



*Государственное бюджетное образовательное
учреждение среднего профессионального
образования*

«Владимирский авиамеханический колледж»

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

к выполнению лабораторных работ

по дисциплине

ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ

СОДЕРЖАНИЕ

| | |
|--|-----------|
| ВВЕДЕНИЕ..... | 5 |
| ЧАСТЬ 1 ПОСОБИЕ ПО РАБОТЕ В ИНТЕГРИРОВАННОЙ СРЕДЕ ПРОГРАММИРОВАНИЯ TURBO PASCAL 7.1 | 6 |
| 1. Интегрированная среда программирования Turbo Pascal 7.1..... | 7 |
| 2. Запуск интегрированной среды программирования Turbo Pascal 7.1..... | 7 |
| 3. Работа в интегрированной среде программирования | 8 |
| 3.1. Окна интегрированной среды Turbo Pascal 7.1 | 8 |
| 3.2. Выбор команд главного меню..... | 9 |
| 3.3. Быстрые способы выбора команд меню | 9 |
| 3.4. Редактор интегрированной среды | 10 |
| 3.5. Подготовка текста программы в интегрированной среде программирования Turbo Pascal 7.1 | 11 |
| 3.6. Компиляция и исполнение программы | 11 |
| 3.7. Рекомендации по исправлению ошибок | 12 |
| 3.8. Сохранение программы на диске..... | 12 |
| 3.9. Открытие текста программы | 13 |
| 3.10. Создание .exe – файла | 14 |
| 3.11. Получение листинга программы | 14 |
| 4. Тестирование и отладка программы..... | 14 |
| ЧАСТЬ 2 ЛАБОРАТОРНЫЕ РАБОТЫ ПО ОСНОВАМ ПРОГРАММИРОВАНИЯ В СРЕДЕ TURBO PASCAL | 19 |
| Требования к выполнению лабораторных работ..... | 20 |
| Методические указания | 20 |
| Содержание лабораторной работы | 20 |
| Содержание отчета | 20 |

| | |
|--|-----------|
| <i>Лабораторная работа № 1 Разработка программы линейной конструкции с использованием стандартных функций.....</i> | <i>21</i> |
| <i>Лабораторная работа № 2 Разработка программы разветвляющейся структуры</i> | <i>26</i> |
| <i>Лабораторная работа № 3 Разработка программ циклической структуры</i> | <i>31</i> |
| <i>Лабораторная работа № 4 Разработка программ усложненной структуры.....</i> | <i>35</i> |
| <i>Лабораторная работа № 5 Разработка программ с использованием перечислимых и ограниченных типов данных</i> | <i>38</i> |
| <i>Лабораторная работа № 6 Обработка одномерных массивов.....</i> | <i>41</i> |
| <i>Лабораторная работа № 7 Обработка многомерных массивов.....</i> | <i>45</i> |
| <i>Лабораторная работа № 8 Разработка программ с использованием строковых типов данных.....</i> | <i>48</i> |
| <i>Лабораторная работа № 9 Разработка программ с использованием множественных типов данных</i> | <i>52</i> |
| <i>Лабораторные работы № 10-12 Разработка программ с использованием процедур и функций</i> | <i>56</i> |
| <i>Лабораторная работа № 13-14 Разработка программ с использованием массива записей.</i> | <i>59</i> |
| <i>(4 часа).....</i> | <i>59</i> |
| <i>Лабораторная работа № 15 Разработка программ с использованием текстовых файлов</i> | <i>63</i> |
| <i>Лабораторная работа № 16 Разработка программ с использованием типизированных файлов</i> | <i>67</i> |
| <i>Лабораторная работа № 17 Разработка программ с использованием файлов записей.</i> | <i>73</i> |
| <i>Лабораторная работа № 18-20.....</i> | <i>75</i> |

| | |
|--|----|
| <i>Графические возможности Turbo Pascal</i> | 75 |
| <i>Лабораторная работа № 21 Разработка программ с использованием рекурсии.</i> | 77 |
| СОДЕРЖАНИЕ | 82 |

ВВЕДЕНИЕ

Предлагаемое методическое пособие предназначено для студентов специальностей 230115 «Программирование в компьютерных системах» и 230113 «Компьютерные комплексы и системы», выполняющих лабораторные работы по дисциплине «Основы алгоритмизации и программирования» в среде программирования Turbo Pascal 7.1.

Необходимость в разработке практикума возникла по причине отсутствия достаточного количества учебников, в которых сочетается справочная информация по практической работе в интегрированной среде программирования Turbo Pascal 7.0 с заданиями к лабораторным работам.

