

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Локтионова Оксана Геннадьевна
Должность: проректор по учебной работе
Дата подписания: 31.12.2020 13:36:44
Уникальный программный ключ:
0b817ca911e6668abb13a5d426d39e5f1c11eabbf73e943df4a4861da56d0b

ПЛМОД

МИНОБРАЗОВАНИЯ И НАУКИ РОССИИ

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Юго-Западный государственный университет»
(ЮЗГУ)

Кафедра биомедицинской инженерии



УТВЕРЖДАЮ

Проректор по учебной работе

О.Г. Локтионова

2017 г.

ОСНОВЫ АВТОМАТИЗИРОВАННОЙ ОБРАБОТКИ РЕЗУЛЬТАТОВ МЕДИКО-БИОЛОГИЧЕСКИХ ИССЛЕДОВАНИЙ

Методические указания к самостоятельной работе по дисциплинам
«Стандартные программные средства в имитационном
моделировании биотехнических систем», «Прикладные пакеты
математической обработки данных»

Курск 2017

УДК 004.93:61

Составители: О.В. Шаталова, К.Д.А. Кассим.

Рецензент

Доктор технических наук, профессор Р.А. Томакова

Основы автоматизированной обработки результатов медико-биологических исследований: методические указания к самостоятельной работе / Юго-Зап. гос. ун-т; сост.: О.В. Шаталова, К.Д.А. Кассим. Курск, 2017. 51 с.

Предназначено для студентов направления подготовки 12.03.04 «Биотехнические системы и технологии» и по специальности 30.05.03 «Медицинская кибернетика» по дисциплинам «Стандартные программные средства в имитационном моделировании биотехнических систем», «Прикладные пакеты математической обработки данных». Может быть использована аспирантами, обучающимися по направлениям 05.11.13 – Системный анализ, управление и обработка информации и 05.11.17 – Приборы, системы и изделия медицинского назначения.

Текст печатается в авторской редакции

Подписано в печать 5.05.17. Формат 60×84 1/16. Бумага офсетная.
Усл. печ. л. 2,96. Уч.-изд. л. 2,68. Тираж 100 экз. Заказ 884.
Юго-Западный государственный университет.
305040, г. Курск, ул. 50 лет Октября, 94.

1 Мультимедийные технологии

В настоящее время наблюдается новый этап информатизации различных видов деятельности человека, вызванный развитием телекоммуникаций, Интернета, Web-технологий (гипертекст, гипермедиа и др.). Графика, анимация, фото, видео, звук, текст в интерактивном режиме работы создают интегрированную информационную среду, в которой пользователь обретает качественно новые возможности.

Мультимедиа окружает нас повсюду – через различные экраны современный человек соприкасается с виртуальными мирами, влияние которых становится все более существенным. Практически все сферы деятельности – наука, культура, образование, бизнес, сегодня немыслимы без продуктов мультимедиа. Презентация доклада, обучающая программа, анимационный рекламный ролик, виртуальное путешествие внутри или вокруг здания – вот небольшой перечень мультимедийных средств передачи информации. Все чаще появляются презентации, адаптированные для сети Интернет, что находит свое отражение в корпоративных сайтах ведущих компаний в различных отраслях.

Каждый современный человек, аналогично тому, как он овладел компьютерной грамотностью, должен овладеть основами мультимедиа технологий.

Мультимедиа (multi – много, media – среда, т.е. много сред) – это «одновременное использование различных форм представления информации и ее обработки в едином объекте-контейнере».

Мультимедиа является современной компьютерной информационной технологией, позволяющей объединить в единой информационной среде различные типы данных, таких как текст, графика, фотография, анимация, видео, звук.

Аппаратные средства

Все оборудование, отвечающее за звук, объединяется в звуковые карты, а за видео – в видеокарты.

Аппаратные средства мультимедиа:

- Средства звукозаписи (звуковые платы, микрофоны);
- Средства звуковоспроизведения (усилитель, колонки, акустические системы, наушники и гарнитуры);
- Манипуляторы (компьютерные мыши, джойстики, миди-клавиатуры);

- Средства «виртуальной реальности» (перчатки, очки, шлемы виртуальной реальности, используемые в играх);
- Носители информации (CD, DVD и HDD);
- Средства передачи (мини видеокамеры, цифровые фотоаппараты);
- Средства записи (приводы CD / DVD-ROM , CDRW / DVD+RW, TV- и FM-тюнеры);
- Средства обработки изображения (платы видеомонтажа, клавиатуры, графические акселераторы).
- Компьютер, телевизор, средства для получения и удобного восприятия информации и др.

Программные средства мультимедиа технологии

Программные средства мультимедиа складываются из трех компонентов:

1. Системные программные средства.
2. Инструментальные программные средства.
3. Прикладные программные средства.

1. Системные программные средства

Системные программные средства – это набор программ, входящих в состав операционной системы компьютера и осуществляющих управление устройствами мультимедиа, причем это управление на двух уровнях – физическое управление вводом-выводом информации на низком уровне с помощью машинных команд и управление пользователем характеристиками устройств с помощью графического интерфейса, изображающего пульт управления устройством, например регулировки громкости звука, тембра, стереобаланса и т. д.

Как правило, программы физического управления устройствами называют драйверами устройств.

2. Инструментальные программные средства

Инструментальные программные средства – программы позволяющие модифицировать мультимедийные файлы и создавать мультимедийные приложения.

Инструментальные программные средства – это пакеты программ для создания мультимедийных приложений:

- редакторы неподвижных графических изображений,
- средства создания анимированных GIF-файлов,
- средства аудио- и видеомонтажа,

- средства создания презентаций,
- средства распознавания текстов, введенных со сканера,
- средства создания обучающих программ,
- системы распознавания голоса и преобразования звуковых файлов в текстовые,
- системы создания приложений виртуальной реальности и другие.

Инструментальные средства существенно расширяют возможности управления мультимедийными устройствами по сравнению с теми, которые предоставляют системные средства, но это всегда платные продукты и некоторые из них стоят очень дорого, например профессиональные системы видеомонтажа.

3. Прикладные программные средства

Прикладные программные средства – это готовые и, как правило, продаваемые программные системы на CD или DVD дисках – фильмы, учебники, энциклопедии, игры, книги, виртуальные музеи, путеводители, рекламные материалы и т. д.

Задания

Задание 1. *Запись аудио компакт-диска в программе Nero*

Примечание. Для первых тренировочных записей можно использовать CD-RW, но если диск записывается для воспроизведения на бытовых плеерах, тогда лучше использовать CD-R.

Вначале нужно вставить в привод чистый CD-R диск.

После этого запустить программу Nero (Nero) и создать в ней новый проект CD - Audio-CD (рисунок 1.1):

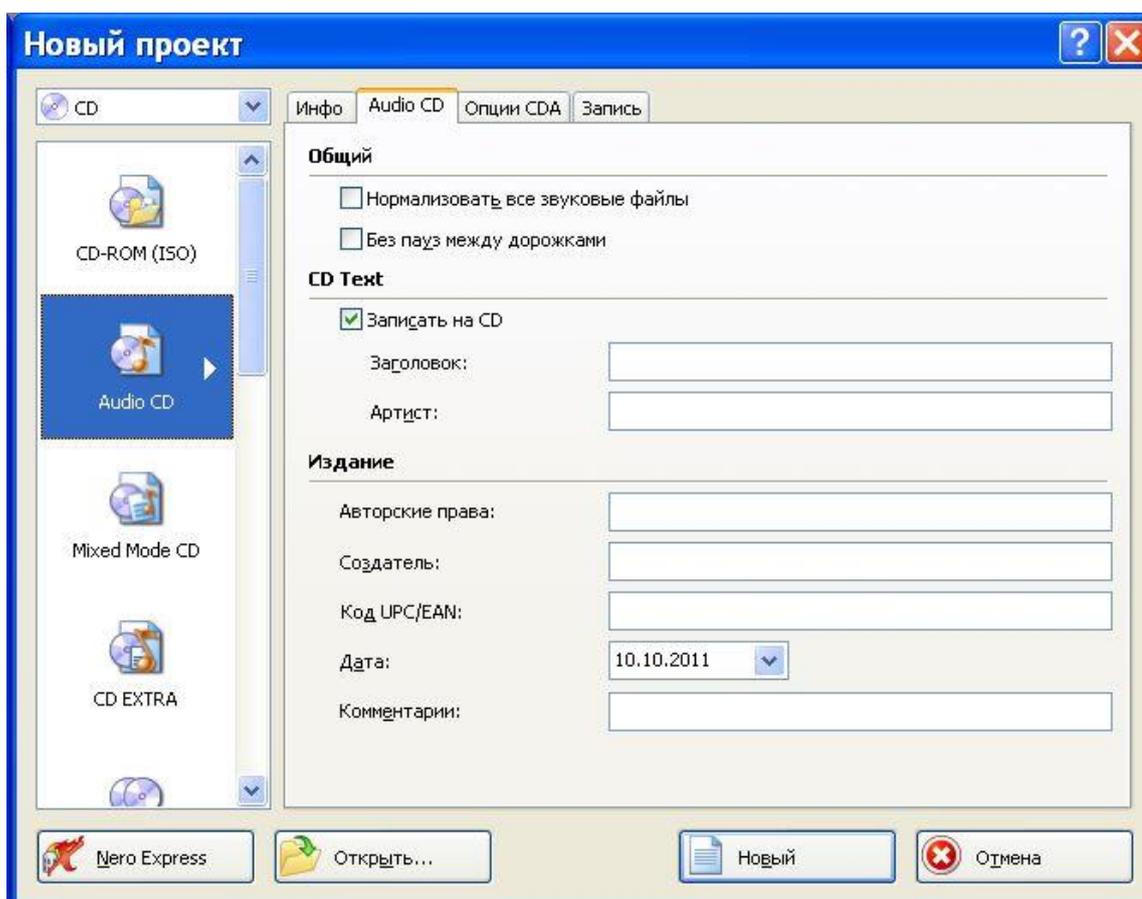


Рисунок 1.1 – Интерфейс программы Nero (Nero)

После этого откроется окно с двумя панелями (рисунок 1.2). В правой панели будут перечислены локальные диски, нужно на этих дисках найти файлы аудио типов (WAV, MP3, WMA и т.п.) и затем мышью перетащить найденные файлы в левую панель. После перетаскивания файлов в левую панель Nero укажет для каждого такого файла его длительность воспроизведения, а в нижней строке окна укажет общую длительность аудио диска:

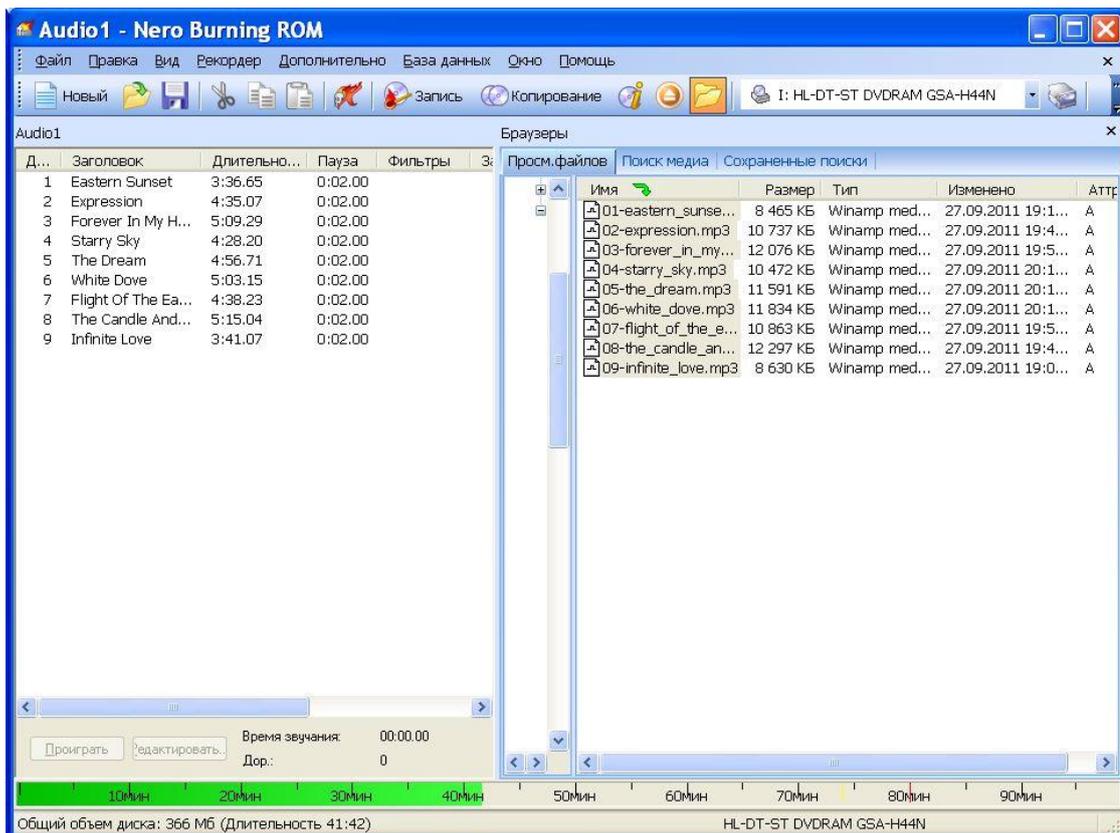


Рисунок 1.2 – Интерфейс программы Неро (Nero) после выбора записи проекта

После того как все файлы собраны, нужно нажать кнопку "Запись" (рисунок 1.3). Или выполнить команду меню "Рекордер - записать проект". В открывшемся окне проверить правильность установки опций записи и нажать кнопку "Прожиг":

