ГОСУДАРСТВЕННОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Кафедра «Транспортные и технологические машины»

ИНФОРМАТИКА

Методические рекомендации к лабораторным работам для студентов направления подготовки 23.03.02 «Наземные транспортно-технологические комплексы» дневной формы обучения

Могилев 2018

УДК 004.4
ББК 32.973
И 74

Рекомендовано к изданию
учебно-методическим отделом
Белорусско-Российского университета

Одобрено кафедрой «Транспортные и технологические машины» «31» августа 2018 г., протокол № 1

Составители: канд. техн. наук, доц. В. В. Береснев;
ст. преподаватель В. И. Семчен

Рецензент канд. техн. наук, доц. А. Е. Науменко

Методические рекомендации к лабораторным работам для студентов направления подготовки 23.03.02 «Наземные транспортно-технологические комплексы» дневной формы обучения.

Учебно-методическое издание

ИНФОРМАТИКА

Ответственный за выпуск И. В. Лесковец

Технический редактор С. Н. Красовская

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 56 экз. Заказ №

Издатель и полиграфическое исполнение:
Государственное учреждение высшего профессионального образования
«Белорусско-Российский университет».
Свидетельство о государственной регистрации издателя,
изготовителя, распространителя печатных изданий
№ 1/156 от 24.01.2014.
Пр. Мира, 43, 212000, Могилев.

Содержание

Введение

Программа курса «Информатика» для специальности 23.03.02 «Наземные транспортно-технологические комплексы» предусматривает выполнение студентами лабораторных работ, целью которых является получение практических навыков работы с операционной системой Windows, текстовым процессором WORD, табличным процессором Excel, созданием презентаций, программированием в среде Visual Studio, разработкой приложений для решения различных инженерных задач, работой в СУБД Access.

Задание выдается преподавателем. Отчет представляется в виде файла с выполненным заданием.

1 Лабораторная работа № 1. Перевод чисел из двоичной системы исчисления в десятеричную и наоборот

Цель работы: изучить методы перевода чисел из двоичной системы счисления в десятеричную и наоборот.

Основные сведения

В двоичной системе всего две цифры – 0 и 1. Особую роль здесь играет число 2 и его степени: 2, 4, 8 и т. д. Самая правая цифра числа показывает число единиц, следующая цифра – число двоек, следующая – число четверок и т. д. Двоичная система счисления позволяет закодировать любое натуральное число – представить его в виде последовательности нулей и единиц. В двоичном виде можно представлять не только числа, но и любую другую информацию: тексты, картинки, фильмы и аудиозаписи. Инженеров двоичное кодирование привлекает тем, что легко реализуется технически.

Для отладки программ и в других ситуациях в программировании актуальной является проблема перевода чисел из одной позиционной системы исчисления в другую.

Если основа новой системы исчисления равняется некоторой степени старой системы исчисления, то алгоритм перевода очень простой: нужно сгруппировать справа налево разряды в количестве, равном показателю степени, и заменить эту группу разрядов соответствующим символом новой системы исчисления:

– перевод из двоичной системы в десятичную:

$$110100101 = 1\cdot 2^8 + 1\cdot 2^7 + 0\cdot 2^6 + 1\cdot 2^5 + 0\cdot 2^4 + 0\cdot 2^3 + 1\cdot 2^2 + 0\cdot 2^1 + 1\cdot 2^0 = 421,$$

$$10{,}11 = 1\cdot 2^1 + 0\cdot 2^0 + 1\cdot 2^{-1} + 1\cdot 2^{-2} = 1\cdot 2 + 0\cdot 1 + 1\cdot 0{,}5 + 1\cdot 0{,}25 = 2{,}75;$$

– из восьмеричной в десятичную:

$$735 = 7\cdot 8^2 + 3\cdot 8^1 + 5\cdot 8^0 = 477;$$

– из шеснадцатиричной в десятичную:

$$92C8 = 9\cdot 16^3 + 2\cdot 16^2 + 12\cdot 16^1 + 8\cdot 16^0 = 37576.$$

Для перевода чисел из системы исчисления с основой p в систему исчисления с основой q с использованием арифметики старой системы исчисления с основой p нужно:

– для перевода целой части последовательно число, записанное в системе основой, делить на основу новой системы исчисления, выделяя остатки. Последние, записанные в обратном порядке, будут образовывать число в новой системе исчисления;

– для перевода дробной части последовательно дробную часть умножать

на основу новой системы исчисления, выделяя целые части, которые и будут образовывать запись дробной части числа в новой системе исчисления.

Пример: 999,35 = 1111100111,01011:

для целой части:

999	2									
1	499	2								
	1	249	2							
		1	124	2						
			0	62	2					
				0	31	2				
					1	15	2			
						1	7	2		
							1	3	2	
								1	1	

для дробной части:

0,	35
	2
0,	70
	2
1,	40
	2
0,	80
	2
1,	60
	2
1,	20

Порядок выполнения

1 Выполнить перевод двоичных чисел в десятичные и десятичных в двоичные в соответствии с заданием.

2 Проверить полученные значения с использованием калькулятора.

Контрольные вопросы

1 Правила перевода двоичных чисел в десятичные.

2 Правила перевода десятичных чисел в двоичные.

2 Лабораторная работа № 2. Действия над двоичными числами

Цель работы: изучить арифметические действия над двоичными числами.

Основные сведения

Арифметические действия в двоичной системе производятся по тем же правилам, что и в десятичной системе счисления. Однако, т. к. в двоичной системе счисления используются только две цифры 0 и 1, то арифметические действия выполняются проще, чем в десятичной системе.

Сложение двоичных чисел. Сложение выполняется поразрядно столбиком, начиная с младшего разряда и используя таблицы двоичного сложения:

$$0 + 0 = 0$$
$$0 + 1 = 1$$
$$1 + 0 = 1$$
$$1 + 1 = 10.$$

При сложении необходимо помнить, что 1 + 1 дают нуль в данном разряде и единицу переноса в старший.

$$\begin{array}{r} 1011010 \\ + \\ 111001 \\ \hline 10010011 \end{array} \qquad \begin{array}{r} 90 \\ + \\ 57 \\ \hline 147. \end{array}$$

Вычитание двоичных чисел. Вычитание выполняется поразрядно столбиком, начиная с младшего разряда и используя таблицы двоичного вычитания:

$$0 – 0 = 0$$
$$1 – 0 = 1$$
$$1 – 1 = 0$$
$$10 – 1 = 1.$$

$$\begin{array}{r} 1011010 \\ - \\ 111001 \\ \hline 100001 \end{array} \qquad \begin{array}{r} 90 \\ - \\ 57 \\ \hline 33 \end{array}$$

т. е. при вычитании двоичных чисел в случае необходимости занимается единица из старшего разряда, которая равна двум единицам младшего разряда.

Умножение двоичных чисел. Умножение в двоичной системе производится по тому же принципу, что и в десятичной системе счисления, при этом используется таблица двоичного умножения:

$$0 \cdot 0 = 0$$
$$0 \cdot 1 = 0$$
$$1 \cdot 0 = 0$$
$$1 \cdot 1 = 1.$$

$$\begin{array}{r} 10101 \\ \times \\ 1001 \\ \hline 10101 \\ + \\ 10101 \\ \hline 10111101 \end{array} \qquad \begin{array}{r} 1101 \\ \times \\ 11 \\ \hline 1101 \\ + \\ 1101 \\ \hline 100111 \end{array}$$

$21 \times 9 = 189$ $\qquad$ $13 \times 3 = 39$

Деление двоичных чисел. Деление в двоичной системе производится вычитанием делителя со сдвигом вправо, если остаток больше нуля.

Делимое больше делителя:

```
 110010 | 1010
 1010   |-----
-------|  101
 001010
-1010
-------
   0000
```

50 : 10 = 5.

Делимое меньше делителя:

```
  11001  | 101000
 110010  |--------
-101000  |  0,101
--------
 -101000
 --------
  101000
  101000
  ------
  000000
```

```
  25  | 40
 250  |-----
-240  | 0,625
-----
 -100
 -----
   80
 -----
  -200
  -----
   200
   ---
     0
```

Порядок выполнения

1 Выполнить действия с двоичными числами в соответствии с заданием.
2 Проверить полученные значения с использованием калькулятора.

Контрольные вопросы

1 Правила сложения и вычитания двоичных чисел.
2 Правила умножения и деления двоичных чисел.

3 Лабораторная работа № 3. Системное программное обеспечение. Работа с Windows

Цель работы: изучить основные элементы и получить навыки работы в операционной системе Windows.

Основные сведения

Операционная система (ОС) – это совокупность программных средств, осуществляющих управление ресурсами ЭВМ, запуск прикладных программ и их взаимодействие с внешними устройствами и другими программами, а также обеспечивающих диалог пользователя с компьютером.

В функции операционной системы входит:

– осуществление диалога с пользователем;

– ввод-вывод и управление данными;

– планирование и организация процесса обработки программ;

– запуск программ на выполнение;

– всевозможные вспомогательные операции обслуживания;

– передача информации между различными внутренними устройствами;

– программная поддержка работы периферийных устройств.

Среда Windows обеспечивает независимый запуск и параллельное выполнение нескольких программ. Каждая из программ имеет свое собственное окно, в котором отражаются результаты ее работы.

После загрузки Windows большую часть экрана занимает так называемый «Рабочий стол».

«Рабочий стол» – это главная область экрана, которая появляется после включения компьютера и входа в операционную систему Windows. Запущенные программы и открытые папки появляются на рабочем столе. На рабочий стол можно помещать различные объекты, например файлы и папки, и выстраивать их в удобном порядке.

Настройка рабочего стола включает размещение на нем ярлыков для наиболее часто используемых программ, документов и принтеров, а также изменение его фона и тому подобные действия.

Справка и поддержка Windows представляет собой встроенную справочную систему Windows. Она позволяет получать быстрые ответы на общие вопросы, предлагает способы выявления неисправностей и инструкции по выполнению тех или иных действий. Если понадобится справка о программе, не являющейся частью Windows, понадобится обратиться к справке программы.

Чтобы открыть центр справки и поддержки Windows, нажмите кнопку «Пуск» и выберите «Справка и поддержка».

Для запуска любого приложения (программы) в Windows можно воспользоваться одним из следующих способов:

1) щелкнуть мышью по команде «Пуск» на панели задач. При этом появится Главное меню. Затем выбрать команду Все программы и в ней выбрать нужное приложение (программу) и щелкнуть мышью по имени приложения (программы);

2) двойной щелчок по ярлыку на рабочем столе, если ярлык для данного приложения (программы) есть.

Для выхода из Windows необходимо выполнить команду «Пуск-Завершение работы».

В Windows имеется папка Мой компьютер и программа Проводник, с помощью которых можно работать с папками и файлами.

