнейших работ — последующих этапов моделирования, тестирования и использования полученных решений.

Разработка структуры документов для всех групп заинтересованных лиц: руководителей организации, заказчиков проекта по моделированию бизнеса, рабочих групп моделирования по различным направлениям: информационное моделирование, описание структуры документов, описание организационной структуры, создание моделей верхнего уровня, описание процессов, сотрудников подразделений, участвующих в проекте, клиентов организации и средств массовой информации.

Возможности ARIS вывести в отчетный документ любую информацию, содержащуюся в базе данных проекта: описание моделей (включая графику), описание объектов с их характеристиками, взаимосвязи объектов не только в рамках выделенных для документирования моделей, но и по всей базе данных.

Автоматизированный процесс создания текстовых отчетов в ARIS. Использование скриптов — программ на языках Visual Basic и ARIS Basic Language Capability. Перенос информации из графических моделей в текстовые файлы в соответствии с определенными правилами. Генерация отчета под управлением скриптов — создание текстового файла и запись его на носитель информации.

## ПЛАН ЛАБОРАТОРНЫХ ЗАНЯТИЙ

### Лабораторная работа № 1. Классификация систем

**Задание:**

Приведите убедительные примеры (классификация дана согласно [10, стр. 40–84]):

- Систем и не-систем
- Абстрактных и конкретных систем
- Открытых и замкнутых систем
- Детерминированных, одушевленных, социальных и экологических систем
- Статических, динамических и гомеостатических систем
- Систем поддержания состояния, поиска цели, многоцелевых и целеустремленных
- Адаптивных систем

### Лабораторная работа № 2. Основные системные понятия

**Задание:**

Для нескольких выбранных вами систем (не менее пяти) укажите, что является их системным эффектом (эмерджентным свойством).

**Задание:**

Проведите структурный анализ выбранной вами системы по следующей схеме:

- 1) Определите цель системы
- 2) Опишите границы системы
- 3) Выделите ее подсистемы и наиболее крупные элементы
- 4) Какая из подсистем является «управляющей»?
- 5) Выделите контуры обратной связи (хотя бы по одному для балансирующей и усиливающей)
- 6) Опишите внешнюю среду системы

**Задание:**

Для нескольких систем (не менее трех) опишите их жизненный цикл.

**Задание:**

Приведите примеры контринтуитивного поведения систем. Например, для борьбы с лесными пожарами эффективно создавать искусственные возгорания навстречу движущемуся огню.

### Лабораторная работа № 3. Бизнес-игра “Beer game”

**Задание:**

Примите участие в командной бизнес-игре “Beer game”. Подробнее об игре и ее правилах вы можете прочитать в [9, стр. 51–73].

Эта игра позволяет на практике «прочувствовать» системный принцип, согласно которому поведение системы гораздо в большей степени определяется структурой системы, а не поведением отдельных ее элементов. Обсудите результаты игры и этот принцип после окончания игры. Приведите примеры проявления этого принципа в других системах.

Игровой сервер: <http://davinci.tamu.edu/beergame/v1/>

### Лабораторная работа № 4. Цели и показатели

**Задание:**

Найдите (в прессе, интернете) примеры (минимум три) плохо сформулированных целей, т. е. не по методике SMART. Поясните, какие необходимые признаки SMART-целей отсутствуют в найденных вами примерах.

**Задание:**

Для выбранной вами системы придумайте единственный(!) показатель, характеризующий текущее состояние системы. Обоснуйте ваш выбор.

**Задание:**

Для выбранной вами системы составьте сбалансированную систему показателей (Balanced Score Card). Все цели должны быть поставлены по методике SMART (или хотя бы SM).

Для каждой цели укажите один или несколько показателей. Для каждого показателя должно быть известно: кто производит его вычисление, с какой периодичностью, кто является ответственным за выполнение этого показателя.

### Лабораторная работа № 5. Бизнес-процессы и цепочка добавленной стоимости

**Задание:**

Для реально существующей системы выделите ее:

- 1) основные бизнес-процессы
- 2) обеспечивающие бизнес-процессы
- 3) бизнес-процессы управления
- 4) бизнес-процессы развития.



Какой из основных процессов, на ваш взгляд, добавляет наибольшую ценность конечному продукту (услуге) системы?

**Задание:**

В выбранной вами системе укажите создаваемый продукт (услугу) и опишите его (ее) жизненный цикл.

**Задание:**

Определите основные бизнес-процессы БГУ. Сравните полученный вами результат с официальной позицией (этот результат получен в рамках процесса внедрения системы менеджмента качества и представлен на сайте БГУ по адресу <http://www.bsu.by/main.aspx?guid=54661>).

Насколько ваш результат совпал с официальным? Какие процессы вы упустили или, наоборот, ошибочно добавили в список основных? Почему, как вам кажется, это произошло?

### **Лабораторная работа № 6. Моделирование цепочки добавленной стоимости в IDEF и ARIS**

**Задание:**

Для выбранной вами системы постройте двухуровневую контекстную модель согласно методологии IDEF0, т. е. постройте модель первого (A0) и второго (A1, A2, ...) уровня.

В качестве средства моделирования используйте на выбор AllFusion Process Modeler или Microsoft Visio.

**Задание:**

Выполните то же задание используя методологию ARIS, т. е. постройте диаграмму цепочки добавленной стоимости (VAD).

В качестве средства моделирования используйте на выбор ARIS Toolset или Microsoft Visio.

### **Лабораторная работа № 7. Моделирование бизнес-процесса в IDEF и ARIS**

**Задание:**

В выбранной вами системе выделите нетривиальный бизнес-процесс и постройте его модель используя методологию IDEF3. Особое внимание уделите нестандартным ситуациям.

В качестве средства моделирования используйте на выбор AllFusion Process Modeler или Microsoft Visio.

**Задание:**

Выполните то же задание используя методологию ARIS, т. е. постройте диаграмму событийной цепочки процесса (eEPC). Особенное внимание уделите нестандартным ситуациям. Обязательно укажите исполнителей и информационные потоки.

В качестве средства моделирования используйте на выбор ARIS Toolset или Microsoft Visio.

**Лабораторная работа № 8. Построение AS IS модели системы в ARIS****Задание:**

Постройте AS IS модель вымышленной компании используя методологию ARIS. Для этого ознакомьтесь с описанием системы (файлы вы получите у преподавателя), после чего выполните следующие шаги моделирования:

- 1) На основе интервью генерального директора опишите организационную структуру этой компании
- 2) Составьте список процессов верхнего уровня (не более 20) и постройте диаграмму VAD
- 3) На основе текстового описания бизнес-процесса постройте его подробную модель в нотации eEPC.

