

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ
АВТОМОБИЛЬНО–ДОРОЖНЫЙ ИНСТИТУТ
ГОСУДАРСТВЕННОГО ВЫСШЕГО УЧЕБНОГО ЗАВЕДЕНИЯ
«ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ»



ДонНТУ

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

к курсовой работе по дисциплине
«Информатика и компьютерная техника»

для студентов специальности 7.050102
“Экономическая кибернетика”

Горловка 2006

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ
АВТОМОБИЛЬНО–ДОРОЖНЫЙ ИНСТИТУТ
ГОСУДАРСТВЕННОГО ВЫСШЕГО УЧЕБНОГО ЗАВЕДЕНИЯ
«ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

МЕТОДИЧЕСКИЕ УКАЗНИЯ

к курсовой работе по дисциплине
«Информатика и компьютерная техника»
(для студентов специальности 7.050102
«Экономическая кибернетика»)

Утверждено
на заседании методической
комиссии факультета «Экономика
и управление»
Протокол № ____ от ____ 2006 г.

Утверждено
на заседании кафедры
«Информационные системы в
экономике»
Протокол № ____ от ____ 2006 г.

Горловка 2006

УДК 337 (071)

Методические указания к курсовой работе по дисциплине «Информатика и компьютерная техника» (для студентов специальности 7.050102 «Экономическая кибернетика») / Сост.: Николаенко Д.В. – Горловка: АДИ ДонНТУ, 2006. – 54 с.

Составлена по программе дисциплины «Информатика и компьютерная техника».

Составил: Д.В. Николаенко, ассистент

Ответственный за выпуск: В. Л. Николаенко, к.т.н., доцент

Рецензент: Л.П. Вовк, д.т.н., доцент

СОДЕРЖАНИЕ

<u>Введение</u>	4
<u>1 Требования к содержанию пояснительной записки к курсовой работе</u>	5
<u>2 Порядок выполнения и оформления курсовой работы</u>	8
<u>3 Указания к выполнению курсовой работы</u>	10
<u>4 Вопросы к защите курсовой работы</u>	35
<u>5 Список литературы</u>	37
<u>Приложение А Индивидуальные задания</u>	38
<u>Приложение Б Практические задания</u>	39

ВВЕДЕНИЕ

Целью курсовой работы является:

- выработка навыков алгоритмического мышления;
- закрепление навыков разработки, кодирования, тестирования алгоритмов;
- освоение приемов работы в интегрированной среде разработчика;
- приобретение навыков визуального и компонентного программирования;
- закрепление навыков практического использования основных конструкций языка;
- выработка навыков разработки Windows приложений;
- освоение технологии объектно-ориентированного программирования;
- приобретение навыков разработки автоматизированных рабочих мест;
- освоение объектно-ориентированных средств разработки приложений системы Visual Basic.

Среди множества инструментальных средств, автоматизирующих работу пользователя по созданию Windows приложений, широкое распространение получила система Visual Basic NET, средствами которой предлагается студентам построить приложение, включающее основные элементы графического интерфейса пользователя: Меню, включающее команды, а так же, подменю; палитры инструментов, диалоговое окно. При выполнении заданий студенты практически используют механизмы объектно-ориентированного программирования (ООП) такие как наследование, инкапсуляция, полиморфизм.

Построенное приложение, может быть положено в основу разработки автоматизированного рабочего места специалиста - экономиста.

1 ТРЕБОВАНИЯ К СОДЕРЖАНИЮ ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ К КУРСОВОЙ РАБОТЕ

1.1 Анализ прецедентов системы (вариантов использования)

В этом параграфе приводится структурная схема проекта.

1.2 Алгоритмизация модулей проекта

В данном параграфе приводятся условия задач-модулей, обсуждения алгоритмов, блок-схемы, исходные коды и результаты тестирования задач-модулей.

В файле отчета – эта совокупность фрагментов текста имеет вид:

Модуль Vector

Задача "Vector"

Объектный анализ. Объектная модель.

Пусть объектный анализ системы приводит к следующей объектной модели:

Приложение

└ Форма

└ Кнопка Start

└ Кнопка Close

Модель потока событий

Нажатие Кнопки_Start – выполнение приложения

Нажатие Кнопки_Close – завершение приложения

Объектное проектирование

Вспользуемся стандартными классами Forms и Button для построения формы windows-приложения и кнопки приложения.

Спецификации классов

Класс Forms

Объект – Объектом является окно приложения.

Формы в Visual Basic .NET являются полноправными классами без каких бы то ни было оговорок. Классы Form содержат конструкторы и

деструкторы и, как любые другие классы, могут создаваться и удаляться.

Визуальное описание формы находится в файле с расширением .resx, а весь ее код – в файле с таким же именем, но с расширением .vb.

Свойства

Name – frmDigit

Text – Vector свойство формы устанавливающее заголовок приложения, отображающийся на строке «заголовок приложения»

Методы

Отсутствуют

События

Отсутствуют

Класс Command Button

Объект – Объектом является кнопка приложения Start.

Свойства

Name – btnStart

Text - Start

Методы

Отсутствуют

События

Click – событие кнопки возникающее при нажатии на объекте. В нашем случае – выполнение приложения.

Объект - Объектом является кнопка приложения Close

Свойства

Name – btnClose

Text - Close

Методы

Нет

События

Click – реакция на нажатие кнопки btnClose – выгрузка приложения

1.3 Идея алгоритма

В данном пункте необходимо привести краткое словесное (вербальное) описание алгоритма.

1.4 Схема алгоритма

В данном пункте необходимо привести блок-схему алгоритма

1.5 Исходный код алгоритма

В данном пункте необходимо привести листинг алгоритма

Листинг должен содержать комментарий вида:

```
,  
'Фамилия И.О. Эк-XXx  
'Задача Matrix  
'Вариант N  
'Найти сумму отрицательных элементов, лежащих выше  
главной  
'диагонали матрицы a (m, n)  
,
```

1.6 Тестирование алгоритма

Тест 1

Если

Здесь необходимо привести исходный набор данных. При подборе исходных данных для проведения тестирования алгоритма необходимо пользоваться свойством алгоритма *массовость - любой набор допустимых данных удовлетворяет алгоритму.*

Например, для тестирования системы уравнений 1.1 необходимо выбирать такие проверочные значения x – $x=-1$, $x=0$, $x=1$, тем самым проверяя работу программы на каждом интервале.

(1.1)

Ожидается ответ:

Здесь приведены ожидаемые результаты, т.е. результаты просчитанные вручную с использованием других вычислительных средств.

Действительно:

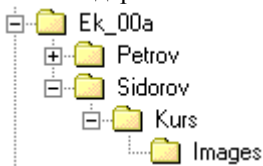
Здесь приведены полученные результаты. Полученные результаты можно вставлять в документ отчета путем фотографирования-копирования окон работы программы. Для этого необходимо во время выполнения программы производить «фотографирование в буфер обмена» текущего окна

программы (комбинация клавиш ALT+PrtScrn), а затем вставку из кармана в документ (комбинация клавиш SHIFT+INS или CTRL+V).

2 ПОРЯДОК ВЫПОЛНЕНИЯ И ОФОРМЛЕНИЯ КУРСОВОЙ РАБОТЫ

2.1 Занятие 1

1. Получение индивидуальных заданий курсовой работы.
2. Построение фрагмента дерева каталогов:



1. Подготовка шаблона файла отчета – пояснительной записки - Report_Kurs_Ek_00b_Sidorov.doc
2. Анализ алгоритма задачи.
3. Разработка алгоритма.

2.2 Занятие 2-10

1. Совместно с преподавателем обсуждение, прорабатывание алгоритмов решения задач.
2. Разработка схем алгоритмов задач. Написание кода алгоритма. Подготовка текстовой части курсовой работы – пояснительной записки.
3. Проведение тестирования алгоритма на компьютере.

2.3 Занятие 11

Формирование файла отчета.

Стилевое оформление файла отчета следует выполнять в соответствии с рекомендациями, изложенными в разделе 1 – «требования к содержанию пояснительной записки к курсовой работе» данного методического пособия. Печать файла отчета.

2.4 Занятие 12 – 17

Защита работы.

Защита работы – это значит представить преподавателю жесткую копию работы и ее электронную форму, ответить на контрольные вопросы

преподавателя (Смотри раздел 4 настоящих методических указаний – «Вопросы к защите курсовой работы»).

Отчет имеет следующую рубрикацию:

ВВЕДЕНИЕ

1 СРЕДСТВА ЯЗЫКА РАЗРАБОТКИ WINDOWS ПРИЛОЖЕНИЙ

2 Автоматизированное рабочее место (АРМ) СТУДЕНТА ПО ДИСЦИПЛИНЕ «ИКТ»

2.1 Постановка задачи и сценарий интерфейса

2.2 Структура приложения

2.3 Алгоритмизация задач-модулей

2.3.1 Модуль Vector

2.3.2 Модуль Matrix 1

2.3.3 Модуль Matrix 2

2.3.4 Модуль Graphic

2.3.5 Модуль Picture

2.4 Листинг исходного кода приложения

2.5 Тестирование проекта

ЗАКЛЮЧЕНИЕ

ЛИТЕРАТУРА

3 УКАЗАНИЯ К ВЫПОЛНЕНИЮ КУРСОВОЙ РАБОТЫ

В этом разделе, на примере, рассмотрим средства языка, позволяющие построить приложение, включающее основные элементы графического интерфейса пользователя: меню, диалоговое окно, управляющие элементы.

