

Министерство сельского хозяйства РФ
ФГБОУ ВО
Кубанский государственный аграрный университет
имени И.Т. Трубилина
Кафедра Системного анализа и обработки информации

АЛГОРИТМИЗАЦИЯ И ПРОГРАММИРОВАНИЕ

Методические рекомендации
по контактной и самостоятельной работе
для обучающихся по дисциплине
«Экономическая информатика»
по направлению
38.03.01 Экономика, «Финансы и кредит»
всех форм обучения

Краснодар-2019

Под редакцией д.э.н., проф. Барановской Т.П.

Вострокнутов А.Е. Алгоритмизация и программирование: методические указания к лабораторным работам для самостоятельной работы по дисциплине «Экономическая информатика» / А.Е. Вострокнутов, Ю.Н. Самойлюков, К.А. Ковалева, Н.М. Нилова. – Краснодар: КубГАУ, 2019 [Электронный ресурс].

Составлены в соответствии с рабочей программой дисциплины «Экономическая информатика» для обучающихся по направлению подготовки 38.03.01 Экономика, «Финансы и кредит».

СОДЕРЖАНИЕ

Лабораторная работа №1. Интегрированная среда разработки программ Turbo Pascal 7.0	4
Лабораторная работа №2. Программы линейной структуры	15
Лабораторная работа №3. Операторы ветвления: условный оператор, оператор варианта	22
Лабораторная работа №4. Циклические конструкции. Арифметические циклы.....	35
Лабораторная работа №5. Регулярный тип данных (массивы)	44
Список использованных источников	54

Лабораторная работа №1. Интегрированная среда разработки программ Turbo Pascal 7.0

1 Цель работы

Изучить основные принципы работы в интегрированной среде разработки программ (*IDE*) *TurboPascal 7.0*.

2 Порядок выполнения работы

Ознакомиться с теоретическими сведениями об *IDE TurboPascal 7.0*. Выполнить задания раздела 5. Составить отчет о лабораторной работе и защитить его у преподавателя.

3 Содержание отчета

- наименование и цель работы;
- задание на лабораторную работу;
- результаты выполнения работы.

4 Краткая теория

Интегрированная среда разработки программ *IDE (Integrated Development Envelopment)* содержит средства, позволяющие программисту выполнить все действия по разработке программы. В первую очередь, это встроенный редактор, ориентированный на редактирование исходных текстов программ на языке *Pascal*. Встроенный транслятор, позволяющий как компилировать, так и интерпретировать программу. Средства отладки программ.

4.1 Основные элементы *IDE TurboPascal 7.0*

Для запуска на исполнение *IDE TurboPascal* необходимо активизировать каталог `\BP\BIN\` (или `\TP\BIN\`) и в командной строке указать `turbo.exe`. При необходимости открыть файлы с текстом программ можно указать их имена через пробелы. После запуска появится окно *IDE TurboPascal*, изображённое на рисунке 1.1.

Пункты главного меню *IDE* содержат все команды. Активизация главного меню выполняется нажатием клавиши *F10*, при этом один из пунктов выделяется зелёным маркером. Маркер может перемещаться по пунктам при помощи клавиш управления курсором (`←` `→`), тем самым, выбирая один из пунктов. Для отмены выбора пунктов достаточно нажать клавишу *Esc*. Эта

клавиша также используется для закрытия диалоговых окон, выхода из подменю. Для инициализации одного из пунктов необходимо после активизации меню, выделить его и нажать клавишу *Enter*. При этом появляется соответствующее подменю с набором команд.

Центральное место занимает окно редактирования. В заголовке окна указывается имя редактируемого файла. Если это вновь создаваемый файл, то его имя *NONAMEXX.PAS*, где *XX* – это порядковый номер нового файла (нумерация начинается с 00). В правом верхнем углу окна расположен номер активного окна и кнопка изменения режима отображения окна при помощи «мышки» (оконный или полноэкранный).



Рисунок 1.1 – IDE Borland Pascal 7.0

Каждому вновь открываемому окну, присваивается следующий по порядку номер. Допускается открытие 99 окон. Переключение между первыми девятью окнами возможно при помощи комбинации клавиш *Alt + <№ окна>*, где *<№ окна>* – цифра от 1 до 9. Для активизации остальных окон необходимо пользоваться командой из пункта меню *Window – List... Alt + 0* (будет рассмотрено ниже). В левом верхнем углу расположена кнопка закрытия окна при помощи мыши [□]. В левом нижнем углу указываются номера строки и столбца, в которой расположен курсор.

В строке статуса отражается состояние *IDE* при выполнении тех или иных операций. При работе с редактором *IDE* в строке статуса располагаются подсказка для наиболее используемых команд.

F1 Help – вызывает контекстную помощь *IDE*.

F2 Save – позволяет сохранить текст программы из активного окна редактирования на диске.

F3 Open – служит для открытия файла с текстом программы для редактирования.

Alt+F9 Compile – компилирует текст исходной программы из активного окна редактирования.

F9 Make – позволяет перекомпилировать текст исходной программы из активного окна редактирования и используемые ей модули.

Alt+F10 Local menu – вызывает локальное меню.

4.2 Главное меню *IDE TurboPascal 7.0*

4.2.1 Пункт меню *File (Alt+F)*

Пункт меню *File* содержит команды для работы с файлами.

Команды, заканчивающиеся многоточием, вызывают диалоговые окна для выполнения своих функций. Справа от некоторых команд изображены клавиши, позволяющие выполнить данную команду из окна редактирования.

New – команда, служащая для создания нового файла с текстом программы. Для этого создаётся новое окно редактирования, которому присваивается следующий по порядку номер.

Open... F3 – служит для вызова диалогового окна открытия файла с текстом программы (рисунок 1.2).

В разделе *Name* можно ввести имя открываемого файла или выбрать его в разделе *Files* при помощи клавиш управления курсором. Для переключения между разделами диалоговых окон служит клавиша *Tab*. Внизу диалогового окна находится информационная строка, показывающая активный каталог, выделенный файл, его размер дату и время создания.

После выделения файла или введения его имени, нажатие кнопки *Open* приводит к открытию файла в новом окне редактирования. При нажатии на кнопку *Replace* файл открывается

в активном окне. Команда *Cancel* во всех диалоговых окнах отменяет выполнение команды (можно использовать клавишу *Esc*). *Help* – служит для вызова справки по данному окну.

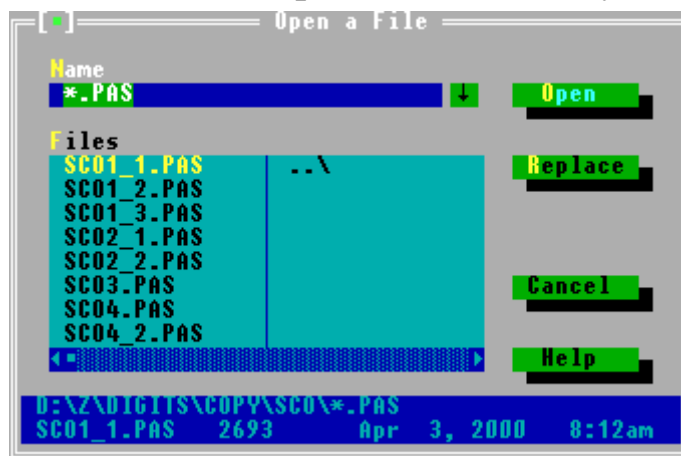


Рисунок 1.2 – Диалоговое окно открытия файла

Save (F2) – сохраняет на внешнем запоминающем устройстве внесённые в файл изменения в процессе редактирования. Если у файла нет имени (*NONAMEXX.PAS*), то вызывается диалоговое окно, позволяющее ввести имя и указать путь для сохраняемого файла.

Save as... – вызывается диалоговое окно, похожее на окно *Open*, позволяющее ввести новое имя и указать путь для сохраняемого файла.

Save all – сохраняет на диске изменения в файлах, находящихся в открытых окнах редактирования.

Change dir... – позволяет изменить рабочий каталог. Диалоговое окно представлено на рисунке 1.3.

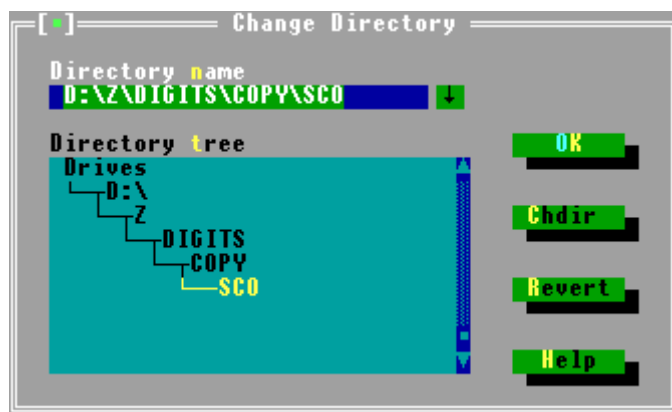


Рисунок 1.3 – Диалоговое окно смены текущего каталога

Выбор каталога осуществляется нажатием *Chdir*. При необходимости вернуть исходное состояние нажимают клавишу *Revert*. После выбора необходимого каталога нажимается клавиша *Ok*.

Print – служит для вывода текста программы из активного окна редактирования на печатающее устройство.

Printer setup... – позволяет настроить печатающее устройство.

DOS shell – временный выход в *DOS*, возврат в *IDE* осуществляется выполнением команды *Exit*.

Exit (Alt+X) – завершение работы с *IDE*. Если в открытых окнах есть не сохраненный текст, то появляются информационные окна, запрашивающие подтверждение на сохранение измененного файла.

4.2.2 Пункт меню *Edit (Alt+E)*

Зачастую при редактировании текста программы возникает необходимость работы с отдельными блоками текста. Блок текста – это любой объем текста (начиная от одного символа), который выделен на экране. В каждый момент времени в окне может быть выделен только один блок. Выделение выполняется следующим образом:

- курсор устанавливается в начало (конец), выделяемого блока;

- при нажатой клавише *Shift*, клавишами управления курсора выделяемая область перемещается в конец (начало) блока.

Меню *Edit* содержит команды, управляющие редактированием.