Методическое пособие состоит из двух частей. Часть 1 – «Пособие для студентов по работе в интегрированной среде программирования Turbo Pascal 7.1», содержит все необходимые практические рекомендации по созданию программ на Паскале в интегрированной среде программирования Turbo Pascal 7.1: использование системы меню, выполнению различных операций различными способами с учетом операционной среды, использованию «горячих клавиш». Пособие позволит по этапам создать исходный модуль программы, выполнить компиляцию, отредактировать текст программы, выполнить ее тестирование и отладку, создать загрузочный модуль.

Часть 2 – «Лабораторные работы по основам программирования в среде Turbo Pascal 7.1» содержит задания для выполнения 12 лабораторных работ по 30 вариантов в каждой.

Целесообразно использование методического пособия на занятиях в качестве раздаточного материала и для самостоятельного изучения.

ЧАСТЬ 1
ПОСОБИЕ ПО РАБОТЕ В ИНТЕГРИРОВАННОЙ СРЕДЕ
ПРОГРАММИРОВАНИЯ TURBO PASCAL 7.1

1. Интегрированная среда программирования Turbo Pascal 7.1

Разработка программ на Паскале включает в себя следующие действия (этапы разработки программы): ввод и редактирование текста программы на языке программирования Turbo Pascal 7.1 (создание исходного модуля), ее трансляцию и отладку.

Для выполнения каждого этапа применяются различные средства:

- Для ввода и редактирования текста используется текстовый редактор;
- Для трансляции программы – компилятор;
- Для построения исполняемого компьютером программного модуля с объединением разрозненных откомпилированных модулей и библиотекой стандартных процедур Turbo Pascal 7.1 – компоновщик (linker);
- Для отладки программы с анализом ее поведения, поиском ошибок, просмотром и изменением содержимого ячеек памяти компьютера – отладчик (debugger).

В интегрированной среде Turbo Pascal 7.1 все перечисленные элементы объединены в единое целое, что обеспечивает повышение скорости и качества разработки программ. Иногда ее кратко называют IDE (Integrated Development Environment – интегрированная среда разработки).

2. Запуск интегрированной среды программирования Turbo Pascal 7.1

Запуск интегрированной среды программирования Turbo Pascal 7.1 можно выполнить различными способами в зависимости от используемой операционной системы и оболочки.

- 1) При работе в ОС MSDOS: в командной строке набрать команду

C:\TP\BIN\TURBO.EXE

- 2) При работе в среде WINDOWS:

- Открыть папку Мой компьютер на Рабочем столе;
- Открыть папку TP на диске C:;
- Открыть папку BIN;

- Запустить файл Turbo.exe, дважды щелкнув по его пиктограмме левой кнопкой мыши.

В результате запускается интегрированная среда программирования Turbo Pascal 7.1. При этом экран компьютера будет иметь вид, как на рис. 1.



Рис. 1

3. Работа в интегрированной среде программирования

3.1. Окна интегрированной среды Turbo Pascal 7.1

Окна – это область экрана, которую можно перемещать, изменять в размере, перекрывать, открывать и закрывать.

Интегрированная среда программирования Turbo Pascal 7.1 позволяет открыто любое количество окон. Но в любой момент времени может быть активным только одно окно. Активное окно – это окно, с которым вы в настоящий момент времени работаете. Любая выбранная команда или набранный текст относится только к активному окну.

Существует несколько типов окон, но большинство из них имеют следующие элементы:

- полоса заголовка;
- закрывающая кнопка;
- полосы прокрутки;
- уголок для изменения размеров окна;
- кнопка масштабирования;
- номер окна

Расположение этих элементов показано на примере окна редактирования текста программ на рис.

3.2. Выбор команд главного меню

Для выбора команды меню с помощью клавиатуры необходимо:

- 1) Нажать клавишу **F10**. Это делает строку главного меню активной.
- 2) Для выбора пункта меню используют клавиши со стрелками, а затем нажимают **Enter**.

При этом, если за командой меню следует знак многоточия (...), то выбор команды приведет к выводу диалогового окна. Если за командой меню следует стрелка (>), то команда выведет другое меню (подменю). При выборе команды без многоточия и без стрелки выполняется соответствующее действие.

Для быстрого выбора можно нажать высвеченную букву команды вместе с клавишей **Alt**.

Для отмены действий используют клавишу **Esc**.

Для выбора команд с помощью мыши:

- 1) Щелкнуть левой кнопкой мыши на заголовке пункта меню;
- 2) Щелкнуть левой кнопкой мыши на имени выбранной команды.

