

МЕЖГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Кафедра «Программное обеспечение информационных технологий»

ОСНОВЫ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

*Методические рекомендации к лабораторным работам
для магистрантов всех специальностей
дневной и заочной форм обучения*



Могилев 2024

УДК 004
ББК 32.97
О75

Рекомендовано к изданию
учебно-методическим отделом
Белорусско-Российского университета

Одобрено кафедрой «Программное обеспечение информационных технологий» «30» августа 2024 г., протокол № 1

Составитель канд. техн. наук, доц. Э. И. Ясюкович

Рецензент канд. техн. наук, доц. В. М. Ковальчук

Методические рекомендации содержат базовые сведения об основах информационных технологий, задания для выполнения лабораторных работ и примеры решения задач.

Учебное издание

ОСНОВЫ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

Ответственный за выпуск	В. В. Кутузов
Корректор	А. А. Подошевка
Компьютерная верстка	Н. П. Полевничая

Подписано в печать . Формат 60×84/16. Бумага офсетная. Гарнитура Таймс.
Печать трафаретная. Усл. печ. л. . Уч. изд. л. . Тираж 16 экз. Заказ №

Издатель и полиграфическое исполнение:
Межгосударственное образовательное учреждение высшего образования
«Белорусско-Российский университет».
Свидетельство о государственной регистрации издателя,
изготовителя, распространителя печатных изданий
№ 1/156 от 07.03.2019.
Пр-т Мира, 43, 212022, г. Могилев.

© Белорусско-Российский
университет, 2024

Содержание

Введение.....	4
1 Лабораторная работа № 1. Подготовка и редактирование текстовых документов в среде редактора MS Word.....	5
2 Лабораторная работа № 2. Решение задач предметной области с использованием надстройки Поиск решения в среде MS Excel.....	7
3 Лабораторная работа № 3. Решение задач предметной области на языке <i>VBA</i> в среде программы MS Excel	13
4 Лабораторная работа № 4. Построение и редактирование графических элементов в среде графических редакторов	20
5 Лабораторная работа № 5. Проектирование Web-документа на языке HTML	23
6 Лабораторная работа № 6. Реализация имитационной модели технической системы в среде программы Simulink пакета MatLab.....	24
7 Лабораторная работа № 7. Разработка презентации в среде программы PowerPoint.....	29
8 Лабораторная работа № 8. Изучение численных методов решения нелинейных уравнений	31
9 Лабораторная работа № 9. Изучение численных методов решения задач, описываемых системой ОДУ	34
10 Лабораторная работа № 10. Решение задач предметной области в среде программы MathCad.....	37
11 Лабораторная работа № 11. Решение задачи минимизации функции многих переменных.....	40
12 Лабораторная работа № 12. Проектирование базы данных на основе реляционной модели в среде СУБД Access	41
Список литературы	44

Введение

Цель методических рекомендаций к лабораторным работам по дисциплине «Основы информационных технологий» заключается в расширении магистрантами практических навыков работы в среде приложений *MS Office*, *MathCad*, *MatLab*, программирования на языках *VBA*, *C#*, *HTML*, а также решения задач с использованием численных методов, методов минимизации и оптимизации.

Информационные технологии характеризуют уровень развития общества, возможность его интеграции в мировую цивилизацию. Именно этим определяются актуальность и необходимость изучения и овладения, по крайней мере, основами информационных технологий. Данное требование в настоящее время является важным элементом подготовки специалистов высшей квалификации.

Дисциплина «Основы информационных технологий» занимает значительное место в подготовке магистров технического и экономического профилей.

Цель изучения дисциплины – подготовка магистрантов к использованию современных информационных технологий как инструмента для решения на высоком уровне научных и практических задач в своей предметной области.

Предметом изучения дисциплины являются: современные информационные технологии; технологии программирования (объектно ориентированного, процедурного, логического), программные средства информационных технологий (офисные, системы математических вычислений *MathCad*, *MatLab*; программные средства для Web-программирования; языки разметки гипертекста и скриптовые языки); сетевые технологии и *Internet*, инструментальные средства создания Web-серверов и сайтов; системы управления базами данных; методы и средства защиты информации; математические модели и численные методы решения задач в различных предметных областях; модели, описываемые обыкновенными дифференциальными уравнениями; методы оптимизации и системы поддержки принятия решений; методы минимизации функций многих переменных; методы решения вариационных задач.

В методических рекомендациях приводится 12 лабораторных работ по основным темам дисциплины.

Выполнение каждой работы рассчитано на два часа и предусматривает подготовку отчета, который содержит: название и цель работы; постановку задачи и исходные данные; краткое описание технологии выполнения работы; полученные результаты и их анализ; выводы.

1 Лабораторная работа № 1. Подготовка и редактирование текстовых документов в среде редактора MS Word

Цель работы: изучение технологии работы с текстом в среде редактора *MS Word*.

1.1 Некоторые приемы эффективной работы в среде редактора MS Word

Для более эффективной работы с текстом в редакторе *MS Word* можно использовать специальные приемы.

Представить прописью число: поставить курсор в место, где необходимо вывести числовое значение прописью, нажать **Ctrl + F9**, в появившемся затененном поле с фигурными скобками {} вставить конструкцию {=число}*CardText}, нажать клавишу **F9**. **Shift + F9** – возвращает исходное поле, например {=2345}*CardText}.

Изменить регистр выделенного фрагмента текста: **Shift + F3**.

Выделить вертикальный блок текста: удерживая **Alt**, выделить мышью необходимый блок, или **Ctrl + Shift + F8**, и далее выделить блок текста клавишами перемещения курсора.

Убрать форматирование области текста: *Главная*, кнопка **Шрифт**, кнопка **Очистить формат**.

Убрать цвет выделенного текста: *Главная*, кнопка **Шрифт**, кнопка **Цвет текста**, кнопка **Авто**.

Открыть диалоговое окно *Найти и заменить*: **Ctrl + H**.

Удалить принудительный разрыв строк в тексте (как правило, в текстах, скаченных из сети Интернет): **Ctrl + H**, кнопка **Больше>>**, очистить все поля и снять все флажки с параметров поиска, установить флажок на параметре **Подстановочные знаки**, поставить курсор в поле **Найти**, нажать кнопку **Специальный**, выбрать **Разрыв строки** в поле **Заменить на**, ввести пробел, нажать кнопку **Заменить** или **Заменить все**.

Установить режим замены выделенного фрагмента новым текстом без удаления выделенного (установлен по умолчанию): кнопка **Office**, **параметры Word**, **Дополнительно**, поставить галочку **Заменять выделенный фрагмент**. Если эта галочка не установлена, то вставляемый текст будет вставляться перед выделенным фрагментом.

Скопировать формат с выделенного фрагмента на другой фрагмент текста: выделить фрагмент текста, форматирование которого используется как образец *Главная*, **Буфер обмена**, **Формат по образцу**, «закрасить» кисточкой фрагмент, к которому требуется применить форматирование.

Преобразовать строку таблицы в текст: выделить строку таблицы *Макет*, **Данные**, **Преобразовать в текст**.

Разделить таблицу: *Макет*, **Объединение**, **Разбить таблицу**.

Выводить шапку таблицы в начале каждой страницы: щелчок на первой

строке таблицы (шапке) **Свойства таблицы**, вкладка **Строка**, **Повторять как заголовок на каждой странице**.

Вставить табулятор в ячейку таблицы: **Ctrl+Tab**, а не **Tab**, т. к. клавиша **Tab** в таблицах служит для перехода между ячейками.

Чтобы набрать текст с большим расстоянием между словами, следует использовать позиции табуляции, значок которых находится над левой линейкой. Существует пять основных типов табуляции: L – выравнивание по левому краю; J – выравнивание по правому краю; ┘ – выравнивание по центру; ┘ – выравнивание текста по разделителю (для чисел); ┘ – табуляция с чертой.

Порядок выполнения работы

1 В **Моих документах** создать папку, в ней – документы Word **Введение** и **Теория**, установить необходимые **Отступы и интервалы**.

2 В документе **Введение** создать три страницы текста с заголовками: **Начало**, **История развития Internet**, **Протоколы Internet**.

3 Заполнить файл **Теория** текстом ответов на вопросы.

4 По указанию преподавателя ввести в текст формулы, построить список и таблицу, вставить картинку.

5 Построить оглавление документа.

6 На строках оглавления в файле **Введение** построить гиперссылки на строки, следующие в тексте непосредственно после соответствующих заголовков, а также на файл **История**.

7 Выполнить другие действия в тексте по указанию преподавателя.

Контрольные вопросы

- 1 Как построить гиперссылку в текстовом документе?
- 2 Что такое закладка в текстовом документе и как ее построить?
- 3 Как выделить вертикальный блок текста?
- 4 Назовите основные элементы панели **Рисование**.
- 5 Как вставить графический объект в текстовый документ?
- 6 Как нарисовать эллипс и залить его каким-либо цветом?
- 7 Как нарисовать штрихпунктирную линию, линию со стрелкой?
- 8 Как изменить толщину уже существующей линии?
- 9 Как построить формулу в текстовом документе?
- 10 Как построить расшифровку числа в текстовом документе?
- 11 Как преобразовать строку таблицы в текст?
- 12 Как перенести шапку таблицы на последующие страницы текста?
- 13 Как построить оглавление текстового документа?
- 14 Как быстро изменить регистр выделенного фрагмента текста?
- 15 Как быстро открыть диалоговое окно **Найти и заменить**?

2 Лабораторная работа № 2. Решение задач предметной области с использованием надстройки Поиск решения в среде MS Excel

Цель работы: изучение технологии решения задач предметной области с использованием надстройки **Поиск решения** в среде **MS Excel**.

2.1 Решение системы линейных алгебраических уравнений

Постановка задачи. В блоке ячеек **A2:D5** записаны коэффициенты системы линейных алгебраических уравнений (СЛАУ), а ее свободные члены – в блоке ячеек **B2:B5** (рисунок 2.1).

	A	B	C	D	E	F	G	H	I	J
1			A			B		X		F
2	4	5,7	8	9		12		-6,781		0,0E+00
3	2	4	5,4	7		17		7,543		0,0E+00
4	7	9	3	4		21		-4,136		0,0E+00
5	8	7,5	4	9		15		3,246		0,0E+00
6										
7			МОБР					МУМНОЖ		
8	0,4598	-0,7517	-0,0971	0,1680				-6,781		
9	-0,3943	0,5643	0,2338	-0,1485				7,543		
10	0,5324	-0,4978	0,0081	-0,1488				-4,136		
11	-0,3167	0,4192	-0,1121	0,1516				3,246		

Рисунок 2.1 – Решение системы линейных алгебраических уравнений

Подготовим лист Excel для решения задачи. Для этого в блоке ячеек **J2:J5** построим так называемые невязки в виде:

$$=A2 * H\$2 + B2 * H\$3 + C2 * H\$4 + D2 * H\$5 - F2$$

$$=A3 * H\$2 + B3 * H\$3 + C3 * H\$4 + D3 * H\$5 - F3$$

$$=A4 * H\$2 + B4 * H\$3 + C4 * H\$4 + D4 * H\$5 - F4$$

$$=A5 * H\$2 + B5 * H\$3 + C5 * H\$4 + D5 * H\$5 - F5$$

Вызовем надстройку **Поиск решения** и введем в его окно параметры. **Установить целевую ячейку:** J2 **Равной:** значению 0: **Изменяя ячейки:** H2:H5; **Ограничения:** J3 = 0; J4 = 0; J5 = 0.

Нажмем кнопку **Параметры**. Откроется окно **Параметры поиска решения**, отметим в нем галочкой **Линейная модель** и нажмем **Ок**.