Undo (Alt+Backspace) – команда отмены действия предыдущей операции редактирования. Если несколько раз выполнять команду *Undo*, редактор продолжает отменять действия. Команда *Undo* вставляет все удаленные символы, удаляет любой вставленный символ, заменяет все перезаписанные символы и перемещает курсор обратно на предыдущую позицию. Если отменяется блочная операция, то файлы выводятся в том виде, в каком они были перед выполнением блочной операции. Команда *Undo* не изменяет установку параметров, влияющих более чем на одно окно. Параметр *Group Undo* в диалоговом окне *Options/Environment|Editor* влияет на поведение команды *Undo* и соответствующей команды *Redo*. При выборе параметра *Group*

Undo (Групповая отмена), и нажатии клавиш *Alt+Backspace* или *Edit/Undo*, редактор изменяет последнюю группу команд. Под группой понимается последовательность команд одного типа.

Redo – отменяет действие команды *Undo*.

Cut (Shift+Del) – служит для вырезки выделенного блока, который удаляется из текста и помещается в буфер обмена (*Clipboard*) – временную область памяти.

Copy (Ctrl+Ins) – служит для копирования выделенного текста в буфер обмена.

Paste (Shift+Ins) – предназначена для вставки содержимого буфера обмена в активное окно, начиная с текущей позиции курсора.

Clear (Ctrl+Del) – удаляет выделенный блок из текста программы. Восстановить удаленный текст можно при помощи команды *Undo*.

На поведение в редакторе выделенных блоков влияют два параметра – *Persistent Blocks* и *Overwrite Blocks*. Они находятся в диалоговом окне *Options/Environment/Editor*. Если включен параметр *Persistent Blocks*, то выделенные блоки остаются выделенными, пока не удалиться или не отмениться выделение (или пока не выделен другой блок). Если параметр *Persistent Blocks* выключен, и курсор перемещен за выделенный блок, то выделение блока отменяется. При выключённом параметре *Persistent Blocks*, параметр *Overwrite Blocks* игнорируется. Если включен параметр *Overwrite Block*, и набирается символ, то выделенный блок заменяется набранным символом. Если параметр *Overwrite Block* включен, а параметр *Persistent Block* выключен, и нажимается клавиша *Del* или *Backspace*, то весь выделенный текст удаляется. Если вставляется текст (нажимая символ или вставляя его из буфера), то весь выделенный текст заменяется вставленным.

Show clipboard – открывает окно с содержимым буфера обмена. Последний помещенный в него блок является выделенным.

4.2.3 Пункт меню *Search (Alt+S)*

Меню поиска *Search* позволяет осуществлять поиск текста, имен процедур и месторасположение ошибок в программе.

Find ... – вызывает диалоговое окно поиска, позволяющее набрать искомый текст и установить параметры, влияющие на поиск:

- [] *Case sensitive* – различать прописные и строчные буквы при поиске;
- [] *Whole words only* – искать только целое слово;
- *Direction* – Направление поиска по тексту...
 - (.) *Forward* – вперед;
 - () *Backward* – назад;
- *Scope* – В какой области текста искать...
 - (.) *Global* – во всем тексте;
 - () *Selected text* – только в отмеченном тексте.
- *Origin* – Начиная с какого места в тексте искать...
 - (.) *From Cursor* – от курсора;
 - () *Entire Scope* – во всей области.

Replace – команда вызывает диалоговое окно, позволяющее набирать образец текста для поиска и образец текста, на который его надо заменить. Работа в диалоговом окне замены похожа на работу в диалоговом окне поиска. Дополнительный параметр, *Prompt on replace*, управляет подсказкой для каждой замены. Кнопка *Change All* производит замены во всем тексте.

Search Again (Ctrl+L) – команда повторяет последнюю *Find* или *Replace* команду. Все установки, которые были сделаны в последнем диалоговом окне (*Find* или *Replace*) остаются действительными при выборе *Search Again*.

Goto to line number – позволяет указать номер строки для быстрого перехода к ней.

Show last compiler error – показывает последнюю ошибку компиляции программы.

Find error – показывает адрес последней ошибки времени исполнения, при возврате в окно редактирования курсор указывает строку, в которой произошла ошибка.

Find Procedure – выводит диалоговое окно, позволяющее ввести имя процедуры для поиска. Команда доступна только во время сеанса отладки. Она позволяет быстро переместить курсор на необходимую подпрограмму.

4.2.4 Пункт меню *Run (Alt+R)*

Команды меню запуска запускают программу из активного окна редактирования, а также начинают и заканчивают сеансы отладки.

Run (Ctrl+F9) – запускает программу, используя параметры, переданные в неё с помощью команды *Run/Parameters...*. Если со времени последней компиляции исходный код был модифицирован, программа будет автоматически перекомпилирована, а затем выполнена.

Step over (F8) – выполняет программу пошагово, без захода в тело процедур и функций. Используется при отладке.

Trace into (F7) – выполняет программу пошагово, с заходом в тело процедур и функций. Используется при отладке.

Go to Cursor (F4) – выполняет программу до строки, на которой стоит курсор в текущем окне редактора. Используется при отладке.

Program reset (Ctrl+F2) – прекращает текущий сеанс работы с программой, освобождает память, размещенную под программу, и закрывает все открытые файлы, используемые программой. Часто используется при отладке.

Parameters... – передаёт в запускаемую программу параметры, которые необходимо вводить с командной строки *DOS*.

4.2.5 Пункт меню *Compile (Alt+C)*

Compile (Alt+F9) – компилирует программу из активного окна редактора. При успешной компиляции, нажимают любую клавишу, для удаления информационного окна компиляции. Если обнаружена ошибка или предупреждение, окно редактирования, содержащее исходный код с ошибкой, становится активным, появляется сообщение об ошибке, а курсор устанавливается на местоположении первой ошибки.

Make (F9) – используется для компиляции программ, состоящих из нескольких модулей. При этом перекомпилируются все модули, в которых были внесены изменения.

Build – используется для компиляции программ, состоящих из нескольких модулей. При этом перекомпилируются все исходные тексты модулей.

Destination – позволяет определить, будет ли выполняемый код храниться на диске (*Disk*) (как файл **.EXE*) или он будет храниться в памяти (*Memory*).

Primary file... – позволяет задать файл **.PAS*, который будет компилироваться, и считаться главным при использовании команд *Compile/Make F9* или *Compile/Build*.

Clear primary file – очищает установку главного файла.

Information – показывает информационное окно состояния программы и памяти.

4.2.6 Пункт меню *Debug Alt +D*

Команды меню *Debug* управляют всеми функциями отладчика.

Breakpoints...– вызывается диалоговое окно, позволяющее установить точку прерывания программы.

Call stack (Ctrl+F3) – открывается окно, в котором показывается содержимое стека (специально организованной области памяти).

Register – появляется окно с содержимым регистров микропроцессора.

Watch – появляется окно наблюдений, в которое можно поместить переменные, наблюдаемые при отладке (*Ctrl+F7*).

Output – активизация окна вывода.

User screen (Alt+F5) – просмотр результатов работы программы на экране.

Более подробно опции пункта меню *Debug* рассмотрены в следующей лабораторной работе.

4.2.7 Пункт меню *Tools (Alt +T)*

В меню *Tools* можно добавить необходимые программы для разработки программ и запускать их потом из *IDE*. Для этого, необходимо выполнить следующее:

1. Для вывода диалогового окна *Tools* выбрать команду *Options/Tools*. В блоке списка заголовков программ *Program Titles* находятся краткие описания уже инсталлированных и готовых к выполнению программ.

2. Для вывода диалогового окна *Modify/New Tool* выбрать команду *New*.

3. В блоке ввода заголовка программы *Program Title* набрать имя программы. Для запуска программы по нажатию клавиш, указать непосредственно перед и после того символа, который должен использоваться в качестве оперативной клавиши, символ тильды (~). Этот символ будет выводиться в меню *Tools* жирным шрифтом и специальным шрифтом, и при нажатии этой клавиши выбирается программа.

4. Для запуска программы по оперативной клавише, выбрать один из параметров *Hot Key*. Когда нажимается присвоенная оперативная клавиша, программа начинает работать.

5. В блоке ввода *Program Path* набрать имя программы.

6. В блоке ввода *Command Line* набрать параметры или макрокоманды, которые необходимо передавать программе. Полную справочную информацию по макрокомандам, которые можно использовать в блоке *Command Line* в *Modify/New Tool* можно найти в оперативном справочнике *Help Borland Pascal*.

4.2.8 Пункт меню *Options Alt+O*

Меню *Options* содержит команды, позволяющие посмотреть и изменить различные установки по умолчанию в *Borland Pascal*. В данной работе команды этого пункта не рассматриваются.

4.2.9 Пункт меню *Window Alt+W*

Из названия пункта меню ясно, что он предназначен для работы с окнами.

Tile – предназначен для расположения окон редактирования на экране «мозаикой» (все окна одного размера и одновременно видны на экране).

Cascade – окна редактирования располагаются каскадом (одно за другим).

Close all – позволяет закрыть все окна редактирования.

Refresh display – обновляет содержимое активного окна редактирования.

Size/Move (Ctrl+F5) – позволяет изменять размер окна редактирования и перемещать его по экрану. При нажатой клавише *Shift* и клавиш управления курсора происходит изменение размера активного окна. Отпустив *Shift* можно переместить окно. Нажатие *Enter* приводит к завершению операций по перемещению и изменению размера.

Zoom (F5) – раскрывает окно в полный экран.

Next (F6) – активизирует следующее по порядку окно.

Previous (Shift+F6) – активизирует предыдущее окно.

Close (Alt+F3) – закрывает активное окно.

List...(Alt+0) – вызывает окно списка активных окон, позволяющее активизировать или закрыть выбранное окно.

4.2.10 Пункт меню *Help Alt+H*

Меню *Help* содержит команды, позволяющие вызывать справочную информацию как по языку *Pascal*, так и по *IDE Borland Pascal 7.0*.

В частности, команда *Contents* вызывает окно с содержанием разделов справки.

Index (Shift+F1) – поиск по ключевым словам.

Topic Search (Ctrl+F1) – контекстно-зависимая справка. Позволяет получить справку о языковой конструкции. Необходимо подвести курсор к какому-либо слову в программе и выбрать эту команду для получения соответствующей справки.