С их помощью Вы можете:

– открывать, закрывать и просматривать содержимое папок;

– копировать, перемещать, удалять и переименовывать файлы и папки;

– можно увидеть содержимое или свойства файла (его размер, дату и время создания и модификации);

– проводник позволяет связывать тип документов с определенным приложением.

Создание папки в окне Проводника. В левой части окна Проводника выбрать диск или папку, где требуется создать новую папку и сделать ее текущей. В правой части окна Проводника в свободном месте вызвать контекстное меню и выбрать команду «Создать – Папку». В окне появляется новая папка с именем, которое присвоено ей по умолчанию – ввести имя новой папки и нажать клавишу Enter.

Для того чтобы выделить файл, папку или группу файлов, папок:

– один файл (папку) – щелкнуть мышкой по имени;

– несколько разных файлов (папок) – нажать клавишу Ctrl и, удерживая ее, щелкнуть все нужные объекты;

– диапазон файлов (папок) – щелкнуть первый файл диапазона нажать клавишу Shift и, удерживая ее, щелкнуть последний файл диапазона;

– диапазон файлов (папок) – указать выделением рамкой с помощью мыши.

Снятие выделения файлов и папок. Снять выделение одного файла или папки – щелкнуть мышкой по имени этого файла.

Снять выделение нескольких файлов – нажать клавишу Ctrl и, удерживая ее, щелкнуть все нужные объекты.

Снять выделение всех файлов или папок – щелкнуть мышкой в поле проводника.

Копирование и перемещение файлов и папок. Выделить файлы и папки. Для копирования в меню выбрать команду «Копировать». Для перемещения в меню выбрать команду «Вырезать».

Открыть папку, куда требуется поместить файлы.

В меню выбрать команду «Вставить».

Изменение имени файла или папки. В окне Проводника выделить файл или папку, которую требуется переименовать. В меню выбрать команду «Переименовать». Ввести новое имя и нажать клавишу Enter.

Удаление файла или папки. В окне Проводника выбрать файл или папку, которую требуется удалить. В меню выбрать команду «Удалить».

Все удаленные файлы и папки помещаются в корзину.

Раскрыть папку «Корзина».

Выделить файлы или ярлыки, которые требуется восстановить.

В выбрать команду «Восстановить».

При восстановлении файла, находившегося в удаленной папке, вначале будет восстановлена сама эта папка.

Файлы, удаленные с сетевых дисков, а также файлы, удаленные со съемных носителей, в папку Корзина не помещаются. Такие файлы удаляются сразу без возможности восстановления.

Копирование и перенос информации между различными документами осуществляется через буфер обмена (Clipboard).

Скопировать (вырезать) элемент можно, выполнив команду «Правка – Копировать (Вырезать)» с помощью кнопок панели инструментов или с помощью контекстного меню. Для этого щелкнуть правой клавишей мыши по выделен-

ному объекту и выбрать команду «Копировать (Вырезать)». Фрагмент помещается в буфер обмена.

Для вставки в любой документ перейти в нужный документ, выполнить команду «Вставить» одним из следующих способов: нажав кнопку на стандартной панели инструментов; вызвать контекстное меню и выполнить команду «Вставить».

Windows позволяет работать одновременно с несколькими приложениями. Поэтому на экране могут одновременно находиться окна нескольких приложений. В каждый момент времени только одно окно является активным, заголовок активного окна выделен синим цветом.

Порядок выполнения

1 Создать две папки в окне Проводника.
2 Создать файлы в одной из папок.
3 Скопировать файлы и вставить во вторую папку.
4 Удалить файлы из первой папки.
5 Войти в папку Корзина.
6 Выделить файлы, которые требуется восстановить.
7 Выбрать команду Восстановить.

Контрольные вопросы

1 Функции операционной системы.
2 Настройка панели задач.
3 Создание, удаление копирование и восстановление файлов и папок.

4 Лабораторная работа № 4. Текстовый процессор MS Word

Цель работы: приобрести навыки работы с текстовым редактором MS Word.

Основные сведения

Microsoft Word 2010 – это текстовый процессор, предназначенный для создания профессионально оформленных документов. Объединяя в себе лучшие средства форматирования текста, приложение MS Word помогает легко создавать и оформлять документы. Кроме того, приложение MS Word обеспечивает удобную среду для совместной работы.

Окно MS Word состоит из элементов, представленных ниже.

Строка заголовка окна. В строке заголовка окна отображается название документа, открытого в данный момент. Новый документ получает по умолчанию название Документ 1 (2, 3 и т. д.).

Кнопки управления окном располагаются в правой части строки заголовка окна. Они позволяют свернуть окно, перевести окно в полноэкранный

или оконный режим, закрыть окно.

Лента – это полоса в верхней части экрана, на которой размещаются все основные наборы команд, сгруппированные по тематикам на отдельных вкладках и в группах.

Панель быстрого доступа – настраиваемая панель инструментов с наиболее часто используемыми командами.

Рабочая область (текстовое поле) находится в центральной части окна MS Word. В этой области набирается текст, создаются и добавляются различные объекты.

Полосы прокрутки – вертикальная полоса прокрутки, находящаяся в правой части окна, и горизонтальная – в нижней части (отсутствует, если ширина окна достаточна для отображения документа).

Линейки. Горизонтальные и вертикальные линейки располагаются вверху и слева. Вертикальная линейка отображается только в режиме Разметка страницы.

Кнопка Линейки, расположенная над полосой прокрутки, позволяет показать или скрыть линейки.

Строка состояния располагается внизу окна. Она позволяет получить сведения о текущем документе, выбрать режим и масштаб отображения документа. Настройка строки состояния осуществляется через контекстное меню.

К элементам интерфейса относятся также **Мини-панели инструментов**, содержащие наиболее часто используемые элементы для оформления текста документа, рисунков, диаграмм и других объектов, состав которых не может быть изменён. Мини-панель для оформления текста появляется автоматически при выделении фрагмента документа. Первоначально отображается полупрозрачная мини-панель, которая становится яркой при наведении на неё указателя мыши.

Лента является главным элементом пользовательского интерфейса MS Word 2010.

Лента позволяет быстро находить и использовать необходимые команды, которые упорядочены в логические группы, собранные на вкладках.

Вкладки:

– **Файл** – обеспечивает доступ ко всем действиям с файлами. Остальные вкладки используются для работы внутри файла;

– **Главная** – содержит команды, связанные с буфером обмена, выбором шрифтов, настройками абзаца, стилями и правкой;

– **Вставка** – включает в себя инструменты для добавления различных объектов в документ;

– **Разметка страницы** – позволяет настраивать параметры страницы и порядок расположения элементов на странице, работать с темами, фоновыми изображениями и интервалами между абзацами в документе;

– **Ссылки** – позволяет вставлять в документ специальные элементы: оглавление, сноски, заголовки, предметный указатель и пр.;

– **Рассылки** – содержит команды для создания, предварительного просмотра и реализации технологии слияния;

– **Рецензирование** – содержит команды, необходимые для проверки документа и предоставления доступа к нему другим пользователям, а также коман-

ды, предназначенные для добавления комментариев, отслеживания и обработки изменений, сравнения версий и защиты документа;

– Вид – содержит все необходимое для отображения документа различными способами, а также для работы с документами в нескольких окнах.

Для перехода к нужной вкладке нужно щёлкнуть левой кнопкой мыши по ее названию.

При наведении на элемент управления указателя мыши отображается всплывающая подсказка с информацией о назначении этого элемента. **Клавиша F1** вызывает окно справки MS Word.

Текстовый курсор представляет собой вертикальную черту. Его местоположение определяет позицию, с которой будет записываться информация, поступающая с клавиатуры, включаться рисунок или таблица. Курсор перемещается с помощью клавиш управления курсором или мыши.

Команда «Параметры» отображает диалоговое окно Параметры MS Word для настройки параметров приложения.

В MS Word 2010 можно быстро настроить ленту в соответствии с выполняемыми задачами.

Настройка ленты (добавление и удаление вкладок, групп, команд) выполняется при помощи вкладки Файл, команды «Параметры – Настройка» ленты.

Для возврата стандартных настроек предназначена кнопка Сброс.

Сохранение активного документа выполняется командой «Файл – Сохранить» или командой «Файл – Сохранить как». Отличие команды «Сохранить как» от команды «Сохранить» в том, что активный документ сохраняется с другим именем. В любом случае появляется диалоговое окно.

В диалоговом окне в левой панели выбрать папку для сохранения. В окне Имя файла ввести имя сохраняемого файла. По умолчанию файлы сохраняются в формате документа MS Word с расширением .docx.

Открытие документа для редактирования осуществляется:

– двойным щелчком по файлу;

– командой «Файл-Открыть».

В диалоговом окне в левой панели выбрать папку, где находится нужный файл. Выделить файл и выполнить команду «Открыть».

Ввод текста начинается с позиции курсора. Верстка текста осуществляется автоматически.

В конце абзаца нажимается клавиша Enter. Абзацем в Word называется любое количество символов, рисунков и т. д. заканчивающееся признаком конца абзаца.

Форматирование символов означает выбор шрифта, его размера и начертания.

Для редактирования и форматирования текста используются, в первую очередь, кнопки команд на ленте вкладки Главная**,** а также диалоговое окно Шрифт, вызываемое либо из Контекстного меню, либо щелчком по кнопке соответсвующей группы в ленте вкладки Главная**.**

Форматирование абзацев заключается в установке отступов между соседними абзацами, а также от краев листа бумаги, создании красной строки и выборе выравнивания текста: по центру, по левому краю, по ширине и т. д.

Для редактирования и форматирования абзаца используются, в первую

очередь, кнопки команд на ленте вкладки Главная, а также диалоговое окно Абзац, вызываемое либо из Контекстного меню, либо щелчком по кнопке соответствующей группы в ленте вкладки Главная.

Чтобы текст располагался на странице наилучшим образом, устанавливают параметры страницы, используя кнопки команд на ленте вкладки Разметка страницы.

Создание и редактирование таблиц. Для создания таблицы необходимо установить курсор в место документа, где должна быть таблица и, используя команду «Таблица», на ленте вкладки Вставка добавить или нарисовать таблицу.

Для добавления столбца или строки в таблицу необходимо установить курсор в столбец или строку, после или пред которыми необходима вставка, а затем выполнить команду контекстного меню Вставить и в выпадающем списке выбрать способ вставки.

Чтобы добавить строку в конец таблицы, можно щелкнуть по последней ячейке последней строки, а затем нажать клавишу Tab.

Для изменения высоты строк и ширины столбцов удобно использовать маркер границ в виде двойной стрелки, который виден при установке указателя мыши на границу.