3.3. Быстрые способы выбора команд меню

Наиболее часто используемые быстрые методы («горячие клавиши») для доступа к строке меню и выбора команд перечислены в таблице 1.

Таблица 1

Общие «Горячие клавиши»

| Клавиша | Элемент меню | Функция |
|---------|------------------|--|
| F1 | Help | Выводит на экран окно подсказки |
| F2 | File/Save | Сохраняет файл, находящийся в активном окне редактора |
| F3 | File/Open | Появляется окно сохранения файла |
| F4 | Run/Go to Cursor | Выполнение программы до строки, на которой стоит курсор |
| F5 | Window/Zoom | Масштабирует активное окно |
| F6 | Window/Next | Переход к следующему открытому окну |
| F7 | Run/Trace Info | Запускает программу в режиме отладки с заходом внутрь процедур |
| F8 | Run/Step Over | Запускает программу в режиме отладки, минуя вызовы процедур |
| F9 | Compile/Make | Запускает Make текущего окна |
| F10 | (none) | Активизация главного меню |

3.4. Редактор интегрированной среды

Составной частью интегрированной среды разработки программ является редактор Turbo Pascal 7.1, для управления которого используются клавиши, описанные в таблице 2:

Таблица 2

«Горячие клавиши» редактирования

| Клавиши | Элемент меню | Функция |
|---------------|---------------------|--|
| Shift+стрелки | (none) | Выделяет фрагмент текста в активном окне редактора |
| Ctrl+Del | Edit/Clear | Удаляет выделенный текст из окна без помещения его в буфер |
| Ctrl+Ins | Edit/Copy | Копирует выделенный текст в буфер |
| Shift+Del | Edit/Cut | Удаляет выделенный текст из окна с помещением его в буфер |
| Ctrl+L | Search/Search Again | Повторяет последнюю команду Find или Replace |
| Shift+Ins | Edit/Paste | Вставляет текст из буфера в окно редактора |

3.5. Подготовка текста программы в интегрированной среде программирования Turbo Pascal 7.1

1. Запустите среду программирования Turbo Pascal 7.1 одним из описанных выше способов. После запуска программы на экране раскроется окно редактора.

2. Введите текст программы, пользуясь обычными правилами работы в текстовых редакторах. Для удаления неверно введенных символов используйте клавиши **BackSpace** и **Delete**, а для перемещения внутри окна редактора используйте клавиши со стрелками. Для завершения ввода нажимайте **Enter** в конце каждой строки.

Для управления редактором используйте команды таблицы 2.

3.6. Компиляция и исполнение программы

1. Для выполнения компиляции программы нажмите клавиши **Alt+F9**. Если текст введен правильно, то на экран будет выведено сообщение об успешной компиляции (рис. 2).

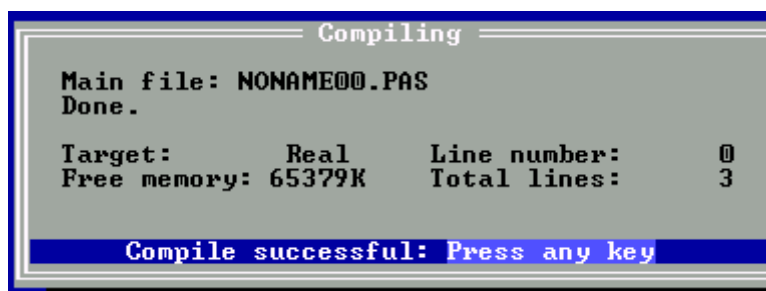


Рис. 2

В этом случае нажмите любую клавишу (Press any key).

2. Запустите программу на исполнение клавишами **Ctrl+F9**. После этого раскроется экран пользователя, в которое нужно ввести исходные данные по запросу программы.

3. Для просмотра результатов работы программы нажать **Alt+F5**. Оценить правильность ожидаемого результата и вернуться в среду Turbo Pascal, нажав любую клавишу.

При обнаружении ошибок, компилятор прерывает работу программы и выводит соответствующее сообщение в верхней строке редактора. Место ошибки в тексте

программы указывается курсором. Ошибку следует найти и исправить, а затем заново выполнить компиляцию.

3.7.Рекомендации по исправлению ошибок

Наиболее частыми ошибками начинающих программистов являются:

1. **Unknown identifier** – неизвестный идентификатор. Идентификатор не был описан. Паскаль требует, чтобы были объявлены все переменные, типы данных, константы, подпрограммы, т.е. все идентификаторы перед их использованием.