3.1 Постановка задачи и сценарий интерфейса

Постановка задачи

Построить приложение, включающее:

Меню

File Array Picture Help

Меню File содержит команду Exit.

Меню Array содержит команду Vector и подменю Matrix.

Подменю Matrix содержит команды Matrix 1 и Matrix 2.

Меню Graphics содержит команды Graphic и Picture.

Меню Help содержит команду About.

Палитра инструментов

Кнопка для команды Exit

Кнопка для команды Vector

Кнопка для команды Matrix_1

Кнопка для команды Matrix_2

Кнопка для команды Graphic

Кнопка для команды Picture

Система помощи

Система помощи включает только ToolTipText – «всплывающие подсказки».

Для кнопки команды Vector

«Решение задачи Vector»

Для кнопки команды Matrix_1

«Решение задачи Matrix_1»

Для кнопки команды Matrix_2

«Решение задачи Matrix_2»

Для кнопки команды Graphic

«Решение задачи Graphic»

Для кнопки команды Picture

«Решение задачи Picture»

Сценарий интерфейса

Выбор пункта меню Exit приводит к завершению приложения –

выгрузке главной формы приложения.

Выбор пункта меню Vector приводит к выполнению процедуры myVector в окне новой формы – решение задачи Vector.

Выбор пункта меню Matrix 1 и Matrix 2 приводит к выполнению процедур myMatrix1 и myMatrix2 соответственно в окнах новых форм – решение задач Matrix 1 и Matrix 2.

Выбор пункта меню Graphic приводит к выполнению процедуры myGraphic в окне новой формы – решение задачи Graphic.

Выбор пункта меню Picture приводит к выполнению процедуры myPicture в окне новой формы – решение задачи Picture.

Выбор пункта меню About приводит к выполнению процедуры myAbout, активизирующей модальное диалоговое окно, как окно новой формы, содержащее строки: «Сидоров Ф.В. ст. гр. Эк 00 б» «My First Windows Application»

Кроме того, диалоговое окно содержит кнопку Ок, нажатие которой приводит к выполнению метода Unload, освобождающего ресурсы, занимаемые окном формы.

Задержка курсора над кнопками палитры инструментов приводит к появлению ToolTipText – «всплывающих подсказок».

Свойства элементов приложения

Таблица 1.1 – Свойства элементов приложения

Элемент	Имя	Надпись	Свойства
1	2	3	4
Форма Главная	frmMainKurs	Сидоров ст. гр. Эк 00 б	по умлч
Меню File	mnuFile	File или Файл	по умлч
Меню Array	mnuArray	Array или Массив	по умлч
КмМеню Vector	mnuArrayVector	Vector или Вектор	по умлч
ПодМеню Matrix	mnuArrayMatrix	Matrix или Матрица	по умлч
КмМеню Matrix 1	mnuArrayMatrixMatrix_1 Matrix_1	Matrix_1 или Матрица 1	по умлч
КмМеню Matrix 2	mnuArrayMatrixMatrix_2 Matrix_2	Matrix_2 или Матрица 2	по умлч
Меню Graphics	mnuGraphics	Graphics или Графики	по умлч
КмМеню Graphic	mnuGraphicsGraphic Graphic по умлч	Graphic или График	по умлч
КмМеню Picture	mnuGraphicsPicture Picture по умлч	Picture или Рисунок	по умлч
Меню Help	mnuHelp	Help или Помощь	по умлч
КмМеню About	mnuHelpAbout	About или О проекте	по умлч

КмМеню Contents	mnuHelpContents	Contents или Ссылки	по умлч
Кнопка RUS	btnRUS	RUS	по умлч
Форма Vector	frmVector	Vector	по умлч
Кнопка Start	btnStart	Start	по умлч

Продолжение таблицы 1.1

1	2	3	4
Кнопка Close	btnClose	Close	по умлч
Форма Matrix 1	frmMatrix_1	Matrix 1	по умлч
Кнопка Start	btnStart	Start	по умлч
Кнопка Close	btnClose	Close	по умлч
Форма Matrix 2	frmMatrix_2	Matrix 2	по умлч
Кнопка Start	btnStart	Start	по умлч
Кнопка Close	btnClose	Close	по умлч
Форма Graphic	frmGraphic	Graphic	по умлч
Кнопка Start	btnStart	Start	по умлч
Кнопка Close	btnClose	Close	по умлч
Форма Picture	frmPicture	Picture	по умлч
Кнопка Start	btnStart	Start	по умлч
Кнопка Close	btnClose	Close	по умлч
Форма About	frmAbout	About	по умлч
Кнопка OK	btnOK	OK	по умлч

Надпись	lblAvtor	Выполнил: студент группы Эк 00 Сидоров	Font:Times New Roman Bold 12 пунктов
Окно рисунка	PictureBox1	-	...\Student\Sidorov\Kurs\Images\Img_1.bmp ges\Img_1.bmp ges\Img_1.bmp

3.2 Структура приложения

Структура приложения выполняется в виде блок-схемы работы приложения.

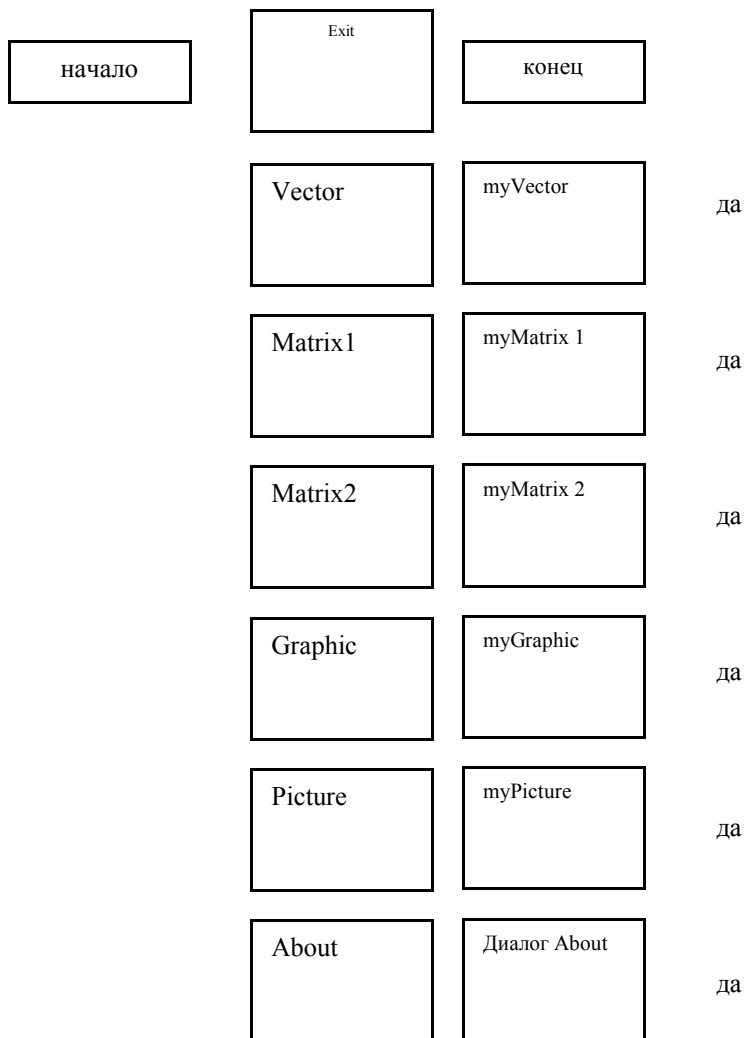


Рисунок 3.1 - Структура приложения

3.3 Указания к построению меню

В Visual Basic.NET существует два компонента меню: MainMenu и ContextMenu

Эти компоненты имеют большее количество возможностей, доступных как во время разработки, так и во время выполнения. В режиме разработки изменять меню можно с помощью редакторов меню, а во время выполнения – с помощью методов и свойств этих двух подклассов класса Menu.

MainMenu представляет собой компонент. Для вставки его в приложение необходимо выбрать MainMenu на панели инструментов Windows Forms и поместить на форму. Фактически меню будет помещено на панель компонентов, находящуюся под формой, но редактор меню в верхнем левом углу формы поместит прямоугольную область, где меню будет находиться. В прямоугольной области меню будет помещен текст «Type Here». Для разработки меню именно это и требуется – вводить текст. Введите название меню вместо «Type Here». При этом для создания «горячей клавиши» перед выбранной буквой необходимо ввести амперсанд (&).