5 Задание для выполнения работы

- 1) Запустить *IDE TurboPascal 7.0*.
- 2) Создать новый файл.
- 3) Набрать в окне текст примера программы:

```
program Lab_1;  
begin  
  WriteLn('Пробуем работать в IDE Borland Pascal  
7.0');  
  WriteLn('5*6 = ',5*6);  
end.
```

- 4) Изменить текущий каталог на корневой.
- 5) Сохранить файл под именем *example.pas*.
- 6) Откомпилировать и запустить программу.
- 7) Просмотреть результаты выполнения программы.
- 8) Выделить весь текст программы и скопировать его в конец (за слово *end*.)
- 9) Найти слово *begin*. Сколько раз встречается?
- 10) Заменить *begin* на *XXX*.
- 11) Отменить сделанные замены.
- 12) Откомпилировать программу еще раз и убедиться, что ошибок нет.
- 13) Удалить фрагмент за словом *end*.
- 14) Сохранить программу.
- 15) Закрыть текущее окно.
- 16) Временно выйти в *DOS* и вернуться в *IDE*.
- 17) Выйти из *IDE TP*.

- 18) Запустить Pascal.
- 19) Открыть файл с именем example.pas.
- 20) Создать новый файл.
- 21) Скопировать содержимое example.pas в новое окно.
- 22) Сохранить новый файл на диске C:\ под именем exampr.pas.
- 23) Переключиться на окно example.pas (3-мя способами).
- 24) Закрывать это окно.
- 25) Изменить размер окна exampr.pas до половины экрана и переместить его.
- 26) Развернуть на весь экран.
- 27) Скопировать слово program и вставить его перед и после begin.
- 28) Удалить эти строки.
- 29) Получить контекстно-зависимую справку для оператора вывода WriteLn.
- 30) Как вызвать обычную справку?
- 31) Заменить WriteLn на Writeln.
- 32) Откомпилировать, сохранить и запустить программу.
- 33) Просмотреть результаты выполнения.
- 34) Закрывать текущее окно.
- 35) Выйти из IDE TP.

Лабораторная работа №2. Программы линейной структуры

1 Цель работы

Изучить структуру программы на языке Паскаль, операторы присваивания, ввода и вывода данных, используемые при проектировании программ линейной структуры.

2 Порядок выполнения работы

Получить задание на выполнение лабораторной работы (раздел 6) согласно своему варианту. Разработать и отладить программу. Составить и защитить отчет о лабораторной работе у преподавателя.

3 Содержание отчета

- наименование и цель работы;
- задание на лабораторную работу согласно варианту;
- схема алгоритма, текст программы на алгоритмическом языке;
- результаты работы программы.

4 Краткая теория

4.1 Структура программы на языке Паскаль

Любая программа, написанная на языке Turbo Pascal 7.0 имеет следующие разделы:

РАЗДЕЛ ОБЪЯВЛЕНИЙ

program	Заголовок программы
{ \$... }	Глобальные директивы компилятора
uses	Подключаемые библиотеки (модули)
label	Раздел объявления глобальных меток
const	Раздел объявления глобальных констант
type	Раздел объявления глобальных типов
var	Раздел объявления глобальных переменных

РАЗДЕЛ ПРОЦЕДУР И ФУНКЦИЙ

procedure (function	Заголовок процедуры (функции)
n)	{тело процедуры}

РАЗДЕЛ ОСНОВНОГО БЛОКА ПРОГРАММЫ

begin	Основной блок программы
end.	

В тексте программы могут присутствовать комментарии. Они важны для документирования (пояснения) программы. В комментариях указывается вся информация, которая предназначена для разъяснения программы и упорядочения программного текста. Комментарии представляют собой

произвольный текст, ограниченный с двух сторон фигурными скобками **{...}** или **(* ...*)**.

4.2 Операторы языка программирования

Оператором называется элементарная структурная единица программы. Оператор предназначен как для записи алгоритмических действий по преобразованию данных, так и для задания порядка выполнения других действий. Операторы выполняются в порядке их следования в программе. Операторы отделяются друг от друга точкой с запятой. Операторы делятся на:

- простые (не содержат в себе других операторов);
- составные (включают в себя один или несколько дополнительных операторов).

4.2.1 Оператор присваивания

С помощью данного оператора определяется или переопределяется значение переменной, стоящей слева от :=.

Синтаксис оператора:

<имя_переменной> := <выражение>;

Выражение должно быть совместимо по присваиванию с типом переменной.

Примеры:

```
X := Pi ;  
Y := Sqr (Z*Z+T*T) ;  
Bool := (I>1) or (I<3) ;
```

4.2.2 Операторы ввода данных

Операторы ввода предназначены для ввода исходных данных в программу.

Операторы ввода имеют вид:

```
READ (X, Y, Z, . . .) ;  
READLN (X, Y, Z, . . .) ;
```

где: X,Y,Z,... - имена переменных. Данные операторы обеспечивают ввод данных с клавиатуры и присваивают переменным X,Y,Z,... соответствующие значения.

Оператор READLN (X,Y,Z,...); отличается от READ(X,Y,Z,...); тем, что после ввода последней переменной курсор переводится в начало новой строки. Допускается использование оператора ввода без параметров: READLN;

4.2.3 Операторы вывода данных

Для вывода данных на экран используются операторы:

```
WRITE (X,Y,Z, . . . ) ;  
WRITELN (X,Y,Z, . . . ) ;  
WRITELN ;
```

где: X,Y,Z,... - имена переменных, значения которых подлежат выводу.

Оператор WRITE(X,Y,Z,...); выполняет вывод на экран в одну строку.

Оператор WRITELN (X,Y,Z,...); кроме вывода на экран значений X,Y,Z,..., осуществляет перевод курсора в начало новой строки после вывода последнего значения.

Оператор WRITELN; осуществляет пропуск строки, в которой находится курсор, и переводит его в начало новой строки.

Операторы вывода допускают использование указания о ширине поля, отводимого под значение в явном виде:

```
WRITE (Y:m:n,X:k:l, . . . ) ;  
WRITELN (Y:m:n, X:k:l, . . . ) ;
```

где m и k - количество позиций, отведенных под запись значения переменных Y и X соответственно; n и l - количество позиций, отведенных под запись дробной части чисел Y и X.

Примеры:

```
WRITE('Сумма = ',Sum); {вывод на экран строки «Сумма = », затем значение суммы, которое хранится в переменной Sum}  
WRITELN('Введите ',I,'-й элемент: ');
```

5 Примеры программ

5.1 Программа нахождения среднего из двух целых чисел и одного вещественного числа:

```
program Middling;  
var  
  X, Y: Integer;  
  Z, Midd: Real;  
begin  
  WriteLn('Введите два целых числа X и Y:');  
  ReadLn(X,Y);  
  WriteLn('Введите вещественное число Z:');
```

```

    ReadLn (Z) ;
    Midd:=(X+Y+Z) /3;
    Write ('Среднее арифметическое = ', Midd) ;
end.

```

5.2 Программа перевода дюймов в сантиметры (1 дюйм = 2,54 см).

```

program Inches;
var
    Inch, Centimeter: Real;
begin
    Write ('Введите величину в дюймах = ');
    ReadLn (Inch) ;
    Centimeter:= Inch * 2.54;
    Write ('Величина           в           сантиметрах
= ', Centimeter:5:2) ;
end.

```

5.3 Программа вычисления значения выражения:
 $Y = \frac{a \cdot \sin(b)}{w^{15}}$, где $a = 12.4$, $b = 15^\circ 25'$, w - ввести. (Для возведения в степень используется следующее соотношение: $a^x = e^{x \ln a}$).

```

program Calculate;
var
    y, a, b, w: Real;
begin
    a:=12.4;
    b:=(15 + 25/60)*Pi/180; {перевод из градусов в радианы}
    Write ('Введите w = ');
    ReadLn (w) ;
    y:= a*sin(b) /exp(ln(w)*15) ;
    WriteLn ('Y = ', Y:5:3) ;
end.

```

6 Варианты заданий для самостоятельного решения

1. В году примерно 3.156×10^7 сек. Написать программу, которая запрашивает возраст в годах и переводит его в секунды.
2. Масса m одной молекулы воды примерно равна 3.0×10^{-23} гр. Кварта воды равна примерно 950 гр. Написать программу,

которая запрашивает количество воды в квартах и выводит число молекул в этом количестве воды.

3. Вычислить: $a = \frac{2 \cos(x - 30^\circ)}{\frac{1}{3} + \sin^2(y + 2)}$, $y = \frac{z^2}{5 + \frac{z^2}{7}}$, $x = 1.236$, $z = 4.5$.

4. Найти периметр и площадь прямоугольника.

5. Вычислить сумму депозита с процентами

$S = K \cdot (1 + P/100)^n$, где K - капитал, P - процентная ставка, n - число периодов начисления процентов.

6. Вычислить: $y = 2 \ln x - \operatorname{tg}(x)$.

7. Время в полете вычисляется по формуле $T = S/V$, где S - расстояние между пунктами, V - скорость полета самолета. Определить время в полете при отсутствии ветра и при наличии встречного ветра (V_1 - скорость ветра).

8. Написать программу, которая запрашивает время в часах и минутах, после чего переводит в минуты, затем в секунды и выводит результаты на экран.

9. Найти площадь всей поверхности цилиндра: $T = 2\pi R(R + H)$.

10. Вычислить: $y = x^5 + \cos(vt)$, $x = \frac{v^3}{-t^2}$, $v = |t|$, t - ввести.

11. Написать программу, которая запрашивает количество дней и переводит в недели и дни. Например, 18 дней = 2 недели и 4 дня.

12. Вычислить:

$$y = 2a - b \cdot c, a = \frac{\sqrt[4]{\frac{1}{2} - v \cdot |w^3 - 100|}}{b/13}, b = \frac{v}{w^2}, c = 1.5, w, v - \text{ввести}.$$

13. Найти площадь боковой поверхности шара: $T = 4\pi R^2$.

14. Русские неметрические единицы длины: 1 аршин = 16 вершков; 1 вершок = 44,45 мм. Длина некоторого отрезка составляет N метров. Перевести ее в аршины.

15. Вычислить значение функции $Y = \sin(a+d) + d \cos(a/2)$

16. Определить, какую долю в процентном соотношении составляет число A_1 от A_2 по формуле $P = A_1 / A_2 \cdot 100$.

17. Найти объем куба (с использованием стандартной функции и без).