В качестве средства моделирования используйте на выбор ARIS Toolset или Microsoft Visio.

**Лабораторная работа № 9. Анализ и оптимизация бизнес-процессов****Задание:**

Приведите примеры каждого из четырех подходов к решению проблем (согласно Акоффу, см. [8, стр. 7–21]), а именно:

- 1) Невмешательство (Absolution)
- 2) Приемлемое решение (Resolution)
- 3) Оптимальное решение (Solution)
- 4) Растворение проблемы (Dissolution)

**Задание:**

Для выбранных вами бизнес-процессов (не менее трех) укажите их проблемные места и возможные пути разрешения проблем.

**Задание:**

Представьте, что вам дали полномочия по совершенствованию процесса. Проведите анализ процесса «как он есть сейчас» и предложите несколько способов как его улучшить. Обязательно используйте предложенные в курсе методы анализа и оптимизации.

**Лабораторная работа № 10. Имитационное моделирование бизнес-процессов в ARIS****Задание:**

Для предложенного вам процесса обработки заявки клиента постройте его диаграмму eEPC. Попробуйте сократить время обработки заявки путем определения оптимальной численности персонала и сокращения времени выполнения работ. При выполнении работы используйте модуль имитационного моделирования ARIS Simulation по схеме:

- 1) Для событий, функций, исполнителей процесса задайте данные для имитационного моделирования
- 2) Задайте частоту запуска процесса
- 3) Проведите моделирование за период 1 месяц
- 4) Проанализируйте результаты
- 5) Постройте альтернативные сценарии выполнения процесса и выполните их имитационное моделирование
- 6) Выберите оптимальный сценарий.

## СПИСОК КОНТРОЛЬНЫХ ВОПРОСОВ

1. Краткая история развития профессий ИТ-сектора, базирующихся на специальности Математика.
2. Роль системного аналитика в процессе автоматизации деятельности компании.
3. Знания, необходимые для выполнения должностных обязанностей Системного аналитика.
4. Умения и навыки, необходимые для выполнения должностных обязанностей Системного аналитика.
5. Плюсы и минусы профессии Системный аналитик, важные качества претендента на должность Системного аналитика.
6. В чем состоят отличия Системного аналитика от Бизнес-аналитика?
7. Предпосылки и история возникновения системного анализа.
8. Перечислите дисциплины, использующие системный подход к изучению объектов и явлений.
9. Раскройте несколько точек зрения на содержание понятия системный анализ.
10. Объект системного анализа в теоретическом и прикладном аспекте.
11. Области применения системного анализа, конкретно.
12. Понятие системы. Примеры.
13. Понятие Цели системы. Примеры.
14. Возможные классификации систем.
15. Понятие сложной системы. Примеры.
16. Понятие обратной связи, ее основные функции.
17. Понятие модели, ее основные свойства.
18. Классификация моделей по различным признакам.
19. Нотация модели, предъявляемые к ней основные требования. Примеры.
20. Общие принципы моделирования.
21. Понятие Структурного анализа, его основные цели.
22. Основные идеи SADT методологии.
23. Методология SADT. Синтаксис диаграмм. Примеры.
24. Методология SADT. Пять типов взаимосвязей между блоками для описания их отношений. Характеризуйте каждый тип. Примеры.
25. Синтаксис SADT-моделей. Принцип ограничения объекта, декомпозиция, коды ICOM. Примеры.
26. Характеристика экономических систем, классификационные признаки.
27. Понятие организации. Примеры. Виды организаций.
28. Связи системы-организации с внешней средой.
29. Система целей организации.

- 30.Разбиение организации на структурные подсистемы.
- 31.Функционально-ориентированный подход в управлении.
- 32.Процессно-ориентированное управление.
- 33.Сравнительная характеристика функционального и процессного управления.
- 34.Понятие «процесс», представленное с помощью модели технических терминов ARIS.
- 35.Раскройте основные понятия теории бизнес-процессов.
- 36.Классификация процессов организации.
- 37.Опишите процесс идентификации бизнес-процесса. Примеры.
- 38.Что должен выполнить СА, участвуя во внедрении процессного подхода к управлению.
- 39.Какую информацию следует выделять при проведении обследования и вербальном описании систем?
- 40.Этапы, необходимые для организации проведения обследования подсистем организации.
- 41.Какую информацию должно содержать вербальное описание системы Организация.
- 42.Напишите, как Вы понимаете фразу: «ARIS — это и архитектура, и концепция, и методология.
- 43.Здание ARIS: типы представления бизнес-процессов.
- 44.Понятие репозитория ARIS.
- 45.Этапы описания системы. Фазовая модель ARIS.
- 46.ARIS. Модели стратегического планирования. Диаграмма целей.
- 47.ARIS. Организационная схема.
- 48.ARIS. Диаграмма цепочки добавленного качества.
- 49.ARIS. Событийная цепочка процесса.
- 50.ARIS. Диаграмма окружения функции.

## СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ

### Основная:

1. Спицнадель В. Н. Основы системного анализа. СПб: Изд. дом «Бизнес-пресса», 2000.
2. Марка Д., МакГоуэн К. Методология структурного анализа и проектирования / Пер с англ. М.: МетаТехнология, 1993.
3. Шеер А.-В. Бизнес-процессы. Основные понятия. Теория. Методы / Пер с англ. М.: Весть-МетаТехнология, 1999.
4. Шеер А.-В. Моделирование бизнес-процессов / Пер с англ. М.: Весть-МетаТехнология, 2000.
5. Каменнова М., Громов А., Ферапонтов М., Шматалюк А. Моделирование бизнеса. Методология ARIS. М.: Весть-МетаТехнология, 2001.