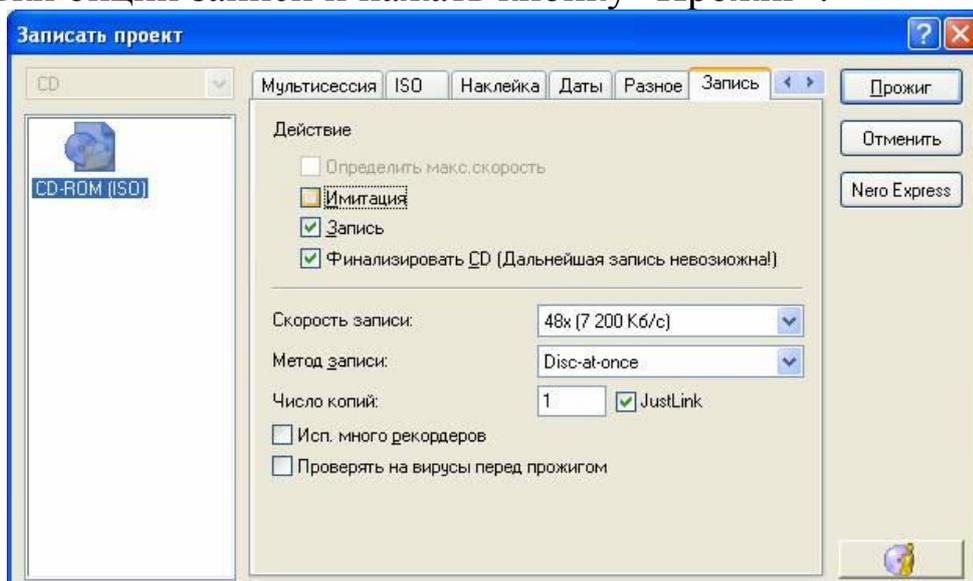


Рисунок 1.3 – Интерфейс записи проекта в программе Неро (Nero)

Важно проверить, чтобы была выбрана опция «Финализировать CD»!

После этого дождаться завершения записи диска. Программа Nero будет отображать ход записи, а по окончании записи выдаст сообщение об успешном завершении.

Примечание:

Хотя возможна запись аудио-CD из файлов форматов MP3 или WMA нужно понимать, что в этом случае несмотря на то что формат диска будет Audio-CD, качество звука будет соответствовать качеству звука исходных MP3 или WMA файлов. Так что если вы хотите записать действительно качественный аудио-компакт диск его нужно записывать из несжатых звуковых файлов формата WAV.

Задание 2. *Запись DVD дисков с фильмами в Windows*

Примечание. Если диск планируется использовать на бытовых плеерах, тогда нужно предварительно изменить имена файлов следующим образом - короткие, без пробелов и латинскими буквами. Не все бытовые плееры могут распознавать имена файлов, в которых есть пробелы или не латинские символы. Также для лучшей совместимости с бытовыми плеерами лучше использовать R, а не RW диски.

Вставляете в дисковод чистый диск. Затем находите нужный видео файл (или файлы) в Проводнике Windows, выделяете, нажимаете правую кнопку мыши и затем выбираете пункты меню "Отправить" и "DVD RW дисковод" (рисунок 1.4):

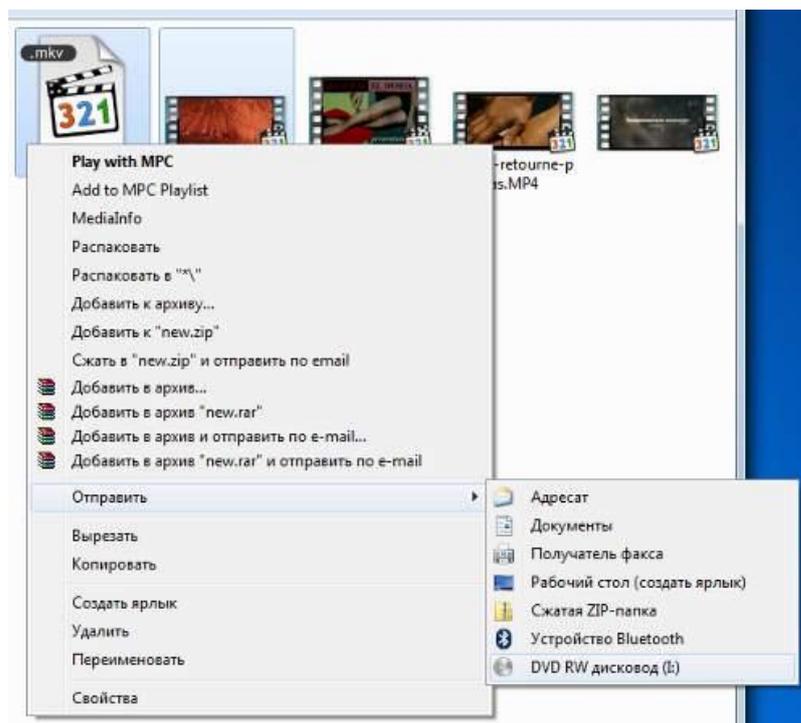


Рисунок 1.4 – Скриншот запись нужного файла на диск

После этого откроется окно, в котором будет отображаться процесс кэширования файлов перед записью (рисунок 1.5):

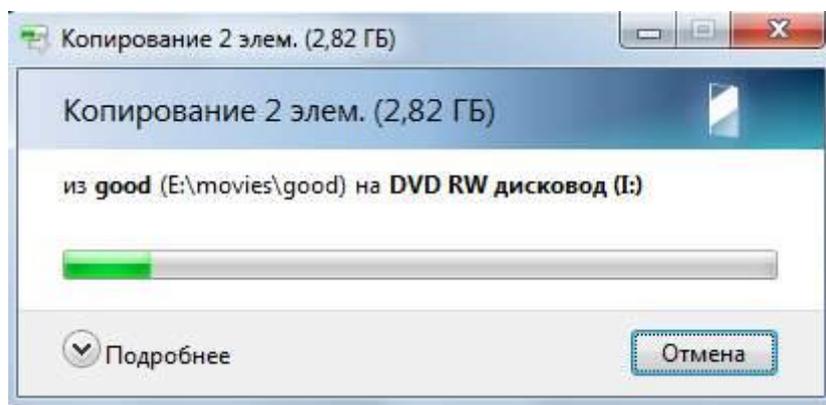


Рисунок 1.5 – Процесс кэширования файлов перед записью

После этого откроется еще одно окно Проводника, в котором будет список файлов для записи. В этом окне нужно нажать правую кнопку мыши и выбрать пункт меню "Записать на диск" (рисунок 1.6):

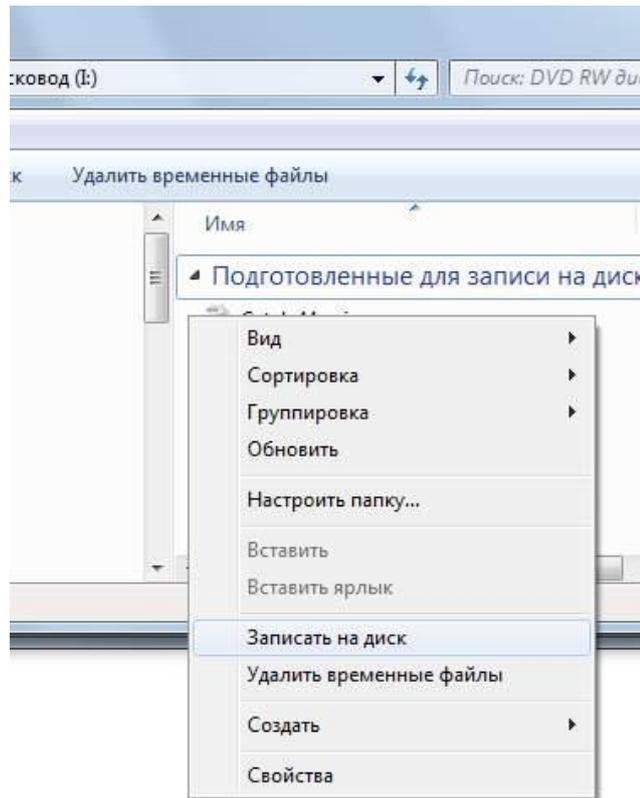


Рисунок 1.6 – Запись файлов на диск с помощью окна Проводника

После этого откроется окно, в котором можно указать метку для диска и нужно нажать кнопку "Далее" (рисунок 1.7):

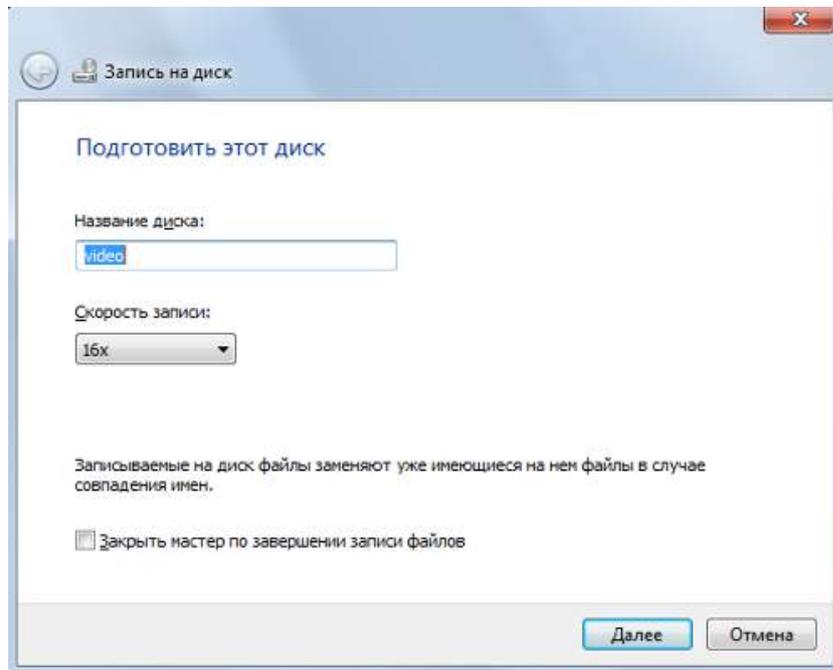


Рисунок 1.7 – Скриншот окна, в котором указывается метка для диска

И дождаться завершения записи (рисунок 1.8):

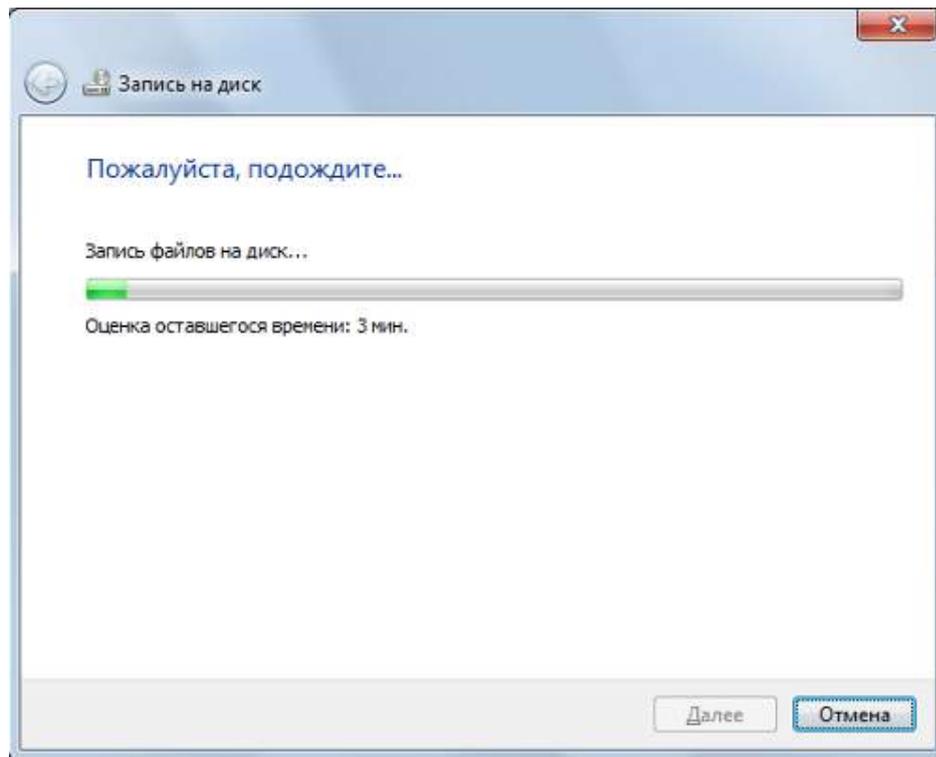


Рисунок 1.8 – Скриншот окна завершения записи диска

Задание 3. *Запись DVD дисков в Nero Burning ROM*

(В любой другой программе последовательность действий будет такая же.)