18. Вычислить:

$$g = a + \left| \frac{\sin(q+a)}{b+c} \right|^{0.5}, \text{ При } b = \frac{3.2a}{a+b+c}, c = 0.07a^2, a = 1.75, q = 48^\circ 24' 18''$$

19. Определить расстояние, пройденное физическим телом за время t , если тело движется с постоянной скоростью v .
20. Найти площадь поверхности куба.
21. Определить расстояние между двумя точками с координатами (x_1, y_1) и (x_2, y_2) .
22. Вычислить: $y = \cos x - \frac{x^2 - 2x}{2 + (x^3 - 1)}$.
23. Найти объем цилиндра по формуле: $V = \pi R^2 H$.
24. Вычислить: $y = a + b + c$, $a = \frac{\sqrt[3]{w-20}}{1 - (v + b/13)}$, $b = \frac{w}{v}$, $c = 25$, w, v – ввести.
25. Время в пути вычисляется по формуле $T = S/V$, где S – расстояние между пунктами, V – скорость движения катера. Определить время в пути при движении по течению реки и против течения (V_1 – скорость течения).
26. Найти расстояние от точки с координатами (x, y) до начала координат.
27. Вычислить: $y = \ln x + tg(z)$, $x = \frac{2.4z}{1 - z^2}$, $z = \sqrt{|2v - 10|}$, $v = 20.3$.
28. Даны длины сторон оснований усеченной пирамиды и высота. Найти объем пирамиды по формуле: $V = \frac{H}{3}(S_1 + S_2 + \sqrt{S_1 S_2})$, где S_1, S_2 – площади оснований усеченной пирамиды, H – высота.
29. Русские неметрические единицы длины: 1 верста=500 сажень; 1 сажень=3 аршина; 1 аршин=16 вершков; 1 вершок=44,45 мм. Длина некоторого отрезка составляет p метров. Перевести ее в русскую неметрическую систему.
30. Трехмерные вектора заданы своими координатами: $A=(x_a, y_a, z_a)$ и $B=(x_b, y_b, z_b)$. Найти угол (в градусах) между векторами A и B , используя формулу: $\cos \varphi = \frac{(A, B)}{|A| \cdot |B|} = \frac{x_a x_b + y_a y_b + z_a z_b}{\sqrt{x_a^2 + y_a^2 + z_a^2} \cdot \sqrt{x_b^2 + y_b^2 + z_b^2}}$.
31. Найти координаты вершины параболы $y = ax^2 + bx + c$.
32. Найти диагональ и площадь квадрата.

Лабораторная работа №3. Операторы ветвления: условный оператор, оператор варианта

1 Цель работы

Изучить операторы, используемые для организации ветвления в программе. Познакомиться с логическими выражениями и операциями.

2 Порядок выполнения работы

Получить задание на выполнение лабораторной работы согласно своему варианту (раздел 6). Разработать и отладить программу. Составить и защитить отчет о лабораторной работе у преподавателя.

3 Содержание отчета

- наименование и цель работы;
- задание на лабораторную работу согласно варианту;
- схема алгоритма, текст программы на алгоритмическом языке;
- результаты работы программы.

4 Краткая теория

С помощью условного оператора и оператора выбора реализуется операция ветвления алгоритма, то есть с помощью этих операторов, в зависимости от результатов проверки условия, выбирается одно из множества потенциально возможных направления дальнейшего хода вычислительного процесса.

4.1 Условный оператор If

Условный оператор **If** позволяет произвести развилку алгоритма, в которой осуществляется выбор одной из двух альтернативных ветвей в зависимости от некоторого условия (при этом одна из ветвей может отсутствовать). В качестве условия выбора используется значение логического выражения. Оно может быть простым или сложным. Сложные условия организуются при помощи логических операций AND, OR, NOT.

В целом синтаксис оператора **If** можно представить в следующем виде:

```
if <логическое_выражение> then <оператор1>
[else <оператор2>]
```

В результате вычисления выражения получается логическое значение типа Boolean. Если результатом является значение TRUE, то выполняется <оператор1>, следующий за служебным словом then. Если же в результате выполнения получается значение FALSE, то выполняется <оператор2>, следующий за Else. Если часть условного оператора, начиная со слова Else, отсутствует, то управление передается оператору, следующему за условным.

Следует отметить, что использование в операторе **If** после служебного слова then второго оператора **If** приводит к синтаксически неоднозначной конструкции вида:

```
if <выражение1> then
    if <выражение2> then
        <оператор1>
    else
        <оператор2>
```

Компилятор Турбо Паскаль разрешает подобную неоднозначность, интерпретируя этот фрагмент следующим образом:

```
if <выражение1> then
begin
    if <выражение2> then
        <оператор1>
    else
        <оператор2>
end;
```

То есть, служебное слово else всегда ассоциируется с ближайшим по тексту служебным словом if, которое еще не связано со служебным словом else.

Примеры:

```
1) if X>Y then Max:=X else Max:=T;
2) if X>0 then X:=Sqrt(X);
3) if Ch=1 then
    begin
        X:=Sin(X);
        WriteLn('Sin(x) = ', X);
    end
```

```

else
begin
  X:=Cos (X) ;
  WriteLn ( 'Cos (x) = ' ,X) ;
end;

```

```

4) if (A=5) and (B=5) then
  if C mod 2 = 0 then
    K:=(A+B)*C
  else
    K:=(A+B)/C
  else
    WriteLn('A и B не равны 5');
5) if D>0 then
  begin
    if C mod 2 = 0 then
      K:=D*C;
    end
  else
    K:=Sqr (D) ;

```

В пятом примере служебное слово else относится к первому if, а не к ближайшему. Порядок изменен при помощи операторных скобок begin...end.

4.2 Оператор варианта CASE

Оператор **Case** производит развилку алгоритма на произвольное множество ветвей. Из этого множества либо выбирается единственная ветвь, отвечающая одному из заданных условий, либо ни одной, если ни одно из условий не выполняется.

Оператор варианта состоит из выражения (селектора) и списка операторов, каждому из которых предшествует одна или более констант (они называются константами выбора) или ключевое слово else. Селектор должен иметь порядковый тип. Строковый тип и длинный целый тип являются недопустимыми типами селектора. Все константы выбора должны быть уникальными и иметь порядковый тип, совместимый с типом селектора.

Синтаксис:

```
case <селектор> of
```



```

<константа_выбора1>: <оператор1>;
<константа_выбора2>: <оператор2>;
...
<константа_выбораN>: <операторN>;
[else <оператор>]
end;

```

Оператор **Case** приводит к выполнению оператора, которому предшествует константа выбора, равная значению селектора или диапазону выбора, в котором находится значение селектора. Если такой константы выбора или такого диапазона выбора не существует и присутствует ветвь **else**, то выполнятся оператор, следующий за ключевым словом **else**. Если же ветвь **else** отсутствует, то никакой оператор не выполняется.

Примеры:

```

1) case Ch of
    'A'..'Z': WriteLn('Латинский алфавит');
    'А'..'Я': WriteLn('Русский алфавит');
end;

2) case I of
    0, 2, 4, 6, 8: WriteLn('Четная цифра');
    1, 3, 5, 7, 9: WriteLn('Нечетная
цифра');
else
    WriteLn('Цифра не принадлежит
диапазону 0..9!');
end;

```

5 Примеры программ

5.1 Решить уравнение: $y = \begin{cases} \cos^2 x & \text{при } 0 < x < 2, \\ 1 - \sin x^2 & \text{иначе.} \end{cases}$

```

program Solution;
var
    x, y: Real;
begin
    Write('Введите x = ');
    ReadLn(x);
    if (x>0) and (x<2) then
        y := sqr(cos(x))

```

```

else
  y:= 1-sin(x*x);
  WriteLn('y = ',y:5:2);
end.

```

5.2 Ввести символ латинского алфавита из диапазона А..Е. Определить, какая буква была введена – гласная или согласная?

```

Program LatSymbols;
var
  c: Char;
begin
  Write('Введите лат.символ из диапазона А..Е:
');
  ReadLn(c);
  case c of
    'A', 'a', 'E', 'e': WriteLn('Гласная');
    'B', 'b', 'C', 'c', 'D', 'd':
WriteLn('Согласная');
  else
    WriteLn('Символ не принадлежит диап.
А..Е');
  end;
end.

```

6 Варианты заданий для самостоятельного решения

6.1 Оператор IF

Вариант 1

1. Может ли шар радиуса r пройти через ромбообразное отверстие с диагоналями p и q ?
2. Ввести с клавиатуры a, b, c, d и вычислить:

$$y = \frac{\min\{a, b\} \cdot \max\{b, c, \min\{a, d\}\}}{\min\{a, \max\{c, d\}\}}.$$

Вариант 2

1. Заданы три числа: a, b, c . Определить, могут ли они быть сторонами треугольника, и если да, то определить его тип:

- равносторонний, равнобедренный, разносторонний. (Условия существования треугольника: $a \leq b + c$; $b \leq a + c$; $c \leq a + b$).
2. Среди заданных целых чисел k, l, m найти пары кратных.

Вариант 3

1. Можно ли коробку размером $a \times b \times c$ упаковать в посылку размером $r \times s \times t$? «Углом» укладывать нельзя.
2. Вычислить:

$$y = \sqrt{\text{среднее}(x, y, z, v)},$$
 (x, y, z, v – вводятся с клавиатуры и принадлежат $[-7; 7]$).

Вариант 4

1. Если значение переменной W не равно 0 и при этом котангенс от W меньше 0.5, тогда поменять знак у W , а если значение W равно 0, то присвоить W значение 1.
2. Вычислить:

$$y = \frac{1 - \max(a, b)}{\min(c, d)} - \text{среднее}(a, b, c, d),$$
 (a, b, c, d вводятся с клавиатуры).

Вариант 5

1. Составить алгоритм и программу вычисления функции, x – вводится с клавиатуры:
- $$y = \begin{cases} \sqrt{\sin^2 x^{1/3} + \cos^3 x^{1/2}} & , \text{при } x \in [-1; 1] \\ \frac{1}{\log_2 x} + x^{\ln x} & , \text{или } x \notin [-1; 1] \end{cases}$$
2. Число делится на 3 тогда и только тогда, когда сумма его цифр делится на 3. Проверить этот признак на примере заданного трехзначного числа.