После этого вернется окно **Поиск решения**, в котором нажмем кнопку **Выполнить**, тогда появится окно **Результаты поиска решения** с сообщением **Решение найдено**. Нажмем **Ок** и увидим результаты решения.

Проверить результаты решения можно, воспользовавшись, например, матричным методом. Для этого запишем систему линейных алгебраических уравнений в следующем виде: $|A| \times |X| = |B|$. Здесь **A** – матрица коэффициентов

при неизвестных x ; X – вектор-столбец неизвестных x ; B – вектор-столбец свободных членов.

Умножим обе части приведенного матричного уравнения на обратную матрицу $|A^{-1}|$. Тогда получим: $|A^{-1}| \times |A| \times |X| = |A^{-1}| \times |B|$. Поскольку $|A^{-1}| \times |A| = |E|$ – единичная матрица, то решение матричного уравнения примет вид $|X| = |A^{-1}| \times |B|$.

Следовательно, для решения задачи достаточно найти обратную матрицу, используя функцию Excel **МОБР**, и умножить ее на вектор $|B|$. При этом следует иметь в виду, что матричные операции завершаются нажатием клавиш **Ctrl + Shift + Enter**.

2.2 Решение оптимизационной (транспортной) задачи

Постановка задачи. Пусть производство некоторой продукции осуществляется на четырех предприятиях: A_1, A_2, A_3, A_4 , а затем развозится в пять пунктов потребления этой продукции: B_1, B_2, B_3, B_4, B_5 . На предприятиях A_i ($i = 1, 2, 3, 4$) продукция находится соответственно в количествах a_i условных единиц. В пункты B_j ($j = 1, 2, 3, 4, 5$) требуется доставить b_j единиц продукции. Стоимость перевозки единицы груза с учетом расстояний из A_i в B_j определена матрицей $C = (C_{ij})$.

Предприятия A_i могут выпускать в день 235, 175, 185 и 175 единиц продукции соответственно. Пункты потребления B_j готовы принимать ежедневно 125, 160, 60, 250 и 175 единиц продукции соответственно. Стоимости перевозки единицы продукции с предприятий в пункты потребления приведены в таблице 2.1.

Таблица 2.1 – Стоимости перевозки единицы продукции

Предприятие	Потребитель					Объем производства
	B_1	B_2	B_3	B_4	B_5	
A_1	3,2	3	2,35	4	3,65	235
A_2	3	2,85	2,5	3,9	3,55	175
A_3	3,75	2,5	2,4	3,5	3,4	185
A_4	4	2	2,1	4,1	3,4	175
Потребности	125	160	60	250	175	–

Требуется минимизировать суммарные транспортные расходы по перевозке продукции.

Решение

Построим математическую модель задачи. Неизвестными будем считать объемы перевозок. Пусть x_{ij} – объем перевозок с i -го пункта поставки в j -й пункт потребления.

Суммарные транспортные расходы – это функция

$$F = \sum_{i=1}^n \sum_{j=1}^m c_{ij} x_{ij},$$

где c_{ij} – стоимость перевозки единицы продукции с i -го предприятия в j -й пункт потребления ($i = \overline{1, 4}$; $j = \overline{1, 5}$).

Неизвестные в этой задаче должны удовлетворять следующим ограничениям:

1) объемы перевозок должны быть неотрицательными, т. е. $x_{ij} \geq 0$;

2) поскольку модель сбалансирована, то вся продукция должна быть вывезена с предприятий, а потребности всех пунктов потребления должны быть полностью удовлетворены, т. е. $\sum_{i=1}^4 x_{ij} = b_j$ и $\sum_{j=1}^5 x_{ij} = a_i$.

Таким образом, имеем следующую задачу линейного программирования:

$$\sum_{i=1}^4 \sum_{j=1}^5 c_{ij} x_{ij} \rightarrow \min,$$

т. е. требуется найти минимум функции F при ограничениях:

$$\sum_{i=1}^4 x_{ij} = b_j, \quad j \in [1, 5];$$

$$\sum_{j=1}^5 x_{ij} = a_i, \quad i \in [1, 4];$$

$$x_{ij} \geq 0, \quad i \in [1, 4], \quad j \in [1, 5].$$

Для решения задачи воспользуемся надстройкой **Поиск решения** программы Excel, на **Лист 1** которой в ячейки **B3:F6** введем стоимости перевозок единицы продукции (рисунок 2.2).

В ячейки **B16:F16** введем формулы для расчета суммарной потребности продукции для j -го пункта, т. е. в ячейку **B16** – формулу **=СУММ(B12: B15)** и размножим ее до столбца **F**, в ячейки **G12:G15** – формулы суммарного объема производства i -го предприятия, т. е. в ячейку **G12** – формулу **=СУММ(B12:F12)** и размножим ее до строки 15.

В ячейки **B18:F18** введем значения потребности продукции соответствующего пункта потребления, в ячейки **H12:H15** – значения объемов производства соответствующих предприятий, а в ячейку **B20** – формулу целевой функции, т. е. **=СУММПРОИЗВ(B3:F6;B12:F15)**.

Выполним команду **Данные $\Rightarrow$ Поиск решения**. Откроется окно **Поиск решения** (см. рисунок 2.2), содержащее два блока: **Стоимости перевозок** из предприятий **A1, A2, A3, A4** в пункты потребления **B1, B2, B3, B4, B5** – блок **A1–F6** и **Неизвестные объемы** – блок **A11–H16**. В блоке **B18–F18** содержатся потребности, а в ячейке **B20** – целевая функция, равная произве-

дению стоимости перевозок производимой продукции на их объемы: ***СУММПРОИЗВ(B3:F6;B12:F15)***.

	A	B	C	D	E	F	G	H
1	Предпр	Пункты потребления						
2		B1	B2	B3	B4	B5		
3	A1	3,2	3	2,35	4	3,65		
4	A2	3	2,85	2,5	3,9	3,55		
5	A3	3,75	2,5	2,4	3,5	3,4		
6	A4	4	2	2,1	4,1	3,4		
7								
8								
9		Неизвестные объемы						
10								
11		B1	B2	B3	B4	B5	Ограничения 1	Объем
12	A1	0	0	60	65	110	=СУММ(B12:F12)	235
13	A2	125	0	0	0	50	=СУММ(B13:F13)	175
14	A3	0	0	0	185	0	=СУММ(B14:F14)	185
15	A4	0	160	0	0	15	=СУММ(B15:F15)	175
16	Огранич	=СУММ(B12:B15)	=СУММ(C12:C15)	=СУММ(D12:D15)	=СУММ(E12:E15)	=СУММ(F12:F15)		
17		Потребность						
18		125	160	60	250	175		
19		Целевая функция:						
20		=СУММПРОИЗВ(B3:F6;	<==	СУММПРОИЗВ(B12:F15)				

Рисунок 2.2 – Подготовка *Листа 1* для решения транспортной задачи

В поле ***Установить целевую ячейку*** укажем ячейку, содержащую формулу целевой функции, – ***B20***. Установим переключатель ***Равной*** в положение ***минимальному значению*** (рисунок 2.3).

Рисунок 2.3 – Окно *Поиск решения*

В поле *Изменяя ячейки* зададим диапазон подбираемых параметров: **B12:F15**, а в поле *Ограничения* – необходимые ограничения (см. рисунок 2.2). Нажмем на кнопку *Добавить*, а после ввода всех ограничений – кнопку *Ок*.

Затем нажмем кнопку *Параметры* и в открывшемся окне отметим галочкой *Линейная модель* и нажмем кнопку *Ок*. Отобразится окно *Поиск решения*, в котором нажмем кнопку *Выполнить*. В результате в блоке ячеек **B12:F15** получим оптимальный набор переменных при заданных ограничениях, т. е. оптимальный план перевозок.

В результате решения задачи получили, что предприятие 1 должно поставлять 60 единиц груза потребителю **B3**, 65 единиц груза – потребителю **B4**, 110 единиц груза – потребителю **B5** и т. д. (см. рисунок 2.2). При этом целевая функция принимает минимальное значение, выводимое в ячейке **B20**, равное 2373,5 денежных единиц.

Порядок выполнения работы

- 1 Для указанного преподавателем варианта (таблица 2.2) подготовить на листе Excel исходные данные для решения СЛАУ.
- 2 Выполнить решение СЛАУ матричным методом.
- 3 Проверить полученные результаты.
- 4 Для этого же варианта найти решение СЛАУ с помощью надстройки *Поиск решения*.

Таблица 2.2 – Варианты заданий для решения СЛАУ

Номер варианта	Порядок СЛАУ	Порядок матрицы <i>B</i>	Номер варианта	Порядок СЛАУ	Порядок матрицы <i>B</i>
1	4	1	7	6	2
2	5	1	8	4	2
3	6	1	9	5	1
4	4	1	10	6	1
5	5	2	11	4	2
6	6	2	12	5	2

- 5 Для указанного преподавателем варианта (таблица 2.3) решить транспортную задачу.

Таблица 2.3 – Варианты заданий для решения транспортной задачи

Номер вари- анта	Задание						Номер вари- анта	Задание							
1	$A = 120, 150, 50, 110;$ $B = 100, 60, 40, 130, 70, 30$						7	$A = 110, 160, 80, 90;$ $B = 100, 65, 35, 120, 80, 40$							
	Та- риф	6	7	3	5	1		10	Та- риф	5	6	3	10	7	20
		2	5	4	1	3		6		21	14	9	6	5	3
		7	6	8	9	3		7		10	8	5	7	6	6
		5	1	3	5	7		5		5	1	2	5	1	6
2	$A = 110, 110, 80, 100;$ $B = 90, 50, 40, 120, 70, 30$						8	$A = 130, 150, 80, 90;$ $B = 110, 60, 50, 140, 70, 20$							
	Та- риф	3	10	20	1	6		7	Та- риф	5	7	6	3	6	10
		1	4	1	3	5		5		6	4	1	3	5	4
		10	20	1	6	7		6		2	18	24	16	27	20
		3	6	3	10	1		1		5	6	3	10	20	6
3	$A = 120, 140, 90, 80;$ $B = 110, 160, 40, 130, 70, 30$						9	$A = 110, 115, 85, 100;$ $B = 90, 50, 40, 120, 70, 40$							
	Та- риф	1	4	1	3	5		6	Та- риф	6	5	10	20	1	3
		5	6	3	10	20		5		6	7	3	5	1	10
		2	5	6	3	10		7		2	5	6	3	10	3
		11	9	25	14	3		1		6	11	10	5	6	8
4	$A = 100, 95, 55, 110;$ $B = 35, 60, 70, 90, 45, 60$						10	$A = 170, 125, 75, 80;$ $B = 100, 60, 40, 130, 70, 50$							
	Та- риф	5	3	5	9	6		56	Та- риф	5	7	5	9	12	9
		6	6	17	6	15		3		6	4	8	6	15	7
		2	10	1	2	27		8		2	18	1	2	27	11
		5	4	7	11	20		11		5	16	7	11	20	15
5	$A = 170, 105, 115, 70;$ $B = 100, 55, 35, 90, 80, 60$						11	$A = 130, 95, 115, 70;$ $B = 140, 65, 43, 120, 40, 30$							
	Та- риф	9	3	6	10	20		15	Та- риф	3	5	13	6	8	7
		6	7	1	23	5		11		9	6	7	1	3	10
		2	5	24	6	13		9		16	2	5	24	16	6
		6	11	3	10	15		14		10	15	11	4	10	12
6	$A = 160, 60, 130, 70;$ $B = 90, 50, 40, 120, 70, 50$						12	$A = 170, 55, 155, 90;$ $B = 110, 60, 50, 140, 70, 40$							
	Та- риф	9	15	7	6	8		17	Та- риф	18	11	6	7	5	14
		7	6	4	1	12		10		3	10	20	8	6	11
		5	2	18	24	24		9		1	4	9	13	5	9
		11	5	6	3	13		11		10	20	14	15	17	12

Контрольные вопросы

- 1 Как включить в группу *Анализ* на вкладке *Данные MS Excel* новую надстройку?
- 2 Прокомментируйте матричную формулу решения СЛАУ.
- 3 Прокомментируйте алгоритм перемножения двух матриц.
- 4 Как вычислить обратную матрицу в *MS Excel*?
- 5 Прокомментируйте технологию решения СЛАУ в *MS Excel*.
- 6 Как завершить выполнение матричной операции в *MS Excel*?
- 7 Как перемножить две матрицы в *MS Excel*?
- 8 Какую размерность имеет матрица двух перемножаемых матриц?
- 9 Какая функция используется для перемножения матриц в *MS Excel*?
- 10 Как проверить достоверность перемножения матриц?
- 11 Как в *MS Excel* перемножить две матрицы?
- 12 Сформулируйте постановку транспортной задачи.
- 13 Прокомментируйте структуру таблицы транспортной задачи.
- 14 Как построить формулу для целевой функции?
- 15 Что следует ввести в поле *Установить целевую ячейку* окна *Поиск решения*?