При наборе названия меню под ним появится подменю. А справа – меню этого же уровня. Щелкнув правой клавишей мыши на пункте меню, появляется контекстное меню, в котором для вставки подменю или вложенного меню можно выбрать Insert New. Таким образом, разработка меню производится путем добавления новых пунктов и ввода их заголовков.

Для изменения свойств меню, например. Его заголовка, можно воспользоваться окном Properties.

3.4 Указания к построению палитр инструментов

Элемент управления ToolBar (Windows Forms) используется в формах в качестве панели управления, на которой выводится ряд раскрывающихся меню и кнопок с точечными рисунками, активизирующими команды. Таким образом, щелчок кнопки на панели управления эквивалентен выбору команды меню. Для кнопок можно настроить режим поведения кнопок, раскрывающихся меню или разделителей. Обычно на панели инструментов содержатся кнопки и меню, соответствующие элементам структуры меню приложения, которые предоставляют быстрый доступ к наиболее часто используемым в приложении функциям и командам.

Примечание. Свойство DropDownMenu элемента управления ToolBar использует в качестве ссылки экземпляр класса ContextMenu. Обратите особое внимание на ссылки, которые передаются при реализации такого вида

кнопок на панели инструментов в приложении, так как свойство допускает использование любого объекта, который унаследован из класса Menu.

Элемент управления ToolBar обычно закреплен в верхней части родительского окна, но его можно также закрепить с любой стороны окна. На панели инструментов могут отображаться подсказки, при наведении указатель мыши на кнопку. *Подсказка* — это небольшое всплывающее окно с кратким описанием назначения кнопки или меню. Чтобы включить отображение подсказок, необходимо задать для свойства ShowToolTips значение true.

В некоторых приложениях представлены элементы управления, очень похожие на панель инструментов; их также можно перемещать в окне приложения. Для элемента управления Windows Forms ToolBar эти действия недоступны.

Если для свойства Appearance задано значение Normal, элементы панели инструментов отображаются в виде объемных кнопок в ненажатом состоянии. Задав для свойства Appearance панели инструментов значение Flat, можно отобразить панель инструментов и ее кнопки в плоском виде. При наведении указателя мыши на плоскую кнопку она становится объемной. Кнопки панели управления можно разделить на логические группы с помощью разделителя. Разделитель — это кнопка панели инструментов, для которой в свойстве Style задано значение Separator. На панели инструментов она отображается в виде пробела. На плоской панели инструментов разделители кнопок отображаются в виде линий, разделяющих кнопки, а не в виде пробелов.

С помощью элемента управления ToolBar можно создавать панели инструментов, добавляя объекты Button в коллекцию Buttons. Для добавления кнопок в элемент управления ToolBar можно также использовать редактор коллекции; каждому объекту Button необходимо присвоить текст, рисунок, или и то и другое одновременно. Рисунок предоставляется соответствующим компонентом ImageList. Во время выполнения можно добавлять кнопки в коллекцию ToolBarButtonCollection или удалять их из нее с помощью методов Add и Remove.

Для программирования кнопок панели управления необходимо добавить код в событие ButtonClick объекта ToolBar, используя свойство Button класса ToolBarButtonClickEventArgs, чтобы определить, какая кнопка нажата.

Неотъемлемой частью элемента управления ToolBar являются добавляемые на этот элемент кнопки. Эти кнопки используются для предоставления легкого доступа к командам меню или, с другой стороны,

могут быть размещены в другой области пользовательского интерфейса приложения для представления команд, недоступных в меню.

Чтобы добавить кнопки в режиме разработки необходимо в раскрывающемся списке в верхней части окна «Свойства» выбрать элемент управления `ToolBar`, который был добавлен в форму.

Щелкните свойство `Buttons`, чтобы выбрать его, и нажмите кнопку `Ellipsis (...)`, чтобы открыть редактор коллекции `ToolBarButton`.

Используйте кнопки `Добавить` и `Удалить`, чтобы добавить или удалить кнопки из элемента управления `ToolBar`.

Настройте свойства отдельных кнопок в окне «Свойства», которое отображается на панели в правой части редактора. Можно настроить следующие важные свойства.

DropDown Menu Определяет, что меню должно отображаться в раскрывающейся кнопке панели инструментов. Свойство `Style` кнопки панели инструментов должно иметь значение `DropDown Button`. Это свойство в качестве ссылки использует экземпляр класса `ContextMenu`. В качестве примера см. фрагмент кода.

PartialPush Определяет, находится ли кнопка-переключатель в частично нажатом состоянии. Свойство `Style` кнопки панели инструментов должно иметь значение `ToggleButton`.

Pushed Определяет, находится ли кнопка-переключатель панели инструментов в нажатом состоянии. Свойство `Style` кнопки панели инструментов должно иметь значение `ToggleButton` или `PushButton`.

Style Определяет стиль кнопки панели инструментов. Должно иметь одно из значений перечисления `ToolBarButtonStyle`.

Text Строка текста, отображаемая на кнопке.

ToolTipText Текст, используемый в качестве всплывающей подсказки для кнопки.

Нажмите кнопку `ОК`, чтобы закрыть диалоговое окно и создать указанные панели.

Чтобы добавить кнопку программными средствами

С помощью процедуры создайте кнопки панели инструментов путем добавления их в коллекцию `ToolBarButtons`.

Определите параметры свойств отдельных кнопок, передав индекс через свойство `Buttons`.

В примере, представленном ниже, подразумевается, что элемент управления `ToolBar` уже добавлен в форму.

Примечание. Индекс коллекции `ToolBarButtons` начинается с нуля, что, соответственно, должно быть учтено в коде.

```

Public Sub CreateToolBarButtons()
' Create buttons and set text property.
    ToolBar1.Buttons.Add("One")
    ToolBar1.Buttons.Add("Two")
    ToolBar1.Buttons.Add("Three")
    ToolBar1.Buttons.Add("Four")
' Set properties of StatusBar panels.
' Set Style property.
    ToolBar1.Buttons(0).Style =
ToolBarButtonStyle.PushButton
    ToolBar1.Buttons(1).Style =
ToolBarButtonStyle.Separator
    ToolBar1.Buttons(2).Style =
ToolBarButtonStyle.ToggleButton
    ToolBar1.Buttons(3).Style =
ToolBarButtonStyle.DropDownButton
' Set the ToggleButton's PartialPush property.
    ToolBar1.Buttons(2).PartialPush = True
' Instantiate a ContextMenu component and menu items.
' Set the DropDownButton's DropDownMenu property to
the context menu.
    Dim cm As New ContextMenu()
    Dim miOne As New MenuItem("One")
    Dim miTwo As New MenuItem("Two")
    Dim miThree As New MenuItem("Three")
    cm.MenuItems.Add(miOne)
    cm.MenuItems.Add(miTwo)
    cm.MenuItems.Add(miThree)
    ToolBar1.Buttons(3).DropDownMenu = cm
' Set the PushButton's Pushed property.
    ToolBar1.Buttons(0).Pushed = True
' Set the ToolTipText property of one of the buttons.
    ToolBar1.Buttons(1).ToolTipText = "Button 2"
End Sub

```

3.5 Указания к построению схем алгоритмов

Установка сетки

В редакторе текстов Word необходимо на панели инструментов

Рисование нажать кнопку Рисование, выбрать пункт Сетка и в появившемся диалоговом окне Привязка к сетке установить следующие параметры:

- Привязка к сетке
- Шаг сетки по горизонтали = 0.5см
- Шаг сетки по вертикали = 0.5см
- Отображать линии сетки на экране.

Выбор символа действия

Нажмите кнопку Автофигуры на панели инструментов *Панель инструментов - строка с кнопками и другими элементами управления, служащими для выполнения команд.*

Чтобы вызвать панель инструментов, в меню Сервис выберите команду Настройка, а затем выберите вкладку Панели инструментов. Рисование, выберите команду Блок-схема, затем щелкните нужную фигуру.

- Щелкните в том месте, где требуется нарисовать фигуру блок-схемы.
- Если требуется добавить фигуры в блок-схему, повторите шаги 1 и 2, а затем расположите фигуры нужным образом.

Добавьте соединительные линии между фигурами.

- Нажмите кнопку Автофигуры на панели инструментов Рисование, выберите команду Соединительные линии, а затем щелкните нужную соединительную линию.
- Наведите указатель мыши на место закрепления соединительной линии.

При перемещении указателя мыши по фигуре места соединения отображаются в виде синих кружков.

- Щелкните первое нужное место соединения, наведите указатель мыши на другую фигуру, а затем щелкните второе место соединения.

Заблокированные соединительные линии будут соединять фигуры даже при перемещении фигур.

Добавление текста в фигуры.

Щелкните нужную фигуру правой кнопкой мыши, выберите команду Добавить текст и введите текст.

Примечание. Добавление текста к линиям или соединительным линиям невозможно; для размещения текста рядом с этими объектами используйте надписи.

Изменение типа линии.