### Дополнительная:

6. Ван Гиг Дж. Прикладная общая теория систем / Пер с англ. М.: Мир, 1981.
7. Мескон М., Альберт М., Хедоури Ф. Основы менеджмента. М.: Дело, 1997.
8. Тарасенко Ф.П. Прикладной системный анализ (учебник). Томск: Изд-во Томского университета, 2004.
9. Сенге П. Пятая дисциплина: искусство и практика самообучающейся организации. М.: ЗАО «Олимп-Бизнес», 1999. 408 с.
10. Акофф Р. Акофф о менеджменте. СПб: Питер, 2002. 448 с.
11. Репин В. В. Бизнес-процессы компании: построение, анализ, регламентация. М.: РИА «Стандарты и качество», 2007.
12. Репин В. В., Елиферов В. Г. Процессный подход к управлению. Моделирование бизнес-процессов. М.: РИА «Стандарты и качество», 2004.
13. Черемных С. В., Семенов И. О., Ручкин В. С. Моделирование и анализ систем. IDEF-технологии: практикум. М.: Финансы и статистика, 2006.
14. Войнов И. В., Пудовкина С. Г., Телегин А. И. Моделирование экономических систем и процессов. Опыт построения ARIS-моделей. Челябинск: Изд-во ЮУрГУ, 2002.

## ТИПОВАЯ УЧЕБНАЯ ПРОГРАММА

**Министерство образования Республики Беларусь  
Учебно-методическое объединение вузов Республики Беларусь  
по естественнонаучному образованию**

**УТВЕРЖДАЮ**

Первый заместитель Министра образования  
Республики Беларусь

\_\_\_\_\_ А.И. Жук

«10» февраля 2009 г.

Регистрационный № ТД-Г.175 / тип.

### **КОМПЬЮТЕРНАЯ МАТЕМАТИКА**

**Типовая учебная программа  
для высших учебных заведений по специальности  
1-31 03 01 Математика (по направлениям)**

**СОГЛАСОВАНО**

Председатель УМО вузов Республики  
Беларусь по естественнонаучному  
образованию

\_\_\_\_\_ В.В. Самохвал

«29» декабря 2008 г.

**СОГЛАСОВАНО**

Начальник Управления высшего и  
среднего специального образования

\_\_\_\_\_ Ю.И. Миксюк

«29» декабря 2008 г.

Первый проректор Государственного  
учреждения образования  
«Республиканский институт высшей  
школы»

\_\_\_\_\_ И.В Казакова

«29» декабря 2008 г.

Эксперт-нормоконтролер

\_\_\_\_\_ С.М. Артемьева

«29» декабря 2008 г.

Минск 2008

## **СОСТАВИТЕЛИ:**

**Голубева Л.Л.** — доцент кафедры дифференциальных уравнений Белорусского государственного университета, кандидат физико-математических наук,

**Липницкий В.А.** — профессор кафедры высшей математики Белорусского государственного университета информатики и радиоэлектроники доктор физико-математических наук, профессор,

**Малевич А.Э.** — доцент кафедры дифференциальных уравнений Белорусского государственного университета, кандидат физико-математических наук, доцент,

**Садовский А.П.** — профессор кафедры дифференциальных уравнений Белорусского государственного университета, доктор физико-математических наук, профессор,

**Щеглова Н.Л.** — доцент кафедры дифференциальных уравнений Белорусского государственного университета, кандидат физико-математических наук, доцент.

## **РЕЦЕНЗЕНТЫ:**

Кафедра высшей математики учреждения образования «Белорусский государственный университет информатики и радиоэлектроники».

Хвошинская Л.А., доцент кафедры высшей математики учреждения образования «Белорусский государственный аграрный технический университет», кандидат физ.-мат. наук, доцент.

## **РЕКОМЕНДОВАНА К УТВЕРЖДЕНИЮ В КАЧЕСТВЕ ТИПОВОЙ:**

Кафедрой дифференциальных уравнений механико-математического факультета Белорусского государственного университета  
(протокол № 6 от 19 февраля 2008 г.)

Научно-методическим советом Белорусского государственного университета  
(протокол № 3 от 27 марта 2008 г.)

Научно-методическим советом по математике и механике Учебно-методического объединения вузов Республики Беларусь по естественнонаучному образованию

(протокол № 3 от 10 апреля 2008 г.);

Ответственный за выпуск: зав. кафедрой дифференциальных уравнений ММФ БГУ профессор В.И.Громак



## ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

В настоящее время в обществе наблюдается рост интереса и внимания к проблемам информатизации общества. Это обусловлено рядом объективных и субъективных факторов.

Научно-технический прогресс, информатизация всех сфер общественной жизни, современные глобальные процессы и проблемы человечества предъявляют новые требования к уровню образованности личности, личностному и профессиональному развитию. В современном обществе, важнейшими характеристиками которого выступают непрерывность и динамизм развития, компьютерная грамотность приобретает особую значимость как общечеловеческая ценность, социальный механизм развития личности, общественного сознания, общества в целом.

В последние годы компьютерная математика становится эффективным значимым инструментом исследования не только в прикладных, но и в фундаментальных областях науки. Крупномасштабные исследования, связанные с расширением технических и углублением интеллектуальных возможностей, инновациями в образовательной практике позволили выделить наиболее эффективные модели, алгоритмы, целостные технологии познания, развития и реализации творческого потенциала личности.

Важным представляется освоение выпускниками высших учебных заведений современных основ информационных технологий, позволяющих сознательно и планомерно организовывать свою профессиональную деятельность, развивать собственный творческий потенциал, добиваться успехов в жизни и труде.

Типовая учебная программа «Компьютерная математика» предназначена для студентов специальности 1-31 03 01 Математика (по направлениям).

## ЦЕЛИ И ЗАДАЧИ ДИСЦИПЛИНЫ

**Целью преподавания** дисциплины «Компьютерная математика» является подготовка специалистов, обладающих знаниями и умениями эффективного использования компьютерной техники, современных информационных систем, а также имеющих навыки работы в различных математических средах.

Преподавание дисциплины **решает следующие задачи:**

- приобретение студентами знаний в области теории и истории развития компьютерной техники,
- приобретение навыков работы в средах современных общеупотребительных операционных систем,
- овладение современными методами и приёмами поиска и использования информации посредством возможностей мировой компьютерной сети,
- выработка умения организации учебной и исследовательской работы с использованием современных числовых и символьных математических пакетов,
- приобретение способностей самостоятельно расширять компьютерные математические знания с дальнейшим их использованием при анализе математических моделей широкого круга прикладных задач.