3 Лабораторная работа № 3. Решение задач предметной области на языке VBA в среде программы MS Excel

Цель работы: изучение технологии объектно ориентированного программирования задач предметной области на языке *VBA* в среде *Excel*.

3.1 Основы языка VBA

Основные типы данных, используемые в языке программирования *VBA*, приведены в таблице 3.1.

Константы, переменные и массивы.

Константа – это величина, которая определяется в программной единице один раз и ее значение не может быть изменено в пределах данной программной единицы. Объявление констант выполняется оператором *Const*, который имеет следующий формат:

***Const* <имя константы> = <значение>**

Например,

***Const nk* = 9**

– константа целого типа;

***Const pi* = 3.14159265358**

– константа вещественного типа;

***Const vr_goda* = "Лето"**

– константа строкового типа.

При определении констант их тип можно задать следующим образом:

% – *Integer*; **&** – *Long*;

! – *Single*;

– *Double*; @ – *Currency*; \$ – *String*.

Например,

Const on% = 1 – 2 байта;

Const w& = 27 – 4 байта;

Const tr# = 45 – 8 байтов.

Таблица 3.1 – Основные типы данных *VBA*

Тип данных	Размер, байт	Диапазон значений
Byte (байт)	1	От 0 до 255
Boolean (логический)	2	True или False
Integer (целое число)	2	От -32768 до 32767
Long (длинное целое)	4	От -2147483648 до 2147483647
Single (число с плавающей точкой обычной точности)	4	От -3,40E38 до -1,40E-45 для отрицательных. От 1,40E-45 до 3,40E38 для положительных
Double (число с плавающей запятой двойной точности)	8	От -1,79E308 до -4,9E-324 для отрицательных. От 4,94E-324 до 1,79E308 для положительных
Date (даты и время)	8	От 1 января 100 г. до 31 декабря 9999 г.
String (строка)	10 + длина строки	От 1 до ... примерно 65 400
Variant (произвольный с числами)	16 + численное значение	Любое численное значение вплоть до типа Double
Variant (произвольный с символами)	22 + длина строки	Такой же, как у строки переменной длины

Переменная – это величина, которая может принимать различные значения в пределах программной единицы.

Массив – это поименованная совокупность величин. Объявление переменных и массивов в программах выполняется с помощью оператора ***Dim*** следующим образом:

Dim <имя переменной или массива> (<размерность>) [*As* <*mun*>]

Например,

Dim *x As Single, i As Integer*

Dim *z(5, 8) As Double, prm As Long, n As integer*

Dim *al21*

Если [*AS* <*mun*>] отсутствует, то используется по умолчанию тип *Variant*.

В *VBA* имеется возможность работать с массивами переменной длины. В этом случае массив объявляется без указания его размерности:

Dim <переменная> () [*As* <*mun*>]

Затем в подходящем месте программы записывается оператор:

Redim [*Preserve*] <переменная> (<макс. размер>) [*As* *mun*>]

Здесь **Preserve** указывает, что при переопределении массива предыдущие значения не уничтожаются.

Явное объявление переменной является хорошим стилем программирования и имеет приоритет: время выполнения программы ускоряется, т. к. в памяти компьютера создаются все объявленные переменные до ее выполнения.

Чтобы обезопасить себя от неявного объявления переменной, необходимо в самом начале программы прописать **Option Explicit**.

В итоге, если в программе используется идентификатор, которого нет в блоке **dim**, то возникнет ошибка.

Выражения.

В языке **VBA** используются следующие виды выражений:

- арифметические выражения;
- логические выражения;
- выражения отношения;
- строковые выражения.

Выражения строятся из операндов и операций.

В качестве операндов используются переменные, константы, элементы массивов и функции.

Операции.

В выражениях используются операции:

- арифметические:
 - а) сложение (+);
 - б) вычитание (–);
 - в) умножение (*);
 - г) деление (/);
 - д) возведение в степень (^);
 - е) целочисленное деление (\);
- логические:
 - а) **NOT** – операция «не» (инверсия);
 - б) **OR** – операция «или» (логическое сложение);
 - в) **AND** – операция «и» (логическое умножение);
 - г) **XOR** – операция «исключающее или»;
 - д) **EQV** – операция «проверка логического равенства двух выражений»;
 - е) **IMP** – операция «импликация» и др.;
- операции отношения (сравнения) (таблица 3.2);
- операции с данными типа **String** (строковые): +, & – слияние строк.

Таблица 3.2 – Операции отношения (сравнения)

Операция	Комментарий	Операция	Комментарий	Операция	Комментарий
>	Больше	>=	Больше либо равно	=	Равно
<	Меньше	<=	Меньше либо равно	<>	Не равно

Краткий обзор основных операторов VBA.

Оператор – это элемент языка, определяющий свойства (типы) данных, либо действия, которые необходимо над ними выполнить.

В языке *VBА* используются следующие типы операторов:

- операторы описания;
- операторы присваивания;
- операторы условные;
- операторы выбора;
- операторы организации циклов;
- операторы (функции) ввода-вывода.

Оператором описания в языке *VBA* является уже рассмотренный оператор *Dim*, а в качестве операторов присваивания используются арифметический и логический операторы присваивания, синтаксис которых имеет вид

$$a = b,$$

где a – переменная или элемент массива;
 b – арифметическое либо логическое выражение.

Например:

- арифметический оператор присваивания

$$Sigma = (a * x^2 - b^{4.25}) / (1 - b * Log(x))$$

- логический оператор присваивания

$$L = ((\alpha - k) > (m + s)) \text{ And } fl \leq a^3$$

Условный оператор **If** используется для изменения естественного порядка выполнения операторов. Он выполняет одну или другую последовательность операторов в зависимости от условия – истинности логического выражения.

Синтаксис оператора *If*:

If <условие> **Then** <группа операторов>

или

```
If <условие> Then
    < группа операторов 1>
Else
    < группа операторов 2>
```

End If

или

```

If <условие 1> Then
    < группа операторов 1>
ElseIf < условие 2> Then
    < группа операторов 2>
Else
    < группа операторов 3>
End If

```

Здесь <условие> – выражение, дающее результат *True* или *False*. (*Null* = *False*).

В качестве <группа операторов> в частном случае может быть использован

один оператор.

Оператор выбора **Select Case** выполняет блок операторов в зависимости от значения выражения.

Синтаксис оператора:

```
Select Case < выражение>
  [Case < список 1>
    < Блок операторов 1>]
  [Case < список 2>
    < Блок операторов 2>]
  ...
  [Case < список n >
    < Блок операторов n>]
  [Case Else
    < Блок операторов n>]
End select
```

Здесь <выражение> – любое числовое или строковое выражение; <список 1>, ... , <список n> список одного или более значений в форме <выражение>; либо

<выражение> **To** <выражение>

либо

Is ≤операция сравнения> <выражение>

Если значение <выражение> совпадает с одним из значений <список i>, то выполняется блок < Блок операторов i> и управление передается оператору, следующему за **End select**.

Операторы цикла.

Операторы цикла позволяют повторить блок операторов до тех пор, пока верно некоторое условие, или заданное число раз. В языке VBA используются следующие операторы цикла:

- **Do...Loop**;
- **While...Wend**;
- **For...Next**.

Оператор **Do...Loop** повторяет блок операторов до тех пор, пока условие верно, либо пока условие не станет верным.

Синтаксис операторов цикла.

Оператор **Do...Loop** может использоваться в двух формах. Первая форма называется формой с предусловием, т. к. в ней условие выхода из цикла указывается в начале, после **Do** [{**While** | **Until**}]:

```
Do [{While | Until} условие]
  [Блок операторов]
[Exit Do]
```

[Блок операторов]

Loop

Вторая форма называется формой с постусловием, т. к. в ней условие выхода из цикла указывается в конце, после **Loop** [**While** | **Until**]:

Do

[Блок операторов]

[**Exit Do**]

[Блок операторов]

Loop [{**While** | **Until**} условие]

Параметр **While** означает: «пока условие верно» область цикла будет выполняться. Параметр **Until** означает: «пока условие не станет верным» область цикла будет выполняться.

Оператор [**Exit Do**] выполняет выход из области цикла при выполнении некоторого условия. Обычно он используется вместе с оператором **If**.

Оператор **While ... Wend** выполняет в цикле последовательность операторов до тех пор, пока верно задаваемое условие.

Синтаксис оператора:

While [условие]

[Блок операторов]

Wend

где [условие] – логическое выражение или выражение отношения.

Операторы **For ... next** и **For Each ... next**.

Оператор **For ... next**.

Синтаксис оператора:

For i = in To ik [Step_h]

[блок операторов]

[**Exit For**]

[блок операторов]

Next [i]

Здесь **i** – управляющая переменная цикла; **in**, **ik** – начальное и конечное значения управляющей переменной цикла; **h** – шаг ее изменения.

Оператор **For ... next** циклически повторяет группу операторов, заключенных между строкой **For i = in To ik** и строкой **Next i**. При этом количество повторений будет равно $(ik - in) / Step_h + 1$.

Возможен досрочный выход из цикла с помощью оператора **Exit For**, который может использоваться, например, совместно с оператором **If**.

Оператор **For Each ... next**.

Синтаксис оператора:

For Each a In group

[блок операторов]

[**Exit For**]

[блок операторов]
Next [element]

Здесь a – переменная, используемая для обозначения элементов коллекции или группы **group**; **group** – имя коллекции объектов, группы элементов или массива.

Это также оператор цикла. Он на каждом шаге своего выполнения присваивает переменной a очередной элемент из коллекции **group** и выполняет [блок операторов]. Здесь также возможен досрочный выход из цикла с помощью оператора **Exit For**, который может использоваться, например, совместно с оператором **If**.

Порядок выполнения работы

Составить процедуру формирования массива $x(25)$ вещественного типа и вычисления указанных в таблице 3.3 параметров (вариант – по указанию преподавателя). Величина r , используемая в выражениях таблицы – это формируемое функцией **Rnd** псевдослучайное число в интервале $[0, 1]$.