- Выделите линию или соединительную линию, которую требуется изменить.
- Измените тип линии или соединительной линии. На панели инструментов Рисование нажмите кнопку Тип линии. Выберите нужный тип либо нажмите кнопку Другие линии, а затем выберите тип.

Установка размеров символа действия

Контекстное меню символа – Формат автофигуры – в диалоговом окне Формат автофигуры выбрать закладку Размер и установить Высоту= 2 см., Ширину= 3 см, а для символов «начало» и «конец» - Высота = 1 см.

3.6 Указания к решению задач Vector и Matrix

Рассмотрим пример, в котором из двумерного массива формируется одномерный массив.

В примере следует обратить внимание на использование переменной L, назначение которой в индексации элементов одномерного массива (формируемого массива), определение ее начального значения, изменение.

Также следует обратить внимание на комментирование кода алгоритма.

Пример «Старый знакомый»

Дан вещественный массив $A(m,n)$. Для каждой тройки найти максимальный элемент из тех, что образуют подматрицу матрицы A так, что найденный элемент является правым нижним углом подматрицы. Найденные максимальные элементы складывать в массив C, заполняя его с конца.

Решение

Построим Windows Приложение, главное окно которого содержит кнопки Start и Close.

Обсуждение алгоритма

Сценарий интерфейса

Нажатие Close-завершение Приложения.

Нажатие Start-выполнение Приложения.

Идея алгоритма:

В двойном цикле по i и j будем искать «3», сравнивая $a(i,j)=«3»$. При «совпадении» решим задачу поиска максимального элемента в подматрице:

$$A=$$

и найденный максимальный элемент положим в массив C : $C(L)=\text{Max}$.

Сразу же выполним $L=L-1$, подготовив этим новое место в массиве C . Очевидно, что до просмотра массива A надо сделать $L=mn-2m-2n+4$ (в массиве C не может быть больше элементов, чем «внутренних» элементов матрицы A).

Свойства объектов проекта

Объект	Имя	Надпись
Форма Сидоров А.О.	frmOldMen	Проект «Старый знакомый»
Кнопка	cmdStart	Эк 00a Start
Кнопка	cmdClose	Close

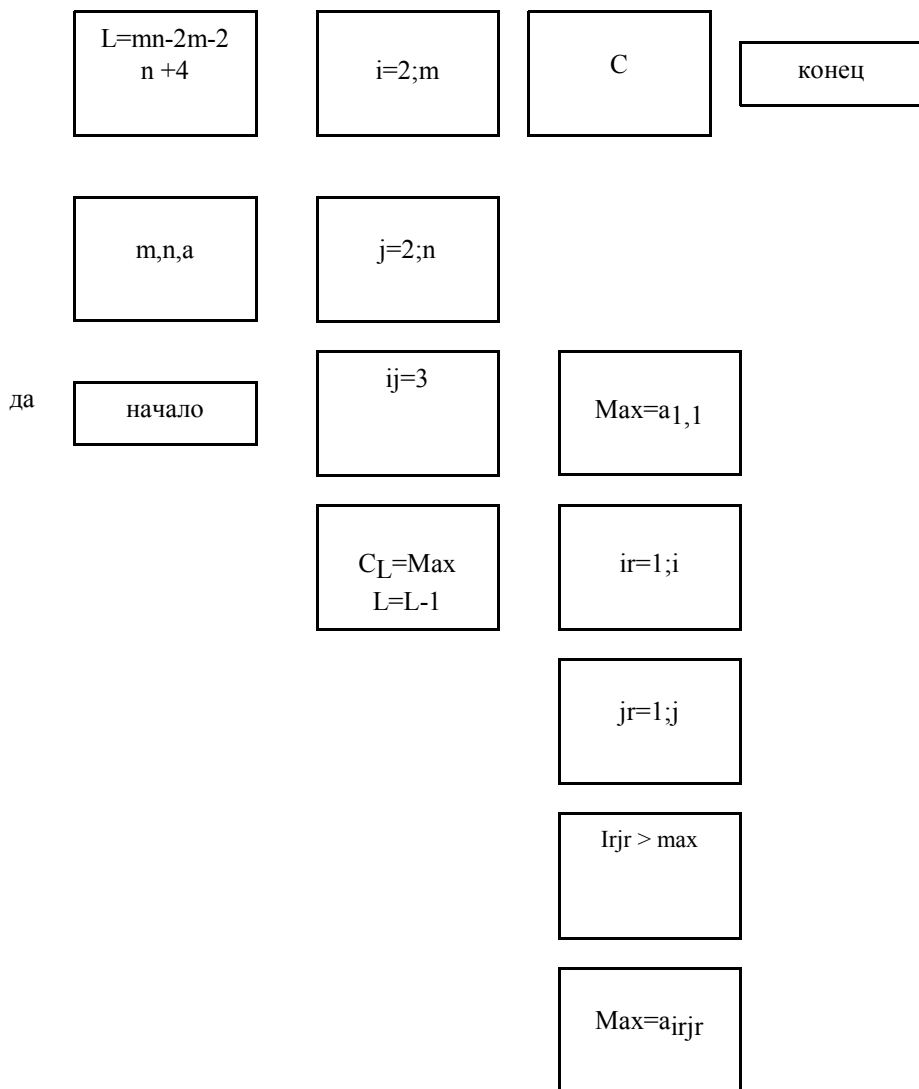


Рисунок 3.2 – Схема алгоритма

Исходный код алгоритма

```
Public Class Form1
Inherits System.Windows.Forms.Form
Private Sub Button1_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Start.Click
    ' Сидоров О.Л. ст. гр. Эк 006
    ' Задача "OldMen"
    ' Вариант 30
    ,
    Dim a(,) As Single
    Dim C() As Single
    Dim Max As Single
    Dim str() As String

    Dim m As Integer, n As Integer
    Dim i As Integer, j As Integer, ir As Integer,
jr As Integer, L As Integer, k As Integer

    m = InputBox("Введите число строк m", "Окно
ввода m", 4)
    n = InputBox("Введите число столбцов n", "Окно
ввода n", 3)
    ReDim a(m, n)
    k = m * n - 2 * m - 2 * n + 4
    ReDim C(k)
    ReDim str(m)
    '=====
    ' Ввод исходного массива
    ,
    Dim msg As String
    For i = 1 To m
        For j = 1 To n
a:          a(i, j) = InputBox("Введите элемент
                матрицы", "Элемент матрицы", 0)
            If a(1, j) = 3 Or a(i, 1) = 3 Or a(i,
                n) = 3 Or a(m, j) = 3 Then
                MsgBox("На границе матрицы не
                должно быть троек")
                msg = MsgBox("Изменить элемент
                матрицы?", vbYesNo)
                If msg = 6 Then
```

```

                GoTo a
            End If
            Close()
        End If
    Next j
Next i
L = k 'm * n - 2 * m - 2 * n + 4
For i = 2 To m - 1
    For j = 2 To n - 1
        If a(i, j) = 3 Then
            Max = a(i, j)
            For ir = 1 To i
                For jr = 1 To j
                    If a(ir, jr) > Max Then
                        Max = a(ir, jr)
                    End If
                Next jr, ir
            Next jr, ir
            C(L) = Max : L = L - 1
        End If
    Next j, i
    '=====
    ' Вывод сформированного массива C
    '
    Dim cep As String
    For i = 1 To k
        cep = cep & C(i) & " "
    Next i
    tbOtvvet.Text = cep
End Sub
End Class

```

3.7 Указания к решению задачи Graphic и Picture

Модуль “Graphics”

Условие

Построить график функции: Лемниската Бернулли

Решение

Обсуждение алгоритма:

Построим Windows Приложение, главное окно которого содержит кнопки Start и Close.

Сценарий интерфейса:

Нажатие Close-завершение Приложения.

Нажатие Start-выполнение Приложения.

Идея алгоритма:

Вывод графики выполним на полотне g (расположено на PictureBox, полотно g принадлежит пространству System.Drawing.Graphics), воспользовавшись его методами: TransformTranslate (для смещения начала координат в центр PictureBox); ScaleTransform (для масштабирования полотна); DrawEllipse (для рисования точки); DrawString (для подписи осей); DrawLine (для рисования сетки).

Для рисования точки создадим класс myPoint с ReadWrite свойствами:

WidthPen (толщина точки) PointColor (цвет точки) инициализация, которых выполняется значениями Color.Black и 1 при создании экземпляра класса, при чем значение 1 является не обязательным параметром и по умолчанию равно 1.

Метод pSet класса myPoint «зажигает» точку по заданным координатам x и y, на полученном объекте из пространства имен System.Drawing.Graphics (в данном случае «g»).