1. Вставить диск (DVD+/-R или DVD+/-RW) в привод
2. Запустить программу Nero (рисунок 1.9):

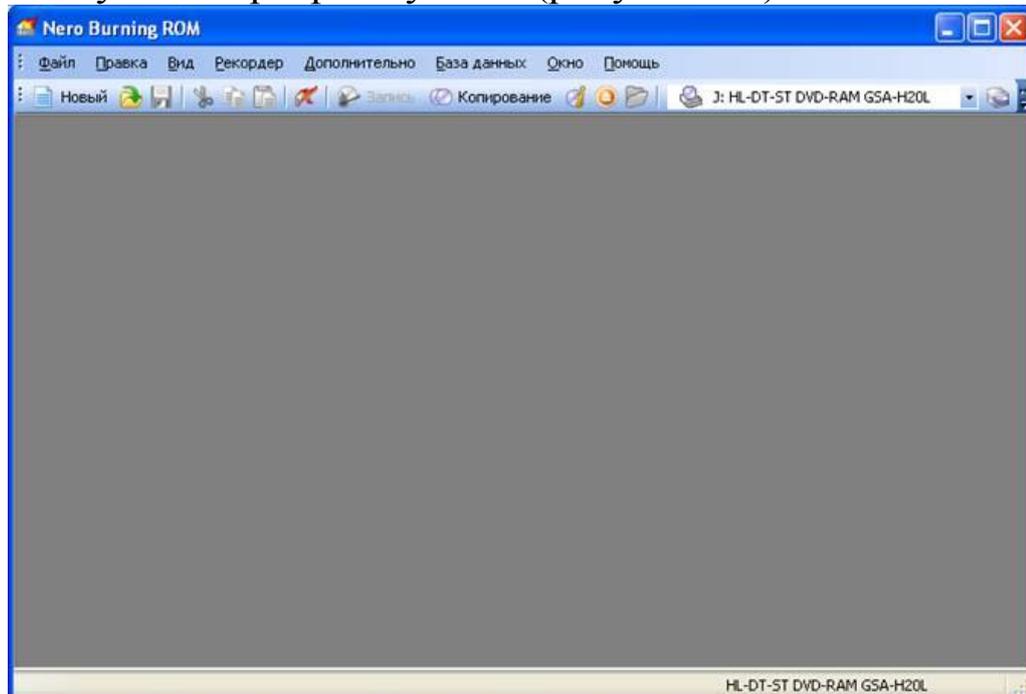


Рисунок 1.9 – Интерфейс окна программы Nero

Если диск DVD+/-RW его нужно предварительно стереть (рисунок 1.10):

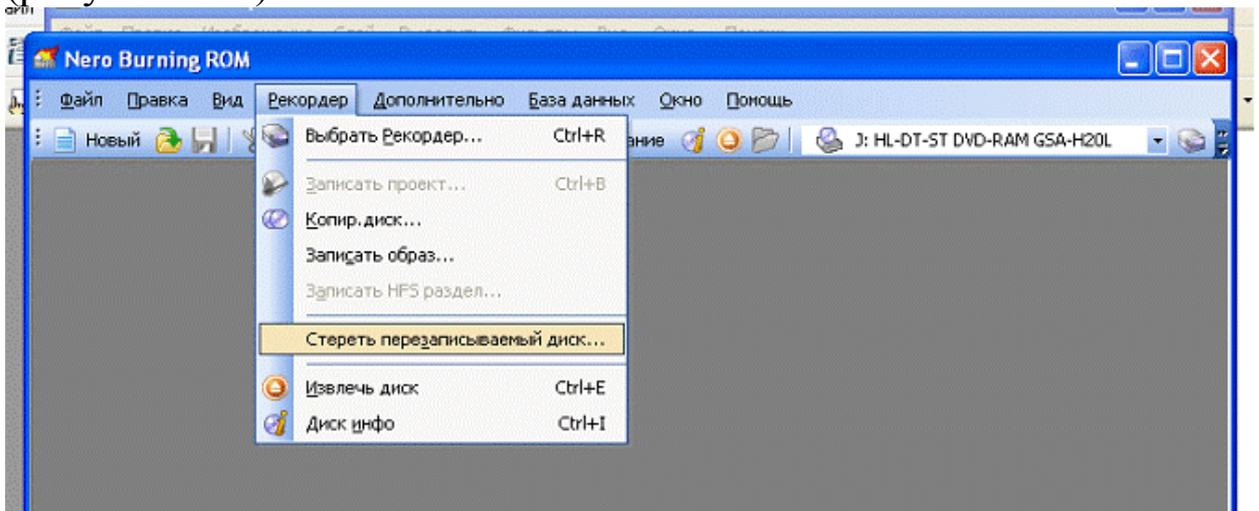


Рисунок 1.10 – Диалоговое окно предварительного стирания данных с диска

Затем появится окно для стирания перезаписываемого диска (рисунок 1.11):

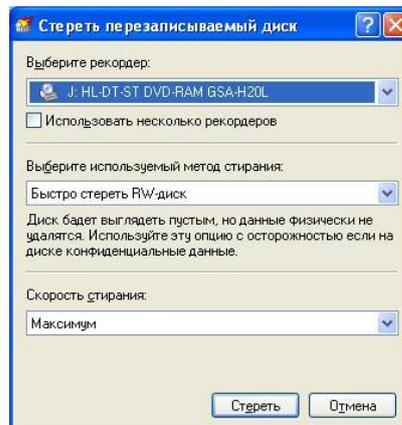


Рисунок 1.11 – Диалоговое окно стирания перезаписываемого диска

Тут нажать кнопку "Стереть".

3. Создать новый проект

Нажать кнопку "Новый" или меню "Файл – Создать" (рисунок 1.12).

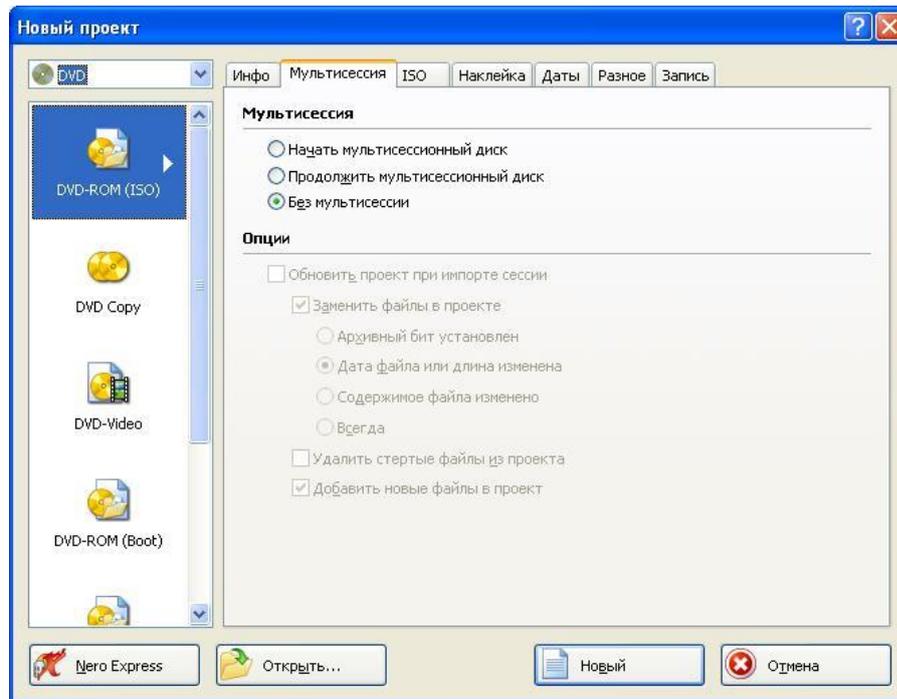


Рисунок 1.12 – Диалоговое окно для записи нового проекта

В этом диалоговом окне нужно выбрать тип диска, который будет записан:

DVD-ROM(ISO) – обычный компьютерный диск с файлами (такой формат в котором создаются диски с программами).

DVD-Video – цифровой видео диск, который может быть использован на бытовых DVD плеерах.

И нажать кнопку «Новый».

4. Собрать файлы для записи

Это делается простым перетаскиванием нужных файлов и папок из правой панели в левую (рисунок 1.13):

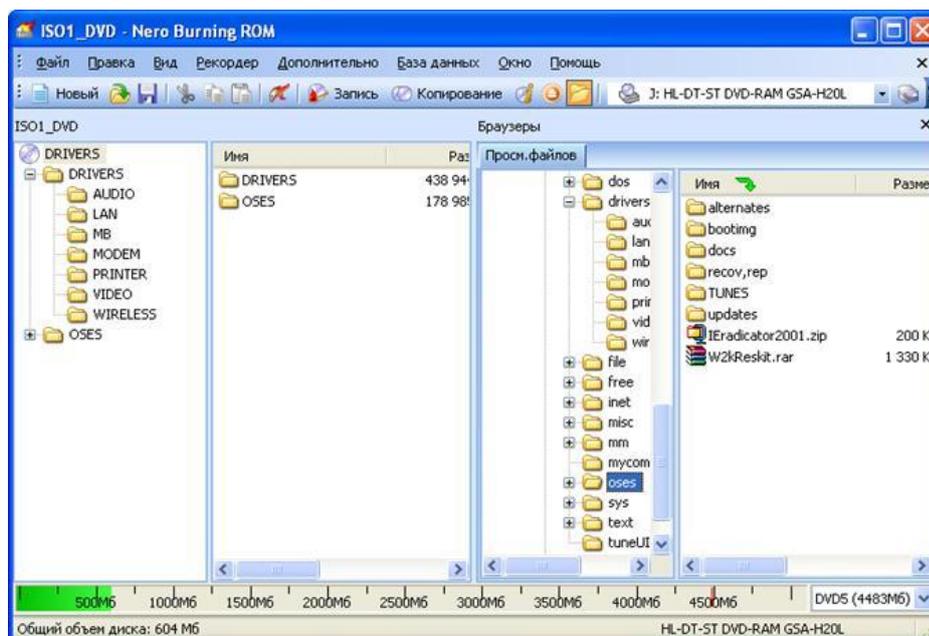


Рисунок 1.13 – Диалоговое окно для записи диска

Можно перетаскивать как отдельные файлы, так и целые папки. Можно перетаскивать файлы и папки из разных исходных папок. Внизу окна расположена шкала, которая показывает, какой размер, будут занимать на диске уже выбранные файлы. В конце шкалы (справа) находится переключатель размера диска (DVD5 или DVD9). В общем случае его нужно установить в DVD5 (это обычные, однослойные DVD диски). Но если необходимо записать двухслойный DVD диск, тогда нужно переключить на DVD9.