Таблица 3.3 – Варианты заданий

Номер варианта	Задание
1	Рассчитать элементы массива по правилу $x_i = \cos(r)$. Определить максимальный элемент массива и его порядковый номер
2	Рассчитать элементы массива по правилу $x_i = \sin(r)$. Вычислить сумму и среднее арифметическое положительных элементов массива
3	Рассчитать элементы массива по правилу $x_i = (i + 1)\cos(r)$. Найти минимальное, максимальное и среднее арифметическое значения элементов массива
4	Рассчитать элементы массива по правилу $x_i = \sin(r) - \cos(r)$. Создать двухмерный массив y , первый столбец которого содержит положительные элементы массива x , а второй – отрицательные
5	Рассчитать элементы массива по правилу $x_i = \sin(2r) - \cos(r)$. Переписать в отдельный массив элементы с номерами, кратными 3, и определить их количество
6	Рассчитать элементы массива по правилу $x_i = \sin(1,5r) + \cos(r)$. Переписать в массив Y элементы массива X с четными номерами и определить их количество
7	Рассчитать элементы массива по правилу $x_i = \sin(r) - \cos(r)$. Поменять местами максимальный и минимальный элементы массива
8	Рассчитать элементы массива по правилу $x_i = \sin(r) - \cos(r)$. Найти сумму абсолютных значений максимального и минимального элементов
9	Рассчитать элементы массива по правилу $x_i = \sin(r) - \cos(r)$. Заменить все отрицательные элементы массива на 0
10	Рассчитать элементы массива по правилу $x_i = \sin(3r) - \cos(r)$. Определить минимальный элемент массива и его порядковый номер
11	Рассчитать элементы массива по правилу $x_i = \sin(3r) + \cos(2r)$. Определить сумму отрицательных элементов массива и их количество
12	Рассчитать элементы массива по правилу $x_i = 2\sin(r) + \cos(r)$. Найти отклонение каждого элемента массива от их среднего арифметического значения

Контрольные вопросы

- 1 Какие типы данных, используемых в языке **VBA**, Вы знаете?
- 2 Какие логические операции и операции отношения Вы знаете?
- 3 Как определяется порядок выполнения операций в арифметических и логических выражениях?
- 4 Какие функции ввода-вывода языка **VBA** Вы знаете?
- 5 Какие операторы организации цикла языка **VBA** Вы знаете?
- 6 Какие варианты оператора цикла **For** языка **VBA** Вы знаете?
- 7 Какие варианты операторы цикла **Do** языка **VBA** Вы знаете?
- 8 Какие варианты использования оператора **If** Вы знаете?
- 9 Как работает оператор **For Each ... next**?
- 10 Назначение оператора **Select Case** языка **VBA**.
- 11 Как работает оператор **Select Case**?
- 12 В чем состоит отличие функции от процедуры?
- 13 Назначение ключевых слов **ByVal** и **ByRef** в языке **VBA**.
- 14 Прокомментируйте алгоритм сортировки массива.
- 15 Прокомментируйте алгоритм перемножения матриц.

4 Лабораторная работа № 4. Построение и редактирование графических элементов в среде графических редакторов

Цель работы: изучение технологий построения графических элементов и работы со слоями в среде программы Photoshop.

4.1 Методические указания

Наиболее эффективная работа в Photoshop производится при использовании слоев. В этом случае формируемый рисунок создается на основе нескольких слоев. По окончании работы с рисунком все слои можно объединить в один. Просмотреть слои можно в окне слоев на вкладке **Layers**.

Задание

Построить логотип, представленный на рисунке 4.1.



Рисунок 4.1 – Логотип

Порядок выполнения работы

- 1 Открыть программу Photoshop.
- 2 Создать новый документ размером 400 × 200 пикселей: Меню **Файл (File) – Новый (new)**.
- 3 Для удобства более точного перемещения графических элементов вывести на экран линейки: **View ⇒ Rulers**, или **Ctrl + R**. В окне появятся вертикальная (вверху окна) и горизонтальная (слева) линейки.
- 4 Переместить курсор мыши на горизонтальную линейку и вытащить ее в центр экрана, аналогично и вертикальную линейку (рисунок 4.2).
- 5 Создать новый слой: слой (**Layer**) ⇒ новый (**New**) ⇒ слой.
- 6 На новом слое при помощи инструмента  создать круг в середине документа (белое окошко).
- 7 С помощью инструмента **Paint Bucket**  (заливка) залить «круглое выделение» синим цветом.
- 8 Внутри синего создать концентрическую окружность меньшего диаметра и нажать клавишу **Del**, чтобы удалить синюю заливку внутреннего круга. Для создания концентрических окружностей можно использовать команду **Select ⇒ Modify ⇒ Contract...**
- 9 Нажать **Ctrl+d**, чтобы отменить выделение внутреннего круга.
- 10 При помощи прямоугольного выделения создать выделение по центру круга и нажать клавишу **Del**, чтобы удалить синий фон в области прямоугольника (рисунок 4.3).
- 11 Щелкнуть правой кнопкой мыши на прямоугольном выделении и выполнить трансформацию выделения (**Transform Selection** в контекстном меню). Повернуть его на 90 град. Нажать **Enter**, затем **Del**. При помощи этого метода создать фигуру, как показано на рисунке 4.4.
- 12 Выбрать инструмент **Прямоугольное выделение** и выделить построенное кольцо с разрезами.
- 13 Выполнить команду **Edit ⇒ Transform ⇒ Distort** (или **Perspective**) и с помощью маркеров (рисунок 4.5), трансформировать кольцо к виду, как на рисунке 4.1, и нажать **Enter**. Команды **Distort** и **Perspective** можно вызвать также из контекстного меню на построенном элементе.
- 14 Скопировать слой **Layer 1 Copy**: выбрать инструмент выделения , при нажатой клавише **Alt** скопировать слой. При этом образуется новый слой **Layer 1 Copy 2**.
- 15 В окне слоев переместить курсор мыши на слой **Layer 1 Copy**, нажать правую кнопку и выбрать команду **Blending Options**, откроется окно **Layer Style**, в котором движком **Opacity** установить прозрачность слоя 61 %.
- 16 Скопировать слой еще раз и уменьшить прозрачность до 30 %.
- 17 При помощи инструмента текст  создать надпись с любым текстом и выбрать шрифт, например **Arial**, нажать кнопку **Ok**.

18 Объединить все слои: **Layer** $\Rightarrow$ **Merge Visible** или **Shift+Ctrl+E**.

19 Сохранить файл для **Web**. Для этого выбрать команду **Save for Web...** в меню **File**, установить восемь цветов (**Colors**) для созданного рисунка. На вкладке **Image Size** в окне **New Size** установить 25 %. При этом размер файла должен составить примерно 1 Кб.

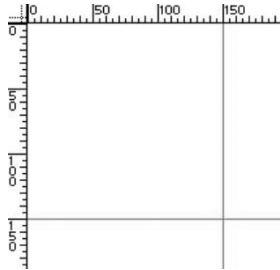


Рисунок 4.2 – Горизонтальная и вертикальная линейки

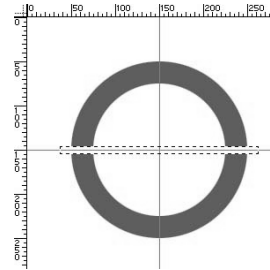


Рисунок 4.3 – Выделение в круге



Рисунок 4.4 – Кольцо с разрезами

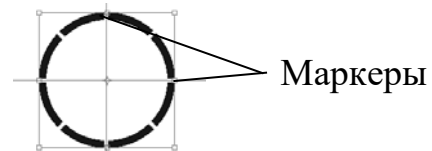


Рисунок 4.5 – Маркеры на выделенном элементе

Контрольные вопросы

- 1 Назначение программы Photoshop.
- 2 Для чего используются слои в программе Photoshop?
- 3 Как создать новый слой?
- 4 Как отменить выделение графического фрагмента в окне программы Photoshop?
- 5 Как сделать слой полупрозрачным?
- 6 Как объединить слои?
- 7 Как построить элемент анимации?
- 8 Как отменить семь последних выполненных действий?
- 9 Как построить концентрические окружности?
- 10 Как построить надпись на области рисунка?
- 11 Как создать текстовый элемент?
- 12 Как сохранить построенный графический объект для **Web**?
- 13 Как уменьшить размер графического файла при его сохранении?
- 14 Как вставить построенный графический элемент в текстовый документ?
- 15 Для чего используются наборы слоев в Photoshop?

5 Лабораторная работа № 5. Проектирование Web-документа на языке HTML

Цель работы: изучение технологии создания простейшей странички *Internet*.

5.1 Краткие теоретические сведения

В сети *Internet* информация представляется в виде сайтов. Сайт может содержать отдельные странички. К сайтам *Internet* предъявляется ряд требований, таких как актуальность, информативность, минимальный размер файлов и др. Особые требования к графическим фрагментам сайтов – высокое качество при их минимальном размере.

HTML (Hyper Text Markup Language) – это язык, предназначенный для создания форматированного текста, который насыщен изображениями, звуком, анимацией и ссылками на другие объекты, например гипертекстовые документы, графические файлы и т. д., разбросанные по всему пространству Web-серверов Интернет/Инtranет.

HTML – язык разметки гипертекста. Он используется для подготовки документов для представления их в *Internet*. В его состав входит ряд тегов (команд) с помощью которых формируются сайты.

Задание

В таблице 5.1 приведены исходные данные к заданию.

Таблица 5.1 – Исходные данные

Иллюстрация	Фирма	Иллюстрация
Общие сведения	Фирма «Финал» создана в 2024 г. Основными направлениями ее деятельности является оказание услуг населению по поставке, техническому обслуживанию и ремонту продукции электротехнической промышленности	Руководство
Услуги		Сотрудники
Товары		Сотрудничество
Цены		Фотогалерея
Контакты		Филиалы
Адрес фирмы		Услуги
Адрес электронной почты		Реклама

Порядок выполнения работы

- 1 Ознакомиться с методическими рекомендациями.
- 2 Получить у преподавателя задание на выполнение работы.
- 3 Составить и отладить HTML-код.

Контрольные вопросы

- 1 Из каких разделов состоит HTML-документ?
- 2 Для чего нужен раздел заголовка HTML-документа?
- 3 Как изменить параметры текста в HTML-документе (размер, цвет, стиль)?
- 4 Как построить таблицу в HTML-документе?
- 5 Как объединить ячейки, строки в таблице?
- 6 Как вставить рисунок в HTML-документ?
- 7 Как построить гиперссылку в HTML документе?
- 8 Как построить сайт фреймовой структуры?
- 9 Как построить заголовок в HTML-документе?
- 10 Какие типы заголовков Вы знаете?
- 11 Как построить список в HTML-документе?
- 12 Какие типы списков Вы знаете?
- 13 Какие версии HTML Вы знаете?
- 14 Какой параметр отвечает за отображение границ таблицы?
- 15 Как построить гиперссылку на графическом элементе?

6 Лабораторная работа № 6. Реализация имитационной модели технической системы в среде программы Simulink пакета MatLab

Цель работы: изучение технологии использования программы **Simulink** пакета **MatLab** для решения задачи имитационного моделирования динамической системы.

6.1 Краткие теоретические сведения

Часто при решении задач численными методами значимую роль оказывает удобство программирования, наличие проверенных математических методов, легкость представления результатов моделирования. Такими особенностями обладает пакет **MatLab**, содержащий в своем составе инструмент визуального моделирования **Simulink**.

Simulink сочетает в себе наглядность аналоговых машин, точность цифровых вычислительных машин и обеспечивает пользователю доступ ко всем возможностям пакета **MatLab**, в том числе к большой библиотеке численных методов.

Подготовка задачи для моделирования в **Simulink** проводится в следующей последовательности.

- 1 Выбор расчетной схемы.
- 2 Составление системы уравнений, описывающих исследуемый процесс.
- 3 Приведение системы к виду, удобному для решения (разрешение

относительно старших производных).