Для вычисления X и Y воспользуемся формулами (т.к. уравнение задано в полярных координатах):

$$x = p * \cos (fi)$$

$$y = p * \sin (fi)$$

Свойства объектов проекта:

Объект	Имя	Надпись
Форма	frmGraphics	Graphics
Кнопка	btnStart	Start
Кнопка	btnClose	Close

Исходный код алгоритма:

```
Imports System.Math
Public Class frmGraphics
Inherits System.Windows.Forms.Form
Windows Form Designer generated code
Private Sub btnStart_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
btnStart.Click
    Dim g As Graphics
    g = PictureBox.CreateGraphics
    g.Clear(Color.White)
    'Переносим начало координат в центр PictureBox
    g.TranslateTransform(PictureBox.Width / 2,
PictureBox.Height / 2)
    'Масштабируем согласно выбранным параметрам
    g.ScaleTransform(tbOX.Text, tbOY.Text)
    Dim oA As New myPoint(Color.Black, )
    Dim y1, y2, x1, x2 As Integer
    Dim penLine As Pen
    penLine = New Pen(Color.FromArgb(100, 125,
125), 0.001)
    'Рисуем сетку
    For y1 = -PictureBox.Width / 2 To
PictureBox.Width / 2 Step 1
        y2 = y1
        g.DrawLine(penLine, -PictureBox.Width, y1,
PictureBox.Width, y2)
    Next
    For x1 = -PictureBox.Height / 2 To
PictureBox.Height / 2 Step 1
        x2 = x1
        g.DrawLine(penLine, x1,
-PictureBox.Height, x2, PictureBox.Height)
    Next
    oA.WidthPen = CSng(tbWidthAxis.Text)
    Dim fi, x, y, Ro As Single
    'Рисуем оси
    For x = -PictureBox.Width / 2 To
PictureBox.Width / 2 Step 0.01
        x = x
        oA.pSet(x, 0, g)
        y = x
```

```

        oA.pSet(0, y, g)
    Next x
    Dim c As Integer = 2
    oA.pointColor = Color.Green
    oA.WidthPen = CSng(tbWidthGraphics.Text)
    'Вычисляем и рисуем точки графика
    For fi = 0 To 2 * PI Step 0.001
        If Cos(2 * fi) > 0 Then
            Ro = c * (Sqrt(2 * Cos(2 * fi))) 'Здесь
необходимо вставить свою формулу, заданную в полярной
системе координат.
            x = Ro * Cos(fi)
            y = Ro * Sin(fi)
            oA.pSet(x, y, g)
        End If
    Next fi
    'Подписываем оси
    Dim myFont As New Font("Times New Roman", 0.3)
    Dim mybrush As New
Drawing2D.LinearGradientBrush(ClientRectangle,
Color.Black, Color.Yellow,
Drawing2D.LinearGradientMode.Horizontal)
    g.DrawString("Y", myFont, mybrush, 0, -3)
    g.DrawString("X", myFont, mybrush, 4, 0)

End Sub

Private Sub btnClose_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
btnClose.Click
    Me.Close()
End Sub
End Class

```

Ниже приведен исходный код класса «MyPoint» или Точка

```

Option Strict On
Module myModule
    Class myPoint
        Dim mPenColor As System.Drawing.Color
        Public mWidthPen As Single
    End Class
End Module

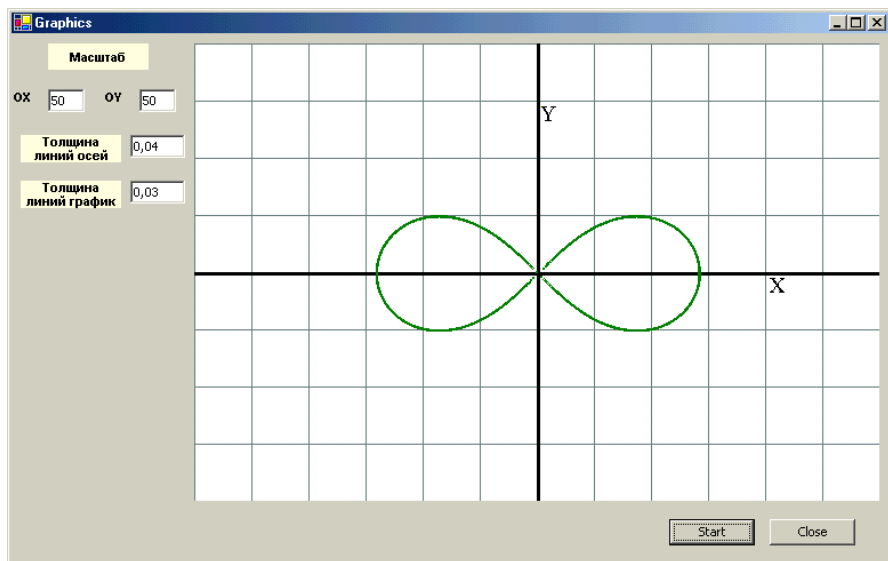
```

```

        Public Sub New(ByVal aPenColor As
System.Drawing.Color, Optional ByVal aWidthPen As
Single = 1)
            mPenColor = aPenColor
            mWidthPen = aWidthPen
        End Sub
        Public Property WidthPen() As Single
            Get
                Return mWidthPen
            End Get
            Set(ByVal Value As Single)
                mWidthPen = Value
            End Set
        End Property
        Public Property pointColor() As
System.Drawing.Color
            Get
                Return mPenColor
            End Get
            Set(ByVal Value As System.Drawing.Color)
                mPenColor = Value
            End Set
        End Property
        Public Sub pSet(ByVal x As Single, ByVal y As
Single, ByVal Sender As System.Drawing.Graphics)
            Dim penPoint As New Pen(mPenColor, 0.0001)
            Dim a As Single = mWidthPen
            Sender.DrawEllipse(penPoint, x, y, a, a)
        End Sub
    End Class
End Module

```

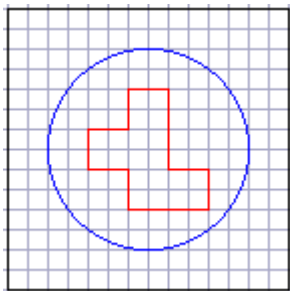
Тестирование алгоритма



Модуль “Picture”

Условие

Нарисовать фигуру:



Решение

Обсуждение алгоритма:

Построим Windows Приложение, главное окно которого содержит кнопки Start и Close.

Сценарий интерфейса:

Нажатие Close-завершение Приложения.

Нажатие Start-выполнение Приложения.

Идея алгоритма:

Вывод графики выполним на полотне g (расположенном на объекте PictureBox), воспользовавшись его методами рисования Line и объектными переменными PictureBox для определения высоты, ширины объекта. Для создания цвета линий воспользуемся пространством имен System.Drawing.Color.

Свойства объектов проекта:

Объект	Имя	Надпись
Форма	frmPicture	Picture
Кнопка	btnStart	Start
Кнопка	btnClose	Close

Исходный код алгоритма:

```
Public Class frmPicture
    Inherits System.Windows.Forms.Form

    Windows Form Designer generated code

    Private Sub btnStart_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
btnStart.Click
        Dim g As Graphics
        g = PictureBox.CreateGraphics
        g.Clear(Color.White)
        Dim m, n As Single
        m = TextBox2.Text
        n = TextBox1.Text
        g.ScaleTransform(m, n)
        Dim Xm As Integer = 140
        Dim Ym As Integer = 140
        Dim Xc, Yc As Integer
        'Увеличивает размеры PictureBox
        PictureBox.Width = PictureBox.Width * m
        PictureBox.Height = PictureBox.Height * n
        'Находит центр рисунка
        Xc = Xm / 2
        Yc = Ym / 2
        'Размер клеточки
        Dim k As Integer = 10
        Dim penRectangle, penEllipse, penPicture As
Pen
        penRectangle = New Pen(Color.Black)
        penEllipse = New Pen(Color.Blue)
        penPicture = New Pen(Color.Red)
        Dim penLine As New Pen(Color.FromArgb(110,
128, 128), 0.1)
        'Рисую сетку
        Dim y1, y2, x1, x2 As Integer
        For y1 = 0 To Xc * 2 Step k
            y2 = y1
            g.DrawLine(penLine, 0, y1, 2 * Xc, y2)
        Next
        For x1 = 0 To Yc * 2 Step k
            x2 = x1
```

```

        g.DrawLine(penLine, x1, 0, x2, 2 * Yc)
    Next
    Dim myRectangle As New Rectangle(0, 0, Xc * 2,
Yc * 2)
    g.DrawRectangle(penRectangle, myRectangle)
    'Рисую эллипс и многоугольник
    g.DrawEllipse(penEllipse, 2 * k, 2 * k, Xc +
(3 * k), Yc + (3 * k))
    Dim point1 As New Point(Xc - k, Yc - 3 * k)
    Dim point2 As New Point(Xc - k, Yc - k)
    Dim point3 As New Point(Xc - 3 * k, Yc - k)
    Dim point4 As New Point(Xc - 3 * k, Yc + k)
    Dim point5 As New Point(Xc - k, Yc + k)
    Dim point6 As New Point(Xc - k, Yc + 3 * k)
    Dim point7 As New Point(Xc + 3 * k, Yc + 3 *
k)
    Dim point8 As New Point(Xc + 3 * k, Yc + k)
    Dim point9 As New Point(Xc + k, Yc + k)
    Dim point10 As New Point(Xc + k, Yc - 3 * k)
    Dim point11 As New Point(Xc - k, Yc - 3 * k)
    Dim arrPoints1 As Point() = {point1, point2,
point3, point4, point5, point6, point7, point8,
point9, point10, point11}
    g.DrawPolygon(penPicture, arrPoints1)
End Sub

Private Sub btnClose_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
btnClose.Click
    Me.Close()
End Sub
End Class