Специалист должен

**знать:**

- идеологию и основные принципы работы с пакетами компьютерной математики: Mathematica, Matlab, Mathcad, Maple;

- компьютерные пакеты подготовки электронных научных документов: LaTeX, Mathematica, MS Word;
- математические основы и алгоритмы компьютерной алгебры;
- математические основы кодирования и защиты информации;
- принципы построения математических и компьютерных моделей;
- методологии моделирования процессов, взаимосвязи данных, систем, объектов;
- основы системного анализа;

**уметь:**

- применять специализированные программные средства для построения моделей процессов, данных, объектов;
- владеть современными приёмами и методами поиска и использования научно-технической информации, редактировать, реферировать, математические и компьютерные разработки, статьи, готовить рукописи к печати;
- проводить научные исследования в области математики, математических методов системной интеграции, проектирования и создания информационных систем, моделирования и проектирования компьютерных комплексов для решения научных, народнохозяйственных и др. задач;
- анализировать, проектировать, разрабатывать и реализовывать программные компоненты этих комплексов.

Изучение дисциплины «Компьютерная математика» планируется за счет часов вузовского компонента цикла естественнонаучных или общепрофессиональных и специальных дисциплин в объеме до 534 часов, в т. ч. 272 часа аудиторных.

Рекомендуется проведение не менее двух контрольных работ в каждом семестре. В качестве итогового контроля предусматривается проведение зачетов и экзамена.

## ПРИМЕРНЫЙ ТЕМАТИЧЕСКИЙ ПЛАН

№ раз дела	№ те мы	Наименование раздела, темы	Объём в часах	
			ЛК	ЛБ/ПЗ
<b>1</b>		<b>Введение в компьютерные технологии</b>	<b>4</b>	<b>2</b>
1	1	Аппаратные и программные средства как единство двух составляющих компьютерных технологий	2	1
1	2	Основы работы в операционной системе MS Windows	2	1
<b>2</b>		<b>Основы компьютерных телекоммуникаций</b>	<b>1</b>	<b>-</b>
2	1	Работа с учебно-методической и научной литературой.	1	-
<b>3</b>		<b>Офисные технологии</b>	<b>5</b>	<b>4</b>
3	1	Обработка текстовой информации	2	1
3	2	Представление информации в виде электронных таблиц	2	1
3	3	Подготовка и демонстрация презентаций	1	2
<b>4</b>		<b>Числовые и символьные математические пакеты</b>	<b>6</b>	<b>12</b>
4	1	Числовой математический пакет Mathcad	3	6
4	2	Символьный математический пакет Maple	3	6
<b>5</b>		<b>Символьный математический пакет Mathematica</b>	<b>16</b>	<b>18</b>
5	1	Структура пакета и сценарий работы с ним	2	1
5	2	Выражение как основная структура данных в Mathematica	2	2
5	3	Знакомство с графикой	2	2
5	4	Образцы, их классификация и использование	1	2
5	5	Символ — основной объект вычислений	1	2
5	6	Функциональный стиль программирования	2	2
5	7	Программирование, основанное на правилах преобразований	3	2
5	8	Вычисление выражений. Конструкции, управляющие ходом вычисления	2	3
5	9	Принципы локализации переменных. Контексты и пакеты	1	2
<b>6</b>		<b>Числовой математический пакет MATLAB</b>	<b>16</b>	<b>18</b>
6	1	Назначение, структура, интерфейс системы MATLAB	2	2
6	2	Данные и вычисления	1	3
6	3	Элементы программирования	1	1
6	4	М-сценарии и М-функции	2	2
6	5	Объектно-ориентированное программирование	2	2
6	6	Высокоуровневая графика	2	2
6	7	Дескрипторная графика	2	2
6	8	Импорт и экспорт данных	2	2

6	9	Приложения с графическим интерфейсом пользователя	2	2
<b>7</b>		<b>Основы компьютерного моделирования</b>	<b>16</b>	<b>18</b>
7	1	Компьютерное моделирование и Simulink	2	2
7	2	Решатели (Solver) и интеграторы (Integrator)	2	3
7	3	Взаимодействие и обмен данными между MATLAB и Simulink	2	2
7	4	S-функции	2	2
7	5	Управляемые подсистемы	2	2
7	6	Системы, управляемые событиями	2	2
7	7	Диаграммы состояний	2	2
7	8	Моделирование виртуальной реальности (VRML) и Simulink	2	3
<b>8</b>		<b>Математические основы компьютерной алгебры</b>	<b>6</b>	<b>28</b>
8	1	Аффинные многообразия и идеалы. Алгоритмы деления	2	6
8	2	Базисы Гребнера	2	12
8	3	Теория исключения	2	10
<b>9</b>		<b>Математические основы защиты информации</b>	<b>12</b>	<b>56</b>
9	1	Основы прикладной алгебры	4	20
9	2	Поля Галуа и их свойства	2	8
9	3	Основы помехоустойчивого кодирования	2	6
9	4	Свойства кодов Хемминга и БЧХ-кодов	-	4
9	5	Проблема декодирования кодов и алгебраические уравнения	-	6
9	6	Введение в криптографию	2	2
9	7	Китайская теорема об остатках	-	4
9	8	Проблема определения простоты числа. Алгоритм Диемитко	2	6
<b>10</b>		<b>Методы моделирования сложных систем и процессов</b>	<b>6</b>	<b>28</b>
10	1	Системный анализ и его место в современном мире	1	-
10	2	Базовые понятия системного анализа	2	4
10	3	Экономические системы: структурное строение и особенности	-	6
10	4	Основы теории процессов	1	4
10	5	Архитектура интегрированных информационных систем	-	8
10	6	Управление проектами	2	6
		Всего аудиторных часов	<b>88</b>	<b>184</b>
		<b>ИТОГО</b>	<b>272</b>	

## СОДЕРЖАНИЕ УЧЕБНОГО МАТЕРИАЛА ДИСЦИПЛИНЫ «КОМПЬЮТЕРНАЯ МАТЕМАТИКА»

### Раздел 1. Введение в компьютерные технологии

#### **Тема 1.1. Аппаратные и программные средства как единство двух составляющих компьютерных технологий.**

Предмет компьютерных технологий. Функции аппаратных и программных средств. Основная конфигурация ПК. Понятие операционной системы. Системные Приложения. Размещение информации на носителях. Архивирование информации. Защита информации от компьютерных вирусов.

#### **Тема 1.2. Основы работы в операционной системе MS Windows**

Операционная система MS Windows. Объектный подход: свойства, реакция на события. Графический интерфейс пользователя. Основные технологические механизмы Windows. Типовой порядок работы с Документом. Совместная работа Приложений. Обмен данными. Связь и внедрение объектов. Методика ознакомления с новыми Приложениями.