Для удаления ячейки, строки или столбца их необходимо выделить, а затем выполнить команду контекстного меню «Удалить ячейки» и в выпадающем окне выбрать способ удаления.

Для удаления всей таблицы необходимо использовать команду «Вырезать» контекстного меню.

Редактор математических выражений Microsoft Equation ориентирован на создание объектов «формула» в тексте документа.

Для вызова редактора формул используется диалоговое окно Вставка объекта, вызываемое командой «Объект» на ленте вкладки Вставка. В диалоговом окне выбирается тип объекта Microsoft Equation.

При этом окно MS Word изменяется. Появляется строка меню редактора формул и панель инструментов Формула.

Порядок выполнения

1 Набрать и отформатировать текст в соответствии с заданием.
2 Создать таблицу в соответствии с заданием.
3 Набрать формулы в соответствии с заданием.

Контрольные вопросы

1 Структура окна приложения.
2 Форматирование текста.
3 Форматирование таблиц.
4 Набор формул.

5 Лабораторная работа № 5. Табличный процессор Excel

Цель работы: изучить структуру интерфейса MS Excel, ознакомиться с основными приемами работы в MS Excel, изучить работу с формулами.

Основные сведения

Табличный процессор MS Excel – это компьютерная программа для хранения и обработки информации, представленной в табличной форме.

С помощью MS Excel можно не только создавать таблицы, но и автоматизировать обработку данных.

Табличный процессор Excel является составной частью программного пакета MS Office.

Создаваемые в Excel файлы называются **рабочими книгами.** Рабочая книга состоит из нескольких типов листов.

После запуска Excel автоматически создает новую книгу с именем **Книга1.xlsx.**

Создание, сохранение и открытие книг в Excel аналогичны тем же действиям, что и в MS Word (см. пример в работе № 3).

Окна Excel имеют структуру, присущую большинству программ пакета MS Office.

Но структура окна Excel содержит следующие элементы, присущие именно этому приложению:

– строка формул;

– рабочая область листа;

– ярлыки листов с кнопками прокрутки для перехода по рабочим листам;

– строка состояния, в которой указываются режимы работы.

Управление и настройка элементов интерфейса аналогична описанной в работе № 3.

Рабочая книга – это файл, который хранится на диске и содержит один или несколько листов. По умолчанию рабочая книга имеет имя ***Книга1, 2***, … .

Рабочий лист – это созданная таблица для решения задачи, диаграмма, макрос, рисунок.

Имена листов находятся на ярлыках, расположенных в нижней части окна книги. Название текущего листа всегда выделено. Листы можно переименовывать, вставлять, удалять, перемещать или копировать в пределах одной книги или из одной книги в другую.

Для перехода с одного листа на другой необходимо указать соответствующий ярлык. Если ярлык нужного листа не виден, то для его поиска используется прокрутка ярлыков.

Таблица Excel содержит 16384 столбца. Строк в таблице – 1048576.

На пересечении строки и столбца находится ячейка. Каждая ячейка имеет уникальный адрес, в котором указываются имя столбца и номер строки, на пересечении которых она расположена.

Ячейка, где находится курсор, называется текущей, и с ней выполняются

определенные действия.

Диапазон (блок) ячеек – это прямоугольник, в котором указываются адреса ячеек левого верхнего и нижнего правого углов, разделенных двоеточием, например A1:D4. Если в выполняемом действии указан блок ячеек, то задействованы все его ячейки.

Выделение на рабочем листе. Выполнение большей части команд и задач в Excel становится возможным после выделения ячеек, с которыми необходимо произвести те или иные действия.

Очистка и удаление ячеек, строк и столбцов. Ячейки либо пусты, либо имеют значения. Можно удалять ячейки или очищать их содержимое.

При удалении ячейки Microsoft Excel так перемещает соседние ячейки, чтобы они заполнили освободившееся место.

При очистке ячейки удаляется ее содержимое (формулы и данные), форматы и примечания, а сама ячейка остается на листе.

Для ввода данных необходимо переместиться в нужную ячейку, набрать данные и нажать клавишу Enter, либо любым способом перейти в другую ячейку.

Для редактирования содержимого ячейки нужно выделить ячейку и нажать функциональную клавишу F2 или дважды щелкнуть по редактируемой ячейке.

Чтобы изменить формат ячеек, следует вызвать соответствующее диалоговое окно из контекстного меню командой «Формат ячеек».

В данном диалоговом окне доступно шесть закладок:

1) **число.** Здесь задается способ отображения числовых значений;

2) **выравнивание.** На этой вкладке можно управлять положением текста. Причем текст можно отображать вертикально или по диагонали под любым углом. Еще обратите внимание на раздел «Отображение». Очень часто используется функция «перенос по словам»;

3) **шрифт.** Задание стилевого оформления шрифтов, размера и цвета текста. Плюс режимы видоизменений;

4) **граница.** Здесь задаются стили и цвета оформления границ. Дизайн всех таблиц лучше оформлять именно здесь;

5) **заливка.** Название закладки говорит само за себя. Доступны для форматирования цвета, узора и способа заливки (например, градиентом с разным направлением штрихов);

6) **защита.** Здесь устанавливаются параметры защиты ячеек, которые активируются только после защиты целого листа.

Формулой в Excel называется последовательность символов (выражение), начинающихся со знака равенства =, и включающая постоянные значения (текст и числа), адреса ячеек (ссылки на ячейки), функции и знаки операций.

Функция – это запрограммированные формулы, позволяющие производить часто встречающиеся последовательности вычислений (стандартные функции разного назначения).

Порядок выполнения арифметических действий в формуле такой же, как принятый в математике.

Результатом работы формулы является новое значение, которое выводится как результат вычисления формулы по уже имеющимся данным.

По умолчанию правильно введенное число выравнивается по правому краю ячейки. Неправильно введенное число считается текстом и выравнивается по левому краю.

Кроме того, в произвольное место формулы можно с помощью кнопки **Мастер функции** вставить любую из многочисленных функций Excel.

В формулах MS Excel используют следующие виды ссылок на ячейки:

– относительные, например A1;

– абсолютные, например F1;

– смешанные, например $F1 или F$1);

– по имени.

По умолчанию Excel использует **относительные ссылки.** В этом случае при копировании ячейки с формулой адреса ячеек в формулах изменяются соответственно перемещению относительно начального положения.

Абсолютная ссылка – это ссылки на ячейки, адреса которых не меняются при копировании содержимого ячеек в любое место относительного начального положения, т. е. абсолютная ссылка в формуле всегда ссылается на ячейку, расположенную в определенном месте.

Смешанная ссылка – это ссылки на ячейки, адреса которых не меняются при копировании содержимого ячеек либо вдоль строк либо вдоль столбцов.

Ссылка по имени соответствует абсолютной адресации. Для ее использования необходимо присвоить ячейке или диапазону ячеек имя. Для этого во вкладке Формулы группы Определенные имена командой «Присвоить имя» либо командой «Присвоить имя из Контекстного меню» вызвать диалоговое окно Создание имени.

Порядок выполнения

1 Создать таблицу в соответствии с заданием.

2 Ввести формулу.

3 Распространить формулу используя мышь.

4 Построить график.

5 Решить систему линейных уравнений.

Контрольные вопросы

1 Абсолютные и относительные ссылки на ячейки.

2 Форматирование ячеек.

3 Как создать график.

4 Методы решения систем линейных уравнений.

6 Лабораторная работа № 6. Power Point

Цель работы: изучить работу в Power Point.

Основные сведения

PowerPoint (полное название – Microsoft Office PowerPoint) – это программа для создания и проведения презентаций, являющаяся частью Microsoft Office и доступная в редакциях для операционных систем Microsoft Windows и Mac OS. PowerPoint является частью Microsoft Office. Это позволило PowerPoint стать наиболее распространенной во всем мире программой для создания презентаций.

Пунктирные линии показывают области, в которые можно ввести текст или вставить другие объекты, такие как Объект, Таблица, Диаграмма, Рисунок, Формула или данные другого типа. Объекты, созданные в одном приложении (например, электронные таблицы), а затем связанные или внедренные в другом приложении, являются объектами OLE.

Вкладка Слайды содержит эскизы всех полноразмерных слайдов, отображаемых в области Слайд. После добавления других слайдов для появления нужного слайда в области Слайд можно щелкнуть соответствующий эскиз на вкладке Слайды. Можно также перетаскивать эскизы, чтобы изменить порядок слайдов в презентации. Кроме того, вкладка Слайды позволяет добавлять и удалять слайды.

Область Заметки позволяет ввести заметки о текущем слайде. Можно раздать заметки аудитории или обращаться к ним во время показа презентации в режиме докладчика. Режимы просмотра презентации PowerPoint предоставляет возможность работать и просматривать информацию в различных режимах. Режим выбирается с учетом вида выполняемых операций. Для установки нужного режима предназначена вкладка Вид.

Обычный режим является основным режимом редактирования, который используется для записи и создания презентации.

Режим сортировщика слайдов представляет слайды в виде эскизов. Заметки к слайдам можно ввести в области заметок, расположенной прямо под областью слайдов в обычном режиме. Однако, если нужно просматривать и работать с заметками в полноэкранном формате, щелкните Страницы заметок в группе Режимы просмотра презентации.

Режим показа слайдов занимает весь экран компьютера, имитируя реальную презентацию. В этом режиме презентация отображается так, как ее будет видеть аудитория. Можно посмотреть, как будут выглядеть рисунки, временные интервалы, видеофрагменты, эффекты анимации (Анимация. Добавление к тексту или объекту специального видео- или звукового эффекта. Например, можно создать элементы текстового списка, влетающие на страницу слева по одному слову, или добавить звук аплодисментов при открытии рисунка.) и эффекты перехода в реальной ситуации.

Создание презентации в Microsoft PowerPoint состоит из следующих эта-

пов: выбор общего оформления; добавление новых слайдов и их содержимого; выбор разметки слайдов; изменение при необходимости оформления слайдов; замена цветовой схемы; применение различных шаблонов оформления; создание эффектов анимации при демонстрации слайдов.

Порядок выполнения

1 Открыть PowerPoint.
2 Создать презентацию в соответствии с заданием.
3 Выйти из PowerPoint.

Контрольные вопросы

1 Понятие шаблона оформления.
2 В каких режимах можно просматривать слайд.
3 Виды форматирования слайдов.

7 Лабораторная работа № 7. Среда программирования VB.NET. Создание простого приложения

Цель работы: ознакомиться с визуальной средой программирования VB.NET. Освоить основные приемы работы в VB.NET.

Основные сведения

VB.NET это среда для разработки приложений на объектно-ориентированном языке программирования **VB.NET**.

Главными составными частями среды программирования VB.Net являются:

Конструктор Форм, Окно Редактора Текста, Панель инструментов, Обозреватель решений, Панель свойств.