Вариант 6

1. Найти среднее арифметическое трех чисел. Если оно больше некоторого числа A , то возвести его в квадрат, иначе удвоить.
2. Вычислить:
$$y = \begin{cases} x \cdot \sqrt{x^2 - 2x}, & \text{при } 3 \leq x \leq 10; \\ \cos x + \sin x, & \text{если } x < 3 \text{ или } x > 10. \end{cases}$$

Вариант 7

1. Вычислить $y = a \cdot \min(x, y, z) - b$, где x, y, z, a, b вводятся, причем a, b должны быть кратны 5, иначе выдать соответствующее сообщение.
2. Ввести возраст человека в годах и в зависимости от числа добавить «лет» или «года». Например, 25 лет, 2 года.

Вариант 8

1. Даны действительные числа a, b, c, x, y . Выяснить, пройдет ли кирпич с ребрами a, b, c в прямоугольное отверстие со сторонами x и y . Просовывать кирпич в отверстие разрешается только так, чтобы каждое из его ребер было параллельно или перпендикулярно каждой из сторон отверстия.
2. Написать программу, которая проверяла бы справедливость неравенства, в результате выводила бы True или False:

$$\frac{1}{n+2} < \sin(n^2 + 5) < \frac{1}{2n}.$$

Вариант 9

1. Составить алгоритм и программу вычисления функции (x вводится)
$$y = \begin{cases} x + \sin^3 x^2, & \text{при } x^2 \geq 0.5; \\ x - \cos^2 x^3, & \text{иначе.} \end{cases}$$
2. Вычислить: $y = \min\{\max\{a, b\}, c\}$, a, b, c, d – даны.

Вариант 10

1. Составить алгоритм и программу вычисления функции, x, a, b, c – вводится с клавиатуры.

$$y = \begin{cases} 2x - 1, & \text{если } a + b + c > 20; \\ 3x^3 + 1, & \text{иначе.} \end{cases}$$

2. Вычислить $y = \min\{a, b, c\} + \max\{d, f\}$, где a, b, c, d, f – даны.

Вариант 11

1. Составить алгоритм и программу нахождения корней уравнения $ax^2 + bx + c = 0$. Предусмотреть варианты двух, одного корня, а также, когда действительных корней нет.
2. Вычислить $y = \max\{a, b, \min\{c, d\}\}$, a, b, c, d – числа.

Вариант 12

1. Составить алгоритм и программу вычисления функции, x вводится с клавиатуры.

$$y_0 = \begin{cases} \sqrt{x^2 - 2x - 3}, & \text{если } x - 1.2 > 0; \\ 1, & \text{если } x - 1.2 = 0; \\ x(8 - x(7 - x)), & \text{если } x - 1.2 \leq 0. \end{cases}$$

2. Вычислить:

$$y = \frac{\max(a, b)}{1 - \min(c, d)} \quad (a, b, c, d - \text{вводятся}).$$

Вариант 13

1. Известно, что из четырёх чисел a, b, c, d одно отлично от трех других, равных между собой. Найти сумму данного числа с любым из трех оставшихся.
2. Вычислить: $U = \max(x, y) + 2 * (\min(x, y, z))$, (x, y, z – вводятся с клавиатуры).

Вариант 14

1. Составить алгоритм и программу вычисления функции, x – вводится с клавиатуры.

$$y = \begin{cases} x^3 - x^2 + 3, & \text{при } x \in [1; 5]; \\ 2x^2 - x^3, & \text{иначе.} \end{cases}$$

2. Вычислить: (a, b, c, d – вводятся с клавиатуры)

$$y = \frac{\max(a, b) + \min^2(b, c)}{1 - \max^3(c, d)}.$$

Вариант 15

1. Составить алгоритм и программу вычисления функции y_0 (x, y – даны)

$$y_0 = \begin{cases} \max(x, y + 5), & \text{если } x > y; \\ \min(x + 1, y + 3), & \text{иначе.} \end{cases}$$

2. Может ли шар радиуса r пройти через прямоугольное отверстие со сторонами A и B .

Вариант 16

1. Составить алгоритм и программу вычисления функции, x , y – вводится с клавиатуры.

$$y_0 = \begin{cases} \sqrt{x - \sin x^2 - 10}, & \text{если } x - \sin(x - 3) < 5; \\ 1, & \text{если } x - \sin(x - 3) = 5; \\ 3 \cdot \cos x, & \text{если } x - \sin(x - 3) > 5. \end{cases}$$

2. Написать программу, которая в зависимости от введенных A , B и C считает их произведение, если они все положительны; или сумму, если хотя бы одно отрицательно.

Вариант 17

1. Написать программу проверки трех введенных чисел на четность/нечетность. Вывести, сколько из них четно, сколько нечетно.
2. Вычислить $y = a + \max\{a, b, \min\{c, d\}\}$, a, b, c, d – числа.

Вариант 18

1. Найти максимум из трех чисел. Если он больше некоторого числа A , то найти сумму этих чисел, иначе – произведение.
2. Ввести двузначное число и найти сумму цифр этого числа, если оно четно, иначе найти разность цифр.

Вариант 19

1. Написать программу, которая проверяла бы справедливость неравенства, в результате выводила бы True или False:

$$\frac{1}{n+1} < \ln\left(\frac{n+1}{n}\right) < \frac{1}{n}, \quad (n \text{ вводится с клавиатуры}).$$

2. Ввести четыре числа. Найти сумму четных и сумму нечетных.

Вариант 20

1. Вычислить сумму A, B и C , если все они четны, разность, если есть хотя бы одно нечетное.

2. Найти $\max(\min(a,b);\min(c,d))$.

Вариант 21

1. Вычислить:

$y = \min(x, y) - \text{среднее}(a, b, z)$, где x, y, z, a, b -вводятся, причем a – нечетно, b кратно 3-м, иначе выдать соответствующее сообщение.

2. Числа a, b, c тогда и только тогда являются сторонами треугольника, когда существуют такие положительные x, y, z что

$$\begin{cases} a = x + y; \\ b = y + z; \\ c = x + z. \end{cases} \text{ . Найти } x, y, z.$$

Вариант 22

1. Вычислить:

$y = b \cdot \max(x, y, z) + a$, где x, y, z, a, b -вводятся, причем a, b четные, иначе выдать соответствующее сообщение.

2. Ввести три числа и определить, есть ли среди них четные и кратные 3-м одновременно.

Вариант 23

1. Ввести три числа x, y, z и проверить выполняется ли условие $x > y > z$.

2. Вычислить (x вводится с клавиатуры).

$$y = \begin{cases} x \cdot \sqrt{x^2 - 16}, & \text{при } 5 \leq x \leq 10; \\ \cos(x) + e \cdot \sin(x), & \text{если } x < 5 \text{ или } x > 10. \end{cases}$$

Вариант 24

1. Известно, что из четырёх чисел a, b, c , одно отлично от двух других, равных между собой. Найти сумму данного числа с любым из двух оставшихся.

2. Вычислить: $U = \min(x, y) + 2 * (\max(x, y, z))$, (x, y, z – вводятся с клавиатуры).

Вариант 25

1. Найти максимум из трех чисел. Если он больше некоторого числа S , то найти среднее арифметическое этих чисел, иначе – произведение.
2. Ввести четыре числа. Найти произведение всех четных и сумму нечетных.

Вариант 26

3. Составить алгоритм и программу нахождения корней уравнения $ax^2+bx+c=0$. Предусмотреть варианты двух, одного корня, а также, когда действительных корней нет.
4. Вычислить $y=\max\{a,b,\min\{c,d\}\}/\max\{b,d\}$, a, b, c, d – числа.

6.2 Оператор варианта

1. Реализовать в программе меню выбора логических операций. В зависимости от варианта посчитать значение произвольного выражения, операндами которого являются 0 и 1.
2. Написать программу, которая ждет ввода знака арифметической операции и, в зависимости от введенного символа, вычисляет значение произвольного арифметического выражения.
3. Написать программу, которая запрашивает ввод знака арифметической операции и выдает ее название.
4. Написать программу, которая запрашивает размер ноги и выдает ориентировочный рост.
5. Ввести код цвета и определить его категорию: теплый или холодный.
6. Написать программу, которая ждет ввода буквы и, в зависимости от введенного значения, сообщает, является ли буква заглавной или строчной.
7. По введенному двузначному числу определить к какому десятку оно относится. Например, 12 – второй десяток.
8. Написать программу, которая запрашивает ваш возраст в годах, а затем отображает информацию о периоде вашей

- жизни: детство, отрочество, юность, молодость, зрелость, старость.
9. Написать программу, которая требует ввода времени дня и, в зависимости от введенного значения, желает доброго утра, доброго дня, доброго вечера или спокойной ночи.
 10. Написать программу определения цвета в зависимости от введенного числа.
 11. Реализовать в программе меню выбора арифметических действий. В зависимости от варианта посчитать значение выражения $Y := X \{ + | - | * | / \} A$. X и A вводятся.
 12. Написать программу, которая ждет ввода первой буквы названия месяца. В зависимости от этого определяет квартал. Учесть: Июнь, Июль – разные кварталы.
 13. Написать программу, которая требует ввода числа месяца и, в зависимости от введенного значения, сообщает номер декады.
 14. Написать программу, которая требует ввода номера месяца, числа и, в зависимости от введенного значения, сообщает, сколько дней осталось до нового года.
 15. Написать программу, которая по номеру курса выдает номера учебных семестров этого курса.
 16. Заданное натуральное число N, не превосходящее 100, записать прописью, то есть вывести соответствующее количественное числительное, например: 25 – «двадцать пять».
 17. По введенной дате (число, месяц) определить знак зодиака.
 18. Написать программу, где ожидается ввод символа с клавиатуры, после чего определяется, является ли этот символ буквой, цифрой или специальным символом.
 19. Написать программу, которая запрашивает ваше имя и рост в сантиметрах, а затем отображает информацию в виде: высокий, средний, низкий.
 20. Написать программу, которая запрашивает у пользователя месяц рождения и выдает сезон, в котором человек родился.
 21. Написать программу, которая ждет ввода буквы русского алфавита и в зависимости от этого определяет гласная или согласная. Если согласная, то звонкая или глухая.