- 4 Определение начальных условий.
- 5 Составление структурной схемы.
- 6 Моделирование возмущающих функций.
- 7 Определение исходных данных.
- 8 Составление модели в среде *Simulink*.
- 9 Включение средств визуализации.
- 10 Тестирование.
- 11 Решение.
- 12 Анализ результатов.

Рассмотрим пример модели *Simulink* имитационного визуального моделирования вертикальных колебаний экипажа двухмассовой динамической системы.

Для исследования влияния основных параметров экипажа на вертикальные колебания используют упрощенную модель с двумя степенями свободы, в которой две массы связаны упругими и диссипативными связями (рисунок 6.1). Такая модель описывает вертикальные колебания экипажей с двухъярусным подвешиванием.

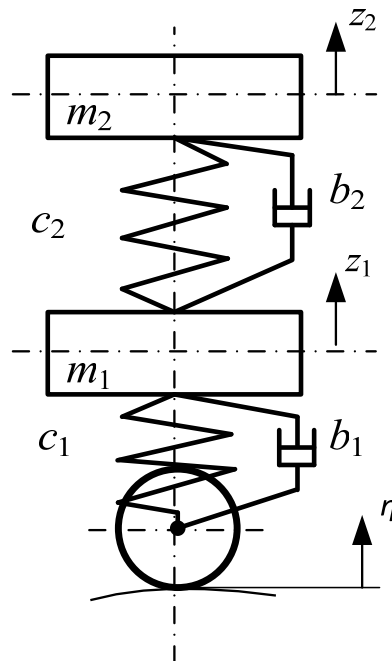


Рисунок 6.1 – Расчетная схема динамической системы

Движение рассматриваемой системы при наличии возмущений со стороны опорной поверхности описывается следующими дифференциальными уравнениями:

$$\begin{cases} m_1 \ddot{z}_1 + b_1 \dot{z}_1 + b_2 (\dot{z}_1 - \dot{z}_2) + c_1 z_1 + c_2 (z_1 - z_2) = b_1 \dot{\eta}(t) + c_1 \eta(t); \\ m_2 \ddot{z}_2 + b_2 (\dot{z}_2 - \dot{z}_1) + c_2 (z_2 - z_1) = 0, \end{cases} \quad (6.1)$$

где m_1 – прорессоренная масса тележки;

m_2 – масса кузова, приведенная к одной тележке;

$z_1, z_2, \dot{z}_1, \dot{z}_2, \ddot{z}_1, \ddot{z}_2$ – обобщенные координаты и их производные по времени;

c_1, b_1 – жесткость и демпфирование в первом ярусе подвешивания;
 c_2, b_2 – жесткость и демпфирование во втором ярусе подвешивания;
 $\eta(t)$ – возмущение со стороны опорной поверхности.

Преобразуем уравнения движения к виду

$$\begin{cases} \ddot{z}_1 = -\frac{1}{m_1}(b_1\dot{z}_1 + b_2(\dot{z}_1 - \dot{z}_2) + c_1z_1 + c_2(z_1 - z_2) - b_1\dot{\eta}(t) - c_1\eta(t)); \\ \ddot{z}_2 = -\frac{1}{m_2}(b_2(\dot{z}_2 - \dot{z}_1) + c_2(z_2 - z_1)) = 0. \end{cases} \quad (6.2)$$

В качестве возмущения используем неровность профессора Н. Н. Кудрявцева. Неровность хорошо описывает изменение прогиба вдоль опорной поверхности. Модель неровности представляет собой сумму полуволны синусоиды частотой ω и трех полуволн синусоиды частотой 3ω , уложенных на длине отрезка пути L . Амплитуды неровностей A_1 и A_2 выбираются в зависимости от типа и состояния опорной поверхности.

$$\eta(t) = |A_1 \sin(\omega t) + A_2 \sin(3\omega t)|, \quad (6.3)$$

где ω – частота возмущения, $\omega = \pi V/L$;

V – скорость движения.

Построим описанную выше модель в среде **Simulink**, при запуске которой открываются два окна.

1 Пустое рабочее окно заготовка для создания новой модели (untitled).

2 Окно библиотеки **Simulink**, содержащей наборы основных разделов (**Library: Simulink**).

Далее можно раскрыть необходимые разделы библиотеки **Simulink** (рисунок 6.2).

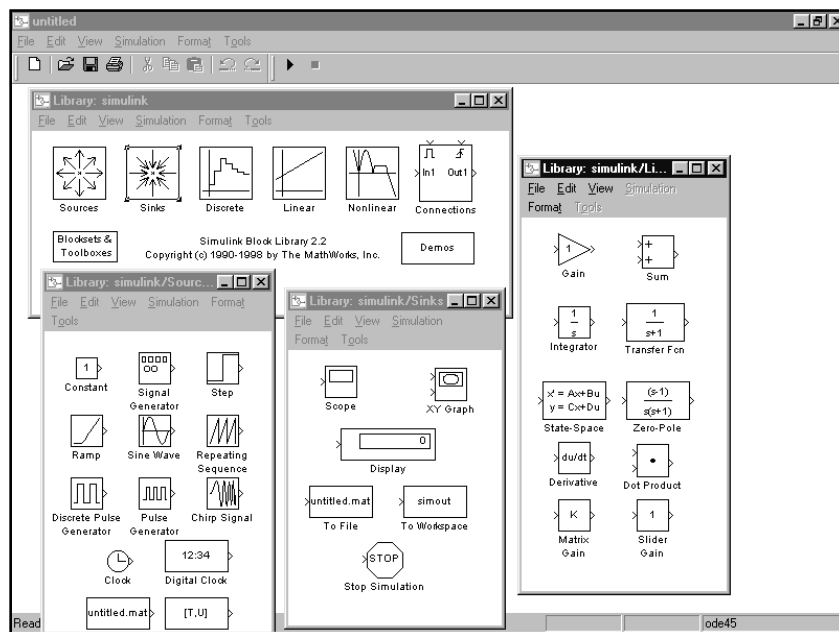


Рисунок 6.2 – Пример начала работы в **Simulink**

Процесс построения модели в системе *Simulink* представляет последовательность выбора необходимых блоков из соответствующих библиотек и соединение их связями.

Построение модели каждого уравнения начинаем с сумматора, имеющего столько входов, сколько членов содержит правая часть уравнения. Для первого уравнения это сумматор **Sum1** (рисунок 6.3). Входы сумматора могут иметь как положительное значение, так и отрицательное. Используемые блоки рекомендуется именовать для облегчения последующей проверки и анализа.

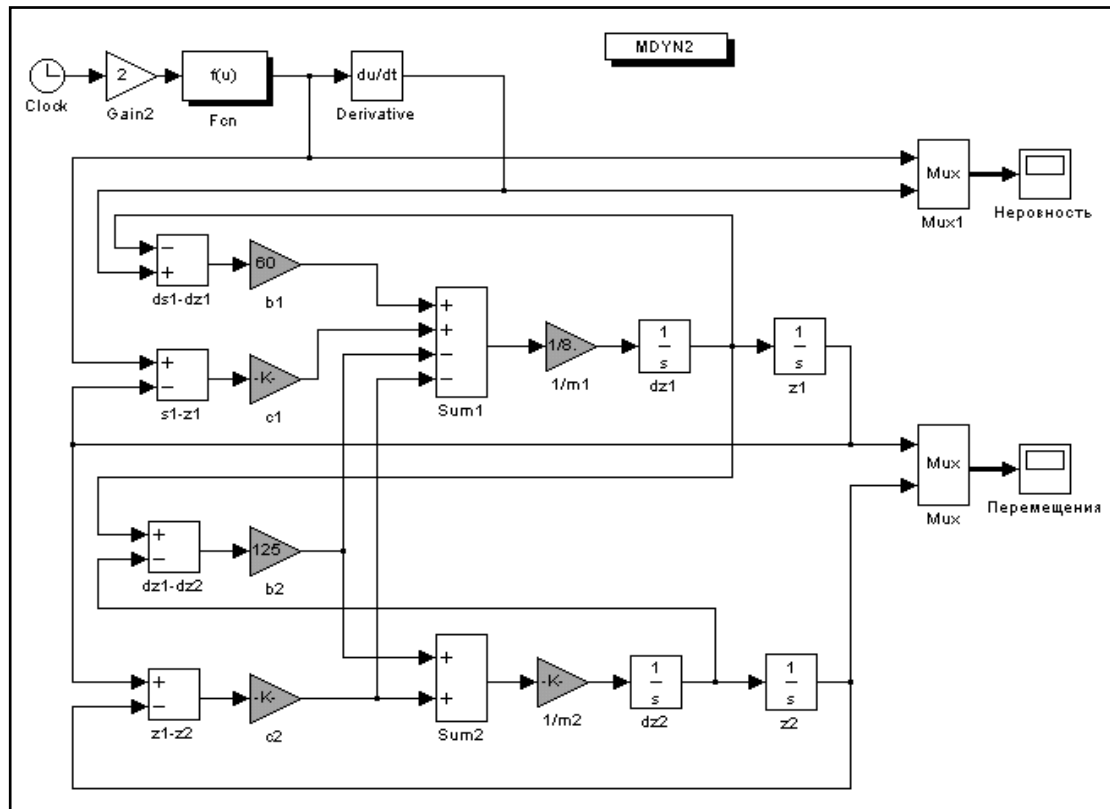


Рисунок 6.3 – Блок-схема модели

К выходу сумматора подключаем линейный преобразователь (множитель $1/m_1$), на выходе которого получаем значение второй производной. Далее включаются последовательно два интегратора ($dz1$, $z1$), на выходе которых получают значения первой производной и самой переменной.

Поскольку решаются два уравнения, то аналогично процесс повторяется и для второго уравнения.

Далее устанавливаем связи между входами и выходами соответствующих блоков, применяя, где необходимо, дополнительные линейные преобразователи и сумматоры.

Для моделирования возмущения используем функциональный блок **Fcn**, в котором в аналитическом виде запишем зависимость перемещения неровности от времени. Блок имеет вход, на который через линейный преобразователь подается от таймера независимая переменная – время.

Для вычисления производной перемещения неровности используется блок дифференцирования.

После установки всех связей следует определить необходимые коэффициенты в используемых преобразователях.

Порядок выполнения работы

- 1 Изучить методические рекомендации.
- 2 Выполнить решение задания, предложенного в таблице 6.1.

Таблица 6.1 – Варианты заданий

Номер варианта	A_1 , м	A_2 , м	L , м	m_1 , т	c_1 , кН/м	b_1 , кН·с/м	m_2 , т	c_2 , кН/м	b_2 , кН·с/м
1	0,005	0,003	25,8	8,75	7000	60,4	25,6	2500	125
2	0,004	0,002	20,5	8,52	6500	55,8	21,8	2200	120
3	0,004	0,003	22,4	7,85	6200	50,5	24,4	2000	132
4	0,005	0,003	28,2	6,97	6800	72,2	32,9	2800	128
5	0,005	0,004	35,4	9,25	7250	65,7	28,5	2600	126
6	0,003	0,002	32,5	7,65	7170	68,3	31,9	2700	126
7	0,003	0,002	38,7	8,15	6400	67,7	30,9	2820	121
8	0,007	0,005	34,7	9,12	6340	66,6	29,7	2770	124
9	0,007	0,004	35,2	8,41	7450	67,5	31,2	2810	127
10	0,006	0,003	35,4	6,17	6550	58,2	30,7	2720	122
11	0,006	0,004	31,2	5,25	5170	62,5	28,9	2800	120
12	0,004	0,002	30,5	7,14	7420	58,2	32,8	2620	118

Контрольные вопросы

- 1 Назначение программы **Simulink**.
- 2 Как запустить на выполнение модель **Simulink**?
- 3 Какие измерительные блоки используются при выполнении лабораторной работы и в каких разделах библиотек **Simulink** они хранятся?
- 4 Какое расширение имеет файл модели **Simulink**?
- 5 Как установить параметры моделирования (симуляции)?
- 6 Какие Вы знаете функции для работы с матрицами?
- 7 Какая функция используется для решения уравнений в символьном виде?
- 8 Прокомментируйте правила моделирования систем в **Simulink**.
- 9 Что такое *m*-файл в системе **Simulink**?
- 10 Какие входные и выходные блоки модели **Simulink** Вы знаете?
- 11 Как создать модель в системе **Simulink**?
- 12 Как изменить направление стрелки линии в модели **Simulink**?
- 13 Как передать результаты моделирования в рабочую область **MatLab**?
- 14 Как добавить контакт к элементу модели **Simulink**?
- 15 Как задать параметры и метод моделирования?