Программисты на VB.Net проводят большинство времени, переключаясь между Дизайнером Форм и Окном Редактора (которое для краткости называют Редактор).

Наиболее часто используемые компоненты.

Label служит для отображения текста на экране. Вы можете изменить шрифт и цвет метки, если дважды щелкнете на свойство Font на Панели свойств. Вы увидите, что это легко сделать и во время выполнения программы, написав всего одну строчку кода.

TextBox – стандартный управляющий элемент Windows для ввода. Он может быть использован для отображения короткого фрагмента текста и позволяет пользователю вводить текст во время выполнения программы.

Button позволяет выполнить какие-либо действия при нажатии кнопки во время выполнения программы. В VB.Net все делается очень просто. Поместив Button на форму, Вы по двойному щелчку можете создать заготовку обработчи-

ка события нажатия кнопки. Далее нужно заполнить заготовку кодом, представленным ниже. Результат выполнения представлен на рисунке 7.1.

```
Public Class Form1
Private Sub Button1_Click(sender As Object,
e As EventArgs) Handles Button1.Click
Dim x, y As Double
x = CDbl(Me.TextBox1.Text)
y = Math.Exp(x) * Math.Sin(x)
Me.TextBox2.Text = CStr(y)
End Sub
End Class
```

Рисунок 7.1 – Результат выполнения программы

Порядок выполнения

1 Разработать блок-схему алгоритма в соответствии с заданием.
2 Создать форму приложения.
3 Набрать текст программы.

Контрольные вопросы

1 Свойства формы.
2 Основные компоненты, устанавливаемые на форму.
3 Использование математических функций.

8 Лабораторная работа № 8. Разработка приложений с использованием условного оператора if

Цель работы: изучить условный оператор if.

Основные сведения

Конструкция If ... Then проверяет истинность некоторого условия и в зависимости от результатов проверки выполняет определенный код:

```
Dim num1 As Integer = 10
Dim num2 As Integer = 9
If (num1 > num2) Then
Console.WriteLine("Число {0} больше числа {1}", num1, num2)
End If
```

Здесь проверяем, больше ли число num1 числа num2, и если num1 больше чем num2, то выводим сообщение.

Но, допустим, мы захотим, чтобы в случае невыполнения этого условия тоже совершалось какое-нибудь действие. Тогда мы добавляем выражение Else

и после него определяем те действия, которые будут совершаться, если программа не удовлетворяет условию в выражении If. Например, мы будем выводить сообщение, что первое число меньше второго:

```
Dim num1 As Integer = 10
Dim num2 As Integer = 9
If (num1 > num2) Then
Console.WriteLine("Число {0} больше числа {1}", num1, num2)
Else
Console.WriteLine("Число {0} меньше или равно числу {1}", num1, num2)
End If
```

Текст программы для определения, в какой четверти находится точка, представлен ниже. Результат выполнения представлен на рисунок 8.1.

```
Public Class Form1
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
Dim X, Y As Double
X = CDbl(Me.KX.Text)
Y = CDbl(Me.KY.Text)
If (X > 0) And (Y > 0) Then Me.Результат.Text = "Точка находится в I четверти"
If (X < 0) And (Y > 0) Then Me.Результат.Text = "Точка находится в II четверти"
If (X < 0) And (Y < 0) Then Me.Результат.Text = "Точка находится в III четверти"
If (X > 0) And (Y < 0) Then Me.Результат.Text = "Точка находится в IV четверти"
End Sub
End Class
```

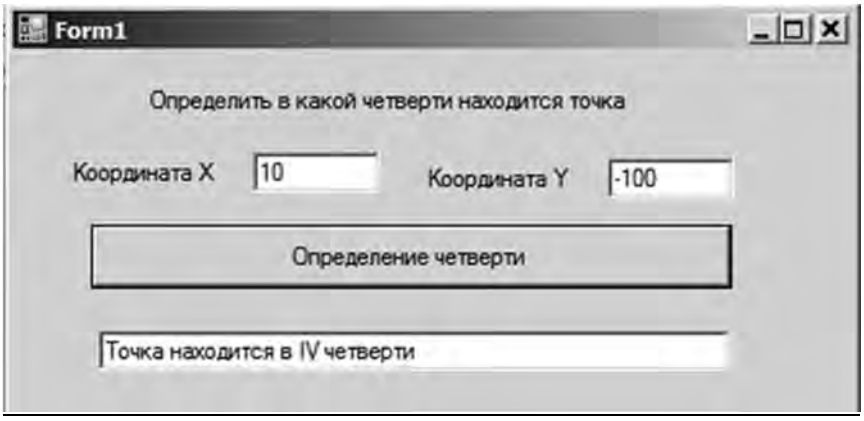

Рисунок 8.1 – Результат выполнения программы

Порядок выполнения

1 Разработать блок-схему алгоритма в соответствии с заданием.
2 Создать форму приложения.
3 Набрать текст программы.

Контрольные вопросы

1 Конструкция оператора if.
2 Задание условий.
3 С какими типами работает оператор if?

9 Лабораторная работа № 9. Разработка приложений с использованием оператора выбора case

Цель работы: изучить оператор выбора case.

Основные сведения

Конструкция Select Case подобна в конструкции If ... Then, т. к. позволяет обрабатывать сразу несколько условий. После слов Select Case указывается сравниваемое выражение. Значение этого выражения последовательно сравнивается со значениями, помещенными после оператора Case. И в случае, если значения совпали, то выполняется блок команд, помещенных после данного оператора Case. Конструкция завершается словами End Select. Если мы хотим определить действия, которые будут выполняться, если совпадений не выявлено, то мы можем использовать оператор Case Else, после которого помещаем блок действий по умолчанию. Блок Case Else необязателен и может не употребляться.

```
Dim num1 As Integer = 10
Select Case num1
   Case 1
      Console.WriteLine("Число num1 равно 1")
   Case 2
      Console.WriteLine("Число num1 равно 2")
   Case 3 To 25
      Console.WriteLine("Число num1 находится на отрезке от 3 до 25")
   Case Else
      Console.WriteLine("Число num1 больше 25")
End Select
```

Num1 должен быть равен одному их простых типов данных (Boolean, Byte, Char, Date, Double, Decimal, Integer, Long, Object, SByte, Short, Single, String, UInteger, ULong и UShort).

Ниже представлена программа для определения дня недели. Результат выполнения представлен на рисунке 9.1.

```
Public Class Form1
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
Dim N As Integer
Dim Naz As String
N = CInt(Me.Номер.Text)
Select Case N
Case 1
Naz = "Понедельник"
Case 2
Naz = "Вторник"
Case 3
Naz = "Среда"
Case 4
Naz = "Четверг"
Case 5
Naz = "Пятница"
Case 6
Naz = "Суббота"
Case 7
Naz = "Воскресенье"
Case Else
Naz = "Введите правильно номер дня недели"
End Select
Me.Название.Text = Naz
End Sub
End Class
```

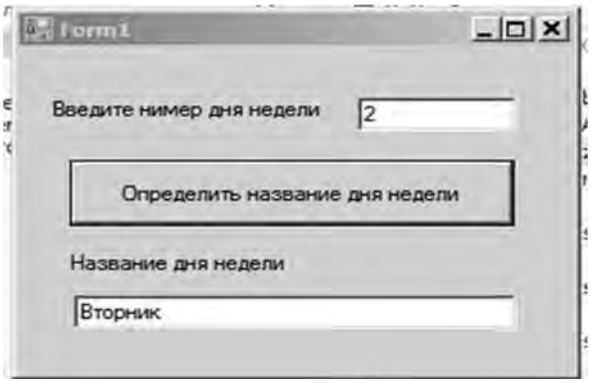

Рисунок 9.1 – Результат выполнения программы

Порядок выполнения

1 Разработать блок-схему алгоритма в соответствии с заданием.
2 Создать форму приложения.
3 Набрать текст программы.

Контрольные вопросы

1 Конструкция оператора Case.
2 Задание условий.
3 С какими типами работает оператор Case?

10 Лабораторная работа № 10. Создание нескольких форм в приложении

Цель работы: ознакомиться с визуальной средой программирования VB.NET – работа с несколькими формами.

Основные сведения

Поставить курсор на решение, щелкнуть правой клавишей и в контекстном меню выбрать Добавить\Существующий проект\<Имя_проекта>. В окне Обозреватель решений добавится проект.

Далее с использованием оператора Imports осуществляем импорт пространства имен или элементов программирования, определенных в текущем проекте. В тексте программы объявляем переменные типа Form вызываемых форм. Результат выполнения представлен на рисунке 10.1.

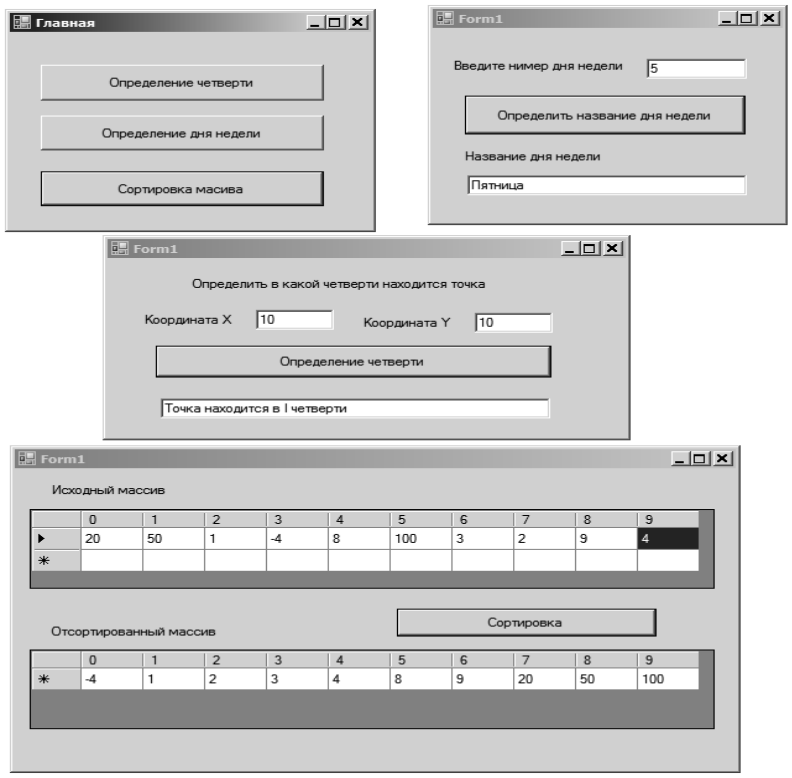

Рисунок 10.1 – Результат выполнения программы

```
Imports Опеделение_четверти
Imports Определение_дня_недели_по_номеру
Imports Сортировка_массива
Public Class Главная
Private Sub _IF_Click(sender As Object, e As EventArgs) Handles _IF.Click
Dim F1 As New Опеделение_четверти.Form1
F1.Show()
End Sub
```