5. Записать файлы на диск

Нажать кнопку "Запись" или меню "Рекордер – Записать проект" (рисунок 1.14):

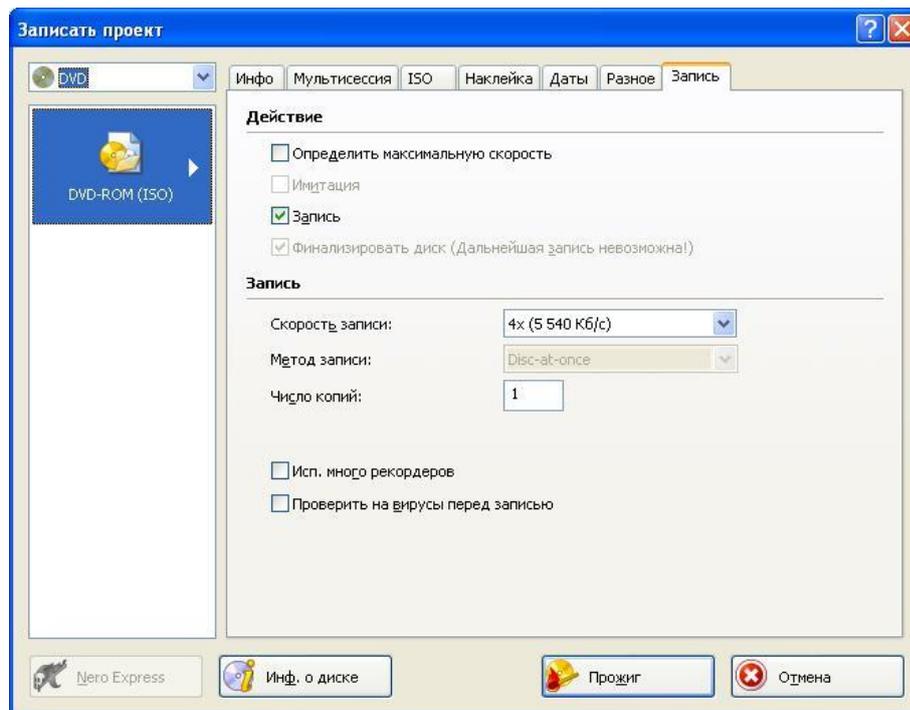


Рисунок 1.14 – Диалоговое окно для записи диска

Тут важно проверить, чтобы были отмечены следующие опции:

Запись - Финализировать диск (если эта опция не отмечена диск не будет читаться на других устройствах)

После этого нужно нажать кнопку «Прожиг» и дождаться конца записи на диск. Об окончании записи будет выдано уведомление. В зависимости от общего размера файлов, скоростной характеристики диска, запись может длиться от нескольких минут до получаса.

Во время записи на диск лучше ничего больше на компьютере не делать. Нельзя выполнять в это время задачи, которые сильно загружают винчестер или процессор это может привести к порче записываемого диска.

Выполните задание:

Ответьте письменно на вопросы

1. Понятие мультимедиа технологии.
2. Каковы аппаратные средства использования мультимедиа технологии?
3. Какими могут быть области применения мультимедиа приложений?

4. Что было идейной предпосылкой возникновения технологии мультимедиа?

5. Назовите программное обеспечение технологии мультимедиа.

Заполните таблицу 1.1.

Таблица 1.1 – Характеристики дисков

Диск	Характеристика		
	Число записи	Объем	Примечание
CD-R			
CD-RW			
DVD-R			
DVD-R DL			
DVD-RW			
DVD+R			
DVD+R DL			
DVD+RW			
DVD-RAM			
BD-R			
BD-RE			

Литература:

1. Запись аудио компакт-диска (Audio-CD) в программе Nero [электронный ресурс]/ режим доступа URL: <http://www.ru.d-ws.biz/articles/burnAudioNero.shtml> (Дата обращения 26.07.2016).

2. Записываемые DVD диски [электронный ресурс]/ Режим доступа URL: <http://www.ru.d-ws.biz/articles/dvdr.shtml> (Дата обращения 4.08.2016)

3. Как записать DVD, CD диск [электронный ресурс]/ режим доступа URL: <http://ru.d-ws.biz/articles/burn-cd-dvd-bd.shtml> (Дата обращения 6.08.2016)

4. Мультимедиа-технологии как основа междисциплинарной интеграции [электронный ресурс]/ режим доступа URL: <http://www.inurb.ru/node/256> (Дата обращения 7.08.2016)

2 Первичная статистическая обработка экспериментальных данных на базе программы «Stadia 6.0»

Порядок выполнения работы

1. Изучите материалы лекции.
2. Ввести исходные данные в таблицу данных. Столбцы отвечают переменным (случайным величинам), а строки – значениям переменных. В задании даны значения одной случайной величины, полученные по выборке объемом $n=100$. Числа записывают в научной нотации, когда очень большие и очень малые значения представлены с десятичным множителем, показатель которого следует за символом E.

Например, $18000=18 \cdot 10^3=18E3$; $0,003=3 \cdot 10^{-3}=3E-3$.

3. В блоке «Статистические методы» выбрать процедуру «Описательная статистика», которая вычисляет следующие выборочные числовые характеристики:

- 1) диапазон значений (размах варьирования) $X_{\min} - X_{\max}$;
- 2) выборочное среднее:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i;$$

- 3) ошибка вычисления среднего:

$$\Delta \bar{x} = \frac{S}{\sqrt{n}};$$

- 4) выборочная дисперсия:

$$S^2 = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n - 1};$$

- 5) стандартное отклонение:

$$S = \sqrt{S^2};$$

- 6) медиана – значение случайной величины, которое делит вариационный ряд пополам;
- 7) доверительный интервал среднего при уровне надежности $\gamma = 0,95$;
- 8) доверительный интервал дисперсии при $\gamma = 0,95$;
- 9) ошибка стандартного отклонения:

$$\Delta S = S \frac{\sqrt{2}}{\sqrt{4(n-1)+1}};$$

- 10) квартили (делят вариационный ряд на 4 части);
- 11) коэффициент асимметрии:

$$A = \frac{\sum_{i=1}^n (x_i - \bar{x})^3}{S^3};$$

- 12) коэффициент эксцесса:

$$\varepsilon = \frac{\sum_{i=1}^n (x_i - \bar{x})^4}{S^4};$$

4. В блоке «Статистические методы» выбрать процедуру «Гистограмма / Нормальность».

3.1. Для построения гистограммы в специальном бланке автоматически указывается число интервалов и размах варьирования: x_{\min} (левая граница) и x_{\max} (правая граница). Для каждого интервала гистограммы на экран выводятся следующие значения:

- 1) левая граница интервала в исходных единицах и в единицах стандартного отклонения $(x-m)/\sigma$;
- 2) частота - число значений, попавших в интервал (в натуральном и процентном выражении);

3) накопленное число выборочных значений до текущего интервала включительно (так называемые накопленные частоты) в натуральном и процентном выражении; если накопленные частоты, выраженные в процентах, разделить на 100%, то получают значения эмпирической функции распределения $F^*(x_i)$.

3.2. Проверка выборочного распределения на нормальность может быть проведена несколькими способами, которые дополняют друг друга:

1) визуальный метод предназначен для предварительной субъективной оценки и может быть осуществлен по виду гистограммы;

2) проверка с помощью критериев согласия Пирсона χ^2 , омега-квадрат ω^2 и Колмагорова D ;

3) проверка с помощью коэффициентов асимметрии и эксцесса.

В процедуре «Гистограмма/Нормальность» проверка нулевой гипотезы об отсутствии различий между выборочным и нормальным распределением проводится с помощью критериев согласия Пирсона χ^2 , омега-квадрат ω^2 и Колмогорова D . Для каждого критерия подсчитывается уровень значимости нулевой гипотезы P . В качестве критического уровня значимости принято значение $\alpha=0,05$. Если $P>0,05$, то нулевая гипотеза принимается.

В процедуре «Описательная статистика» при выдаче значений выборочных коэффициентов асимметрии и эксцесса проводится проверка нулевой гипотезы об отсутствии различий выборочного распределения от нормального по каждому из коэффициентов. Для нормального распределения теоретические значения коэффициента асимметрии $A=0$, коэффициента эксцесса $\varepsilon=3$, эксцесса $E=\varepsilon-3=0$. Для выборочных A и ε приводятся уровни значимости нулевой гипотезы P . Если P больше критического значения $\alpha=0,05$, то нулевая гипотеза может быть принята.

3.3. Графическая выдача результатов содержит изображение гистограммы с наложенной кривой нормального распределения.

4. Если выборочное распределение не является нормальным, то можно попытаться подобрать другое теоретическое распределение с помощью процедуры «Согласие распределений». Данная процедура предназначена для проверки гипотезы об отсутствии различий между эмпирическим и теоретическим распределением. Для этого в меню распределений

необходимо выбрать предполагаемый тип теоретического распределения.

На экран выдаются:

1) значение критерия согласия Колмогорова D с уровнем значимости P нулевой гипотезы об отсутствии различий между выборочным и теоретическим распределениями;

2) значение критерия омега-квадрат ω^2 с уровнем значимости P нулевой гипотезы.

ПРИМЕР. Провести первичную статистическую обработку экспериментальных данных по измерению твердости по Бринелю образцов из легированной стали. Результаты измерений приведены в таблице 2.1.

Таблица 2.1 – Твердость образцов из легированной стали

№ п.п.	НВ						
1	47	26	63	51	54	76	60
2	55	27	54	52	58	77	60
3	63	28	56	53	54	78	58
4	62	29	50	54	58	79	61
5	48	30	59	55	55	80	60
6	52	31	58	56	57	81	55
7	62	32	52	57	59	82	56
8	55	33	54	58	57	83	52
9	56	34	53	59	56	84	55
10	57	35	52	60	58	85	55
11	58	36	52	61	55	86	53
12	59	37	54	62	54	87	57
13	58	38	54	63	55	88	60
14	58	39	48	64	55	89	61
15	60	40	55	65	59	90	54
16	58	41	60	66	59	91	57
17	54	42	60	67	54	92	60
18	59	43	56	68	57	93	59
19	59	44	60	69	60	94	60
20	58	45	63	70	57	95	60
21	57	46	60	71	59	96	60
22	57	47	60	72	58	97	54
23	53	48	58	73	57	98	58

24	57	49	58	74	58	99	58
25	55	50	55	75	54	100	60

Результаты статистического анализа

Описательная статистика. Файл: Твердость образцов.

Переменная

Размер ← Диапазон → Среднее ← Ошибка ← Дисперс → Ст.откл

x1 100 47 63 56,8 0,322 10,4 3,22

Переменная

Медиана

← Квартили → ДовИнтСр ← ДовИнтДисп → Ош.СтОткл

x1 57 55 59 0,631 9,28 1,89E4 0,858

Переменная Асимметр. Значим Экссесс Значим

x1 -0,58 0,0073 3,41 0,15

Гистограмма и тест нормальности

Х-лев.	Х-станд	Частота	%	Накопл.	%
47	-3,05	3	3	3	3
49	-2,42	1	1	4	4
51	-1,8	8	8	12	12
53	-1,18	24	24	36	36
55	-0,562	16	16	52	52
57	0,059	25	25	77	77
59	0,68	18	18	95	95
61	1,3	5	5	100	100
63	1,92				

Колмогоров=0,124, Значимость=0,00098, степ.своб = 100

Гипотеза 1: <Распределение отличается от нормального>

Омега-квадрат=0,203, Значимость=0,0044, степ.своб = 100

Гипотеза 1: <Распределение отличается от нормального>

Хи-квадрат=14,7, Значимость=0,0227, степ.своб = 6

Гипотеза 1: <Распределение отличается от нормального>

Согласие распределений

Распределение экспоненциальное: 56,8, 3,22 (таблица 2.2)

Колмогоров=0,563, Значимость=1, степ.своб = 99

Гипотеза 0: <Распределение не отличается от теоретического>

Омега-квадрат=8,86, Значимость=0, степ.своб = 99

Гипотеза 1: <Распределение отличается от теоретического>

Таблица 2.2 - Статистическое распределение частот n_i , относительных частот n_i/n и накопленных частот $n_{\text{нак}}$.

Δx_i	47-49	49-51	51-53	53-55	55-57	57-59	59-61	61-63
n_i	3	1	8	24	16	25	18	5
n_i/n	0,03	0,01	0,08	0,24	0,16	0,25	0,18	0,05
$n_{\text{нак}}$	0,03	0,04	0,12	0,36	0,52	0,77	0,95	1,00

Пример построения гистограммы частот выборочного распределения представлен на рисунке 2.1.