7 Лабораторная работа № 7. Разработка презентации в среде программы PowerPoint

Цель работы: приобретение практических навыков создания презентаций.

7.1 Краткие теоретические сведения

PowerPoint – это программа, входящая в состав **MS Office**, и представляющая собой графический пакет подготовки и демонстрации презентаций и слайд-фильмов. PowerPoint содержит инструменты, позволяющие представить указанные элементы на экране монитора пользователю или удаленной широкой аудитории средствами Интернет не покидая офиса. Презентации могут содержать неограниченное количество хранящихся в одном файле страниц, называемых слайдами.

Программа **PowerPoint** создает файл презентации с расширением **.ppt**, который содержит набор слайдов и предоставляет пользователю большое количество шаблонов на различные темы. Такие шаблоны содержат слайды, оформленные определенным образом. В поле слайда можно вставить текст, графику, анимацию, таблицы, диаграммы, видеосюжеты, звуковые эффекты и т. д.

Презентации можно оформить в виде однослайдовых рекламных элементов, объявлений и многослайдовых выступлений.

Программа **PowerPoint** позволяет также представить презентации в Интернет и составить тезисы выступлений с необходимыми иллюстрациями.

Достоинства **PowerPoint** состоят в наличии в ней набора готовых вариантов дизайна, позволяющих на каждом этапе управлять процессом создания слайдов, а также в простоте ее освоения и наличии мастеров, содержащих инструменты широкого выбора звука, музыки, текстуры, фотографий, средств мультимедиа и шаблонов, обеспечивающих высокое качество создаваемых презентаций.

Порядок выполнения работы

1. Выбрать общее оформление.
2. Добавить новые слайды и их содержимое.
3. Выбрать разметку слайдов.
4. Изменить и при необходимости оформить слайды.
5. Заменить цветовую схему.
6. Применить различные шаблоны оформления.
7. Создать эффекты анимации при демонстрации слайдов.

Варианты заданий представлены в таблице 7.1.

Таблица 7.1 – Варианты заданий

Номер варианта	Тема презентации
1	Компьютерные игры и их классификация. Применить новое оформление, цветовую схему и фон для каждого слайда
2	Гороскоп для всех знаков зодиака на следующую неделю. Для каждого слайда применить новое оформление, цветовую схему и фон
3	Краткая биография какой-либо известной личности XX столетия. Применить новое оформление, цветовую схему и фон для каждого слайда
4	Программа телепередач и фильмов с анонсами на следующую неделю. Для каждого слайда применить новое оформление, цветовую схему и фон
5	Распорядок дня. Применить возможности <i>ClipGallery</i> , новое оформление, цветовая схема и фон для каждого слайда
6	Презентация «История вычислительной техники». Применить новое оформление, цветовую схему и фон для каждого слайда
7	Структура и состав персонального компьютера
8	Программное обеспечение компьютера
9	Форматирование текста в <i>MC Word</i>
10	Таблицы в редакторе <i>MC Word</i>
11	Ввод и форматирование данных в <i>MC Excel</i>
12	Создание и использование формул в <i>MC Excel</i>
13	Использование функций в <i>MC Excel</i>
14	Создание и редактирование таблиц в <i>MC Access</i>
15	Условные операторы и операторы цикла <i>VBA</i>

Контрольные вопросы

- 1 Для чего предназначена программа *MS PowerPoint*?
- 2 Из каких действий состоит процесс создания презентаций?
- 3 Что такое слайд?
- 4 Как добавить в презентацию новый слайд?
- 5 Что такое шаблон? Какие существуют виды шаблонов в *PowerPoint*?
- 6 Как установить автоматическое время изменения элементов презентации?
- 7 Какое расширение имеет файл презентации, шаблон презентации?
- 8 Объясните назначение *Режима слайдов*.
- 9 Назовите основные характеристики *Режима сортировщика* слайдов.
- 10 Как применить шаблон оформления ко всем слайдам?
- 11 Как поменять местами слайды презентации?
- 12 Установите разные шаблоны оформления для разных слайдов.
- 13 Что такое *Цветовая схема* слайда? Как ее выбрать?
- 14 Как создать для двух слайдов один шаблон оформления, но разную цветовую схему?
- 15 Как настроить симметричное расположение объектов на слайде?

8 Лабораторная работа № 8. Изучение численных методов решения нелинейных уравнений

Цель работы: изучение и практическая реализация численных методов решения нелинейных уравнений.

8.1 Краткие теоретические сведения

Метод Ньютона, известный как метод касательных, – это итерационный численный метод нахождения корня заданной функции, который впервые был предложен английским физиком, математиком и астрономом Исааком Ньютоном. Поиск решения в данном методе выполняется построением последовательных приближений и основан на принципах простой итерации.

Геометрическая интерпретация метода Ньютона заключается в следующем: задается начальное приближение решения x_0 вблизи предполагаемого корня, после чего строится касательная к исследуемой функции в точке приближения и определяется точка ее пересечения с осью абсцисс. Эта точка является следующим приближением. И так далее, пока не будет достигнута необходимая точность (рисунок 8.1).

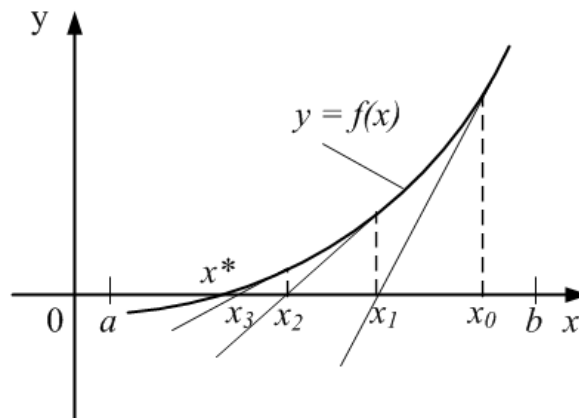


Рисунок 8.1 – Геометрическая интерпретация метода касательных

Для вычисления $n + 1$ -го приближения решения заданной функции на отрезке $[a, b]$ используется следующая итерационная формула:

$$x_{n+1} = x_n - \frac{f(x)}{f'(x)},$$

где $f'(x)$ – производная заданной функции $f(x)$.

Для реализации алгоритма необходимо задать две процедуры-функции для вычисления функции $f(x)$ (*F(x As Double)*) и ее производной $f'(x)$ (*DF(x As Double)*).

Рассмотрим пример решения уравнения $y = \sin(x)$. Полный код программы решения данного уравнения имеет вид:

```

Sub mnt()
  Dim x1 As Double
  Dim x2 As Double
  Dim e As Double
  x1 = Val(TextBox("Введите x1"))
  e = Val(TextBox("Введите точность: e"))
  x2 = x1 - (f(x1) / df(x1))
  Do While (Abs(x2 - x1) > e)
    x1 = x2
    x2 = x1 - (F(x1) / DF(x1))
  Loop
  MsgBox ("x = " + Str((x1 + x2) / 2))
End Sub

```

```

Function f(x As Double) As Double
  Dim y As Double
  y = Sin(x)
  F = y
End Function

```

```

Function df(x As Double) As Double
  Dim y As Double
  y = Cos(X)
  DF = y
End Function

```

Порядок выполнения работы

- 1 Изучить методические рекомендации.
- 2 Выбрать метод решения задачи из таблицы 8.1.
- 3 Выбрать вид функции для решения задачи из таблицы 8.2.
- 4 Составить и отладить программное обеспечение для решения задачи.

Таблица 8.1 – Метод решения задачи

Номер варианта	Метод	Номер варианта	Метод
1; 7	Ньютона	4; 10	Хорд
2; 8	Стефенсена	5; 11	Простой итерации
3; 9	Секущих	6; 12	Дихотомии

Таблица 8.2 – Вид функции

Номер варианта	Функция	Номер варианта	Функция
1	$f(x) = x^{1/2} - x^{-1} \ln(x + 4) - 1,5$	7	$f(x) = \exp(-0,5x) - 0,2x^2 + 1$
2	$f(x) = \cos(x) - \exp(-x) + 0,5$	8	$f(x) = \exp(-0,4x) - 0,5x^2 + 1$
3	$f(x) = 1,5 - 0,4x^{1/3} - 0,5$	9	$f(x) = 1,5 - 0,4x^{1/3} - e^{-x} \sin(x)$
4	$f(x) = 2 - x^{1/3} - 2 \ln(x)$	10	$f(x) = 2 - 0,5x^2 - 0,5x^{-1} \sin(x)$
5	$f(x) = 1 - 0,5x^2 \ln x + 0,3x^{1/2}$	11	$f(x) = 0,3 \exp(-x^2) + x \cos(x)$
6	$f(x) = 1 - x \ln(x) + 0,3x^{1/2}$	12	$f(x) = 0,5 \exp(-x^2) + x \cos(x)$

Контрольные вопросы

- 1 Какие численные методы решения нелинейных уравнений Вы знаете?
- 2 Для чего используется метод деления отрезка пополам?
- 3 Как определить количество решений нелинейного уравнения?
- 4 Прокомментируйте используемый метод решения задачи.
- 5 Как в основной процедуре организована передача параметров в функцию?
- 6 Какие языки программирования можно использовать для решения нелинейного уравнения?
- 7 Какие условные операторы использовались для решения задачи?
- 8 Как повысить точность определения корней нелинейного уравнения?
- 9 В каких единицах измерения задаются аргументы тригонометрических функций?
- 10 Какие методы можно использовать для решения систем нелинейных уравнений?
- 11 Как можно повысить быстродействие алгоритма решения задачи поиска корней уравнения многих переменных?
- 12 Как передается результат вычисления из функции в вызывающую процедуру?
- 13 Какие методы решения системы дифференциальных уравнений Вы знаете?
- 14 Прокомментируйте используемый в процедуре оператор цикла.
- 15 Как организован вывод результатов решения задачи?

9 Лабораторная работа № 9. Изучение численных методов решения задач, описываемых системой ОДУ

Цель работы: изучение технологии решения задач, описываемых системой обыкновенных дифференциальных уравнений (ОДУ), в среде программы *MathCad*.

9.1 Краткие теоретические сведения

Для численного интегрирования обыкновенного дифференциального уравнения в *MathCad* имеется выбор – либо вычислительный блок *Given/odesolve*, либо встроенные функции. Первый путь предпочтительнее из соображений наглядности представления задачи и результатов, а второй дает больше возможностей воздействия на параметры численного метода.

Вычислительный блок *Given/Odesolve* состоит из трех частей:

- 1) *Given* – ключевое слово;
- 2) ОДУ и начальное условие, записанное с помощью логических операторов, причем начальное условие должно быть в форме $y(t_1) = b$;
- 3) *Odesolve(t, t₁, [step])* – встроенная функция для решения ОДУ относительно переменной t на интервале (t_0, t_1) . Здесь *step* – шаг интегрирования.