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
_CASE.Click
Dim F2 As New Определение_дня_недели_по_номеру.Form1
F2.Show()
End Sub
Private Sub Массив_Click(sender As Object, e As EventArgs) Handles
Массив.Click
Dim F3 As New Сортировка_массива.Form1
F3.Show()
End Sub
End Class
```

Порядок выполнения

1 Разработать блок-схему алгоритма в соответствии с заданием.
2 Создать форму приложения.
3 Подключить проекты.
4 Набрать текст программы.

Контрольные вопросы

1 Порядок подключения проектов.
2 Создание объектной переменной требуемой формы подключенного проекта.
3 Визуализация требуемой формы подключенного проекта.

11 Лабораторная работа № 11. Создание приложения с использованием циклических алгоритмов

Цель работы: изучить основные операторы и конструкции языка VB.NET для программирования циклических алгоритмов.

Основные сведения

Цикл For ... Next

В цикл выполняется определенное число раз, причем это число задается счетчиком:

```
For i = 1 To 9
   Console.WriteLine("Квадрат числа {0} равен {1}", i, i * i)
Next
```

Здесь переменная i выполняет роль счетчика. После слова To мы помещаем максимальное значение счетчика. При каждом цикле значение счетчика увеличивается на единицу. И это значение сравнивается со значением после To.

Если эти два значения равны, то цикл прекращает свою работу.

При работе с циклами мы можем увеличивать значение счетчика при каждом проходе не только на единицу, но и вообще на любое число. Для этого нужно либо использовать ключевое слово Step и после него указать шаг цикла, на который будет увеличиваться значение счетчика, либо можно увеличивать счетчик непосредственно в цикле.

Текст программы для вычисления факториала представлен ниже. Результат выполнения представлен на рисунке 11.1.

```
Public Class Form1
Private Sub Button1_Click(sender As Object, e
As EventArgs) Handles Button1.Click
Dim n, i, R As Integer
n = CInt(Me.TextBox1.Text)
R = 1
For i = 1 To n
R = R * i
Next i
Me.TextBox2.Text = CStr(R)
End Sub
End Class
```

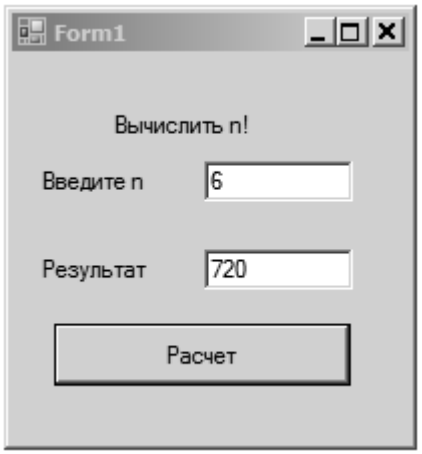

Рисунок 11.1 – Результат выполнения программы

Цикл Do выполняется, пока соблюдается определенное условие. Однако он имеет разные формы. Так, в следующем примере сначала проверяется условие, а затем выполняется блок кода, определенный в цикле:

```
Dim j As Integer = 10
Do While j > 0
   Console.WriteLine(j)
   j -= 1
Loop
```

В данном случае цикл выполняется, пока значение j больше нуля. Если изначально условие, заданное в цикле, неверно, то цикл не будет работать.

Но мы можем определить проверку в конце цикла, и таким образом, наш цикл как минимум один раз отработает:

```
Do
   Console.WriteLine(j)
   j -= 1
Loop While j > 0
```

Текст программы вычисления суммы ряда представлен ниже. Результат выполнения – на рисунке 11.2.

```
Public Class Form1
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
Dim S As Double = 0
Dim S1 As Double = 0
Dim n As Integer
Dim k As Integer = 0
Do
S1 = S
S = 0
k = k + 1
For n = 1 To k
S = S + 20 / (5 * (n ^ 2) + 10 * n + 15)
Next n
Loop Until Math.Abs(S - S1) <= 0.001
Me.TextBox1.Text = CStr(S)
Me.TextBox2.Text = CStr(k)
End Sub
End Class
```

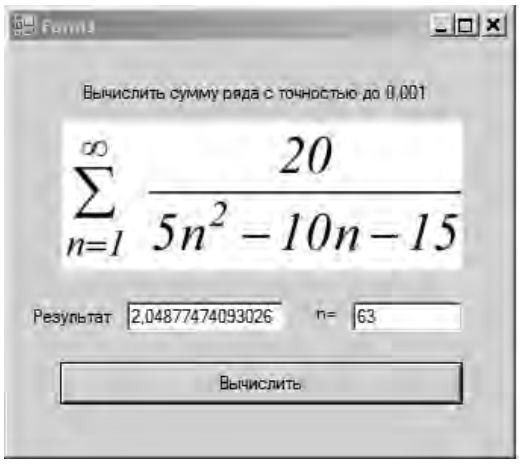

Рисунок 11.2 – Результат выполнения программы

Порядок выполнения

1 Разработать блок-схему алгоритма в соответствии с заданием.
2 Создать форму приложения.
3 Набрать текст программы.

Контрольные вопросы

1 Конструкция оператора for.
2 Конструкция оператора do.
3 С какими типами работают операторы for и do?

12 Лабораторная работа № 12. Создание приложения для вычисления численными методами определенного интеграла

Цель работы: изучить использование основных операторов и конструкций языка VB.NET для программирования циклических алгоритмов при решении задач численными методами.

Основные сведения

Многие инженерные задачи, задачи физики, геометрии и многих других областей человеческой деятельности приводят к необходимости вычислять

определенный интеграл.

Пусть требуется вычислить с заданной точностью значение интеграла функции $f(x)$ на отрезке $[a; b]$.

Найти значение определенного интеграла – это значит найти площадь заштрихованной области. Известно много способов вычисления подобных интегралов либо с помощью первообразной, либо используя приближенные методы. Вычисление интеграла будем производить, используя приближенный метод – метод прямоугольников.

При вычислении интеграла следует помнить, каков геометрический смысл определенного интеграла. Если $f(x) >= 0$ на отрезке $[a; b]$, то он численно равен площади фигуры, ограниченной графиком функции $y = f(x)$, отрезком оси абсцисс, прямой $x = a$ и прямой $x = b$ (рисунок 12.1) Таким образом, вычисление интеграла равносильно вычислению площади криволинейной трапеции.

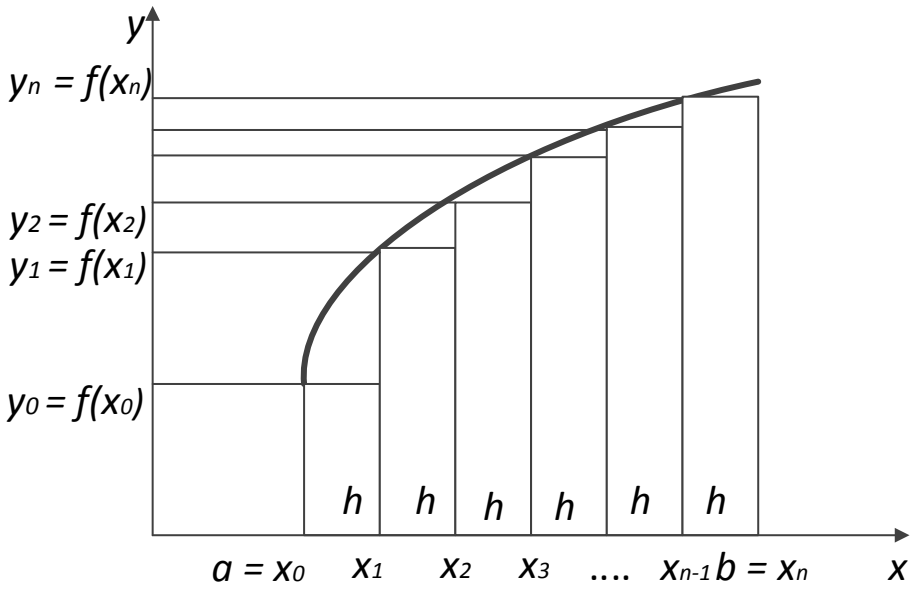

Рисунок 12.1 – Вычисление интеграла с помощью метода прямоугольников.

Разделим отрезок $[a; b]$ на n равных частей, т. е. на n элементарных отрезков. Тогда длина каждого элементарного отрезка

$$h = (b - a) / n.$$

Порядок решения задачи на ЭВМ следующий.

Как правило в первом приближении отрезок интегрирования разбивают на n частей (можно начать с двух), значение интеграла, вычисленное при n-разбиений обозначим через S_1.

Одно это приближение не позволяет оценить точность, с которой вычислено значение интеграла, необходимо найти второе приближение. Для этого увеличим n в 2 раза. Значение интеграла, вычисленное при n-разбиений обозначим через через S_2.

Далее, чтобы установить, достигли мы заданной точности ε или нет,

проверим условие

$$|S_1 - S_2| \le \varepsilon . \qquad (12.1)$$

Далее действуем по алгоритму:

1) если условие выполняется (true, истинно), то S_2 принимается за искомое значение интеграла;

2) если не выполняется (false, ложно), то последнее полученное значение S_2 считается предыдущим, т. е. $S_1 = S_2$ и удвоим число точек деления отрезка $[a; b]$, $n = 2n$;

3) вычислим новое значение S_2.

4) процесс удвоения n и вычисления S_2 и замены его с S_1 будем продолжать до тех пор, пока условие (12.1) не станет истинным.

Текст программы для вычисления интеграла представлен ниже. Результат выполнения – на рисунке 12.2.