### Раздел 2. Основы компьютерных телекоммуникаций

#### **Тема 2.1. Работа с учебно-методической и научной литературой**

Понятие о компьютерных телекоммуникациях. Что такое Internet и как он работает. Internet Explorer: адресация ресурсов, навигация, работа с файлами. Поиск информации в Internet. Поиск и отбор учебно-методической и научной литературы посредством компьютерных технологий. Электронная почта. Основные принципы работы в Центре информационных технологий конкретного вуза.

### Раздел 3. Офисные технологии

#### **Тема 3.1. Обработка текстовой информации**

Обработка текстовой информации. Текстовые файлы и текстовые документы. Текстовый процессор MS Word. Иерархия объектов MS Word. Редактирование объектов - символа, строки, абзаца. Форматирование объектов. Функциональные возможности MS Word при работе с текстом. Редактор формул. Работа с таблицами. Создание указателей и оглавления Документа. Проверка орфографии. Особенности печати.

#### **Тема 3.2. Представление информации в виде электронных таблиц**

Электронные таблицы. Система объектов MS Excel. Адресация объектов. Особенности адресации ячеек. Типы данных. Создание и использование формул. Редактирование и форматирование объектов MS Excel. Основные приемы оптимизации работы с таблицей. Создание диаграмм. Особенности печати.

#### **Тема 3.3. Подготовка и демонстрация презентаций**

Способы представления графики: растровый и векторный, особенности каждого. Графические форматы, используемые в Windows. Общие принципы создания презентаций. Возможности создания презентаций посредством MS Power Point.

## **Раздел 4. Числовые и символьные математические пакеты**

### **Тема 4.1. Числовой математический пакет Mathcad**

Числовой математический пакет Mathcad. Особенности интерфейса. Структура документа. Ввод, редактирование и форматирование выражений. Способы определения и вычисления переменных и функций. Работа с массивами, векторами, матрицами. Графика и анимация. Элементы программирования. Решение уравнений и систем. Символьные вычисления.

### **Тема 4.2. Символьный математический пакет Maple**

Символьный пакет Maple. Выражение как базовый объект символьных пакетов. Атомарные объекты. Анализ структуры выражения. Изменение типа выражения. Наборы и списки как наиболее активно используемые выражения. Таблицы. Массивы. Индексированные переменные.

Преобразование выражений. Работа с частями выражения. Функции подстановки. Преобразование многочленов, рациональных дробей, сложных радикалов, тригонометрических выражений.

Средства программирования. Создание функций пользователя. Простейшие процедуры. Базисные конструкции языка, управляющие вычислением. Методы отладки программ.

Специализированные пакеты. Графика и анимация.

## **Раздел 5. Символьный математический пакет Mathematica**

### **Тема 5.1. Структура пакета и сценарий работы с ним**

Назначение пакета Mathematica. Сценарий работы. Структура пакета. Сессия (Session) как объект работы Ядра (Kernel). Накопление знаний во время сессии и хранение знаний между сессиями и сеансами работы с пакетом. Метафора записной книжки (Notebook). Ячейки (Cells) как основные обобщенные объекты Блокнотов. Основной цикл (In[xx]->Out[xx]), протокол MathLink работы пакета.

### **Тема 5.2. Выражение как основная структура данных в Mathematica.**

Выражение как основная структура данных. Анализ структуры выражения. Типы выражений. Атомарные объекты. Функции-конверторы типов выражений. Списки как наиболее используемое выражение. Встроенные функции для работы со списками.

### **Тема 5.3. Знакомство с графикой**

Базовые функции двумерной и трехмерной графики. Режимы вычисления графических функций. Графические примитивы. Специализированные графические пакеты.

### **Тема 5.4. Образцы, их классификация и использование.**

Образцы (Patterns). Понятие образца. Основные правила записи. Возможности описания различных множеств выражений посредством использования образцов. Образцы, соответствующие условию. Образцы, содержащие альтернативу. Установка значений образцов по умолчанию. Образцы, допускающие повторение выражений. Построение функции пользователя.

### **Тема 5.5. Символ — основной объект вычислений**

Символ как основное средство вычислений. Предопределенные константы. Атрибуты, или свойства символа. Опции, или режимы выполнения функций.

### **Тема 5.6. Функциональный стиль программирования**

Чистые и анонимные функции. Оператор Apply и семейство операторов Map. Возможности последовательного применения функций (Nest, Fold, FixedPoint). Рекурсивное задание функций.

### **Тема 5.7. Программирование, основанное на правилах преобразований**

Программирование, основанное на правилах глобальных преобразований. Функции семейства Set. Механизм верхних и нижних значений символа. Программирование, основанное на правилах локальных преобразований. Функции семейства Replace. Использование образцов в правилах преобразований. Подстановки.

### **Тема 5.8. Вычисление выражений. Конструкции, управляющие ходом вычисления**

Порядок вычисления выражений. Выполнение стандартного процесса вычисления выражения. Выражения, вычисляемые нестандартно. Возможности изменения стандартного порядка вычисления выражения. Конструкции, управляющие ходом вычисления, операторы ветвления и операторы повторения (If, Which, Switch, Cases, While, For, Break, Continue, Return, Do, Table, Sum). Управляемый выход (Catch-Throw).

### **Тема 5.9. Принципы локализации переменных. Контексты и пакеты**

Принципы локализации переменных: Module, With, Block. Контексты (Contexts). Создание пользовательских пакетов.

## **Раздел 6. Числовой математический пакет MATLAB**

### **Тема 6.1. Назначение, структура, интерфейс системы MATLAB**

Назначение и области применения системы MATLAB как высокоэффективной системы инженерных и научных расчетов. Операционная среда и язык программирования системы MATLAB. Концепция Пакетов Прикладных Программ. Сравнительная характеристика пакета MATLAB и других числовых и символьных математических пакетов. Структура пакета. Интерфейс системы.

### **Тема 6.2. Данные и вычисления**

Типы данных в MATLAB. Вектор, матрица, тензор. Операции с массивами. Операции конкатенации, индексации и формирования диапазона значений. Доступ к элементам и группе элементов массивов. Работа со строками и столбцами. Операции отношения и логические операции. Символьные массивы. Структуры и массивы структур. Массивы ячеек.

### **Тема 6.3. Элементы программирования**

Управление вычислением. Операторы управления. Операторы ветвления. Операторы цикла. Обработка исключительных ситуаций.