Пример решения задачи Коши для ОДУ первого порядка $y' = y - y^2$ посредством вычислительного блока *Given*:

Given

$$\frac{d}{dt} y(t) = y(t) - y(t)^2$$

$$y(0) = 0.1$$

$$y = \text{Odesolve}(t, 10)$$

Вставлять логические операторы следует при помощи панели инструментов *Boolean* (булевы операторы). Для ввода с клавиатуры логического знака равенства используется сочетание клавиш **<Ctrl> + <=>**. Символ производной можно ввести как средствами панели *Calculus*, так и в виде штриха, набрав его с помощью сочетания клавиш **<Ctrl> + <F7>**.

MathCad требует, чтобы конечная точка интегрирования ОДУ лежала правее начальной: $t_0 < t_1$ ($t_0 = 0$, $t_1 = 10$), иначе будет выдано сообщение об ошибке. Результатом применения блока *Given/odesolve* является функция $y(t)$, определенная на промежутке (t_0, t_1) .

Пользователь имеет возможность выбирать между двумя модификациями численного метода Рунге – Кутты. Для смены метода необходимо нажатием правой кнопки мыши на области функции *odesolve* вызвать контекстное меню, а в нем – один из двух пунктов: *Fixed* – фиксированный шаг или *Adaptive* – адаптивный. По умолчанию применяется метод с фиксированным шагом.

Встроенные функции *rkfixed*, *Rkadapt*, *Bulstoer*. Альтернативный метод решения ОДУ заключается в использовании одной из встроенных функций: *rkfixed*, *Rkadapt* или *Bulstoer*. Этот способ несколько проигрывает первому в простоте и наглядности, Поэтому более предпочтительным является вычислительный блок *Given/odesolve*.

Пример использования функции *rkfixed*:

```

y = 0.1
D(t, y) = y - y^2
M = 100
y: = rkfixed(y, 0, 10, M, D)

```

Здесь явно задается количество точек интегрирования ОДУ, например, $M = 100$ в третьей строке листинга, а результат получается не в виде функции, а в виде матрицы, состоящей из двух столбцов. В первом столбце находятся значения аргумента, а во втором – соответствующие значения искомой функции $y(t)$.

Для решения задач в *MathCad* можно использовать пользовательские функции. Пример функции RK решения задачи методом Рунге – Кутты:

```

RK(u0, x0, x1, f, N) :=
    tau ← (x1 - x0) / N
    t0 ← x0
    y0 ← u0
    for i ∈ 0.. N - 1
        t_{i+1} ← t_i + tau
        k1 ← f(t_i, y_i)
        k2 ← f(t_i + tau, y_i + tau · k1)
        y_{i+1} ← y_i + tau · (k1 + k2) / 2
    return augment(t, y)

```

Функция RK содержит следующие аргументы: u_0 – начальные условия; x_0 – начало диапазона; x_1 – конец диапазона; f – имя функции; N – количество шагов.

Для вызова функции RK введем переменную M , которой будет присваиваться решение дифференциального уравнения

$$M := RK(u_0, 0, 15, f, 10).$$

Порядок выполнения работы

- 1 Изучить методические рекомендации.
- 2 Выбрать исходные данные из таблицы 9.1.
- 3 Выполнить моделирование процесса нагретого тела на основе модели, предложенной И. Ньютоном:

$$dT/dt = r(T - T_s), \quad (9.1)$$

где T – температура нагретого тела;

T_s – температура окружающей среды;

r – коэффициент остывания тела.

Значение коэффициента остывания зависит от механизма передачи, площади нагретого тела и тепловых свойств самого тела.

- 4 Выполнить решение этого дифференциального уравнения методом Рунге – Кутта в виде пользовательской функции *MathCad*.

- 5 Для изучения точности методов численного решения дифференциальных уравнений воспользоваться аналитическим решением дифференциального уравнения (9.1), которое записывается в виде

$$T(t) = T_s (T_s - T_0) e^{-rt}. \quad (9.2)$$

- 6 Построить зависимости изменения температуры жидкости от времени, полученные аналитически и численно.

Таблица 9.1 – Варианты заданий

Номер варианта	Начальная температура жидкости T_0 , °C	Температура среды T_s , °C	Коэффициент r	Временной интервал
1	95	15	0,024	0...20
2	97	25	0,023	0...30
3	83	15	0,023	0...25
4	87	4	0,023	0...20
5	88	1	0,025	0...30
6	81	22	0,025	0...25
7	85	18	0,025	0...40
8	80	4	0,024	0...35
9	86	30	0,024	0...30
10	82	21	0,024	0...15
11	92	24	0,027	0...24
12	88	12	0,026	0...27

Контрольные вопросы

- 1 Для чего используются обыкновенные дифференциальные уравнения?
- 2 Какие численные методы решения обыкновенных дифференциальных уравнений Вы знаете?

- 3 Какие дифференциальные уравнения используются для моделирования процессов перемещения механических систем?
- 4 Прокомментируйте метод Эйлера решения дифференциального уравнения.
- 5 Какие функции программы **MathCad** используются для решения обыкновенных дифференциальных уравнений?
- 6 Прокомментируйте метод Рунге – Кутты второго порядка.
- 7 Что является решением дифференциального уравнения?
- 8 Для чего используется вычислительный блок *Given MathCad*?
- 9 Какие методы решения ОДУ Вы знаете?
- 10 Как построить систему из семи дифференциальных уравнений второго порядка в системе **MathCad**?
- 11 Как построить циклический блок в системе **MathCad**?
- 12 Как построить функцию в системе **MathCad**?
- 13 Как вставить новый оператор в программу **MathCad**?
- 14 Какие методы решения обыкновенных дифференциальных уравнений Вы знаете?
- 15 Как решить систему дифференциальных уравнений второго порядка?

10 Лабораторная работа № 10. Решение задач предметной области в среде программы MathCad

Цель работы: изучение методов решения уравнений в среде программы **MathCad**.

10.1 Краткие теоретические сведения

Решение нелинейных уравнений.

Для простейших уравнений вида $f(x) = 0$ решение в **MathCad** выполняется с помощью функции **root**:

$$\text{root}(f(x_1, x_2, \dots, x_1, a, b)),$$

которая возвращает значение x_1 , принадлежащее отрезку $[a, b]$, при котором выражение или функция $f(x)$ обращается в 0. Оба аргумента этой функции должны быть скалярами. Функция возвращает скаляр.

Аргументы:

$f(x_1, x_2, \dots)$ – функция, определенная где-либо в рабочем документе, или выражение. Выражение должно возвращать скалярные значения;

x_1 – имя переменной, которая используется в выражении. Этой переменной перед применением функции **root** необходимо присвоить числовое значение. **MathCad** использует его как начальное приближение при поиске корня;

a, b – необязательные параметры, если используются, то должны быть вещественными числами, причем $a < b$.

Нахождение корней полинома.

Для нахождения корней выражения, имеющего вид

$$v_n x^n + \dots + v_2 x^2 + v_1 x + v_0,$$

лучше использовать функцию ***polyroots***, нежели ***root***. В отличие от функции ***root***, функция ***polyroots*** не требует начального приближения и возвращает сразу все корни, как вещественные, так и комплексные.

Polyroots(v) возвращает вектор длины ***n***, содержащий корни полинома степени ***n***. Коэффициенты полинома должны быть записаны в вектор ***v*** длиной ***n + 1***. Рисунок 10.1 иллюстрирует решение уравнений средствами MathCad.

Решение систем уравнений.

MathCad позволяет решать системы уравнений. Максимальное число уравнений и переменных равно 50.

Для решения системы уравнений необходимо:

- 1) задать начальное приближение;
- 2) ввести ключевое слово ***Given***. Оно указывает ***MathCad***, что далее следует система уравнений;
- 3) ввести уравнения системы в любом порядке. Для ввода символа «=» следует использовать ***Ctrl=***;
- 4) ввести любое выражение, которое включает функцию ***Find***, например: ***a:= Find(x, y)***.

Find(z1, z2, ...) возвращает точное решение системы уравнений. Число аргументов должно быть равно числу неизвестных.

Пример использования функции ***Find*** приведен на рисунке 10.1.

Для решения систем линейных уравнений применяется также функция ***lsolve***.

Символьное решение уравнений.

Решение уравнений в символьном виде позволяет найти точные или приближенные значения корней уравнения.

Для решения системы уравнений в символьном виде необходимо:

- 1) ввести ключевое слово ***Given***;
- 2) ввести уравнения в любом порядке ниже слова ***Given***. Удостовериться, что для ввода знака «=» используется ***Ctrl =***;
- 3) ввести имя функции ***Find***, соответствующей системе уравнений;
- 4) нажать ***Ctrl*** (клавишу ***Ctrl***, сопровождаемую точкой). ***MathCad*** отобразит символьный знак равенства;
- 5) щелкнуть мышью на функции ***Find***.

Пример 2 на рисунке 10.1 иллюстрирует символьное решение системы уравнений в ***MathCad***.

Порядок выполнения работы

- 1 Изучить методические рекомендации.
- 2 Выполнить решение варианта задания, предложенного в таблице 2.3.

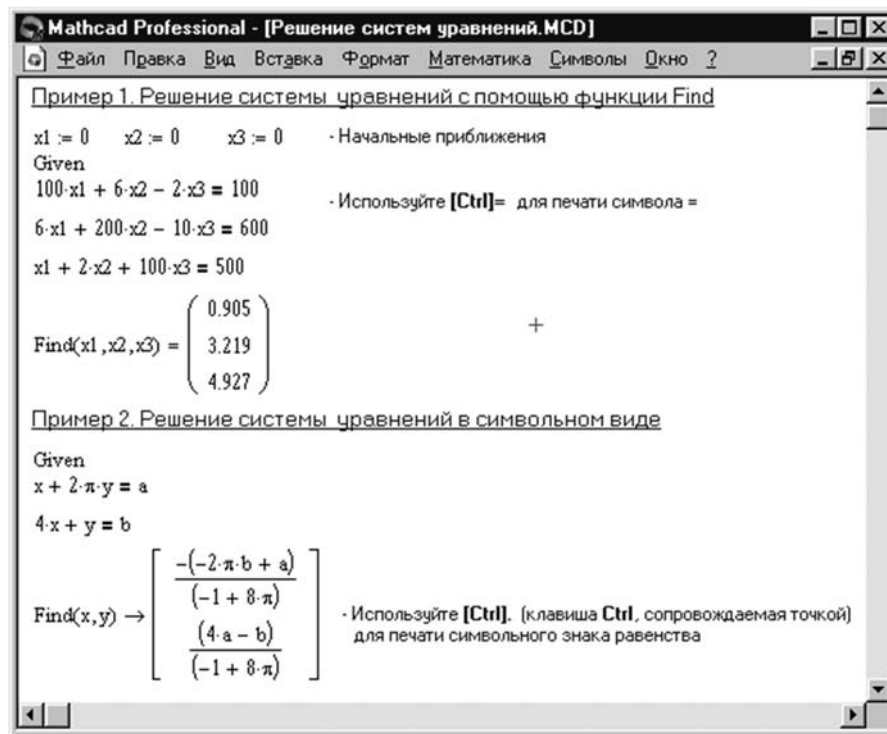


Рисунок 10.1 – Демонстрация решения системы уравнений

Контрольные вопросы

- 1 Какие команды содержит главное меню системы *MathCad*? Их назначение.
- 2 Какие кнопки управления содержит стандартная панель инструментов и какие функции они выполняют?
- 3 Как задаются в *MathCad* числовые константы, комплексные числа, строковые константы, математические выражения?
- 4 Что такое ранжированные переменные?
- 5 Что такое размер и размерность массива в *MathCad*?
- 6 Какие способы построения графиков в *MathCad* Вы знаете?
- 7 Какие типы трехмерных графиков позволяет строить *MathCad*?
- 8 Что такое символьный процессор и каковы его функции?
- 9 Как найти решение нелинейного уравнения с помощью команды *Solve*?
- 10 Как решить нелинейное уравнение с помощью функции *Root*?
- 11 В каких случаях целесообразно использовать функцию *Polyroots*?
- 12 Как решать системы нелинейных уравнений с использованием блока *Given* и функции *Find*?
- 13 Какие уравнения называются дифференциальными?
- 14 Какая функция используется для численного решения обыкновенного дифференциального уравнения?
- 15 Как получить решение уравнения в символьном виде?