```
Public Class Form1
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click

Dim y As Single, i As Integer
Dim x, dx As Single
Dim a As Single, b As Single, S As Single, N As Integer
Dim S1 As Double
a = CDbl(Me.НП.Text)
b = CDbl(Me.ВП.Text)
S = 0
S1 = 0
N = 1
Do
S1 = S
S = 0
x = a
N = N * 2
dx = (b - a) / N
For i = 1 To N
y = (x ^ 2) / (1 + x)
S = S + y * dx
x = x + dx
Next i
Loop Until Math.Abs(S1 - S) <= 0.0001
Me.Результат.Text = CStr(S)

End Sub
End Class
```

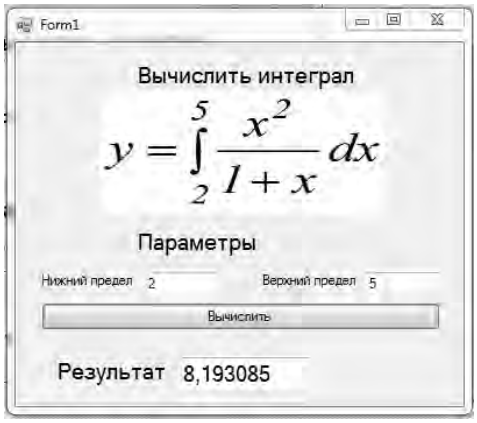

Рисунок 12.2 – Результат выполнения программы

Порядок выполнения

1 Разработать блок-схему алгоритма в соответствии с заданием.
2 Создать форму приложения.
3 Набрать текст программы.

Контрольные вопросы

1 Каков геометрический смысл определенного интеграла?
2 Порядок вычисления определенного интеграла на ЭВМ.
3 Понятие метода прямоугольников.

13 Лабораторная работа № 13. Создание приложения для создания и редактирования табличных данных

Цель работы: изучить работу с табличными данными, используя циклические алгоритмы.

Основные сведения

Объект **DataGridView** предназначен для отображения всей информации из таблиц, запросов или фильтров на форме в виде таблицы. Этот объект может быть создан как вручную (с последующим его подключением), так и перетаскиванием всего источника данных из окна **Data Sources**. Однако наиболее часто его создают перетаскиванием всей таблицы, запроса или фильтра из окна **Data Sources** на форму.

При перетаскивании этого объекта на форму, как и в случае с другими объектами, появляется панель навигации. Она выполняет функции: перемещение по записям, добавление, удаление и сохранение записей. После создания объекта **DataGridView** можно настраивать как свойства всего объекта, так и свойства отдельных столбцов. Начнём с настройки свойств всего объекта. Настройка данных свойств осуществляется в основном через меню действий.

Если в меню действий выбрать пункт **Edit Columns**, то появляется окно, где можно добавлять, удалять и редактировать столбцы. Для этого в списке столбцов левой части окна выбираем столбец, а в правой – настраиваем его свойства. Наиболее часто настраиваются следующие свойства:

– **Name** – имя столбца;
– **DataPropertyName** – имя, отображающего в столбце поля;
– **HeaderText** – текст заголовка столбца;
– **Width** – ширина поля;
– **ReadOnly** – блокировка столбца для редактирования данных;
– **Resizable** – разрешает менять ширину столбца;
– **SortMode** – сортировка данных в таблице по этому столбцу;

– **ToolTipText** – всплывающая подсказка для столбца;

– **Visible** – делает столбец невидимым.

Для добавления новых столбцов в окне **Edit Columns** необходимо нажать кнопку **Add**, а для удаления кнопку **Remove**.

Если необходимо настроить внешний вид всех ячеек таблицы, то для этого необходимо выделить объект **DataGridView** и на панели свойств зайти в свойство **DefaultCellStyle**. Появится окно со свойствами всех ячеек таблицы.

Доступ к отдельным ячейкам таблицы можно получить через подобъект **Item**. Обращение к нему осуществляется следующим образом:

DataGridView.Item(i, j).<Свойство>

Здесь DataGridView – это имя объекта, i – горизонтальная координата ячейки, а j – вертикальная, <Свойство> – это настраиваемое свойство ячейки.

Пример – В верхнюю левую ячейку таблицы записать слово «Привет» и сделать цвет текста в ячейке красным.

```
DataGridView.Item(0, 0).Value = "Привет"
DataGridView.Item(0, 0).Style.ForeColor = Color.Red
```

Здесь DataGridView – это имя объекта, свойство Value определяет содержимое ячейки таблицы, свойство Style.ForeColor определяет цвет текста в ячейке. Нумерация столбцов и строк в таблице начинается с нуля.

Поместить на форму компонент DataGridView, свойству Name присвоить значение Таблица и добавить три столбца с именами и подписями Стб1, Стб2 и Стб3. Поместить элементы Button и TextBox. Свойству TextBox Name присвоить Сумма (рисунок 13.1).

Ниже представлен текст программы. Результат – на рисунке 13.2.

```
Public Class Form1
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
Dim КолСтб, КолСтр, i, j As Integer
Dim Сум As Double = 0
КолСтб = Me.Таблица.ColumnCount
КолСтр = Me.Таблица.RowCount - 1
For i = 0 To КолСтб - 1
For j = 0 To КолСтр - 1
Сум = Сум + CDbl(Таблица(i, j).Value)
Next j
Next i
Сумма.Text = CStr(Сум)
End Sub
End Class
```

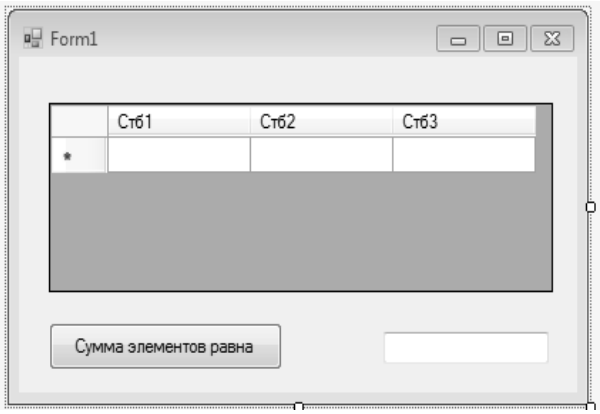

Рисунок 13.1– Размещение элементов на форме

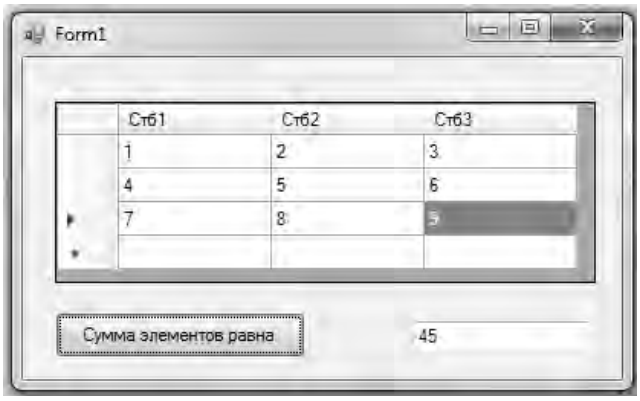

Рисунок 13.2 – Результат выполнения программы

Порядок выполнения

1 Разработать блок-схему алгоритма в соответствии с заданием.
2 Создать форму приложения.
3 Набрать текст программы.

Контрольные вопросы

1 Назначение объекта DataGridView.
2 Свойства объекта DataGridView.
3 Методы ввода текста в ячейки.

14 Лабораторная работа № 14. Создание приложения для ввода-вывода массивов и их сортировки

Цель работы: изучить работу с массивами, используя циклические алгоритмы для их сортировки.

Основные сведения

Массив представляет собой набор данных одного типа. Кроме размера массив характеризуется таким понятием, как размерность (dimension). Например:

– одномерный массив Dim num1 As Integer(5);
– двухмерный массив Dim num2 As Integer(2,2).

Под сортировкой массива подразумевается процесс перестановки элементов массива, целью которого является размещение элементов массива в определенном порядке. Существует много методов (алгоритмов) сортировки массивов. Рассмотрим метод прямого выбора.

Алгоритм сортировки массива по возрастанию методом прямого выбора:

1) сравнить первый элемент со вторым, найти минимум;

2) записать минимум в буфер, во второй элемент значение первого, а в первый значение буфера;

3) сравнить первый элемент с третьим, найти минимум;

4) записать минимум в буфер, в третий элемент значение первого, а в первый значение буфера;

5) и так далее до предпоследнего элемента.

Ниже представлен текст программы.

```
Public Class Form1
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
Dim Масс(9) As Integer
Dim i, j As Integer
Dim Буфер As Integer
For i = 0 To 9
Масс(i) = CInt(Me.ИсхМассив.Item(i, 0).Value)
Next i
For j = 0 To 9
For i = j To 9
If Масс(j) > Масс(i) Then
Буфер = Масс(i)
Масс(i) = Масс(j)
Масс(j) = Буфер
End If
Next i
Next j
For i = 0 To 9
Me.СортМассив.Item(i, 0).Value = CStr(Масс(i))
Next i
End Sub
End Class
```

Форма приложения после выполнения приведена на рисунке 14.1.

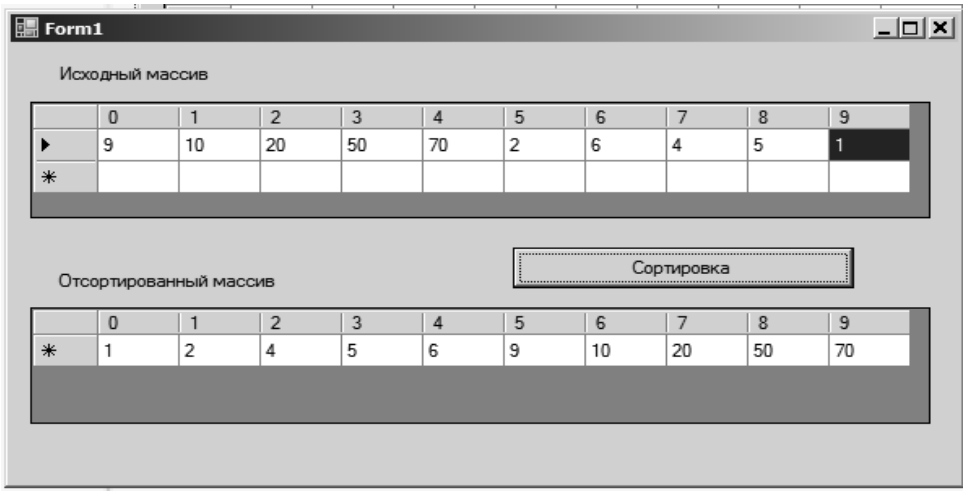

Рисунок 14.1 – Результат выполнения программы

Порядок выполнения

1 Разработать блок-схему алгоритма в соответствии с заданием.
2 Создать форму приложения.
3 Набрать текст программы.

Контрольные вопросы

1 Размерность массива.
2 Создание массива.
3 Понятие сортировки массива методом прямого выбора.

15 Лабораторная работа № 15. Создание приложений с использованием графики

Цель работы: ознакомиться с графическими возможностями визуальной среды программирования VB.NET.

Основные сведения

Рисовать можно с помощью методов класса Graphics. Рассмотрим следующие методы:

– **DrawLine(Pen, Point1, Point2)** – проводит линию, соединяющую две структуры Point;

– **Point** – структура. Представляет упорядоченную пару целых чисел – координат X и Y, определяющую точку на двумерной плоскости;

– параметр **Pen** – задает цвет линии. **Pens.Black** – черный, **Pens.White** – белый;

– **DrawEllipse(Pen, Rectangle)** – рисует эллипс, определяемый ограничивающей структурой **Rectangle** (рисунок 15.1, *а*);

– **Rectangle** задает прямоугольник с координатами верхнего левого угла, высоты и ширины;

– **DrawArc(Pen, Rectangle, Nangle, Angle)** – рисует дугу (рисунок 15.1, *а*). Дуга является частью эллипса. Параметры DrawArc метода не отличаются от параметров **DrawEllipse** метода, за исключением того, что **DrawArc** требует начальный угол (**Nangle**) и угол поворота (**Angle**). В следующем примере рисуется дуга с начальным углом 0° и угол поворота 225° по часовой стрелке:

Текст программы, рисующей графические примитивы, представлен ниже. Результат выполнения на рисунке 15.2.

а) *б*)

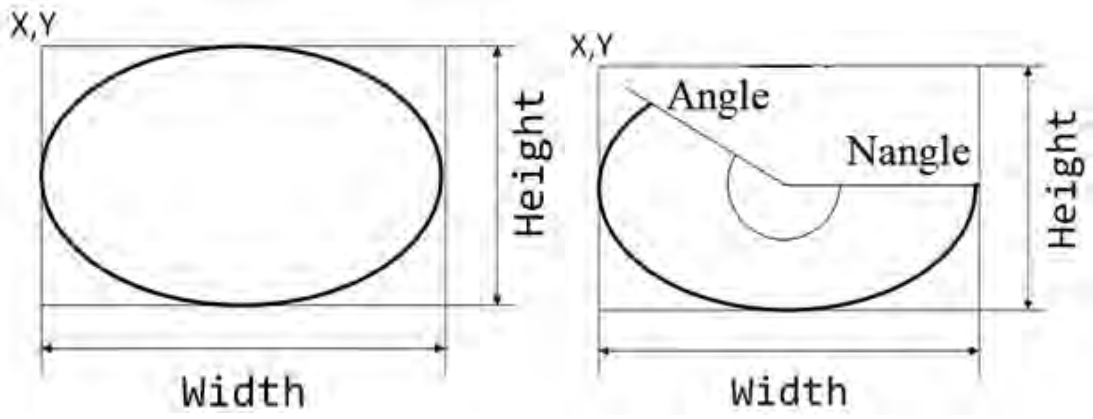

Рисунок 15.1 – Изображения эллипса и дуги