2. **Type mismatch** – несоответствие типов. Например, присваивание вещественного значения целой переменной.

3. **BEGIN expected** или **END expected** – нужен BEGIN или END. Несоответствие пар BEGIN ... END.

4. **“;” expected** – нужно указать “;” и др.

Часто ошибки бывают порожденными. Положение курсора при наличии такой ошибки может не соответствовать ее настоящему месту нахождения. Поэтому ошибку надо искать выше, внимательно изучив и проанализировав сообщение об ошибке.

Для получения подсказки по данной ошибке можно использовать справочную систему Turbo Pascal 7.1, нажав клавишу **F1**.

3.8.Сохранение программы на диске

Пока файлу не присвоено конкретное имя, по умолчанию система Turbo Pascal 7.1 присваивает ему имя NONAME00. Чтобы записать файл с заданным именем на диск, нужно:

1. Перейти в **File-меню**, нажав клавишу **F10**.

2. Выбрать пункт **Save as...** (записать под новым именем); в результате на экране появится диалоговое окно **Save File As** (рис. 3).

3. Перейти к списку **Files** с помощью клавиши **Tab**, выбрать каталог, в который будет сохранен файл;

4. В окне ввода задать имя файла с учетом правил составления имен в MSDOS: не более 8 латинских символов в имени с расширением **.pas**.

Например: **lab1.pas**

Замечание: имя файла не должно начинаться с цифры.

Если файл записан благополучно, то в строке заголовка окна редактора появится введенное Вами имя.

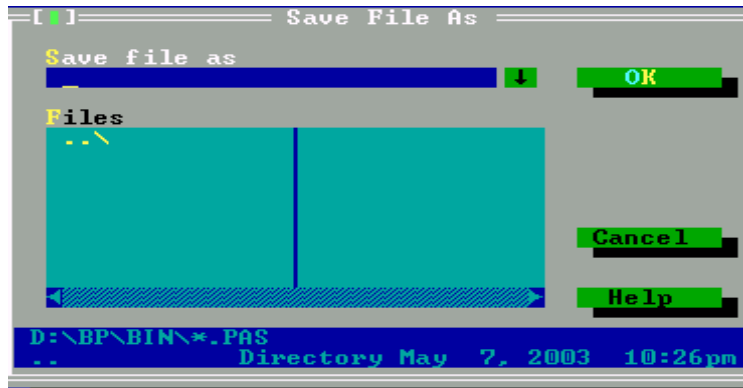


Рис. 3

3.9. Открытие текста программы

Чтобы открыть созданный ранее файл с текстом какой-либо программы, нужно:

1. Запустить интегрированную среду программирования Turbo Pascal 7.1;
2. В главном меню выбрать **File – Open** (открыть) или нажать **F9**. На экране появится окно выбора открываемого файла **Open a File** (рис. 4);

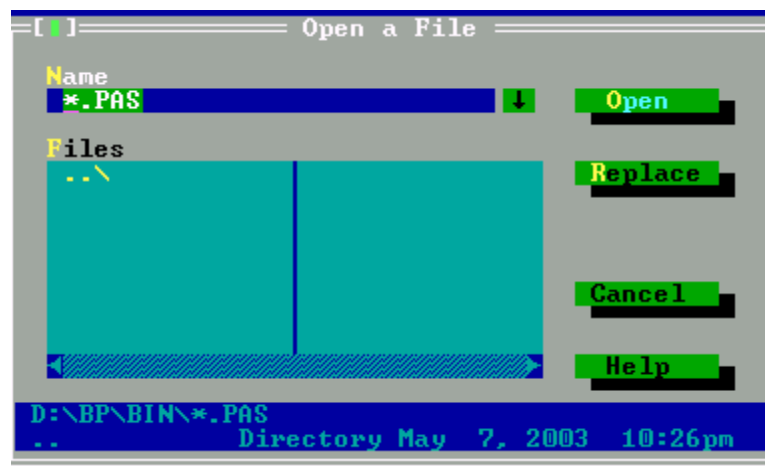


рис. 4

3. Перейти к окну списка **Files**, выбрать нужный каталог и в нем имя открываемого файла;
4. Установить курсор на имя нужного файла, выбрать кнопку **Open**. В результате раскроется окно редактора с текстом выбранного файла.