```

Тестирование алгоритма:

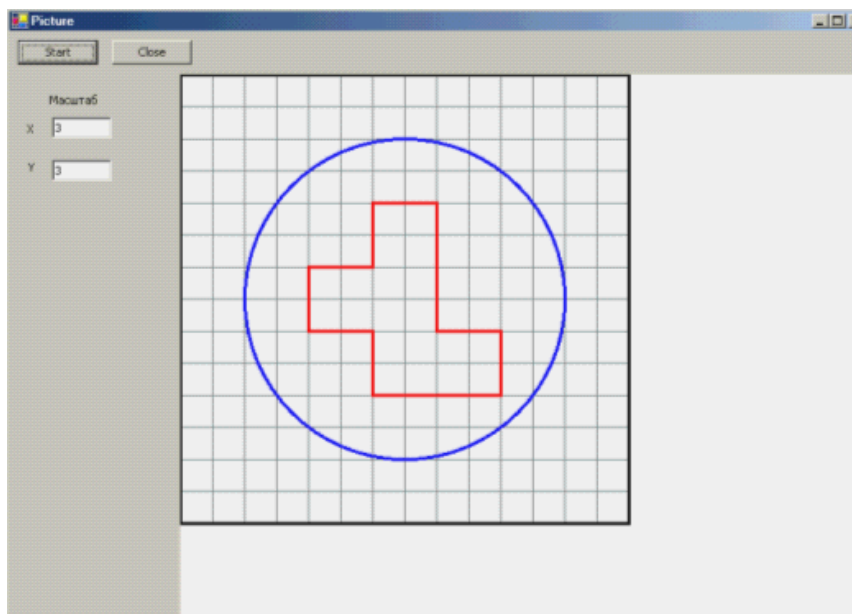
Если:

масштаб 3:3

то ожидается ответ:

Прорисовка картинки на полотне, расположенном на PictureBox

действительно:



4 ВОПРОСЫ К ЗАЩИТЕ КУРСОВОЙ РАБОТЫ

1. Приведите рубрикацию отчета
2. Оформление отчета
3. Как задать стиль?
4. Как вставить «жесткую» разбивку на страницы?
5. Чем отличается шрифт Courier от Times?
6. Как автоматически собрать оглавление?
7. О чем пишут в Реферате?
8. О чем пишут во Введении?
9. О чем пишут в Заключение?
10. Приведите фрагмент дерева каталогов, соответствующий структуре курсовой работы.
11. Как вызвать редактор меню?
12. Можно ли элементам меню задать ToolTipText?
13. Как включить в меню сепаратор?
14. Как включить на панель ToolBox элементы ToolBar?
15. Зачем на ToolBox первой кнопкой помещают кнопку-сепаратор?
16. Какую роль играет ImageList при построении палитры инструментов?
17. Как включить на панель ToolBox элементы CoolBar?
18. Какова идея построения панели CoolBar на основе ToolBar?
19. Как включить сетку на документ Word?
20. Как установить размеры символов действий блок-схем в соответствии со стандартом?
21. Приведите структурную схему приложения.
22. Расскажите сценарий интерфейса.
23. Для каких элементов и как заданы ToolTipText?
24. Алгоритм модуля Vector
25. Алгоритм модуля Matrix 1
26. Алгоритм модуля Matrix 2
27. Алгоритм модуля Graphic
28. Алгоритм модуля Picture
29. Как включить в проект новую форму?
30. Графические методы какого стандартного класса использованы?
31. Как установить декартову систему координат?
32. Какая модель образования цвета использована в конструкторе цвета?
33. Общий вид и работа оператора for – next

- 34. Общий вид и работа оператора do while - loop
- 35. Общий вид и работа оператора do until- loop
- 36. Общий вид и работа оператора do loop - while
- 37. Общий вид и работа оператора do loop - until
- 38. Общий вид и работа оператора if-then-else
- 39. Общий вид и работа оператора select - case
- 40. Общий вид и работа оператора circle
- 41. Общий вид и работа оператора line
- 42. Общий вид и работа оператора inputbox

5 СПИСОК ЛИТЕРАТУРЫ

1. Visual Basic 6.0: пер. с англ.- СПб.: БХВ – Санкт-Петербург, 1999.- 992с., ил.
2. Корнелл Г., Моррисон Дж. Программирование на VB.NET. - СПб.: Питер, 2002. - 490 с.: ил.
3. Кит Франклин VB.NET для разработчиков : Пер. с англ. – М.: Издательский дом „Вильямс”, 2002 – 272 с. : ил. – Парал. тит. англ.
4. Освой самостоятельно Visual Basic .NET за 24 часа. : Пер. с англ. – М.: Издательский дом „Вильямс”, 2002 – 416 с. : ил. – Парал. тит. англ.
5. Васильев П.П. Встроенные функции языка программирования Visual Basic 6.0. – М.: Диалог – МИФИ, 200 – 160 с.
6. Объектно-ориентированное программирование в Visual Basic .NET. Библиотека программиста / Д. Кларк. – СПб.: Питер; 2003. – 352 с.: ил.
7. Постолит А.В. Visual Studio .NET: разработка приложений баз данных. – СПб.: БХВ-Петербург, 2003. – 544 с.: ил.
8. Гарнаев А. Visual Basic .NET: разработка приложений. – СПб.: БХВ-Петербург, 2002. – 620 с.
9. Дэвид С. Плат. Знакомство с Microsoft NET. Русская редакция Microsoft, 2001. – 218 с.
10. Крис Пейн. ASP.NET Освой самостоятельно. –М., СПб., Киев: «Вильямс», 2002. – 824 с.
11. Поль Киммел – СПб: ООО «ДиасофтЮП», 2003. – 720 с.
12. Официальный сайт справочной системы Microsoft Development www.msdn.microsoft.com

ПРИЛОЖЕНИЕ А

ИНДИВИДУАЛЬНЫЕ ЗАДАНИЯ

Теоретические вопросы

1. Общий вид и работа оператора For – Next
2. Общий вид и работа оператора Do – While - Loop
3. Общий вид и работа оператора Do – Until- Loop
4. Общий вид и работа оператора Do – Loop - While
5. Общий вид и работа оператора Do – Loop - Until
6. Общий вид и работа оператора If – Then - Else
7. Общий вид и работа оператора Select - Case
8. Общий вид и работа оператора InputBox
9. Общий вид и работа оператора Print
10. Общий вид и работа оператора MsgBox
11. Общий вид и работа оператора Format
12. Общий вид и работа оператора Circle
13. Общий вид и работа оператора Line
14. Общий вид и работа оператора Pset Задание цвета
15. Форма.
16. Меню
17. Палитра инструментов
18. Кнопка
19. Текстовое поле
20. Надпись
21. Переключатель
22. Флажок
23. Рамка
24. Список
25. Комбинированное поле
26. Рисунок
27. Графическое поле
28. Строка состояния
29. Списки дисков
30. Списки каталогов
31. Списки файлов
32. Дерево
33. Табличные списки
34. Таймер

ПРИЛОЖЕНИЕ Б ПРАКТИЧЕСКИЕ ЗАДАНИЯ

Задача «Vector»

Дан одномерный массив a размерности n целого типа.

1. Переставить отрицательные элементы в начало.
2. Убывают ли элементы массива?
3. Какой элемент чаще встречается?
4. Переставить положительные элементы в конец.
5. Переставить так, чтобы положительные и отрицательные элементы чередовались.
6. Переставить максимальный и минимальный элементы местами.
7. Переставить нули в конец.
8. Переставить единицы в начало.
9. Сколько раз встречается заданный элемент?
10. Сколько элементов встречается более одного раза?
11. Переставить положительные элементы на четные места.
12. Переставить элементы четных и нечетных мест.
13. Переставить нули и единицы местами.
14. Переставить первый элемент на последнее место, второй на предпоследнее и т.д.
15. Переставить первую и вторую половину массива.
16. Переставить так, чтобы все элементы возрастали.
17. Переставить так, чтобы все элементы убывали.
18. Переставить так, чтобы возрастали элементы, стоящие на нечетных местах.
19. Переставить так, чтобы убывали элементы, стоящие на нечетных местах.
20. Переставить так, чтобы возрастали элементы, стоящие на четных местах.
21. Сколько слов в массиве?
22. Какое элемент встречается только раз?
23. Переставить так, чтобы отрицательные элементы убывали.
24. Переставить положительные и отрицательные элементы местами.
25. Найти среднее арифметическое из положительных элементов, предшествующих 3-ке
26. Найти среднее геометрическое из положительных элементов, следующих за 3-кой

27. Есть ли одинаковые элементы?
28. Найти сумму весов элементов, встречающихся более одного раза.
29. Сжать массив (вместо подряд идущих одинаковых элементов записать один из них).
30. Заменить на 7-ку неповторяющиеся элементы.