```
Public Class Form1
Private Sub Рисование_Click(sender As Object, e As EventArgs) Handles Рисование.Click
Dim Рисунок As Graphics
Dim Прямоугольник As Rectangle
Dim N, K, S As Point
Рисунок = Полотно.CreateGraphics
Прямоугольник.X = 10
Прямоугольник.Y = 10
Прямоугольник.Width = 100
Прямоугольник.Height = 100
Рисунок.DrawRectangle(Pens.Black, Прямоугольник)
Рисунок.DrawEllipse(Pens.Black, Прямоугольник)
Прямоугольник.X = 130
Прямоугольник.Y = 10
Прямоугольник.Width = 100
Прямоугольник.Height = 100
Рисунок.DrawRectangle(Pens.Black, Прямоугольник)
Рисунок.DrawArc(Pens.Black, Прямоугольник, 0, 225)
N.X = 10
N.Y = 130
K.X = 110
K.Y = 230
Рисунок.DrawLine(Pens.Black, N, K)
N.X = 180
N.Y = 130
K.X = 230
K.Y = 230
S.X = 130
S.Y = 230
Рисунок.DrawLine(Pens.Black, N, K)
```

```
Рисунок.DrawLine(Pens.Black, S, K)
Рисунок.DrawLine(Pens.Black, N, S)
End Sub
End Class
```

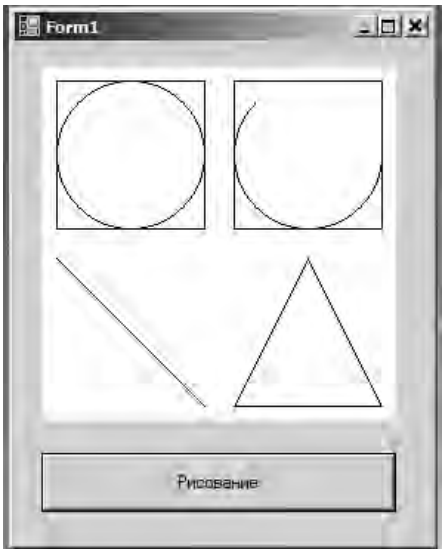

Рисунок 15.2 – Результат выполнения программы

Порядок выполнения

1 Разработать блок-схему алгоритма в соответствии с заданием.
2 Создать форму приложения.
3 Набрать текст программы.

Контрольные вопросы

1 Основные методы класса Graphics.
2 Понятие структуры Point.
3 Понятие структуры Rectangle.

16 Лабораторная работа № 16. Создание приложений с использованием объектов

Цель работы: ознакомиться с визуальной средой программирования VB.NET.

Основные сведения

Visual Basic.NET является полноценным объектно-ориентированным языком, а это значит, что программа может быть представлена в виде взаимосвязанных объектов, которые взаимодействуют между собой. Описанием объекта является класс, в то время как объект – экземпляр этого класса. Класс опреде-

ляется с помощью ключевого слова Class:

```
Class Book
Поля и методы
End Class
```

Всю функциональность класса обеспечивают его члены – поля, свойства, методы, конструкторы, события. Поля представляют обычные переменные, обычным образом также определяются процедуры и функции (в этом плане классы во многом похожи на структуры):

```
Public Class TLine
Public Im As Graphics = Form1.Pol
Public MoveTo As Point
Public Sub RisLine(ByVal P As Point, ByVal Reg As Byte)
If Reg = 1 Then
Im.DrawLine(Pens.Black, MoveTo, P)
MoveTo = P
Else
Im.DrawLine(Pens.White, MoveTo, P)
MoveTo = P
End If
End Sub
End Class
```

Создание экземпляра производится следующей командой:

```
Dim Line  As New TLine().
```

Доступ к полям и методам осуществляется следующим образом:
Имя_ класса.Имя_поля или Имя_ класса.Имя_метода
Например:

```
Line.MoveTo=Нач_Точка
Line. RisLine(P1, 1)
```

Одним из ключевых аспектов объектно-ориентированного программирования является наследование (inheritance). Сущность наследования заключается в том, что мы можем расширить функционал уже имеющихся классов с помощью создания наследников этого класса. Чтобы наследовать один класс от другого, нужно использовать ключевое слово Inherits:

```
Public Class TPrugina
Inherits Tline          'Tprugina наследует класс Tline
Protected P1 As Point
Protected PruginaEnd As Point
Private PointArr(0 To 10) As Point
Public Sub Nach(ByVal P As Point)
```

```
MoveTo.X = P.X – 10      ‘Использование MoveTo класса Tline
MoveTo.Y = P.Y
P1.X = P.X + 10
P1.Y = P.Y
RisLine(P1, 1)        ‘Использование RisLine класса Tline
MoveTo = P
P1.X = P.X
P1.Y = P.Y + 10
RisLine(P1, 1)
End Sub
Public Sub PruginaRis(ByVal P As Point, ByVal dY As Integer, ByVal Reg As Byte)
Dim i, os As Integer
PointArr(1).X = P.X - 10
PointArr(1).Y = P.Y + 10
For i = 2 To 10
os = i Mod 2
If os = 0 Then
PointArr(i).X = PointArr(i - 1).X + 20
Else
PointArr(i).X = PointArr(i - 1).X - 20
End If
PointArr(i).Y = PointArr(i - 1).Y + dY
Next i
Nach(P)
For i = 1 To 10
RisLine(PointArr(i), Reg)
Next i
P1.X = PointArr(10).X - 10
P1.Y = PointArr(10).Y
RisLine(P1, Reg)
PruginaEnd.X = PointArr(10).X - 10
PruginaEnd.Y = PointArr(10).Y + 10
RisLine(PruginaEnd, Reg)
End Sub

End Class
```

По умолчанию все классы могут наследоваться. Однако здесь есть ряд ограничений:

– во-первых, не поддерживается двойное наследование, класс может наследоваться только от одного класса. Хотя проблема множественного наследования реализуется с помощью концепции интерфейсов, о которых мы поговорим позже;

– во-вторых, при создании производного класса мы должны учитывать тип

доступа к базовому классу – тип доступа к производному классу должен быть таким же, как и у базового класса, или более строгим. То есть если базовый класс у нас имеет тип доступа Friend, то производный класс может иметь тип доступа Friend или Private, но не Public.

Полиморфизм – расширение принципа наследования в объектно-ориентированном программировании.

Полиморфизм – наиболее туманный принцип ООП. Суть его состоит в том, что классы-потомки могут иметь методы с тем же самым названием, что и в родительском классе, но при этом выполнять другие действия, специфические для каждого конкретного класса.

Для этого нам надо в базовом классе пометить изменяемый метод модификатором **Overridable**. А уже в производном классе этот же метод использовать с модификатором **Overrides**:

```
Public Class TMehanizm1
Inherits Tprugina            ‘Наследуем класс Tprugina
‘Разрешаем методу GruzRis перезапись
Public Overridable Sub GruzRis(ByVal P As Point, ByVal Reg As Byte)
P1.X = P.X - 20
P1.Y = P.Y
RisLine(P1, Reg)
P1.X = P.X - 20
P1.Y = P.Y + 20
RisLine(P1, Reg)
P1.X = P.X + 20
P1.Y = P.Y + 20
RisLine(P1, Reg)
P1.X = P.X + 20
P1.Y = P.Y
RisLine(P1, Reg)
RisLine(P, Reg)
End Sub

Public Class TMehanizm2
Inherits TMehanizm1        ‘Наследуем класс TMehanizm1
‘Перезаписываем метод GruzRis, т.е. меняем его содержимое
Public Overrides Sub GruzRis(ByVal P As Point, ByVal Reg As Byte)
P1.X = P.X - 20
P1.Y = P.Y + 20
RisLine(P1, Reg)
P1.X = P.X
P1.Y = P.Y + 40
RisLine(P1, Reg)
P1.X = P.X + 20
P1.Y = P.Y + 20
```