#### **Тема 6.4. М-сценарии и М-функции**

Понятие М-сценария и М-функции. Особенности сценария и функции. Синтаксис определения и вызова М-функций. Проверка входных параметров и выходных значений М-функций. Видимость имен переменных и имен функций. Локальные и глобальные переменные. Разработка и отладка М-функций.

#### **Тема 6.5. Объектно-ориентированное программирование**

Принципы объектно-ориентированного программирования (ООП), реализуемые в системе MATLAB: инкапсуляция, одиночное и множественное наследование, агрегирование. Понятие класса и объекта. Основные правила определения класса и создания объекта.

#### **Тема 6.6. Высокоуровневая графика**

Двумерная графика. Функция plot. Оформление графиков функций. Изменение свойств линий. Надписи на графиках. Оформление осей системы координат. Трехмерная графика. Положение камеры и графического объекта. Сохранение графических изображений.

#### **Тема 6.7. Дескрипторная графика**

Уровни работы с графическими объектами. Дескрипторная графика Handle Graphics в MATLAB. Иерархия графических объектов. Функции создания и управления графическими объектами.

#### **Тема 6.8. Импорт и экспорт данных**

Сохранение данных рабочего пространства. Файловый ввод-вывод. Обмен данными между MATLAB и другими файлами (верхний уровень). Низкоуровневое управление файлами (нижний уровень). Работа с текстовыми файлами. Работа с графическими файлами (мультимедиа).

#### **Тема 6.9. Приложения с графическим интерфейсом пользователя**

Понятие графического интерфейса пользователя. Графические элементы управления системы MATLAB. Специальные функции MATLAB для работы с графическими объектами. Интерактивное управление графическими объектами.

### **Раздел 7. Основы компьютерного моделирования**

#### **Тема 7.1. Компьютерное моделирование и Simulink**

Компьютерное моделирование и компьютерная модель. Математическое моделирование, математическая модель. Имитационное моделирование. Simulink как программа имитационного моделирования динамических и событийно-управляемых систем. Понятие динамической системы. Основные составляющие модели Simulink.

#### **Тема 7.2. Решатели (Solver) и интеграторы (Integrator)**

Аналитическое и численное моделирование. Параметры конфигурации Simulink-модели Configuration Parameters. Дискретные и непрерывные системы. Численные методы решения дифференциальных уравнений. Блок Integrator.



### **Тема 7.3. Взаимодействие и обмен данными между MATLAB и Simulink**

Уровни обмена данными между MATLAB и Simulink. Использование переменных Workspace для задания параметров блоков. Установка параметров обмена с рабочей областью Workspace. Блоки передачи информации в модель Simulink и блоки для передачи результатов из модели Simulink. Ручной запуск m-файлов из Simulink-модели. Автоматическое выполнение команд m-файлов до и после выполнения модели. Запуск Simulink-модели из m-файла.

### **Тема 7.4. S-функции**

Назначение S-функций, цель их использования. Определение S-функций. Языки написания S-функций. Блок S-Function, его параметры. Математическое описание S-функции. Этапы Simulink-моделирования. Создание S-функций на языке MATLAB. Определение блока характеристик S-функции. Callback-методы S-функции. Применение S-функции для создания анимации.

### **Тема 7.5. Управляемые подсистемы**

Виртуальные и монолитные подсистемы. Управляемые и неуправляемые подсистемы. Типы управляемых подсистем. Связь подсистемы с моделью. Способы создания подсистем.

### **Тема 7.6. Системы, управляемые событиями**

Сложные системы. Гибридные системы. Трансформационные системы. Реактивные системы. Основные компоненты диаграммы состояний. Определение конечного автомата.

Графический редактор Stateflow. Связь Stateflow с Simulink и Matlab. Виды объектов Stateflow. Графические объекты. Неграфические объекты. События, состояния, переходы.

### **Тема 7.7. Диаграммы состояний**

Язык описания систем управляемых событиями, принятый в Stateflow. Отладка модели в Stateflow. Состояния, их характеристики. Типы действий, связанные с состояниями. Нотации меток состояний. Переходы. Действия, связанные с переходами. Условия, связанные с переходами. Нотации меток переходов. Безусловный переход. Соединительные переходы. Хронологические соединения.

### **Тема 7.8. Моделирование виртуальной реальности (VRML) и Simulink**

Виртуальная реальность. Язык описания виртуальной реальности VRML. Описание, проектирование и создание трехмерных сцен в MATLAB. Основные узлы в VRML. Использование VRML в Simulink. Пакет Virtual Reality Toolbox.

## **Раздел 8. Математические основы компьютерной алгебры**

### **Тема 8.1. Аффинные многообразия и идеалы. Алгоритмы деления.**

Определение аффинного многообразия и полиномиального идеала. Базис идеала. Параметризация аффинных многообразий. Задача описания идеала. Задача о принадлежности к идеалу. Задача решения полиномиальных уравнений. Задача неявного представления. Типы упорядочений мономов в  $k[x_1, x_2, \dots, x_n]$ . Алгоритм деления в  $k[x_1, x_2, \dots, x_n]$ .

### **Тема 8.2. Базисы Гребнера**

Мономиальные идеалы и лемма Диксона. Теорема Гильберта о базисе. Базисы Гребнера, свойства базисов Гребнера. Алгоритм Бухбергера. Минимальный базис Гребнера, редуцированный базис Гребнера. Усовершенствования алгоритма Бухбергера.

### **Тема 8.3. Теория исключения**

Теоремы об исключении и продолжении,  $k$ -й исключающий идеал. Единственность разложения на множители и результаты. Результаты и теорема о продолжении. Обобщенные результаты.

## **Раздел 9. Математические основы защиты информации**

### **Тема 9.1. Основы прикладной алгебры**

Делимость целых чисел. Соотношение Безу. Простые числа и их свойства. Основная теорема арифметики. Сравнения. Классы вычетов и их свойства. Малая теорема Ферма. Функция Эйлера и её свойства. Теорема Эйлера. Обращение теоремы Эйлера и числа Кармайкла. Сложность проверки на простоту и разложения на простые множители. Алгебраические системы. Группы и подгруппы. Смежные классы по подгруппе и их свойства. Теорема Лагранжа о подгруппах конечных групп. Действие группы на множестве. Циклическая классификация двоичных векторов. Кольца и их типы. Мультипликативная группа кольца. Делители нуля. Подкольца и идеалы. Максимальные идеалы. Свойства кольца целых чисел. Кольцо полиномов и его свойства. Вычисления в кольцах. Фактор-кольца. Гомоморфизмы колец и их свойства.