Замечание

Вес элемента – это его номер.

Слово – это подряд идущие одинаковые элементы.

Задача «Matrix_1»

Дан двумерный массив a размерности m на n целого типа.

1. "Квадратики"

Сформировать массив C , состоящий из сумм элементов "контуров", образующих квадраты с центрами, совпадающими с геометрическим центром матрицы.

2. "Миноры"

Для каждого ее элемента записать в массив C сумму элементов матрицы A , остающихся после вычеркивания строки и столбца, на пересечении которых лежит данный элемент.

3. "Будущие лучшие"

Для каждой восьмерки найти наибольший элемент из тех, что образуют подматрицу матрицы A , где найденная восьмерка - это левый верхний угол подматрицы матрицы A . Наибольшие элементы записать в массив C на четные места. На границе матрицы восьмерок нет.

4. "Все бывшие звезды"

Для каждой четверки найти наибольший элемент из тех, что образуют подматрицу матрицы A , где найденная четверка - правый нижний угол подматрицы матрицы A . Наибольшие элементы записать в массив C на нечетные места. На границе матрицы четверок нет.

5. "Подающий надежды"

Для каждой шестерки выбрать элемент и записать его в массив C . Выбираемый элемент находится на пересечении диагонали и границы матрицы и имеет индексы большие, чем у найденной шестерки. На границе матрицы шестерок нет.

6. "Старый знакомый"

Для каждой тройки выбрать элемент и записать его в массив C . Выбираемый элемент находится на пересечении диагонали, параллельной главной, и границы матрицы и имеет индексы меньшие, чем у найденной тройки. На границе матрицы троек нет.

7. "Батут"

Записать в массив C модули разности между наибольшим и наименьшим элементами строк.

8. "Определители девятки"

Подсчитать число девяток и записать в массив C подряд с конца значения определителей девятки. На границе матрицы девяток нет.

9. "Большой крест"

Подсчитать число единиц и записать в массив C с конца и подряд суммы элементов, образующих большой крест. На границе матрицы единиц

нет.

10. "Седловая точка"

Найти "седловые" точки и переписать их в массив С.

11. "Цветы"

Подсчитать число отрицательных элементов, лежащих ниже главной диагонали, и записать их в массив С на четные места.

12. "Цветы"

Подсчитать число положительных элементов, лежащих в полосе, а сами положительные элементы записать в массив С на нечетные места.

13. "Цветы"

Подсчитать число нулевых элементов, лежащих в полосе, а номера их строк записать в массив С на нечетные места.

14. "Цветы"

Подсчитать число положительных элементов, лежащих выше побочной диагонали и записать их в массив С подряд с конца.

15. "Цветы"

Подсчитать число отрицательных элементов, лежащих выше побочной диагонали, включая диагональ, и записать их в массив С.

16. "Избранные"

Переписать в массив С первые К ($K < M$, $K < N$) элементов матрицы А меньших нуля. Массив С заполнить с конца и подряд.

17. "Свой"

Переписать в массив С те элементы, индексы которых удовлетворяют условию: $M+1 \leq i+j \leq N+1$.

18. "Положительные герои"

Определить количество положительных элементов в каждой строке и записать полученные значения в массив С с конца и подряд.

19. "Отрицательные герои"

Найти сумму отрицательных элементов в каждом столбце и записать полученные значения в массив С с конца и подряд.

20. "Ворота"

Для каждой четверки матрицы А записать в массив С произведение элементов, лежащих в первой и последней строке матрицы А и в том же столбце, что и найденная четверка.

21. "Адреса героев"

Найти произведение положительных элементов матрицы А, а в массив С с конца и подряд записать номера строк этих элементов.

22. "Враги"

Записать в массив С номера строк отрицательных элементов матрицы А, заменяя отрицательные элементы их модулями. Массив С заполнить по четным местам.

23. "5-ки"

Подсчитать количество пятерок и записать в массив С на четные места произведение их индексов.

24. "Кресты семерки"

Подсчитать число семерок и записать в массив С на четные места произведение элементов, образующих крест семерки.

На границе матрицы семерок нет.

25. "Добрые люди"

Найти количество положительных элементов в каждом столбце матрицы А и записать полученные значения в массив С на четные места.

26. "Весы"

Для каждой девятки записать в массив С с конца подряд сумму элементов, лежащих в первом и последнем столбце той же строки, что и найденная девятка.

27. "Скорпион"

Как только среди внутренних элементов матрицы А найдется двойка, так сразу же записать в массив С сумму элементов матрицы А, лежащих на пересечении прямых, проходящих параллельно границам и диагоналям матрицы.

28. "333"

Записать в массив С кубы элементов матрицы А, делящихся на 3 без остатка. Массив С заполнить по нечетным местам.

29. "Остатки"

Записать в массив С квадраты элементов матрицы А, остаток от деления которых на два есть единица.

30. "Ворота"

Для каждой четверки матрицы А записать в массив С произведение элементов, лежащих в первой и последней строке матрицы А.

Замечание

Элемент лежит на «границе», если $i=1$ или $i=m$ или $j=1$ или $j=n$.

Элемент лежит «внутри», если $i \neq 1$ или $i \neq m$ или $j \neq 1$ или $j \neq n$.

Задача «Matrix_2»

Сформировать двумерный массив С по заданному правилу.

Вариант 1

Вариант 2

Вариант 3

Вариант 4

Вариант 5

Вариант 6

Вариант 10

Вариант 11

Вариант 7

Вариант 12

Вариант 13

Вариант 8

Вариант 14

Вариант 9

Вариант 15

Вариант 16

Вариант 17

Вариант 18

Вариант 19

Вариант 20

Вариант 21

Вариант 22

Вариант 23

Вариант 24

Вариант 25

Вариант 26

Вариант 27

Вариант 28

Вариант 29

Вариант 30

Задача «Graphic»

Построить график функции

1 Строфоида

2 Циссоида Диокла

3 Декартов лист (Цветок жасмина)

4 Верзьера Анъези

5 Конхоида Никодема

6 Улитка Паскаля

7 Кардиоида

8 Овал Кассини

9 Лемниската Бернулли

10 Эвольвента окружности

11 Архимедова спираль

12 Логарифмическая спираль

13 Спираль ферма

14 Гиперболическая спираль

15 Циклоида

16 Удлиненная циклоида

17 Эпициклоида

18 Гипоциклоида

19 Астроида

20 Трактриса

21 Цепная линия

22 Трехлепестковая роза (Кривая Гвидо Гранди)

23 Черырехлепестковая роза (Кривая Гвидо Гранди)

24 Кохлеоида

25 Кривая затухающего колебания

26 Кривая Гаусса (Нормальная кривая)

27 Полукубическая парабола

28 Гипербола

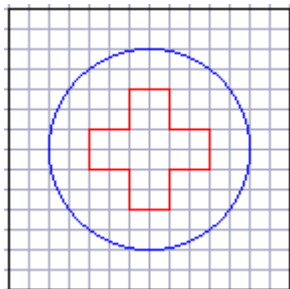
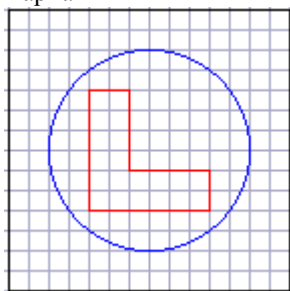
29 Парабола

30 Некоторая кривая

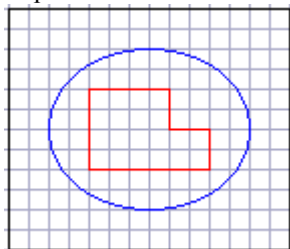
Задача «Picture»

Нарисовать рисунок

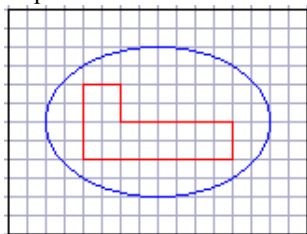
Вариант 1



Вариант 3

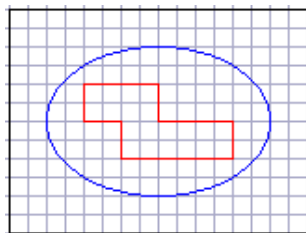
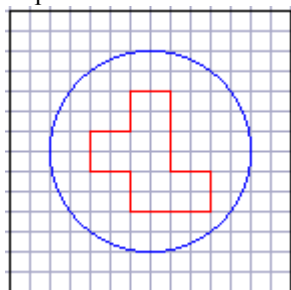