```
RisLine(P1, Reg)
RisLine(P, Reg)
End Sub
End Class

Public Class Form1
Dim Vst As Point
Public Graf, Pol As Graphics
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
Graf = Me.CreateGraphics                ‘Для рисования на форме
Pol = PB.CreateGraphics
Dim Mehanizm1 As TMehanizm1 = New TMehanizm1
Dim Mehanizm2 As New TMehanizm2()
Dim Mehanizm3 As New TMehanizm3
Dim Mehanizm4 As New TMehanizm4
Mehanizm4.Im = Graf                     ‘Для рисования на форме
Pol.Clear(Color.White)
Graf.Clear(Color.FromKnownColor(KnownColor.Control))
Application.DoEvents()
Vst.X = 50
Vst.Y = 20
Mehanizm1.Rabota(Vst)
Vst.X = 100
Vst.Y = 20
Mehanizm2.Rabota(Vst)
Vst.X = 150
Vst.Y = 20
Mehanizm3.Rabota(Vst)
Vst.X = 200
Vst.Y = 20
Mehanizm4.Rabota(Vst)
End Sub
End Class
```

Результат выполнения программы показан на рисунке 16.1.

Порядок выполнения

1 Разработать блок-схему алгоритма в соответствии с заданием.
2 Создать форму приложения.
3 Набрать текст программы.

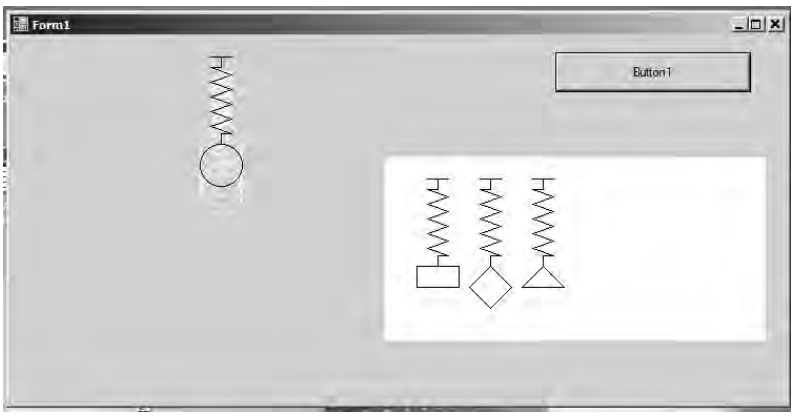

Рисунок 16.1 – Результат выполнения программы

Контрольные вопросы

1 Понятие типа «класс».
2 Понятие инкапсуляции, наследования и полиморфизма.

17 Лабораторная работа № 17. Работа в СУБД Access

Цель работы: ознакомится с системой создания и управления базами данных Access, получить навыки работы с объектами баз.

Основные сведения

Системы управления базами данных (СУБД) – это программные средства, с помощью которых можно создавать базы данных, наполнять их и работать с ними. С помощью Access обычные пользователи получили удобное средство для создания и эксплуатации достаточно мощных баз данных без необходимости что-либо программировать. В то же время работа с Access не исключает возможности программирования. При желании систему можно развивать и настраивать собственными силами.

После создания новой базы данных на экране появляется диалоговое окно Имя: база данных. Это диалоговое окно состоит из нескольких вкладок: *Таблицы*, *Запросы*, *Формы*, *Отчеты*, *Макросы*, *Модули*. Каждая из вкладок посвящена одной составной части или объекту базы данных. В этом окне, которое является центральным пультом управления базой данных, можно выбрать, с каким объектом мы собираемся работать. На каждой вкладке, нажав кнопку Создать, можно создать новый объект соответствующей категории. Кнопка Открыть предназначена для открытия имеющихся объектов, например, для ввода в них данных. Кнопка Конструктор позволяет отредактировать объект активной вкладки.

Таблица – это набор сведений определенной категории, состоит из столбцов (полей) и строк (записей).

Запись – это набор данных об одном объекте в таблице.

Поле – некоторая определенная категория информации в каждой записи. Поле – это минимальная единица информации.

Запрос – это вопрос, с которым мы обращаемся к таблице. Повторяя этот запрос в разное время, мы будем получать разные ответы – в зависимости от состояния данных, включенных в таблицы, на основании которых формируется запрос.

Форма – это средство для работы с базами данных. Формы – электронный аналог бумажного бланка, используемый для внесения в базу данных информации, ее изменения и просмотра на экране. Их можно использовать для ввода, редактирования и просмотра информации, а можно и распечатывать. В большинстве случаев поля формы соответствуют полям используемой таблицы.

Отчет – используется для распечатки информации из базы данных, включая вычисления по большому объему данных.

Таблицы состоят из столбцов и строк. В программе Access столбцы называются полями, которым для идентификации присваиваются уникальные имена. Имена полей отображаются в заголовках столбцов. В строках располагаются взаимосвязанные данные, распределенные по соответствующим полям. Набор информации одной строки называется записью данных.

При создании базы данных необходимо тщательно продумать ее структуру. Разумеется, можно внести изменения в структуру базы и после ее создания, но правильное построение базы данных и таблиц с самого начала значительно сэкономит время и усилия.

Для создания таблицы в окне базы данных активизируем вкладку Таблицы и нажмем на кнопку Создать. В появившемся диалоговом окне Новая таблица предлагается выбор различных режимов создания таблицы: *Режим таблицы*, *Конструктор*, *Мастер таблиц*, *Импорт таблиц* и *Связь с таблицами*. Для изучения процедуры создания таблицы наиболее подходит режим Конструктор, т. к. именно в этом режиме задается структура таблицы. Итак, выберем режим Конструктор и нажмем на кнопку ОК. Открывается окно Таблица.

В режиме Конструктора вводятся имена полей и задаются типы данных полей, но не вводятся пока никакие данные. Речь идет только о создании структуры таблицы. По этой структуре Access сформирует таблицу, в которую можно будет ввести данные.

По окончании ввода структуры таблицы нажимаем на кнопку Вид на панели инструментов, чтобы перейти в режим заполнения таблицы. Access попросит сохранить таблицу и назначить ей имя.

В большинстве случаев все данные не хранятся в одной таблице. Обычно информацию разбивают на несколько таблиц, а затем устанавливают между ними связи.

Например, в одной таблице содержится адрес клиента, а в другой – данные о кредите. Эти таблицы должны быть связаны, т. к. касаются одного клиента.

По такому принципу строится реляционная база данных. Информация разбивается на несколько таблиц, а затем между ними устанавливаются связи с помощью совпадающих полей.

В базе данных можно устанавливать несколько типов связей. В случае связи с отношением один-к-одному каждой записи в одной таблице соответствует

одна запись в другой. Так, например, для каждой записи с адресом клиента существует одна запись, касающаяся данных о его кредите.

Одна запись в какой-либо таблице может быть связана с одной или несколькими записями в другой. Такой тип отношений называется один-ко-многим.

Можно просмотреть связи, установленные между таблицами в базе данных. Находясь в окне базы данных, щелкнем на кнопке *Схема данных,* третьей справа на панели инструментов. Access открывает окно с тем же названием.

Порядок выполнения

1 Создать таблицу в соответствии с заданием.
2 Создать запрос.
3 Создать форму.

Контрольные вопросы

1 Понятие таблицы.
2 Понятие запроса.
3 Понятие формы.

18 Лабораторная работа № 18. Поиск информации в компьютерной сети с соблюдением правил защиты информации

Цель работы: научиться использовать поисковые службы Интернет и поисковые серверы WWW для поиска необходимой информации.

Основные сведения

Поиск информации в Интернете осуществляется с помощью специальных программ, обрабатывающих запросы – информационно-поисковых систем (ИПС). Существует несколько моделей, на которых основана работа поисковых систем, но исторически две модели приобрели наибольшую популярность – это поисковые каталоги и поисковые указатели. Поисковые каталоги устроены по тому же принципу, что и тематические каталоги крупных библиотек. Они обычно представляют собой иерархические гипертекстовые меню с пунктами и подпунктами, определяющими тематику сайтов, адреса которых содержатся в данном каталоге, с постепенным, от уровня к уровню, уточнением темы. Поисковые каталоги создаются вручную. Высококвалифицированные редакторы лично просматривают информационное пространство WWW, отбирают то, что по их мнению представляет общественный интерес, и заносят в каталог. Основной проблемой поисковых каталогов является чрезвычайно низкий коэффициент охвата ресурсов WWW. Чтобы многократно увеличить коэффициент охвата ресурсов

Web, из процесса наполнения базы данных поисковой системы необходимо исключить человеческий фактор – работа должна быть автоматизирована.

Автоматическую каталогизацию Web-ресурсов и удовлетворение запросов клиентов выполняют поисковые указатели. Работу поискового указателя можно условно разделить на три этапа:

1) сбор первичной базы данных. Для сканирования информационного пространства WWW используются специальные агентские программы – черви, задача которых состоит в поиске неизвестных ресурсов и регистрация их в базе данных;

2) индексация базы данных – первичная обработка с целью оптимизации поиска. На этапе индексации создаются специализированные документы – собственно поисковые указатели;

3) рафинирование результирующего списка. На этом этапе создается список ссылок, который будет передан пользователю в качестве результирующего. Рафинирование результирующего списка заключается в фильтрации и ранжировании результатов поиска. Под фильтрацией понимается отсев ссылок, которые нецелесообразно выдавать пользователю (например, проверяется наличие дубликатов). Ранжирование заключается в создании специального порядка представления результирующего списка (по количеству ключевых слов, сопутствующих слов и др.).

В России наиболее крупными и популярными поисковыми указателями являются:

– «Яndex» (www.yandex.ru);
– «Рамблер» (www.rambler.ru);
– «Google» (www.google.ru).

Порядок выполнения

1 Запустить обозреватель MS Internet Explorer.

2 В адресной строке набрать адрес поискового WWW-сервера.

3 Открыть новое окно браузера, выполнив последовательность команд в главном меню Файл-Создать-Окно или использовав сочетание клавиш Ctrl + N.

4 Повторить пп. 2 и 3 не менее четырех раз. В разные окна браузера загрузите главные страницы поисковых машин.

5 Сравнить интерфейсы поисковых WWW-серверов.

Контрольные вопросы

1 Поиск документов, в которых обязательно присутствует выделенное слово.

2 Поиск по цитате с пропущенным словом.

3 Поиск документов, в которых присутствует любое слово из запроса.

Список литературы

1 **Майо, Д.** Самоучитель Microsoft Visual Studio 2010 / Д. Майо. – Санкт-Петербург : БХВ-Петербург, 2011. – 464 с.

2 **Левинсон, Д.** Тестирование ПО с помощью Visual Studio 2010 : пер. с англ. / Д. Левинсон. – Москва : ЭКОМ Паблишерз, 2012. – 336 с. : ил.

3 **Хорев, П. Б.** Объектно-ориентированное программирование : учебное пособие / П. Б. Хорев. – 4-е изд., стер. – Москва : Академия, 2012. – 448 с.

4 **Акулов, О. А.** Информатика. Базовый курс : учебник для вузов / О. А. Акулов. – 5-е изд., испр. и доп. – Москва : Омега-Л, 2008. – 574 с.

5 Информатика (общий курс) : учебник для вузов / Под ред. В. И. Колесникова. – 3-е изд. – Москва ; Ростов-на-Дону : Дашков и К ; Наука-Спектр, 2009. – 400 с.