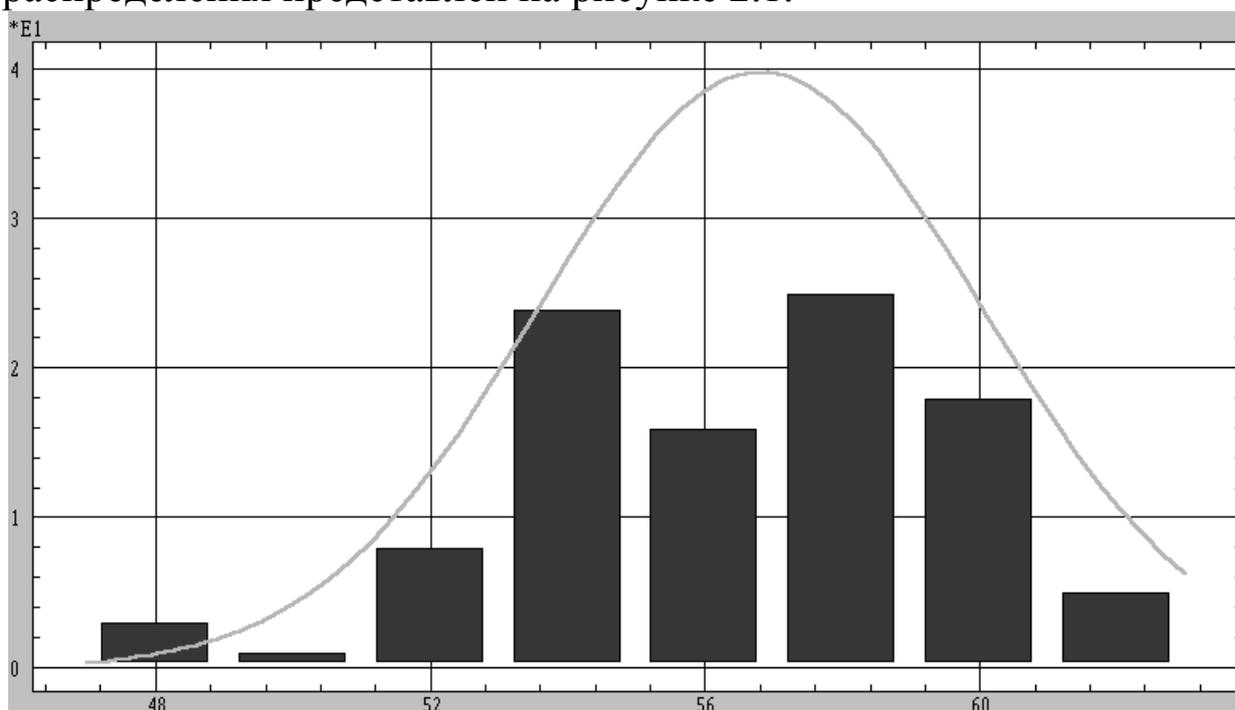


Рисунок 2.1 - Гистограмма частот выборочного распределения

Выводы

- 1) Выборочное среднее $\bar{x}=56,8$;
- 2) Выборочная дисперсия $S^2=10,4$;

- 3) Стандартное отклонение $S=3,22$;
- 4) Доверительный интервал для среднего:
 $56,8 \pm 0,631 = [57,431; 56,169]$;
- 5) Доверительный интервал для дисперсии: $[9,28; 18,9]$;
- 6) Медиана $Me=57$;
- 7) Нижний квартиль $x_{0,25}=55$; верхний квартиль $x_{0,75}=59$;
- 8) Коэффициент асимметрии $A=-0,58$;
- 9) Коэффициент эксцесса $\varepsilon=3,41$;
- 10) Проверка на нормальность:
 - по критериям согласия Пирсона, Колмогорова и омега-квадрат нулевую гипотезу отвергаем, так как уровни значимости для каждого критерия $P \ll \alpha = 0,05$: $P=0,0227$ по критерию Пирсона, $P=0,00098$ по критерию Колмогорова, $P=0,0044$ по критерию омега-квадрат; следовательно, распределение отличается от нормального;
 - по коэффициенту асимметрии $P=0,0073 < \alpha = 0,05$; следовательно, нулевую гипотезу отвергаем, распределение отличается от нормального;
 - по коэффициенту эксцесса $P=0,15 > \alpha = 0,05$; следовательно, нет оснований отвергнуть нулевую гипотезу.
- 11) Проверка выборочного распределения с теоретическим в процедуре «Согласие распределений» показала, что по критерию Колмогорова выборочное распределение не отличается от экспоненциального, в то время как проверка по критерию омега-квадрат не выявила согласия с этим распределением.

Контрольные вопросы

1. Какие существуют формы записи статистического материала?
2. Что такое полигон и гистограмма?
3. Назовите выборочные числовые характеристики, дайте их определения.
4. Дайте определения точечной и интервальной оценок.
5. Какие требования предъявляются к точечным оценкам?
6. Назовите наилучшие оценки математического ожидания и дисперсии генеральной совокупности.
7. Что называется доверительным интервалом и доверительной вероятностью?

8. Как строится доверительный интервал для математического ожидания?
9. Как строятся доверительные интервалы для дисперсии и среднего квадратического отклонения?
10. Дайте определение эмпирической функции распределения. Как построить ее график по данным выборки?
11. Какое распределение называется нормальным?
12. Как построить нормальную кривую по опытным данным?
13. Что называется статистической гипотезой?
14. Назовите общие принципы проверки гипотез.
15. Какими способами можно проводить «тест на нормальность»?

Литература

1. Вентцель, Е.С. Теория вероятностей и ее инженерные приложения/ Е.С. Вентцель, Л.А. Овчаров. -М.: Наука, 1988. - 480 с.
2. Вентцель, Е.С. Теория вероятностей. Задачи и упражнения/ Е.С. Вентцель, Л.А. Овчаров. – М.: Наука, 1969. - 368 с.
3. Гурский, Е.И. Теория вероятностей с элементами математической статистики/ Е.И. Гурский. –М.: Высшая школа, 1971. - 328 с.
4. Гмурман, В.С. Теория вероятностей и математическая статистика/ В.С. Гмурман. -М.: Высшая школа, 1972. - 368 с.
5. Гмурман, В.Е. Руководство к решению задач по теории вероятностей и математической статистике/ В.Е. Гмурман. – М.: Высшая школа, 1975. - 333 с.
6. Колде, Я.К. Практикум по теории вероятностей и математической статистике/ Я.К. Колде. – М.: Высшая школа, 1991. - 157 с.

3 Введение в Simulink

Краткие сведения о пакете

Система структурного моделирования Simulink предназначена для компьютерной реализации математических моделей динамических систем и устройств, представленных функциональной блок-схемой или системой уравнений. При этом возможны различные варианты моделирования: во временной области, в частотной области, с событийным управлением и т.д.

Для построения функциональной блок-схемы моделируемых устройств Simulink имеет обширную библиотеку блочных компонентов и удобный редактор блок-схем. Он основан на графическом интерфейсе пользователя и по существу является типичным средством визуально-ориентированного программирования. Используя библиотеку компонентов решающих блоков, пользователь с помощью мыши переносит нужные блоки из библиотеки в рабочее окно пакета Simulink и соединяет линиями связи входы и выходы блоков. Таким образом, создается блок-схема системы или устройства, то есть компьютерная модель.

Порядок работы с пакетом Simulink следующий:

- 1 На рабочем столе открываем пиктограмму MATLAB.
- 2 В открывшемся командном окне на панели инструментов нажимаем кнопку Simulink.
- 3 Открывается Simulink. Кроме рабочего окна с общим именем Untitled (“Безымянный”) открывается окно библиотеки Simulink с разделами:

- *Sources* – источники;
- *Sinks* – приемники;
- *Discrete* – дискретные;
- *Linear* – линейные;
- *Nonlinear* – нелинейные
- *Connections* – связи;
- и т.д.

Порядок выполнения работы:

В качестве примера рассмотрим демонстративный пример. Для этого:

1. Нажимаем кнопку *Demos* в меню *Help*.
2. В левом окошке выбираем пункт *Simulink*.
3. Двойным щелчком выбираем пункт *Simple models* (в версии 6.1 и выше пункт *General*).
4. В правом окне выбираем модель *Spring-mass system simulation* (рисунок 3.1).
5. Запускаем ее командой *Run* или двойным щелчком.

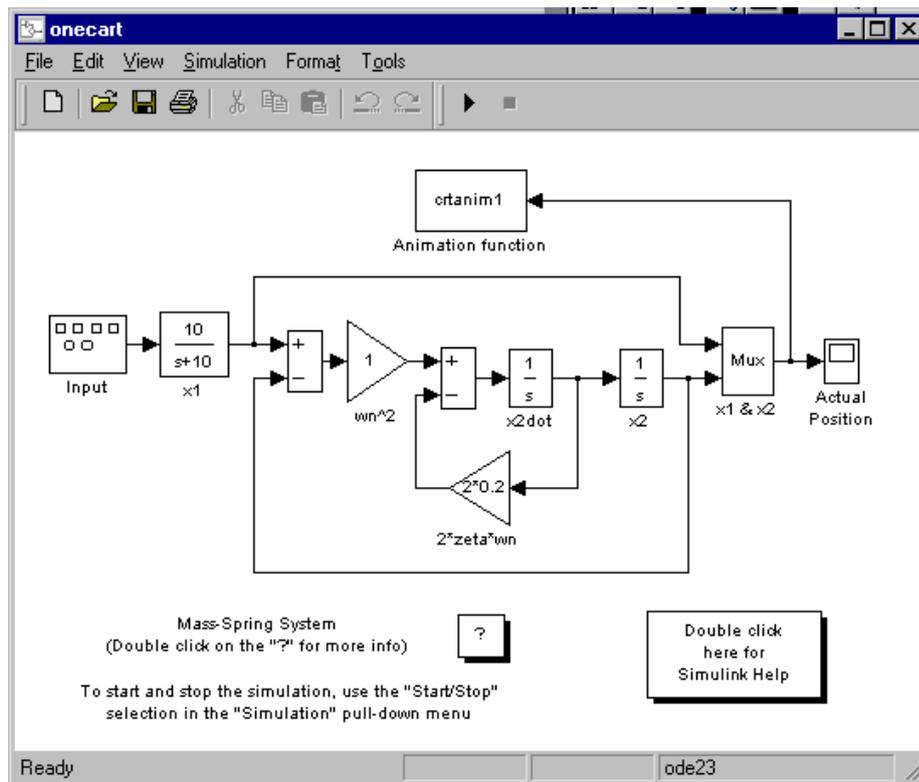


Рисунок 3.1 - Окно Демонстрации “Spring–mass system simulation”

Данная модель реализует колебательную механическую систему с одной степенью свободы. Физическим аналогом модели является переменное по направлению движение, закрепленной пружиной к “прыгающей” заделке (слева) массы по гладкой поверхности с трением. Внешняя сила приложена к заделке. Модель реализует дифференциальное уравнение, описывающее движение кубика. В MATLAB в передаточных функциях вместо p – оператора Лапласа пишется s .

Рассмотрим блоки (слева направо):

Input – генератор, **x1** – звено первого порядка, сумматор, усилитель, $\frac{1}{s}$ – интегратор, **Mux** - смеситель (позволяет вывести более одного сигнала на **Scope**), **Scope (Actual position)** – осциллограф, **Animation function** - блок анимации.

Запуск моделирования осуществляется кнопкой .

На экране – анимационная картинка, представленная на рисунке 3.2:

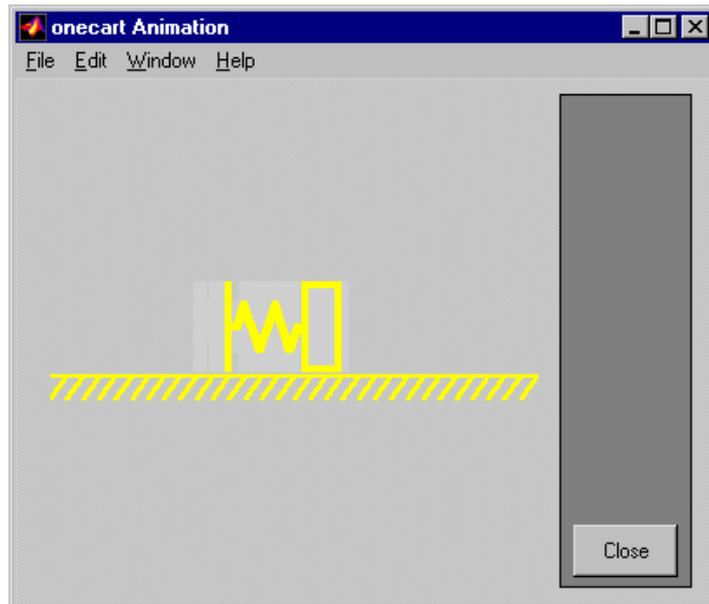


Рисунок 3.2 - Движение груза на пружине

Двойной щелчок на **Scope** после запуска моделирования открывает окно **Scope** (рисунок 3.3), в котором отображаются графики, характеризующие колебания правой массы во времени и знакопеременную внешнюю силу.

Остановить моделирование с помощью значка  и закрыть окна демонстрации.