### **Тема 9.2. Поля Галуа и их свойства**

Характеристика поля. Минимальные подполя и их классификация. Расширение полей. Трансцендентные, алгебраические и конечные расширения. Теорема о башне расширений полей. Конечные поля и их свойства: количество элементов, порождающий полином, цикличность мультипликативной группы, примитивные элементы, методы формирования. Группа Галуа. Норма и след. Следовый критерий разрешимости квадратных уравнений в полях Галуа.

### **Тема 9.3. Основы помехоустойчивого кодирования.**

Теорема Шеннона. Код Хемминга и ASCII-формат. Линейные коды. Первичное кодирование и декодирование. Проверочная и порождающая матрицы кода. Метрика Хемминга. Минимальное расстояние кода и его декодирующие возможности. Декодирование по методу максимального правдоподобия. Декодирование по смежным классам. Синдромы ошибок и их свойства. Идея синдромного декодирования.

### **Тема 9.4. Свойства кодов Хемминга и БЧХ-кодов.**

Минимальное расстояние кода Хемминга, синдромный метод и его декодирования. Коды Боуза-Хоквингема-Чоудхури (БЧХ-коды), исправляющие двойные и тройные ошибки, их матрицы и основные параметры. Общее определение БЧХ-кода. Реверсивные коды. Свойства синдромов ошибок в БЧХ-кодах.

### **Тема 9.5. Проблема декодирования кодов и алгебраические уравнения.**

Декодирование двойных ошибок в реверсивных и БЧХ-кодах сведение к решению квадратного уравнения в поле Галуа характеристики 2. проблема решения уравнения в полях Галуа. Декодирование многократных ошибок в БЧХ-кодах сведением к решению алгебраических уравнения в полях Галуа.

### **Тема 9.6. Введение в криптографию**

Краткая история криптографии. Шифры Цезаря и Вижинера. Криптосистема RSA. Криптосистемы Рабина, Эль Гамала, DES. Современный международный криптографический стандарт шифрования AES.

### **Тема 9.7. Китайская теорема об остатках.**

Китайская теорема об остатках. Формула Гарнера. Работа с большими числами. Применение китайской теоремы об остатках в числовых криптографических системах.

### **Тема 9.8. Проблема определения простоты числа. Алгоритм Диемитко.**

Вероятностные и детерминированные алгоритмы проверки числа на простоту. Числа Ферма и Марсена. Алгоритм Диемитко и его применение в белорусских и российских стандартах шифрования. Перспективы развития современной криптологии и криптографии. Квантовая криптография.

## **Раздел 10. Методы моделирования сложных систем и процессов**

### **Тема 10.1. Системный анализ и его место в современном мире**

Системный анализ и его место в современном мире. Предпосылки возникновения системного анализа. Системный анализ и системный подход в управлении. Системный анализ и другие системные дисциплины. Области применения системного анализа.

### **Тема 10.2. Базовые понятия системного анализа**

Системы и их классификация. Типы и виды систем. Жизненный цикл системы. Взаимоотношения систем. Понятие модели. Общепринятые принципы моделирования. Уровни моделирования. Степени структурирования и детализации. Варианты моделей.

### **Тема 10.3. Экономические системы: структурное строение и особенности**

Структурный анализ как разновидность системного анализа. Характеристики экономических систем. Подсистема организация, ее структура. Функционально — ориентированное и процессно-ориентированное управление.

### **Тема 10.4. Основы теории процессов**

Процессный подход при построении систем управления технологиями предприятия. Исходная, объединенная и обобщенная модели бизнес-процесса. Инжиниринг и оптимизация бизнес-процессов. Основные направления реинжиниринга.

### **Тема 10.5. Архитектура интегрированных информационных систем.**

Архитектура интегрированных информационных систем (АРИС) как методология системного анализа. Виды и типы моделей. Уровни моделирования. Фазовая модель. Предварительная

информационная модель. Предварительная процедурная модель. Инструментальные программные средства ARIS. Примеры моделирования систем и процессов.

**Тема 10.6. Управление проектами.**

Понятие проекта и управления им. Жизненный цикл проекта и организация. Группы процессов управления проектом. Взаимодействия процессов. Диаграммы зависимостей процессов управления проектами. Основные документы проекта. Области знаний по управлению проектами: управление интеграцией, содержанием, сроками, стоимостью, качеством, человеческими ресурсами, коммуникациями, рисками, поставками проекта.

## ПЕРЕЧЕНЬ РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ

№№ п/п	Основная литература
1.	Ю.Шафрин Информационные технологии. В 2 частях. Часть 1. Основы информатики и информационных технологий. Бином. Лаборатория знаний, 2004.
2.	В.Дьяконов, Ю.Новичков, В. Рычков. Компьютер для студента. СПб: Питер, 2001.
3.	Д.Кириянов. Самоучитель Mathcad 13. СПб: БХВ-Петербург, 2006.
4.	Heck A. Introduction to Maple. Third Edition. Springer, 2003.
5.	Stephen Wolfram. The Mathematica Book. Fourth Edition. Cambridge, University Press, 1999.
6.	Л.Л.Голубева, А.Э.Малевич, Н.Л.Щеглова. Компьютерная математика. Символьный пакет Mathematica. Курс лекций. Минск, БГУ, 2005
7.	Л.Л.Голубева, А.Э.Малевич, Н.Л.Щеглова. Компьютерная математика. Числовой пакет MATLAB. Курс лекций. Минск, БГУ, 2007
8.	Л.Л.Голубева, А.Э.Малевич, Н.Л.Щеглова. Компьютерная математика. Числовой пакет MATLAB. Лабораторный практикум. Минск, БГУ, 2009
9.	Потемкин В.Г. Система инженерных и научных расчетов MATLAB 5.x. В 2-х т. Том 1, Том 2. М.: ДИАЛОГ — МИФИ, 1999.
10.	Мартынов Н.Н. Введение в MATLAB 6. КУДИЦ-ОБРАЗ, 2002.
11.	Кокс Д., Литтл Дж., О'Ши Д. Идеалы, многообразия и алгоритмы. Введение в вычислительные аспекты алгебраической геометрии и коммутативной алгебры. — М.: Мир, 2000.
12.	Adams W., Loustaunau P. An Introduction to Gröbner Bases. Graduate Studies in Mathematics. Amer. Math. Soc. — Providence, 1994.
13.	Becker T., Weispfenning V. Gröbner Bases: A Computational Approach to Commutative Algebra. Springer Verlag, Berlin and New York, 1993.
14.	Спицнадель В.Н. Основы системного анализа. — СПб.: Изд. дом «Бизнес-пресса», 2000.
15.	Шеер А.-В. Бизнес-процессы. Основные понятия. Теория. Методы. — М.: Весть-МетаТехнология, 1999.
16.	Шеер А.-В. Моделирование бизнес-процессов. — М.: Весть-МетаТехнология, 2000.
17.	Липницкий В.А. Современная прикладная алгебра. Математические основы защиты информации от помех и несанкционированного доступа. Минск, БГУИР, 2006.
18.	Липницкий В.А., Конопелько В.К. Норменное декодирование помехоустойчивых кодов и алгебраических уравнений. Минск, Издательский центр БГУ, 2007.
19.	Черемушкин А.В. Лекции по арифметическим алгоритмам в криптографии. М.: МЦНМО, 2002.
20.	Мещеряков В.В. Задачи по математике с MATLAB & Simulink. Диалог-МИФИ, 2007.
21.	Дьяконов В. П. MATLAB 6.5 SP1/7.0 + Simulink 5/6 в математике и моделировании.