11 Лабораторная работа № 11. Решение задачи минимизации функции многих переменных

Цель работы: изучение численных методов минимизации функции многих переменных.

11.1 Краткие теоретические сведения

Поиск минимума функций. Задачи поиска максимума эквивалентны задачам поиска минимума, т. к. требуется лишь поменять знак перед функцией. Поиск минимума функции следует начинать с определения интервала, на котором она имеет наименьшее значение.

Для этого можно использовать:

- графическое представление функции;
- аналитический анализ аппроксимирующей функции;
- сведения о математической модели исследуемого процесса.

Порядок выполнения работы

- 1 Изучить методические рекомендации.
- 2 Разработать алгоритм минимизации функции двух переменных (a и d):

$$u_x = a\sqrt{2}\sin(bt + c) + d\sqrt{2}\sin(et + f). \quad (11.1)$$

Значения переменных b , c , e и f функции (11.1) следующие: $b = -0,55$, $c = 7,25$, $e = 0,74$, $f = 2,27$.

- 3 Выполнить минимизацию функции (11.1), начальные и конечные значения переменных a и d которой заданы в таблице 11.1.

Таблица 11.1 – Значения параметров a , d и их интервалы

Номер варианта	a	a_n	a_k	d	d_n	d_k
1	1,0	0,5	1,75	-1,3	-1,24	-0,04
2	2,0	1,25	2,75	-1,5	-2,47	-0,16
3	3,0	2,45	4,27	-1,7	-2,06	-1,06
4	4,0	3,25	5,09	-1,0	-3,02	-0,24
5	5,0	5,05	7,46	-1,0	-2,28	-0,02
6	6,0	4,55	8,42	-0,8	-2,02	-0,12
7	7,0	6,05	9,08	-0,6	-2,14	-0,02
8	8,0	8,15	10,14	-0,4	-1,84	-0,05
9	9,0	7,35	12,28	-0,2	-2,28	-0,01
10	10,0	8,55	12,05	0,0	-1,70	3,60
11	11,0	9,75	12,24	0,1	-1,17	1,28
12	12,0	10,95	14,65	0,3	-1,30	2,45

Контрольные вопросы

- 1 Как можно локализовать положение минимума функции одной переменной?
- 2 В чем состоит идея метода прямого поиска значения функции?
- 3 Прокомментируйте метод золотого сечения.
- 4 Для чего используют аналитический анализ аппроксимирующей функции?
- 5 Какие алгоритмы минимизации функции многих переменных Вы знаете?
- 6 Для чего используется функция аппроксимации?
- 7 Как выбирают величину шага в методе координатного спуска?
- 8 Для чего можно использовать сведения о математической модели исследуемого процесса?
- 9 В чем состоит сложность процесса минимизации функции многих переменных?
- 10 С чего следует начинать поиск минимального значения функции?
- 11 Для чего используется графическое представление функции аппроксимации?
- 12 В чем суть метода минимизации функции многих переменных?
- 13 Какие методы минимизации функции многих переменных Вы знаете?
- 14 Как вычислить значения коэффициентов функции аппроксимации?
- 15 В чем разница функций минимизации и максимизации?

12 Лабораторная работа № 12. Проектирование базы данных на основе реляционной модели в среде СУБД Access

Цель работы: изучение технологии создания баз данных и работы с ними.

12.1 Краткие теоретические сведения

Программа **MS Access** предназначена для создания и ведения базы данных (**БД**). Главным элементом интерфейса **Access** является лента, на которой собраны все команды, сгруппированные по тематическим вкладкам. Лента имеет вкладки двух типов: основные (**Главная, Создание, Внешние данные, Работа с базами данных**) и контекстные, открывающиеся нажатием правой кнопки мыши на нужном объекте.

Основными объектами, с которыми работает **СУБД Access**, являются таблицы, запросы, формы, отчеты, макросы, модули и страницы.

Таблицы – основные объекты **Access**, в которых хранится вся информация базы данных.

Запросы применяются для выборки данных из таблиц базы данных с использованием условий отбора данных.

Формы необходимы для просмотра данных таблиц или запросов, а также для ввода данных в таблицы **БД**.

Отчет предназначен для формирования выходного документа в пригодном для вывода на принтере виде.

Макрос – это объект, дающий возможность с помощью одной операции

выполнять некоторую последовательность команд.

Модуль представляет собой программу на каком-либо языке программирования, реализующую некоторый алгоритм обработки данных в базе.

Создание базы данных состоит из следующих этапов.

1 Проектирование **БД**. На данном этапе определяется, какие таблицы будут содержаться в **БД**, а также структура таблиц, т. е. их поля, тип, размеры и первичные (главные) ключи таблицы.

2 Создание структуры **БД**, т. е. структуры таблиц, входящих в состав **БД**.

3 Ввод записей – заполнение таблиц базы данных конкретной информацией.

Формирование запросов.

Запросы в **Access** являются основным инструментом выборки, обновления и обработки данных в таблицах базы данных. Для выполнения запросов используется язык структурированных запросов **SQL** (Structured Query Language). С помощью инструкций языка **SQL** реализуются запросы, позволяющие производить поиск, отбор и простейшую обработку данных. В **Access** можно создать запросы: на выборку данных, на создание таблицы, на добавление записей, на удаление записей, на обновление полей.

Запросы **Access** могут быть сформированы на двух языках.

1 **QBE (Query By Example)** – язык запросов по образцу, предназначен для пользователей не программистов. Этот язык используют и квалифицированные пользователи-программисты с целью удешевления работы по созданию пользовательского приложения.

2 **SQL (Structured Query Language)** – структурированный язык запросов, используемый пользователями-программистами в программах приложений на языке программирования **VBA**. Язык **SQL** содержит больше возможностей по сравнению с языком запросов **QBE**. Все, что можно сформулировать на языке **QBE**, можно сформулировать и на языке **SQL**, но не наоборот. Например, язык **QBE** не позволяет создавать и редактировать структуру таблицы.

Создание форм, используемых для удобной работы с данными **БД**. Формы можно использовать также для редактирования данных в таблице. Источником данных для формы может быть как таблица, так и запрос.

Отчеты являются удобным объектом для вывода необходимых данных на печать. Источником данных для отчета может служить таблица или запрос.

При загрузке на выполнение **Access** на экран сначала выводится диалоговое окно, а затем появляется главное окно **Access**, в котором располагаются переключатели, указывающие на возможные действия: **Новая база данных**, **Запуск мастера**, **Открыть базу данных**.

Порядок выполнения работы

1 Разработать базу данных согласно варианту, содержащую три таблицы (таблица 12.1).

2 Определить ключевые поля и построить связи между таблицами.

3 Занести в таблицы данные: не менее 10 записей в каждую таблицу.

4 Разработать три формы для ввода данных в каждую таблицу.

5 Разработать запросы: на выборку книг одного издательства; на выборку книг, изданных в указанный период.

6 Разработать отчеты: список книг по каждому издательству; список книг, изданных после 2015 г., в порядке возрастания цены.

7 Разработать стартовую кнопочную форму, которая должна запускаться при запуске файла базы данных.

Таблица 12.1 – Структура таблиц базы данных

Номер варианта	Таблица <i>БД</i>	Поле
1	Книги	Название книги, автор, год издания, издательство, страна издательства, город издательства, цена, количество книг
	Издательства	Код издательства, название, страна, город
	Города	Код города, название
2	Сведения о клиентах	Код клиента, имя, фамилия, город, код страны, должность, телефон
	Сведения о заказах	Код заказа, код клиента, номер заказа, адрес получателя, телефон, дата исполнения, сумма
	Страны	Код страны, название, тип (ближнее или дальнее зарубежье)
3	Операции	Код операции, дата, сумма по операции, код клиента
	Клиенты	Код клиента, фамилия, имя, отчество, место работы, номер паспорта, номер счета
	Вид операции	Код операции, название операции

Контрольные вопросы

- 1 В чем отличие базы данных от электронной таблицы?
- 2 Что такое реляционная модель организации данных?
- 3 Какие способы создания таблиц Вы знаете?
- 4 Какие основные типы полей используются в *СУБД Microsoft Access*?
- 5 Как построить схему данных *СУБД Access*?
- 6 Какое поле называется ключевым?
- 7 Что такое первичный ключ?
- 8 Поясните технологию создания запроса в режиме конструктора.
- 9 Как построить выражение в запросе?
- 10 В какой форме хранятся данные целого типа?
- 11 Для чего используются формы *Access*?
- 12 Для чего используются отчеты и как их построить?
- 13 Какие разделы отчета Вы знаете? Поясните их назначение.
- 14 В какой форме хранятся данные вещественного типа?
- 15 Поясните технологию создания запроса в режиме мастера.

Список литературы

- 1 **Трохова, Т. А.** Информатика для инженера : учебное пособие / Т. А. Трохова, Е. И. Гридина. – Минск : РИВШ, 2022. – 156 с. : ил.
- 2 Информационные технологии в маркетинге : учебник и практикум для вузов / Под общ. ред. С. В. Карповой. – Москва : Юрайт, 2020. – 367 с.
- 3 **Бессмертный, И. А.** Интеллектуальные системы : учебник и практикум для вузов / И. А. Бессмертный, А. Б. Нугуманова, А. В. Платонов. – Москва : Юрайт, 2021. – 243 с.
- 4 **Садовская, М. Н.** Информационные системы в экономике : учебное пособие / М. Н. Садовская, М. А. Акинфина, К. А. Забродская ; под общ. ред. М. Н. Садовской. – Минск : БГЭУ, 2018. – 316 с.
- 5 **Хомоненко, А. Д.** Базы данных : учебник / А. Д. Хомоненко, В. М. Цыганков, М. Г. Мальцев; под ред. А. Д. Хомоненко. – Санкт-Петербург: КОРОНАПринт, 2010. – 736 с.
- 6 **Власовец, А. М.** Введение в VBA : учебное пособие / А. М. Власовец. – Санкт-Петербург : СПбГУЭФ, 2010. – 135 с.
- 7 **Коваленко, В. В.** Проектирование информационных систем: учебное пособие / В. В. Коваленко. – 2-е изд., перераб. и доп. – Москва: ИНФРА-М, 2021. – 357 с.
- 8 **Гладилин, Г. А.** Практикум по эконометрике / А. В. Гладилин, А. Н. Герасимов, Е. И. Громов. – Ростов-на-Дону : Феникс, 2011. – 132 с.
- 9 **Гукин, Д.** Word 2010 для чайников / Д. Гукин. – Москва : Диалектика, 2010. – 252 с.
- 10 **Колеснев, В. И.** Экономико-математические методы и модели. Практикум : учебное пособие / В. И. Колеснев. – Минск : Минфин, 2010. – 296 с.
- 11 **Струченков, В. И.** Методы оптимизации в прикладных задачах / В. И. Струченков. – Москва : Солон-Пресс, 2009. – 320 с.