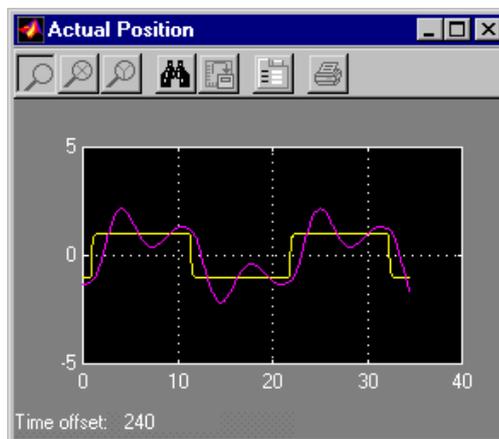


Рисунок 3.3 - Графики, характеризующие работу системы

Ход работы:

1. На экране – открыть окно библиотеки и рабочее окно Simulink.

2. В окне библиотеки двойным щелчком раскрывается каждый из разделов. В разделе библиотеки *Sources* (Источники) выбираем генератор *Signal Generator*, перетаскиваем его в рабочее окно и закрываем окно *Sources*.

3. В разделе *Sinks* (Приемники) выбираем осциллограф *Scope*, перетаскиваем в рабочее окно и закрываем окно *Sinks*.

4. В разделе *Linear* (Линейные блоки) или *Math* (для версии 6.1 и выше) выбираем блок усилитель – *Gain* с регулируемым коэффициентом усиления (КУ) и сумматор – *Sum*, а также интегратор – *Integretor* (в версии 6.1 и выше данный блок находится в разделе *Continuous*). Все блоки последовательно перетаскиваем в рабочее окно и закрываем окно *Linear* (*Math*).

5. В разделе *Connections* (Связи) или *Signals & Systems* (для версии 6.1 и выше) выбираем смеситель – *Mux* и перетаскиваем в рабочее окно.

6. В рабочем окне приступаем к соединению блоков. Схема модели должна иметь вид представленный на рисунке 3.4.

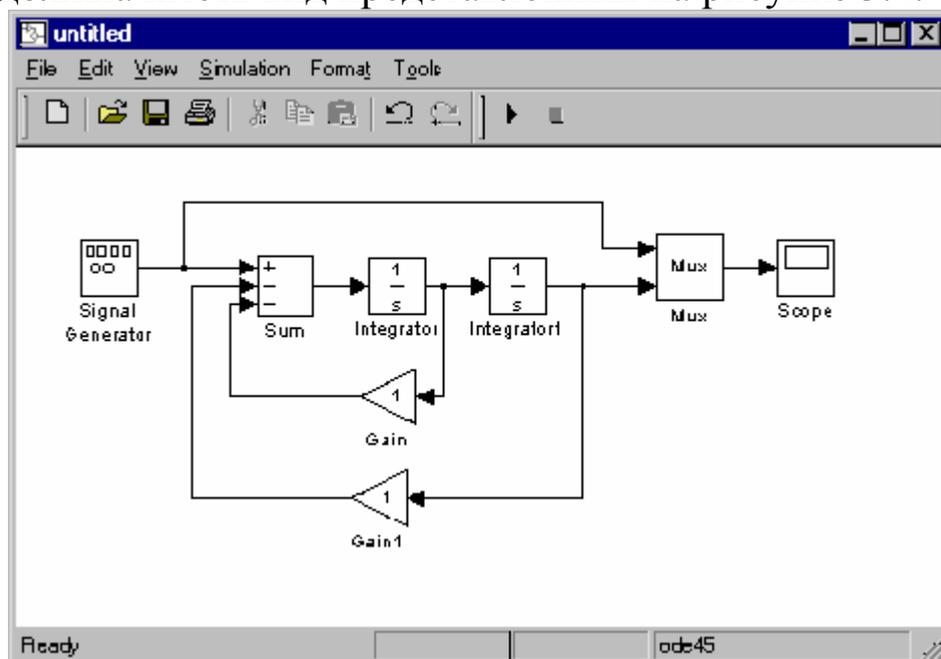


Рисунок 3.4 - Схема модели лабораторной работы

Копирование блока осуществляется путем перетаскивания с нажатой клавишей Ctrl. Поворот блока осуществляется путем выделения блока и нажатием комбинации клавиш Ctrl+F.

7. Перейдем к настройке системы. Открытие окна настройки блока осуществляется путем двойного щелчка. В блоке *Signal Generator* рассмотрим доступные нам сигналы. Выбираем *Squire* – частоту в Гц – 0.02. Смотрим, как система отреагирует на скачкообразное возмущение. Система не является колебательной.

8. Уменьшим КУ по внутреннему контуру (**Gain**). $KU_{Gain}=0.1$, т.е. уменьшим демпфирующую силу в 10 раз. Моделируем. Имеем колебательную систему. Теперь нужно поменять *собственную частоту*, которую определяет нижний KU_{Gain1} . Увеличим его в 10 раз. Видим, что увеличилась частота и уменьшилась амплитуда колебаний.

9. В *Signal Generator* увеличим амплитуду в 5 раз. На внутреннем **Gain** вместо 0.1 сделаем 0.3 – увеличим коэффициент демпфирования.

Мы убедились, что частота колебаний регулируется усилителем на внешнем контуре, а затухание колебаний регулируется на внутреннем контуре.

10. Ознакомиться с демонстрационными примерами *Tracking a bouncing boll, Simple pendulum simulation, Toilet bowl flushing animation* и описать один из них по указанию преподавателя:

- назначение системы;
- состав модели;
- особенности движения системы.

4 Моделирование колебательных систем

Теоретическая часть

Простейшая колебательная система (например, грузик на пружине, LC - колебательный контур) имеет одну степень свободы и описывается обыкновенным дифференциальным уравнением первого порядка. Так простейшая одномассовая поступательная система описывается уравнением равновесия сил, действующих на подвижную массу m :

$$m \cdot \ddot{x} + b \cdot f(\dot{x}) + c \cdot x = F(t), \quad (4.1)$$

где x - перемещение массы; $m \cdot \ddot{x}$ - сила инерции; $b \cdot f(\dot{x})$ - демпфирующая сила сопротивления движению, зависящая от скорости \dot{x} ; $c \cdot x$ - усилие пружины (восстанавливающая сила); $F(t)$ - внешняя сила.

В зависимости от характера демпфирующей силы уравнение (4.1) движения принимает следующие формы:

1. В случае вязкого трения, характерного для механических крутильных систем, демпфирующая сила является линейной функцией скорости:

$$m \cdot \ddot{x} + b \cdot \dot{x} + c \cdot x = F(t), \quad (4.2)$$

где $b \cdot \dot{x}$ - линейная сила вязкого трения.

2. В случае квадратичного закона трения, характерного для гидравлических систем, демпфирующая сила является квадратичной функцией скорости:

$$m \cdot \ddot{x} + b \cdot \dot{x}^2 \cdot \text{sign}(\dot{x}) + c \cdot x = F(t), \quad (4.3)$$

где $b \cdot \dot{x}^2 \cdot \text{sign}(\dot{x})$ - квадратичная демпфирующая сила.

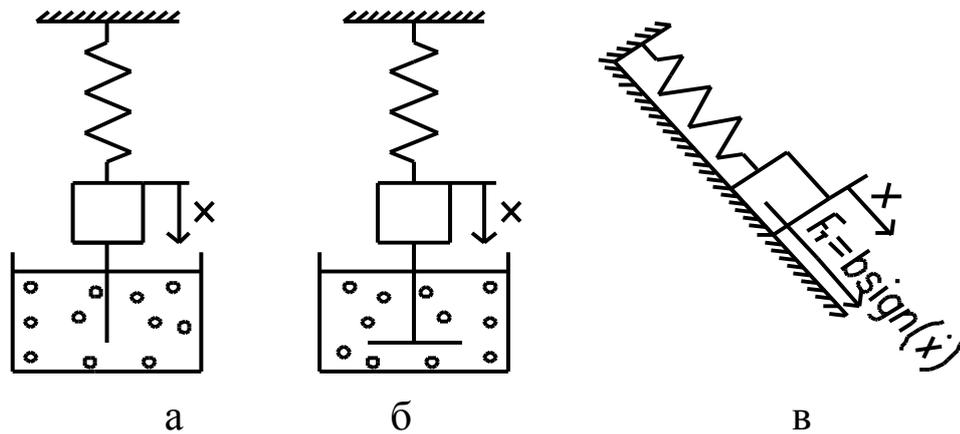


Рисунок 4.1 - Колебательная система: а. с вязким линейным трением, б. с вязким квадратичным трением, в. с сухим трением

3. В случае сухого трения, характерного для механических поступательных систем, демпфирующая сила является функцией знака скорости, а величина ее постоянна:

$$m \cdot \ddot{x} + b \cdot \text{sign}(\dot{x}) + c \cdot x = F(t), \quad (4.4)$$

где $b \cdot \text{sign}(\dot{x})$ - сила сухого трения.

На рисунок 4.1 приведены примеры колебательных систем с вязким линейным, вязким квадратичным и сухим трением.

Свободные колебания системы возникают под действием ненулевых начальных условий ($x(0) \neq 0$, $\dot{x}(0) \neq 0$), либо при скачкообразном приложении силы $F(t)$. Свободные колебания затухают под действием демпфирующей силы, а характер затухания зависит от вида демпфирования (трения).

Порядок составления структурной модели

Используя уравнения (4.2), (4.3), (4.4) построить блок-схему колебательной системы с различными демпфирующими силами. Для этого необходимо:

1. Разрешить дифференциальное уравнение относительно слагаемого со старшей производной.

2. С помощью цепочки сумматоров собрать модель правой части уравнения.

3. Используя цепочки из двух последовательно включенных интеграторов, получить переменные $x(t)$ и $\dot{x}(t)$.

4.С помощью усилителей с регулируемым коэффициентом передачи или соответствующих нелинейных блоков получить слагаемые правой части уравнения по п. 1 и подключить их к цепочке сумматоров.

Таблица 4.1 – Индивидуальные задания

№ вар	m	b	c	$\dot{x}(0)$	$x(0)$	№ вар	m	b	c	$\dot{x}(0)$	$x(0)$
1	10	0.5	10	10	1	13	8	0.48	32	10	2
2	20	1.2	0.8	20	3	14	22	1.32	1.98	20	3
3	15	1.2	60	30	2	15	24	1	24	30	1
4	25	1.25	0.25	10	1	16	30	2.7	1.2	10	2
5	12	1.08	60	20	3	17	10	0.7	40	20	3
6	40	2.8	0.4	30	2	18	20	1.6	0.2	30	1
7	15	1.35	15	10	2	19	36	1.8	1.44	10	2
8	4	0.28	36	20	1	20	16	0.8	64	20	3
9	20	1.4	80	30	3	21	50	4	4.5	30	1
10	18	1.8	18	10	3	22	25	2.5	100	10	1
11	10	0.6	0.9	20	2	23	32	1.6	0.32	20	3
12	25	1.75	25	30	1	24	14	1.12	56	30	2

5. Проверить блок-схему - в каждом замкнутом контуре число перемен знака должно быть нечетным.

6. Внешнюю силу $F(t)$ моделировать соответствующим генератором.

7. В полученной схеме установить коэффициенты в соответствии с вариантом (таблица 4.1).

8. Набрать модель в рабочем окне SIMULINK.

Порядок выполнения работы

1. Составить и набрать структурную модель колебательной системы с вязким трением (по уравнению (4.2)). Задать:

- возмущение - единичный скачок;
- начальные условия - нулевые;
- метод моделирования - Рунге-Кутта;
- точность - 0.01;
- длительность моделирования $T_k=50$ с.

2. Запустить моделирование командой “Start” или “Счет” (функциональная клавиша F3), по окончании вычисления вывести на экран график процесса $x(t)$ командой “Грф” (F6) и зарисовать.

3. Изменяя коэффициент демпфирования (уменьшая в 2, 4, 10 раз), получить графики соответствующих переходных процессов и зарисовать. Сделать вывод о влиянии коэффициента демпфирования на затухание колебаний.

4. Для заданного коэффициента демпфирования исследовать влияние коэффициента передачи во внешнем контуре на частоту собственных колебаний системы. Для этого увеличивать и уменьшать указанный коэффициент в 2 и 4 раза. Сделать выводы.

5. Исследовать влияние ненулевых начальных условий на движение системы. Для этого установить исходные значения коэффициентов, а сигнал на выходе генератора установить равным 0. Задать последовательно начальные условия $\dot{x}(0)$ и $x(0)$. Зарисовать графики переходных процессов при ненулевых начальных условиях. Сделать выводы.

6. Исследовать поведение системы с вязким трением при подключении в виде возмущающего воздействия синусоидального сигнала $F(t) = k \cdot \sin(\omega \cdot t)$ при нулевых начальных условиях в районе резонанса. Для этого необходимо найти резонансную частоту системы ω_p .

$$\omega_p = \sqrt{\frac{c}{m}} \quad (4.5)$$

Определить амплитуду установившихся колебаний при частоте возмущающего сигнала $0.5 \cdot \omega_p$, ω_p , $2 \cdot \omega_p$. Сделать выводы.