	Солон-пресс, 2005.
22.	И. В. Черных. Simulink. Среда создания инженерных приложений. Диалог-МИФИ, 2004.
23.	Джеймс Дэбни, Томас Харман. Simulink 4. Секреты мастерства. Бином. Лаборатория знаний, 2003.

№№ п/п	Дополнительная литература
1.	Т.С. Петрушина, Т.И. Рабцевич. Основы операционной системы Windows. Текстовый редактор Word. Практикум. Мн.: БГУ, 2002.
2.	Г.А.Рассолько, Е.В.Кремень, Ю.А.Кремень. Методы программирования. Использование Excel для выполнения математических расчетов. Учеб.-метод. пособие. Мн.: БГУ, 2002.
3.	Г.А.Рассолько, Е.В.Кремень. Система тестов по математике и информатике на базе пакета Mathcad 2000. В 3 частях. Ч.1 , Ч. 2, Ч. 3. Учеб.-метод. пособие. Мн.: БГУ, 2002.
4.	Вирт Н. Алгоритмы и структуры данных. М.: Мир, 1989.
5.	Кушниренко, Лебедев. Программирование для математиков. М.: Наука, 1988.
6.	Мартынов Н.Н., Иванов А.П. MATLAB 5х. Вычисления, визуализация, программирование. КУДИЦ-ОБРАЗ, 2000.
7.	Лазарев Ю.Ф. MATLAB 5.x Для студентов. BHV — Киев, 2000.
8.	Кондрашов В.Е., Королев С.Б. MATLAB как система программирования научно-технических расчетов. М: Мир, 2002.
9.	Глушаков С.В., Жакин И.А., Хачиров Т.С. Математическое моделирование. Mathcad 2000. Matlab 5.3. Учебный курс. Издательства: Фолио, АСТ; Серия: «Домашняя библиотека», 2001.
10.	Виноградов И.М. Основы теории чисел. М.: Наука. 2006.
11.	Ноден П., Китте К. Алгебраическая алгоритмика Москва, Мир. 1999
12.	В.Дьяконов. Simulink 4. Специальный справочник. Питер. 2001.
13.	В.Дьяконов, В.Круглов. MATLAB. Анализ, идентификация и моделирование систем. Специальный справочник. Питер. 2001.
14.	А.Гультяев. Визуальное моделирование в среде Matlab: Учебный курс. Питер. 2000.
15.	А.Гультяев. MATLAB 5.2. Имитационное моделирование в среде Windows. Корона принт.1999.
16.	В.Дьяконов, В.Круглов. Математические пакеты расширения MATLAB. Специальный справочник. Питер. 2001.
17.	В.Дьяконов, И.Абраменкова, В.Круглов. MATLAB с пакетами расширений. Нолидж. 2001.
18.	М. С. Каменнова, А. И. Громов, М. М. Ферапонтов, А. Е. Шматалюк.

	Моделирование бизнеса. Методология ARIS. — М.: МетаТехнология, 2001.
19	Дьяконов В. П. Matlab 6.5 SP1/7.0 + Simulink 5/6. Основы применения. Солон-пресс · 2005.
20	Уикхэм Ф. Консалтинг в управлении проектами. — Санкт-Петербург, 2005.

### РЕКОМЕНДУЕМОЕ УЧЕБНО-ЛАБОРАТОРНОЕ ОБОРУДОВАНИЕ

Для проведения занятий требуются следующее программное обеспечение: операционная система MS Windows, MS Office, Mathcad, Maple, Mathematica, MATLAB, ARIS, LaTeX.

## **ВОСПИТАТЕЛЬНО-ИДЕОЛОГИЧЕСКАЯ ФУНКЦИЯ**

Учебно-методический комплекс по дисциплине «Компьютерная математика» составлен в соответствии с основными направлениями государственной молодежной политики, отраженными в Концепции непрерывного воспитания детей и учащейся молодежи в Республике Беларусь, в Плате идеологической и воспитательной работы БГУ на 2012-2013 годы и других государственных программах, нормативно-правовых и инструктивно-методических документах, определяющих приоритетные направления идеологии белорусского государства.

Учебно-методический комплекс по дисциплине «Компьютерная математика» способствует созданию условий для формирования нравственно зрелой, интеллектуально развитой личности обучающегося, которой присущи социальная активность, гражданская ответственность и патриотизм, приверженность к университетским ценностям и традициям, стремление к профессиональному самосовершенствованию, активному участию в экономической и социально-культурной жизни страны.

Основными задачами идеологической и воспитательной составляющей Учебно-методического комплекса по дисциплине «Компьютерная математика» являются:

- 1) Максимальное использование потенциальных возможностей кафедры по формированию гражданско-правовой устойчивости профессорско-преподавательского состава и студентов;
- 2) Содействие становлению личности, духовно-нравственное и интеллектуальное развитие студентов;
- 3) Совершенствование информационного сопровождения организации жизнедеятельности студентов, содействие социальной адаптации, оказание им помощи в усвоении и выполнении учебного материала, установленных норм и правил внутреннего распорядка, прав и обязанностей.