22. Написать программу, которая ждет ввода первой буквы названия месяца. В зависимости от этого определяет время года.
23. По углу определить вид треугольника: прямоугольный, тупоугольный, остроугольный.
24. Представить заданное целое число от 1 до 10 в римской системе счисления.
25. Написать программу, которая запрашивает ввод знака операции отношения и выдает ее название.
26. Написать программу, которая запрашивает рост и выдает предполагаемый вес.
27. Необходимо преобразовать целое число N в зависимости от величины остатка от его деления на 17 следующим образом:
 если $N \bmod 17=0$, то $N=0$;
 если $N \bmod 17=1$ или 6, то $N=-N$;
 если $N \bmod 17=2, 3$ или 5, то $N=E*N$;
 если $N \bmod 17=4$, то $N=3*N$;
 иначе $N=5*N$.

28. По количеству артистов коллектива определить его тип:
 1 – соло;
 2 – дуэт;
 3 – трио;
 4 – квартет и т.д.

29.
$$\left\{ \begin{array}{ll} \cos(x) & x = 1 \\ \sin(x) & x = 2 \\ x & x = 3 \\ x^2 & x = 4 \\ mx & x = 5 \\ 1/x & x = 6 \\ x^3 & x = 7 \\ x^4 & x = 8 \\ \sqrt{x} & x = 9 \end{array} \right.$$

30. Вычислить:

$$y = \left\{ \begin{array}{ll} x * x, \text{при} & x \bmod 3 = 0 \\ x + x, \text{при} & x \bmod 3 = 3 \\ x + 10, \text{при} & x \bmod 3 = 9 \\ x - 5, \text{при} & x \bmod 3 = 6 \end{array} \right.$$

$y=x$ в других случаях.

31. Написать программу, которая запрашивает размер ноги и выдает ориентировочный рост.

155 и менее = 36;

160-165 = 37;

166-170 = 38;

171-175 = 39 и т.д.

Лабораторная работа №4. Циклические конструкции. Арифметические циклы

1 Цель работы

Изучить операторы циклов, используемые для организации повторяющихся процессов в программах.

2 Порядок выполнения работы

Получить задание на выполнение лабораторной работы согласно своему варианту. Разработать и отладить программу. Составить отчет о лабораторной работе и защитить его у преподавателя.

3 Содержание отчета

- наименование и цель работы;
- задание на лабораторную работу согласно варианту;
- схема алгоритма, текст программы на алгоритмическом языке;
- результаты работы программы.

4 Краткая теория

Оператор цикла задает повторное выполнение определенных операторов. Паскаль предусматривает использование трех операторов цикла:

- оператор цикла с постусловием **repeat**;
- оператор цикла с предусловием **while**;
- оператор цикла с параметром **for**.

Если число повторений заранее известно, то подходящей конструкцией является оператор **for**. В противном случае следует

использовать операторы `while` или `repeat`. Для управления повторением операторов можно использовать стандартные процедуры `Break` и `Continue`. `Break` завершает оператор цикла и передает управление первому непосредственно следующему за циклом оператору, а `Continue` завершает текущую итерацию и продолжает со следующей итерации этого оператора.

4.1 Оператор цикла с постусловием

Оператор цикла с постусловием начинается со служебного слова `repeat`, заканчивается после выполнения условия, следующего за служебным словом `until`.

Синтаксис записи оператора:

```
repeat  
  <последовательность операторов>;  
until <выражение>;
```

Результат выражения должен быть булевского типа. Операторы, заключенные между ключевыми словами `repeat` и `until`, выполняются последовательно до тех пор, пока результат выражения не примет значение `True` (то есть если выражение есть `false`, то цикл повторяется). Последовательность операторов выполнится, по крайней мере, один раз, поскольку вычисление выражения производится после каждого выполнения последовательности операторов.

Применяя оператор `repeat`, следует соблюдать определенную осторожность, чтобы не допускать закливания. Это может произойти, если проверяемое выражение постоянно сохраняет значение `False`. Поэтому необходимо следить, чтобы в теле цикла были выражения, которые влияют на изменение значения проверяемого выражения.

Примеры оператора цикла с постусловием:

```
1) repeat  
  K := I mod J;  
  I := J;  
  J := K;  
until J = 0;
```

(Цикл будет выполняться ДО ТЕХ ПОР, ПОКА J не станет равно нулю.)

```
2) repeat
```

```
Write('Введите значение (0..9):');  
Readln(I);  
until (I >= 0) and (I <= 9);
```

4.2 Оператор цикла с предусловием

Оператор цикла с предусловием (начинающийся с ключевого слова `while`) содержит в себе выражение, которое управляет повторным выполнением оператора (который может быть составным оператором).

Синтаксис записи оператора:

```
while <выражение> do  
  <оператор>;
```

Выражение, с помощью которого осуществляется управление повторением оператора, должно иметь булевский тип. Вычисление его производится до того, как внутренний оператор будет выполнен. Внутренний оператор выполняется повторно до тех пор, пока выражение имеет значение `True`. Если выражение с самого начала принимает значение `False`, то цикл не выполняется ни разу.

Применяя оператор `while`, также следует соблюдать осторожность, чтобы не допускать заикливания. Поэтому необходимо следить, чтобы в теле цикла были выражения, которые влияют на изменение значения проверяемого выражения.

Примеры оператора цикла с предусловием:

1) (нахождение факториала числа X)

```
Fact:=1;  
while X<=N do  
begin  
  Fact:=Fact*X;  
  X:=X+1;  
end;
```

2) `while I <> 10 do`

`I:=Random(10);` {функция `Random(10)` генерирует случайное число $0 \leq I < 10$ }

4.3 Оператор цикла с параметром

Оператор цикла с параметром (начинающегося со слова `for`) вызывает повторяющееся выполнение оператора (который может быть составным оператором) пока переменной цикла присваиваются последовательно возрастающие или убывающие значения.

Оператор имеет два варианта записи:

- инкрементный (с возрастанием переменной цикла);

```
for <переменная цикла>:=<начало> to <конец>  
do  
    <оператор>;
```

- декрементный (с убыванием переменной цикла);

```
for  
<переменная цикла>:=<начало> downto <конец> do  
    <оператор>;
```

В качестве переменной цикла должна использоваться переменная порядкового типа, объявленная в том блоке, где содержится оператор `for`. Параметры `<начало>` и `<конец>` задаются выражениями, типы которых должны быть совместимы по присваиванию с типом переменной цикла. Когда начинает выполняться оператор `for`, начальное и конечное значения определяются один раз, и эти значения сохраняются на протяжении всего выполнения оператора `for`. Оператор, который содержится в теле оператора `for`, выполняется один раз для каждого значения в диапазоне между начальным и конечным значением. Переменная цикла всегда инициализируется начальным значением.

Когда работает оператор `for`, значение переменной цикла (счетчика циклов) увеличивается (в случае инкрементного варианта) при каждом повторении на единицу. Если начальное значение превышает конечное значение, то содержащийся в теле оператора `for` оператор не выполняется. (Операторы в теле цикла не выполняются ни разу, если конечное значение меньше начального).

В случае декрементного варианта цикла значение переменной цикла уменьшается при каждом повторении на единицу. Если начальное значение в таком операторе меньше, чем конечное значение, то содержащийся в теле оператора цикла оператор не выполняется. (Операторы в теле цикла не выполняются ни разу, если конечное значение больше начального).

Если оператор, содержащийся в теле оператора `for`, изменяет значение управляющей переменной, то это является ошибкой. После выполнения оператора `for` значение управляющей переменной становится неопределенным.

Примеры оператора цикла с параметром.

1) *Инкрементный вариант.*

Вычислить сумму десяти последовательно введенных чисел.
Для этого напишем следующий фрагмент программы:

```
...
Sum:=0;
for I:=1 to 10 do
begin
  Write('Введите число №' ,I, ' = ');
  ReadLn(X);
  Sum:= Sum + X;
end;
```

...
2) *Декрементный вариант.*

Обратный отсчет времени при запуске ракет:

```
...
WriteLn('Начинается          обратный          отсчет
времени');
for Seconds:=10 downto 1 do
  WriteLn('Осталось' ,Seconds:3, ' сек. ');
WriteLn('Пуск !');
```

5 Примеры программ

Вычислить значение факториала $f=10!$ с использованием трех вариантов цикла. ($f = 10! = 1*2*3*...*10$).

5.1 Оператор цикла с предусловием:

```
program First;
var
  i, f: Integer;
begin
  f:=1;
  i:=2;
  while i<=10 do
  begin
    f:= f*i;
    i:= i+1;
  end;
  WriteLn('10! = ' f);
end.
```

5.2 Оператор цикла с постусловием:

```
program Second;
var
  i, f: Integer;
begin
  f:=1;
  i:=2;
  repeat
    f:= f*i;
    i:= i+1;
  until i>10;
  WriteLn('10! = ' f);
end.
```

5.3 Оператор цикла с параметром:

```
program Third;
var
  i, f: Integer;
begin
  f:=1;
  for i:=2 to 10 do
    f:= f*i;
  WriteLn('10! = ' f);
end.
```

6 Варианты заданий для самостоятельного решения

1. Количество вещества ν (в молях) равно отношению числа молекул N в данном теле к постоянной Авогадро $N_A=6 \cdot 10^{23}$ моль⁻¹, т.е к числу молекул в 1 моле вещества. Не используя оператор цикла с параметром найти ν при $N=1..10$.
2. Найти сумму целых положительных чисел из промежутка от A до B , кратных 4 (A и B вводятся с клавиатуры).
3. Найти количество делителей натурального числа, больших K (K вводится).
4. Дана функция $y=\text{tg}x+1$. Написать программу для поиска максимального значения y в промежутке $x \in [2; 2.8]$ с шагом 0,1.
5. Дана функция $y=2x^3+1$ Написать программу для поиска максимального значения y в промежутке $x \in [-1; 1]$ с шагом 0,2.