7. Исследовать систему с вязким трением при подключении в виде возмущающего воздействия меандра (прямоугольных периодических колебаний) при нулевых начальных условиях. Зарисовать процесс $x(t)$ при частоте возмущающего сигнала $0.25 \cdot \omega_p$, ω_p , $4 \cdot \omega_p$. Описать характер движений системы. Сделать выводы.

8. Исследовать колебательные системы с вязким линейным, с вязким квадратичным и с сухим трением (уравнения (4.2), (4.3), (4.4) соответственно) с нулевыми начальными условиями при единичном возмущающем воздействии. Зарисовать полученные

выходные характеристики и сделать вывод о влиянии различных демпфирующих сил на вид переходной характеристики.

Контрольные вопросы

1. По каким законам уменьшается амплитуда колебаний при вязком линейном трении, при вязком квадратичном трении и при сухом трении?

2. Как изменяется амплитуда вынужденных колебаний в зависимости от частоты возмущающей силы?

3. Сколько собственных частот имеют колебательные системы, описываемые обыкновенными уравнениями 3-го, 4-го, 5-го, 6-го порядка?

4. Как изменяется затухание колебаний при изменении коэффициента демпфирования?

5. Как изменится собственная частота колебательной системы при изменении массы?

Литература

1. Пановко, Я.Г. Введение в теорию колебаний/ Я.Г. Пановко. - М.: Наука, 1991. – 320 с.

2. Тимошенко, С.П. Колебания в инженерном деле/ С.П. Тимошенко. - М.: Физматгиз, 1989. – 474 с.

5 VBA. Встроенные функции ввода/вывода

Теоретические сведения

Модули

Все программы Visual Basic for Application сохраняются в рабочих файлах Microsoft Office, это могут быть документы Word, электронные таблицы Excel и файлы других программ пакета. Обычно программы VBA находятся в специальной части файлов, называемой Modules (модули). Модуль VBA содержит *исходный код* программы – текстовое представление инструкций по выполнению вычислений. Каждый файл может иметь до нескольких модулей. Модули объединены в общий объект Project (проект). Существует несколько типов модулей:

- модули объектов документа, которые содержат нужные программные реализации для событий этих объектов. Например, если мы создали экранную форму, которую надо запускать одновременно с открытием документа, то процедуру ее показа надо записать в событие книги - Open;
- экранные формы, которые позволяют создавать свои рабочие окна для реализации ваших собственных задач. Формально данные модули состоят из двух – собственно экранной формы и программного кода для реализации необходимых событий ее объектов;
- модули программ – основной элемент VBA, который содержит программный код для функций пользователя, макросов и внутренних подпрограмм, необходимых для работы как функций и макросов, так и разработанных экранных форм;
- модули классов (объектов), которые создают новые объекты (классы) на основе уже имеющихся объектов приложения.

Первая группа модулей существует в документе всегда, для других групп имеются методы для их создания или загрузки из соответствующих файлов.

Чтобы создать модуль в проекте необходимо выполнить команду **Insert => Module (UserForm, Class Module)**, что приведет к созданию выбранного модуля и изменению содержимого окна проекта. Появится строка **Module1**, соответствующая вставленному

модулю в окне проекта. Модуль может быть создан автоматически, когда пакет (Word, Excel) выполняет запись макроса, ему присваивается имя **ModuleN**, где *N* – первый свободный номер в проекте модулей во время текущего сеанса работы. Например, в первый раз, когда вы сохраняете записанный макрос в Книге1.xls, Excel создает модуль с именем **Module1**. Если вы продолжаете записывать макросы в том же сеансе работы и сохранять их там же, Excel продолжает сохранять записанные макросы в том же модуле **Module1** до тех пор, пока вы не выберете другую рабочую книгу. Если позже в том же сеансе работы вы опять захотите сохранить записанные макросы в Книге1.xls, то будет добавлен другой модуль с именем **Module2**.

Любой из добавляемых модулей может быть импортирован в проект, используя команду **File => Import File...**, затем надо показать нужный файл для загрузки. Эта команда бывает очень удобной, для внесения в проект набора стандартных процедур, которые вы используете при создании своих программ. Чтобы создать такие файлы для загрузки, нужно необходимые модули экспортировать в файл по команде **File => Export File...**

Для удаления модуля следует выполнить следующие команды:

- 1) кликнуть правой кнопкой мышки на имени этого модуля в окне проекта, например, на **Module1**;
- 2) в открывшемся контекстном меню выполнить команду **Remove** (Удалить);
- 3) выбрать предлагаемую операцию экспорта модуля перед его удалением **Yes** (Да) или кликнуть на кнопке **No** (Нет), если экспортировать модуль перед удалением не надо.

Для просмотра модулей и работы с программным кодом вам необходимо использовать редактор Microsoft Visual Basic. Этот компонент предоставляет инструментальные средства, которые используются для создания новых модулей, просмотра содержимого существующих модулей, создания и редактирования исходного кода программ, разработки пользовательских диалоговых окон и выполнения других задач, относящихся к написанию и обслуживанию программ на VBA.

Независимо от того, работаете ли вы в Word или в Excel, редактор VBA запускается одним и тем же способом. Используйте для этого один из следующих приемов:

- Выберите на ленте инструментов вкладку **Разработчик**, а на ней инструмент **Visual Basic**.

- Нажмите сочетание клавиш **Alt+F11**.

Типы данных

Тип данных определяет диапазон возможных значений переменной, количество памяти для ее размещения и набор допустимых операций. Базовые типы данных VBA приведены в таблице 5.1.

Таблица 5.1 – Типы данных VBA

<i>Тип данных</i>	<i>Описание и диапазон значений</i>
Array	Массив переменных любого встроенного типа данных
Boolean	True (истина) или False (ложь)
Byte	Положительное число от 0 до 255
Currency	Используется для денежных вычислений с фиксированным количеством десятичных знаков. От -922 337 203 685 477,5808 до 922 337 203 685 477,5807
Date	Дата и время. Диапазон дат: от 01.01.0100 г. до 31.12.9999 г. Диапазон времени: от 00:00:00 до 23:59:59
Decimal	Десятичное представление данных в целочисленной или вещественной форме
Double	Число с плавающей точкой двойной точности. Отрицательные числа: от -1,79769313486232E+308 до -4,94065645841247E-324. Положительные числа: 4,94065645841247E-324 до 1,79769313486232E+308
Integer	Целое число от -32 768 до 32 767
Long	Длинное целое число от -2 147 483 648 до 2 147 483 647
Object	Ссылка на объект
Single	Число с плавающей точкой обычной точности. Отрицательные числа от -3,402823E+38 до 1,401298E-45. Положительные числа от 1,401298E-45 до 3,402823E+38

Продолжение таблицы 5.1

String (переменной длины)	Длина строки от 0 до, приблизительно, 2 миллиардов символов
String (фиксированной длины)	от 0 до ~65 000 символов
Variant	Может использоваться для хранения любого типа данных, кроме строк фиксированной длины. Диапазон зависит от фактически сохраняемых данных.
Определяемый пользователем тип данных	Используется для описания сложных данных на основе базовых типов.

Переменные

VBA поддерживает 2 способа объявления переменных:

Неявный - VBA создает переменную и резервирует память для ее хранения, когда эта переменная в первый раз появляется в каком-либо операторе VBA (обычно в операторе присваивания).

Явный - имя и тип переменной определяются до первого обращения.

Явное объявление переменных предпочтительнее, так как:

- ускоряет выполнение кода;
- уменьшается количество ошибок;
- код становится более понятным.

Для явного объявления переменных используется оператор Dim со следующим синтаксисом:

Dim <имяПеременной> [As <типДанных>]

Здесь:

имяПеременной - любой допустимый идентификатор;

типДанных - любой поддерживаемый VBA тип данных.

Для принудительного включения явного и обязательного объявления переменных используется оператор Option Explicit. Он должен быть расположен в самом начале модуля (раздел Declarations). При *неявном объявлении переменной* можно задавать ее тип, добавляя в конец имени специальные символы определения типа (type definition character). Эта возможность сохранена для совместимости с устаревшими версиями языка Basic.

Константы

VBA поддерживает *неименованные* и *именованные* константы. *Неименованные* константы — это фактические значения данных определенного типа. Их можно использовать без какого-либо объявления, непосредственно в выражениях.

Именованные константы — это мнемонические обозначения неименованных констант. Для использования в программе именованные константы должны быть предварительно объявлены с ключевым словом `Const`.

Синтаксис объявления именованных констант:

```
Const <имяКонстанты> [As <типДанных>] = <значение1>
```

Где:

имяКонстанты – имя константы;

значение – значение константы.

Пример:

```
Const intPersonCount As Integer = 100
```

```
Const maxLen% = 50
```

Библиотеки типов VBA представляют множество встроенных констант. Такие константы используются обычно при работе с объектами приложения. Эти константы не требуют предварительного описания. Имена встроенных констант начинаются с префикса, который указывает, к объекту какого приложения Microsoft Office они относятся: xl(Excel), wd(Word), ac(Access), pp(Power Point), ol(Outlook), vb(VBA).

Операции

В программах на VBA можно использовать весь типовой для универсального языка программирования набор операций:

- математические (или арифметические) – выполняются над числами и их результатом являются числа (таблица 5.2);
- отношения – применяются не только к числам и их результатом являются логические значения (таблица 5.3);
- логические – используются в логических выражениях и их результатом являются логические значения (таблица 5.4).

Операции над строками представлены в таблице 5.5.

Таблица 5.2 – Математические операции

<i>Операция</i>	<i>Формат</i>	<i>Название</i>
+	[Операнд1] + [Операнд2]	Сложение
-	[Операнд1] - [Операнд2]	Вычитание
-	- [Операнд1]	Перемена знака (или унарный минус)
*	[Операнд1] * [Операнд2]	Умножение
/	[Операнд1] / [Операнд2]	Деление
\	[Операнд1] \ [Операнд2]	Целочисленное деление
Mod	[Операнд1] Mod [Операнд2]	Остаток от деления
^	[Операнд1] ^ [Операнд2]	Возведение в степень

Таблица 5.3 – Операции отношения

<i>Операция</i>	<i>Формат</i>	<i>Название</i>
<	[Операнд1] < [Операнд2]	Меньше
>	[Операнд1] > [Операнд2]	Больше
<=	[Операнд1] <= [Операнд2]	Меньше или равно
>=	[Операнд1] >= [Операнд2]	Больше или равно
<>	[Операнд1] <> [Операнд2]	Не равно
=	[Операнд1] = [Операнд2]	Равно
Is	[Операнд1] Is [Операнд2]	Сравнение двух операндов, содержащих ссылки на объекты
Like	[Операнд1] Like [Операнд2]	Сравнение двух строковых выражений

Таблица 5.4 – Логические операции

<i>Операция</i>	<i>Формат</i>	<i>Название</i>
And	[Операнд1] And [Операнд2]	Логическое умножение (И)
Or	[Операнд1] Or [Операнд2]	Логическое сложение (ИЛИ)
Xor	[Операнд1] Xor [Операнд2]	Исключающее Or (Исключающее «ИЛИ»)
Not	Not [Операнд1]	Логическое отрицание (НЕ)
Imp	[Операнд1] Imp [Операнд2]	Логическая импликация
Eqv	[Операнд1] Eqv [Операнд2]	Логическая эквивалентность

Таблица 5.5 – Операции над строками

<i>Операция</i>	<i>Формат</i>	<i>Название</i>
&	[Строка1] & [Строка2]	Конкатенация (сцепление строк). Для сцепления строк допустимо использование операции вида [Строка1] + [Строка2]

Оператор присваивания

Оператор присваивания предназначен для задания ("присваивания") значения переменным, инициализации констант или изменения свойств объекта. Формат оператора присваивания:

$$[Let] \langle \text{имяЭлемента} \rangle = \langle \text{выражение} \rangle$$

где:

Let - необязательная инструкция, которая сохранена для совместимости;

$\langle \text{имяЭлемента} \rangle$ – это идентификатор переменной, константы (при объявлении) или свойства объекта;

$\langle \text{выражение} \rangle$ состоит из переменных, констант, операций и функций.

Примеры использования оператора присваивания:

Place="d:\windows\system"

File="GameTree.Exe"

Student_Card(100).Group=33

Для присваивания переменной ссылки на объект применяется инструкция Set:

Set *<объектная*
Переменная = [*New*] *<объектноеВыражение>* | *Nothing*

где:

New – опция (ключевое слово), которая используется при создании нового экземпляра класса;

Nothing – опция (ключевое слово), которая позволяет освободить все системные ресурсы и ресурсы памяти, выделенные для объекта.