Вариант 5

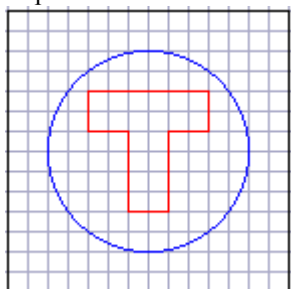


Вариант 7

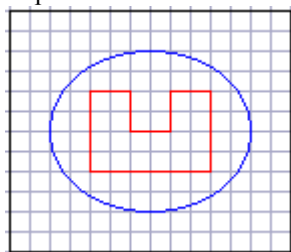
Вариант 2



Вариант 4

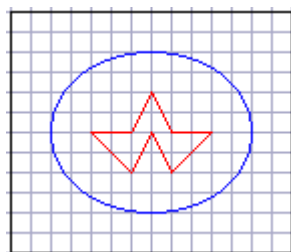
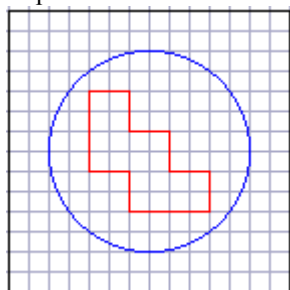


Вариант 6

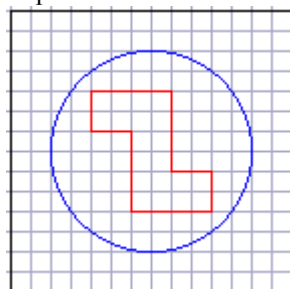


Вариант 8

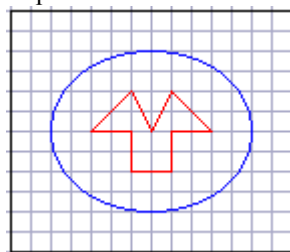
Вариант 9



Вариант 11

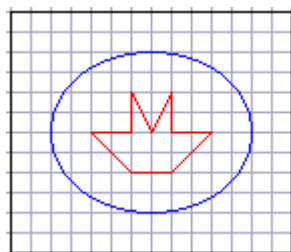
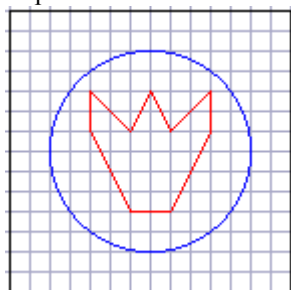


Вариант 13

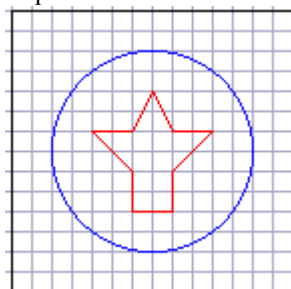


Вариант 15

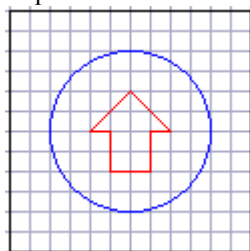
Вариант 10



Вариант 12

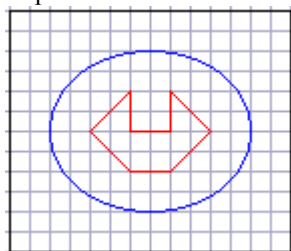


Вариант 14

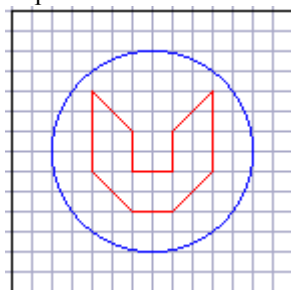


Вариант 16

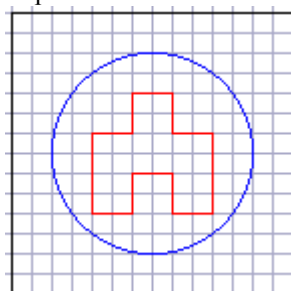
Вариант 17



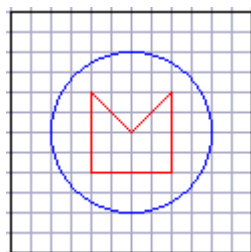
Вариант 19



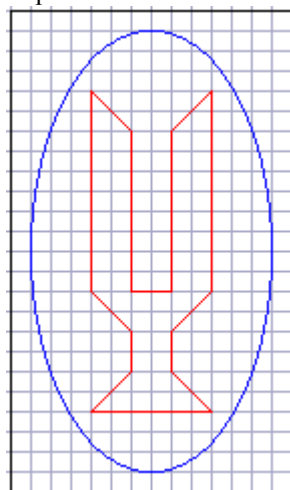
Вариант 20



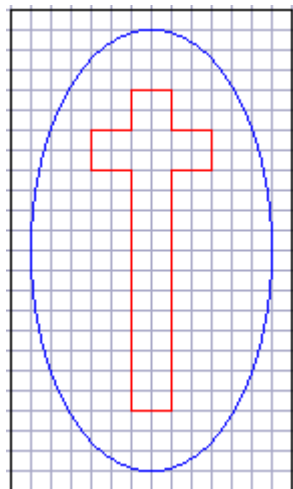
Вариант 22



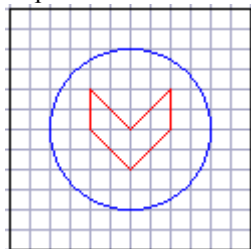
Вариант 18



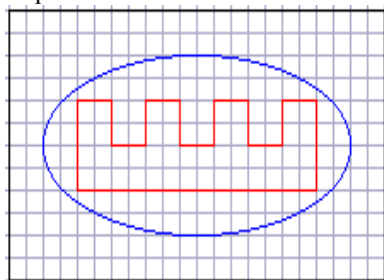
Вариант 21



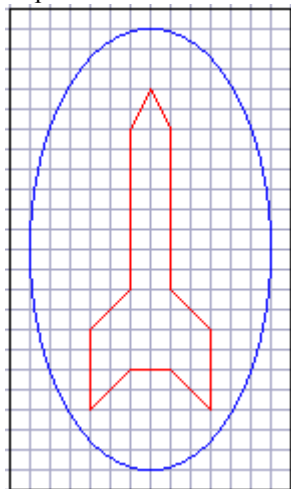
Вариант 23



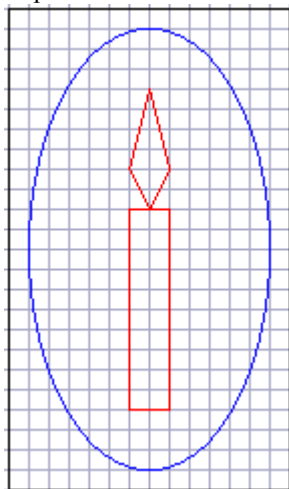
Вариант 25



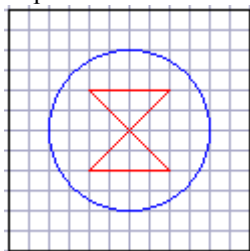
Вариант 26



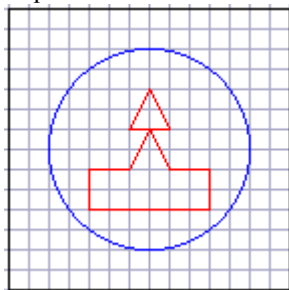
Вариант 24



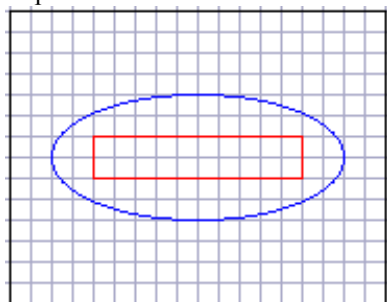
Вариант 27



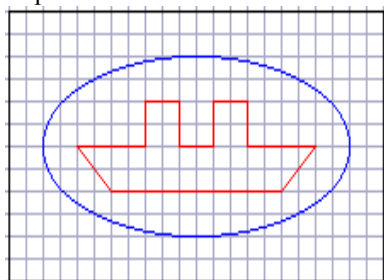
Вариант 28



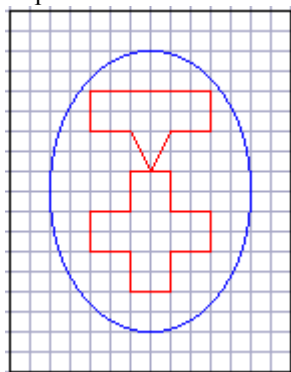
Вариант 29



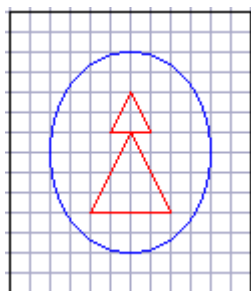
Вариант 30



Вариант 31



Вариант 32



Вариант 33



Методические указания к курсовой работе по дисциплине «Информатика и компьютерная техника» (для студентов специальности 7.100403 «Экономическая кибернетика»)

Денис Владимирович Николаенко

Подписано в печать
Заказ 43-06
Тираж 75 экз.

Формат 70х90/16
Усл. печ. лист. 3,51

АДИ ДонНТУ
84646, г. Горловка, ул. Кирова 51