6. Программа ждет ввода числа и в зависимости от количества цифр в числе выдает сообщение об их разрядности: трехзначное, пятизначное и т.д.
7. В учебном заведении задается начало учебного дня, продолжительность «пары», продолжительность обычного и большого перерывов (и их «место» в расписании), количество «пар». Получить расписание звонков на весь учебный день.
8. Леспромхоз ведет заготовку деловой древесины. Первоначальный объем ее на территории леспромхоза составлял p кубометров. Ежегодный прирост составляет k %. Годовой план заготовки – t кубометров. Через сколько лет в бывшем лесу будут расти одни опята?
9. Предприниматель, начав дело, взял кредит размером k рублей под p процентов годовых и вложил его в свое дело. По прогнозам, его дело должно давать прибыль r рублей в год. Сможет ли он накопить сумму, достаточную для погашения кредита, и если да, то через сколько лет?
10. Написать программу нахождения суммы чисел по правилу (n -нечетно): $a^1 + a^3 + a^5 + \dots + a^n$.
11. Найти сумму целых положительных чисел, больших 20, меньше 100 и кратных 9.
12. Написать программу нахождения суммы чисел по правилу (n -четно): $a^2 + a^4 + a^6 + \dots + a^n$.
13. Найти сумму четных делителей натурального числа.
14. Найти количество делителей натурального числа. Сколько среди них четных?
15. Найти сумму десяти чисел кратных 3.
16. Дана функция $y = x^3 - x^2 + x - 1$. Написать программу для поиска максимального значения y в промежутке $x \in [0; 2]$ с шагом 0.2.
17. Дано натуральное число. Получить все его натуральные делители.
18. Известно время начала и окончания (например, 6:00 и 24:00) работы некоторого пригородного автобусного маршрута с одним автобусом на линии, а также протяженность маршрута в минутах (в один конец) и время отдыха на конечных остановках. Составить суточное расписание (моменты отправления с конечных пунктов) без учета времени на обед и пересменку.

19. Составить алгоритм для расчета функции y при $x=1..10$:

$$y = \begin{cases} \frac{2 - \sin x}{\cos x + 1}, & \text{если } 1 \leq x \leq 9, \Delta x = 1; \\ \operatorname{tg} x, & \text{если } x > 9, \Delta x = 0,1. \end{cases}$$

20. Даны натуральные $n, b_0, b_1, b_2 \dots b_n$. Вычислить $f(b_0), f(b_1), f(b_2) \dots f(b_n)$, где

$$f(x) = \begin{cases} x^2, & \text{если } x \text{ кратно } 3; \\ x, & \text{если } x \text{ при делении на } 3 \text{ дает остаток } 1; \\ \left(\frac{x}{3}\right), & \text{в остальных случаях.} \end{cases}$$

21. Частица движется случайным образом согласно следующим условиям: в момент времени $t=0$ частица находится в исходной точке, $x=0, y=0$. В моменты времени $t=1,2,3\dots$ она делает случайный шаг в одном из четырех направлений: $x:=x+1, x:=x-1, y:=y+1, y:=y-1$. Движение заканчивается, когда $x^2+y^2 \geq R^2$. Определить отношение между временем движения и значением R .

22. Написать программу возведения натурального числа в квадрат, используя следующую закономерность :

$$1^2=1$$

$$2^2=1+3$$

$$3^2=1+3+5$$

$$4^2=1+3+5+7$$

...

$$n^2=1+3+5+7+\dots+2n-1$$

Указание. Не использовать цикл с параметром.

23. Составить алгоритм для расчета функции y при $x=0..10$:

$$y = \begin{cases} x^2 - 2x, & \text{если } x \leq 1, \Delta x = 0,2; \\ 2.5(x-2)^2, & \text{если } x > 1, \Delta x = 0,5. \end{cases}$$

24. Составить алгоритм для расчета функции y при $x=0..15$:

$$y = \begin{cases} x^3 - 3x, & \text{если } x \leq 1, \Delta x = 0,2; \\ \frac{(x-3)^3}{x}, & \text{если } x > 1, \Delta x = 0,5. \end{cases}$$

25. Утверждается, что функция $y=f(x)$ периодическая с периодом T . Проверить это численно, вычислив функцию с постоянным шагом на отрезке $[0;5T]$. Проверить на примере функций: $y=\sin^2 x$, $y=\operatorname{tg} x$ ($T=\pi$); $y=\frac{1}{x}\sin x$ ($T=2\pi$).

26. Дана функция $y=\frac{\sin x - \cos x}{1-x}$. Написать программу нахождения значения y в промежутке $x\in[0;1)\cup(1;2)$ с шагом 0,1.

27. Возвести натуральное число в квадрат, используя закономерность:

$$1^2=1$$

$$2^2=1+3$$

$$3^2=1+3+5$$

$$4^2=1+3+5+7$$

...

$$n^2=1+3+5+7+\dots+2n-1$$

Указание. Не использовать цикл с параметром.

28. Получить таблицу пересчета миль в километры и обратно (1 миля = 1,609344 км) для расстояний, не превышающих k км, в следующем виде:

МИЛИ	КМ
0,6214	1,0000
1,0000	1,6093
1,2428	2,0000
1,8641	3,0000
2,0000	3,2187
...	...

Лабораторная работа №5. Регулярный тип данных (массивы)

1 Цель работы

Изучить методы алгоритмизации и программирования с использованием регулярных типов (массивов).

2 Порядок выполнения работы

Получить задание для выполнения лабораторной работы (раздел 6) согласно своему варианту. Разработать и отладить программу. Составить отчет о лабораторной работе и защитить её у преподавателя.

3 Содержание отчета

- наименование и цель работы;
- задание на лабораторную работу согласно варианту;
- схема алгоритма, текст программы на алгоритмическом языке;
- результаты работы программы.

4 Краткая теория

Массив объединяет фиксированное количество элементов одного типа данных. Более формально массив можно определить как последовательную упорядоченную совокупность элементов некоторого типа, которые адресуются с помощью некоторого индекса.

Наряду с термином «массив» часто используются термины «матрица», «таблица», «вектор».

Определение регулярного типа данных «массив» имеет следующий вид:

type

Tarr = array [T1] of T2,

где **array, of** – служебные слова, T1 – означает тип индекса массива, T2 – тип элементов массива.

В качестве типа индекса может быть использованы все порядковые типы, за исключением длинного целого (`longint`) и поддиапазонов длинного целого.

Если тип элемента в типе массив также является массивом, то результат можно рассматривать как массив массивов или как один многомерный массив. Например, `array[1..5] of array[1..10] of Real`; интерпретируется компилятором точно так же, как массив: `array[1..5,1..10] of Real`. То есть допускается запись типов индексов, разделенных запятыми, в одних скобках. Число индексов в определении (то есть размерность массива) не ограничивается.

Примеры описания массивов:

```
type
  V1 = array[1..100] of real;
  V2 = array[char] of boolean;
  M1 = array[1..10] of array [1..15] of
integer;
var
  Vector: V1;
  Symb_Table: V2;
  Matrix1, Matrix2: M1;
```

Допускается объявление массива непосредственно в разделе описания переменных. Например:

```
var
  B: array[1..5] of byte;
  S: array[(Red, Yellow, Green)] of Boolean;
  M: array[1..5,1..7] of integer;
```

Иногда требуется задать массив с помощью типизированной константы. Например:

```
const
  Marks: array[1..4] of integer = (2,3,4,5);
  Tmatr: array[1..2,1..3] of byte = ((3,1,5),
(7,2,4));
```

Pascal допускает единственное возможное действие над массивом в целом: использование его в операторе присваивания, например

```
Vect1:=Vect2;
```

причем типы обоих массивов в данном случае должны быть эквивалентны.

Доступ к элементам массива строится традиционно: после идентификатора массива в квадратных скобках указывается индекс нужного элемента или список индексов, определяющий элемент многомерного массива. В качестве индексов могут выступать произвольные выражения, тип которых должен соответствовать типу индексов в описании массива.

Например:

```
Vector[1]
Vector[(i+1)*2]
M[i,j]
M[3][1]
M[3,1]
```

Элемент массива считается переменной; он может получать значения (например, в операторе присваивания), а также участвовать в выражениях.

Примеры:

```
M[i,j]:=M[i,j-1]*2;
Symb_Table['a']:=true;
Vector[i]:=M[i,1] mod 2;
```

5 Примеры программ

5.1 Найти сумму элементов массива

```
const
  N = 5;
var
  A: array[1..N] of Integer;
  I, Sum: Integer;
begin
  {ввод элементов}
  for I:=1 to N do
    begin
      Write('Введите ',I,'-й элемент массива: ');
      ReadLn(A[I]);
```

```

    end;
    Sum:=0;
    for I:=1 to N do
        Sum:=Sum+A[I];
        Writeln('Сумма равна ', Sum);
    end.

```

5.2 Вывести номера строк матрицы, которые содержат нулевой элемент.

```

var
    A: array[1..5,1..7] of Byte;
    I, J: Byte;
begin
    {инициализация генератора случайных чисел}
    Randomize;
    {заполнение и печать матрицы}
    for I:=1 to 5 do
        begin
            for J:=1 to 7 do
                begin
                    A[I,J]:=Random(15);
                    Write(A[I,J]:4);
                end;
            WriteLn;
        end;
    WriteLn;
    for I:=1 to 5 do
        for J:=1 to 7 do
            if A[I,J]=0 then
                begin
                    Write(I, ' ');
                    Break;
                end;
        WriteLn;
    end.

```

6 Варианты заданий для выполнения работы

6.1 Векторы

1. Даны два массива. Найти среднее арифметическое элементов каждого и сравнить эти значения.
2. Даны два целочисленных массива одинаковой размерности n . Получить третий массив той же размерности, каждый элемент которого равен большему из соответствующих элементов данного массива.

Например, даны массивы А и В, получим массив С:

i	1	2	3	4	5
А	14	2	7	8	9
В	3	6	5	12	4
С	14	6	7	12	9

Указания: n задать константой. Массивы А и В заполнить с помощью генератора случайных значений.

3. Найти сумму элементов, больших данного числа А (А вводить с клавиатуры).
4. Вставить в массив два числа: первое со значением N перед всеми элементами, большими N, и второе со значением M – после всех элементов, меньших M.
5. Вставить число А в массив целых чисел после всех элементов, значения которых кратны 5.
6. Найти максимальный среди всех элементов массива, имеющих четные номера.
7. Дан массив целых чисел, состоящий из 20 элементов. Заполнить с клавиатуры (случайно). Вывести индексы тех элементов, значения которых больше заданного А;
8. В массиве $A(n)$ найти и напечатать номера (индексы) локальных максимумов, то есть таких a_i , что $a_{i-1} < a_i > a_{i+1}$.
9. Дан массив целых чисел, состоящий из 10 элементов. Заполнить с клавиатуры (случайно). Вывести индексы тех элементов, значения которых больше значения предыдущего элемента (начиная со второго).
10. В массиве $T(k)$ найти номера первого и последнего нулевых элементов.
11. Даны два целочисленных массива $K(m)$ и $L(n)$. Найти элементы массива К, не имеющие себе равных в массиве L.