В следующем примере инструкция Set присваивает переменной MyRange ссылку на диапазон ячеек A1:B1.

Set MyRange = Range("A1:B1")

Приоритеты операций

Порядок выполнения операций определяется расстановкой круглых скобок и приоритетом (старшинством) операций. Это обеспечивает однозначность в трактовании значений выражений. В таблице 5.6 приведены приоритеты выполнения операций.

Таблица 5.6 – Приоритеты операции VBA

<i>Приоритет</i>	<i>Операция</i>
1	Вызов функции и скобки
2	^
3	- (смена знака)
4	*, /
5	\
6	Mod
7	+, -
8	>, <, >=, <=, <>, =
9	Not
10	And
11	Or
12	Xor
13	Eqv
14	Imp

Для иллюстрации вышесказанного приведем пример программы, вычисляющей диаметр, периметр и площадь круга, заданного радиусом (листинг 1).

Листинг 1. Переменные, константы, операции

Private R As Single 'радиус

Public D As Single, S As Single, P As Single 'диаметр, периметр, площадь

Const PI = 3.14 'Константа PI (имеется встроенная функция VBA)

Sub sample1()

Dim res As String

R = 20

*D = 2 * R 'вычисляем диаметр*

*S = PI * R ^ 2 'вычисляем площадь*

*P = D * PI 'вычисляем периметр*

'формируем строку сообщения

res = "Для окружности с радиусом " & CStr(R) & " диаметр равен " & CStr(D) &

_ " , периметр - " & CStr(P) & " , площадь - " & CStr(S)

Debug.Print res ' вывод результата в окно отладчика

End Sub

Функции взаимодействия с пользователем

Для организации диалога с пользователем VBA представляет две встроенные функции - MsgBox и InputBox. Окно сообщений MsgBox выводит сообщения для пользователя, а окно ввода InputBox обеспечивает возможность получения информации от пользователя.

Функция MsgBox() выводит на экран диалоговое окно, содержащее сообщение, устанавливает режим ожидания нажатия кнопки пользователем, а затем возвращает значение типа Integer, указывающее, какая кнопка была нажата.

Формат функции MsgBox:

MsgBox (Prompt [, Buttons] [, Title] [, HelpFile, Context])

Назначение параметров:

Prompt – строковое выражение, отображаемое как сообщение в диалоговом окне;

Title – строковое выражение, отображаемое в строке заголовка диалогового окна. Если этот параметр опущен, в строку заголовка

помещается имя приложения, из которого запускается программа VBA;

`HelpFile` – строковое выражение, определяющее имя файла справки, содержащего справочные сведения о данном диалоговом окне; обычно это файл, который вы уже должны были создать с помощью `Windows Help Compiler`. Если этот параметр указан, необходимо также указать параметр `Context`;

`Context` – числовое выражение, определяющее номер соответствующего раздела справочной системы. Если этот параметр указан, необходимо указать также параметр `HelpFile`;

`Buttons` – числовое выражение, которое задает количество и тип кнопок в диалоговом окне `MsgBox`. `Buttons` указывает также кнопку по умолчанию в диалоговом окне и содержит ли это диалоговое окно стандартные значки `Windows` для предупредительных сообщений и запросов пользователя. Значения `Buttons` можно получить из справочной системы.

Простой пример использования функции `MsgBox`:

```
Sub sample2()
    MsgBox "Видите ли вы две кнопки?", vbYesNo +
vbInformation, "Сообщение"
End Sub
```

В данном примере `MsgBox` применяется не в виде функции, а в виде процедуры (т.е. не возвращает никакого значения). Следовательно, код выбранной кнопки нигде не сохраняется и не может быть использован. Чтобы определить, какая кнопка была нажата, `MsgBox` необходимо вызвать как функцию, т.е. сохранить возвращаемое значение в переменную (Листинг 2).

Листинг 3. Использование `MsgBox`

```
Sub sample3()
    Dim res ' объявляем переменную
    ' вызываем MsgBox и сохраняем значение в переменной
    res = MsgBox("Видите ли вы две кнопки?", vbYesNo +
vbInformation, "Сообщение") Debug.Print res ' печатаем
полученное значение
End Sub
```

При выполнении этого макроса, когда пользователь выбирает кнопку `Yes` или `No` в переменной `res` сохраняется число, соответствующее выбранной кнопке.

Вместо возвращаемых функцией MsgBox целочисленных значений удобнее пользоваться predefined константами VBA. В таблице 5.7 приведены возвращаемые значения констант функции MsgBox.

Таблица 5.7 – Возвращаемые значения функции MsgBox

<i>Константа</i>	<i>Означает, что пользователь нажал кнопку</i>
vbAbort	Стоп (Abort)
vbCancel	Отмена (Cancel)
vbIgnore	Пропустить (Ignore)
vbNo	Нет (No)
vbOk	Ок
vbRetry	Повтор (Retry)
vbYes	Да (Yes)

Дополним код листинга 3 проверкой возвращенного значения (листинг 4).

Листинг 4. Проверка возвращаемого значения MsgBox

Sub sample4()

' вызываем MsgBox и сохраняем значение в переменной

res = MsgBox("Видите ли вы две кнопки?", vbYesNo + vbInformation, "Сообщение")

' проверяем, какая кнопка нажата

If res = vbYes Then: MsgBox "Вы нажали Yes", "Результат выбора"

Else : MsgBox "Вы нажали No", "Результат выбора"

End If

End Sub

Функция InputBox() выводит на экран диалоговое окно, содержащее сообщение и поле ввода, устанавливает режим ожидания ввода текста, а затем возвращает значение типа String, содержащее текст, введенный в поле. Формат функции InputBox:

InputBox (Prompt [, Title] [, Default] [, XPos] [, Ypos] [, HelpFile, Context])

Назначение параметров:

Prompt – строковое выражение, отображаемое как сообщение в диалоговом окне;

Title – строковое выражение, отображаемое в строке заголовка диалогового окна. Если этот параметр опущен, в строку заголовка помещается имя приложения;

Default – строковое выражение, отображаемое в поле ввода и используемое по умолчанию, если пользователь не введет другую строку. Если этот параметр опущен, поле ввода изображается пустым;

Xpos и Ypos – числовые выражения, которые указывают местоположение окна ввода и являются координатами верхнего левого угла диалогового окна; Xpos – горизонтальное расстояние от левого края окна; Ypos – вертикальное расстояние от верхнего края окна. Если эти параметры опущены, диалоговое окно выравнивается по центру экрана;

HelpFile – строковое выражение, определяющее имя существующего файла справки, содержащего справочные сведения о данном диалоговом окне. Если этот параметр указан, необходимо также указать параметр Context;

Context – числовое выражение, определяющее номер соответствующего раздела справочной системы. Если этот параметр указан, необходимо указать также параметр HelpFile.

Приведем пример использования функции InputBox для получения имени пользователя.

```
Sub sample5()
```

```
    Dim username As String
```

```
    username = InputBox("Введите ваше имя ", "Пример 5")
```

```
    MsgBox ("Здравствуйте, "+username)
```

```
End Sub
```

В результате выполнения этого макроса на экран последовательно выводятся диалоговые окна ввода и вывода (рисунок 5.1).

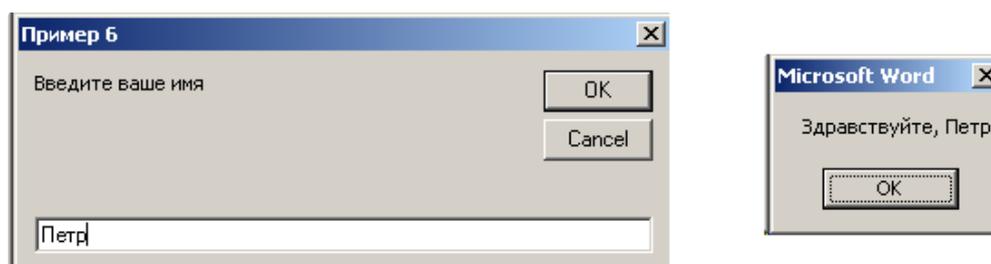


Рисунок 5.1 – Интерактивные функции VBA.

Пример 1.1: Перевести минуты в часы.*Sub lr1()**Const min_1h = 60**Dim min As Integer**Dim Hours As Single**min = Val(InputBox("Введите количество минут"))**Hours = min/min_1h**MsgBox min & " минут составляет " & Hours & " час (а, ов)"**End Sub***Порядок выполнения работы:**

- 1.1. Изучить основные теоретические сведения.
- 1.2. Составить программу, которая переводит одни единицы измерения в другие. Исходные данные вводятся с клавиатуры, результат выводится на экран.
- 1.3. С помощью функций взаимодействия с пользователем доработайте (преобразуйте) Вашу программу. Добавьте комментарии. Текст и результат выполнения программы занести в отчёт.
- 1.4. Сформулируйте выводы по проделанной работе. Оформите отчёт.

Варианты заданий

Написать программу перевода мерных единиц:

1. Километры в метры.
2. Метры в сантиметры.
3. Дециметры в километры.
4. Тонны в килограммы.
5. Центнеры в граммы.
6. Гектары в квадратные метры.
7. Квадратные километры в ары. (1 ар=100 м²).
8. Кубические метры в кубические дециметры.
9. Кубические дециметры в кубические сантиметры.
10. Литры в кубические сантиметры.
11. Гектолитры в кубические дециметры.
12. Сажени в аршины.
13. Аршины в футы.

14. Футы в метры
15. Метры в дюймы.
16. Аршины в сантиметры
17. Дюймы в сантиметры
18. Морские мили в метры
19. Пуды в килограммы.
20. Фунты в килограммы
21. Пуды в фунты.
22. Метры в километры.
23. Сантиметры в метры.
24. Километры в дециметры.
25. Килограммы в тонны.
26. Граммы в центнеры.
27. Квадратные метры в гектары.
28. Ары в квадратные километры (1 ар=100 м²).
29. Кубические дециметры в кубические метры.
30. Кубические сантиметры в литры.

Таблицы перевода единиц измерения

Таблица 5.8 - Единицы длины

Наименование	Перевод в метрические меры
Верста	1,0668 км
Сажень	2,1336 м
Сотка	21,336 мм
Аршин	711,2 мм
Вершок	44,45 мм
Фут	304,8 мм
Дюйм	25,4 мм
Линия	2,54 мм
Точка	0,254 мм

Таблица 5.9 - Единицы площади

Наименование	Перевод в метрические меры
Квадратная верста	1,138 км ²
Десятина	10,925 м ²
Квадратная сажень	4,552 м ²
Квадратный аршин	0,05058 м ²
Квадратный вершок	19,758 см ²
Квадратный фут	9,290 дм ²
Квадратный дюйм	6,452 см ²
Квадратная линия	6,452 мм ²

Таблица 5.10 - Единицы объема, вместимости

Наименование	Перевод в метрические меры
Кубическая сажень	9,713 м ³
Кубический аршин	0,3597 м ³
Кубический вершок	87,824 см ³
Кубический фут	28,317 дм ³
Кубический дюйм	16,387 см ³
Кубическая линия	16,387 мм ³
Ведро	12,299 л
Штоф	1,230 л
Бутылка винная	0,7687 л
Бутылка водочная	0,6150 л
Чарка	123,0 см ³
Четверть (для сыпучих тел)	0,2624 м ³
Гарнец	3,280 л

Таблица 5.11 - Единицы массы

Наименование	Перевод в метрические меры
Берковец	163,8 кг
Пуд	16,38 кг
Фунт	409,5 г
Лот	12,80 г
Золотник	4,266 г
Доля	44,43 мг

Таблица 5.12 - Перевод взаимный метрических единиц измерения площади: см², дм², м², ар (сотка), гектар (га), км²

Перевести и из:	Перевести в:					
	см ²	дм ²	м ²	аров (соток)	га (гектаров)	1 км ² это:
1 см ² это:	1	0,01	0,0001	0,000001	0,00000001	0,0000000001
1 дм ² это:	100	1	0,01	0,0001	0,000001	0,00000001
1 м ² это:	10000	100	1	0,01	0,0001	0,000001
1 ар это: (=1 сотка это:)	1000000	10000	100	1	0,01	0,0001
1 га это:	100000000	1000000	10000	100	1	0,01
1 км ² это:	10000000000	100000000	1000000	10000	100	1

Контрольные вопросы:

- 1 Модуль – это? Назовите типы модулей VBA, которые Вы знаете.

- 2 Как создать модуль?
- 3 Какие типы данных Вы знаете?
- 4 Использование и описание констант.
- 5 Использование и описание переменных.
- 6 Формат функции InputBox.
- 7 Формат функции MsgBox.