12. Каждый из элементов массива $X(n)$ заменить средним значением первых K элементов этого массива.
13. В массиве $A(n)$ каждый элемент, кроме первого, заменить суммой всех предыдущих элементов.
14. Даны два целочисленных массива $K(n)$ и $L(n)$. Найти парные максимумы (максимальные среди элементов, стоящих на одинаковых местах массивов K и L).
15. В массиве $Z(2n)$ каждый элемент с четным индексом поменять местами с предыдущим, то есть получить последовательность чисел $Z_2, Z_1, Z_4, Z_3, \dots, Z_{2n}, Z_{2n-1}$.
16. Элементы заданного массива $T(k)$ расположить в обратном порядке: $t_k, t_{k-1}, \dots, t_2, t_1$. Вычислить среднее значение.
17. Удалить из массива $A(n)$ нулевые элементы, передвинув на их место следующие элементы без нарушения порядка их следования. В результате должен получиться массив меньшего размера, не содержащий нулей.
18. Дан массив целых чисел, состоящий из 25 элементов. Заполнить с клавиатуры (случайно). Найти номер первого отрицательного элемента, делящегося на 5 с остатком 2.
19. Найти сумму и среднее значение элементов, принадлежащих промежутку от A до B (A и B вводить с клавиатуры).
20. К четным элементам массива прибавить число A , а из элементов с четными номерами вычесть число B (A и B вводятся с клавиатуры).
21. Найти сумму положительных и одновременно четных элементов массива.
22. Получить первые n чисел Фибоначчи – первые два числа равны 1, а каждое следующее равно сумме двух предыдущих. Например, $\{1, 1, 2, 3, 5, 8\}$.
23. Заменить все четные элементы массива на их квадраты, а нечетные удвоить.
24. Вычесть из положительных элементов массива элемент с номером $k1$, а к отрицательным прибавить элемент с номером $k2$, нулевые элементы оставить без изменения.
Указания: номера $k1$ и $k2$ вводятся с клавиатуры.
25. Дан вектор X , состоящий из n компонентов x . Написать программу вычисления значений:

$$N = \sqrt{\sum_{i=1}^n |x_i|^2}.$$

Указания: n задать константой (например, $n=15$). Значения x_i ввести.

26. Дан массив целых чисел, состоящий из 30 элементов. Заполнить с клавиатуры (случайно). Найти количество тех элементов, значения которых положительны и не превосходят заданного числа A .
27. Найти сумму и произведение элементов с $k1$ -го по $k2$ -ой, где $k1$ и $k2$ вводятся с клавиатуры. Сделать проверку корректности их ввода.
28. Заменить каждый элемент, кроме первого и последнего, на сумму элементов, стоящих до и после него.
29. Дан массив целых чисел, состоящий из 15 элементов. Заполнить с клавиатуры (случайно). Определить, есть ли в данном массиве пара соседних элементов с суммой, равной заданному числу.
30. Написать программу, реализующую перестановку наибольшего и наименьшего элементов одномерного массива A размерностью m .

Указания: m задать константой (например, $m=20$). Значения a_i ввести.

31. Дан массив X размерности n . Сформировать массив Y той же размерности, элементы которого вычисляются по следующему правилу:
 $Y_1 = X_1;$
 $Y_2 = X_1 + X_2;$
 \dots
 $Y_n = X_1 + X_2 + \dots + X_n.$

32. Выполнить *центрирование массива*, то есть от каждого из заданных m чисел X_1, X_2, \dots, X_m отнять их среднее арифметическое:

$$X_{cp} = \frac{1}{m} \sum_{i=1}^m X_i; \quad \tilde{X}_i = X_i - X_{cp}, \quad i=1, 2, \dots, m.$$

33. Даны два одномерных массива A и B . Найти их скалярное произведение. Скалярным произведением двух массивов одинаковой размерности называется сумма

произведений соответствующих элементов. Это можно записать так:

$a[1]*b[1] + a[2]*b[2] + \dots + a[n-1]*b[n-1] + a[n]*b[n]$, где n – количество элементов в массивах.

Указания: n задать константой (например, $n=5$). Значения a_i ввести.

6.2 Матрицы

1. Дан двумерный массив размерностью 4×6 , заполненный целыми числами с клавиатуры. Сформировать одномерный массив, каждый элемент которого равен количеству элементов соответствующей строки, больших данного числа.
2. Определить, есть ли в данном массиве строка, состоящая только из элементов, принадлежащих промежутку от A до B . A и B вводятся с клавиатуры.
3. Дан двумерный массив размерностью 6×5 , заполненный целыми числами с клавиатуры. Сформировать одномерный массив, каждый элемент которого равен первому четному элементу соответствующего столбца, если такого нет, то равен нулю.
4. Дана матрица размером 5×4 . Поменять местами первую строку и строчку, в которой находится первый нулевой элемент.
5. Найти сумму двух матриц размером $n \times m$.
6. Дан двумерный массив размером $n \times m$, заполненный случайными числами. Определить, есть ли в данном массиве столбец, в котором равное количество положительных и отрицательных элементов.
7. Дана матрица A размерностью $n \times m$. Сформировать одномерный массив B , элементами которого являются номера первых отрицательных элементов каждой строки массива A . (0 - отрицательный элемент отсутствует).
8. Дан двумерный массив размерностью 5×6 , заполненный целыми числами с клавиатуры. Сформировать одномерный массив, каждый элемент которого равен наибольшему элементу соответствующего столбца.
9. Найти количество элементов в каждой строке, больших среднего арифметического элементов данной строки.

10. Найти среднее арифметическое элементов каждой строки матрицы Q и вычесть его из элементов этой строки.
11. Дан двумерный массив размером $n \times m$, заполненный случайными числами. Определить, есть ли в данном массиве строка, содержащая больше положительных элементов, чем отрицательных.
12. Матрица $K(m, m)$ состоит из нулей и единиц. Найти в ней номера строк и столбцов, не содержащих единицы, либо сообщить, что таких нет.
13. Целочисленный массив $K(n, n)$ заполнить нулями и единицами, расположив их в шахматном порядке.
14. Дан двумерный массив размерностью 5×6 , заполненный целыми числами с клавиатуры. Сформировать одномерный массив, каждый элемент которого равен произведению четных положительных элементов соответствующего столбца.
15. Дан двумерный массив размером 8×7 , заполненный случайным образом. Заменить все элементы первых трех столбцов на их квадраты, в остальных столбцах изменить знак каждого элемента на противоположный.
16. Дана матрица размером 8×7 , заполненный случайным образом. Поменять местами две средние строки с первой и последней.
17. Дан двумерный массив размером 5×6 , заполненный случайным образом. Заменить максимальный элемент каждой строки на противоположный по знаку.
18. Определить, есть ли в данном массиве строка, состоящая только из отрицательных элементов.
19. Дана матрица размером 4×5 , заполненная случайным образом. Поменять местами первый и последний столбцы.
20. Дан двумерный массив размерностью 4×5 , заполненный целыми числами с клавиатуры. Сформировать одномерный массив, каждый элемент которого равен количеству отрицательных элементов, кратных 3 или 5, соответствующей строки.
21. В каждой строке, заполненной случайным образом, матрицы размером $n \times m$ поменять местами первый элемент и максимальный.
22. Дан двумерный массив размером 6×7 , заполненный случайным образом. Поменять местами средние строки.

23. Дан двумерный массив размером $n \times m$, заполненный случайным образом. Определить, есть ли в данном массиве строка, в которой ровно два отрицательных элемента.
24. В матрице $Z(m,m)$ каждый элемент разделить на диагональный, стоящий в том же столбце.
25. Определить, есть ли в данном массиве столбец, состоящий только из положительных или нулевых элементов.
26. Элементы одномерного массива $A(n^2)$ построчно расположить в матрице $B(n,n)$.
27. Статистические данные представляются таблицей с заданным количеством строк. Два первых столбца таблицы содержат номер предприятия и численность персонала. Составить программу обработки статистических данных с расчетом средних значений по строкам и столбцам.

Список использованных источников

Основная

1. Метелица Н.Т. Экономическая информатика [Электронный ресурс]: учебно-методическое пособие/ Метелица Н.Т.— Электрон. текстовые данные.— Краснодар: Южный институт менеджмента, 2014.— 42 с.— Режим доступа: <http://www.iprbookshop.ru/26000>.— ЭБС «IPRbooks», по паролю

2. Андреева Т.А. Программирование на языке Pascal [Электронный ресурс]/ Т.А. Андреева— Электрон. текстовые данные.— М.: ИнтернетУниверситет Информационных Технологий (ИНТУИТ), 2016.— 277 с.— Режим доступа: <http://www.iprbookshop.ru/52215.html>.— ЭБС «IPRbooks»

Дополнительная

3. Метелица Н.Т. Основы информатики [Электронный ресурс]: учебное пособие/ Метелица Н.Т., Орлова Е.В.— Электрон. текстовые данные.— Краснодар: Южный институт менеджмента, 2012.— 113 с.— Режим доступа: <http://www.iprbookshop.ru/9751>.— ЭБС «IPRbooks», по паролю.

4. Алексеев А.П. Сборник задач по дисциплине «Информатика» для ВУЗов [Электронный ресурс]: методические указания к проведению практических занятий по дисциплине «Информатика», для студентов первого курса специальностей 10.03.01 и 10.05.02/ А.П. Алексеев— Электрон. текстовые 33 данные.— М.: СОЛОН-ПРЕСС, 2016.— 104 с.— Режим доступа: <http://www.iprbookshop.ru/53849.html>.— ЭБС «IPRbooks»

5. Нечта И.В. Введение в информатику [Электронный ресурс]: учебно-методическое пособие/ И.В. Нечта— Электрон. текстовые данные.— Новосибирск: Сибирский государственный университет телекоммуникаций и информатики, 2016.— 31 с.— Режим доступа: <http://www.iprbookshop.ru/55471.html>.— ЭБС «IPRbooks»

ЭКОНОМИЧЕСКАЯ ИНФОРМАТИКА

Методические рекомендации

Составители:

А.Е. Вострокнутов

Ю.Н. Самойлюков

К.А. Ковалева,

Н.М. Нилова

Усл. печ. л. – 0,98.

Кубанский государственный аграрный университет.
350044, г. Краснодар, ул. Калинина, 13