3.10. Создание .exe – файла

Если необходимо записать программу как исполняемый файл (с расширением .exe) на диск, то нужно:

1. Выбрать в главном меню пункт **Compile**;
2. В появившемся списке выбрать опцию **Destination** (назначение);
3. Если справа от нее стоит слово **Memory** (память), указывающее, что выполняемый код будет храниться в памяти, нажать клавишу **Enter** или щелкнуть левой кнопкой мыши. В результате установка назначения изменится на **Disk** (диск). Если опция **Destination** установлена в **Disk**, что указывает на запись выполняемого кода на диск в виде файла с расширением .exe, то нужно перейти к опции **Make** этого пункта меню.

3.11. Получение листинга программы

Чтобы вывести текст программы на бумагу (получить листинг), нужно:

1. Проверить, подключен ли принтер к компьютеру;
2. Вставить бумагу в печатающее устройство;
3. Выбрать команду **File – Write to ...**
4. В рабочей строке диалогового окна набрать имя устройства **PRN** и нажать клавишу **Enter**;

Чтобы напечатать результат работы программы, нужно:

1. Запустить программу на исполнение, ввести необходимые данные, получить результат и открыть окно пользователя (**Alt+F5**);
2. Вставить бумагу в печатающее устройство;
3. Нажать клавиши **Shift + Print Screen**

4. Тестирование и отладка программы

Для проверки правильности функционирования программы выполняется **тестирование** – исполнение программы с использованием некоторого набора входных данных, охватывающего весь спектр возможных значений для данного типа задач и проверяющего граничные условия, а также позволяющего через контроль промежуточных и конечных результатов решения задачи в ходе

исполнения программы проверить выполнение операторов программы в требуемой последовательности и правильность действия всех алгоритмических конструкций (ветвлений, циклов, обращений к программам и т.п.).

Процесс тестирования удостоверяет качество программы, поэтому он должен быть документирован, т.е. будущие пользователи должны быть, как и при каких обстоятельствах программа тестировалась, каковы были входные данные и результаты, с тем, чтобы тестирование можно было повторить.

Если цель тестирования – только выявить ошибки в программе, то для обнаружения и устранения и устранения ошибок в программе выполняют ее *отладку*.

В процессе создания новой программы программисту приходится сталкиваться с несколькими видами ошибок. Во-первых, это синтаксические ошибки, связанные с неправильным употреблением различных элементов и конструкций языка Паскаль. Причиной возникновения таких ошибок являются недостаточно хорошее знание языка программирования и опечатки при наборе текста программы. Такие ошибки определяются уже на этапе компиляции и серьезной опасности не представляют.

Второй вид ошибок доставляет программисту гораздо больше неприятностей. Это ошибки при выполнении программы. Сообщение о такой ошибке имеет вид:

Run-time error <errnum> at <segment>: <offset>

Где <errnum> - это код ошибки,

<segment>: <offset> - адрес в памяти, где находится ошибка.

Довольно часто оказывается, что синтаксически правильная программа завершает свое выполнение аварийно, с сообщением, например, о попытке деления на ноль, либо она закикливается по непонятной причине. Т.е. программа в процессе выполнения ведет себя не так, как предполагал программист. Выяснить это можно применяя методы отладки программы.

Отладка в интегрированной среде программирования Turbo Pascal 7.1 заключается в том, что с помощью встроенного в интегральную среду

программирования специального средства – **отладчика**, анализируется поведение программы в «окрестностях» ошибки. С этой целью интегрированной среде программирования обеспечивается возможность трассировки программы, т.е. выполнения по шагам с остановкой в указанных точках или при выполнении заданных условий просмотра и изменения содержимого ячеек памяти компьютера, регистров процессора.

Эти возможности отладки представляются программисту через пункт **Debug** главного меню интегрированной среды программирования.

Debug

| | |
|--------------------|---------|
| Breakpoints | |
| Call stack | Ctrl+F3 |
| Register | |
| Watch | |
| Output | |
| User screen | Alt+F5 |
| Evaluate/modify... | Ctrl+F4 |
| Add watch... | Ctrl+F7 |
| Add breakpoint... | |

Для просмотра значений переменных в процессе выполнения программы активизируйте опцию **Add watch**.

В окно просмотра **Add watch** введите выражение (например, $V=0$) для наблюдения за его значением в ходе выполнения программы. Для включения переменной (выражения) в список просмотр можно, установив курсор на актуальной переменной или в начале выражения, нажать клавиши **Ctrl+F7**. Во введенном на экране окна ввода выражений для просмотра, нажимая клавишу со стрелкой вправо, задать нужное значение (оно копируется из окна редактирования). Для его завершения следует нажать **Enter** или выбрать кнопку **[OK]**.

Для того, чтобы в процессе отладки одновременно с окном редактирования было выведено окно просмотра в главном меню, выберите пункт **Window** и задайте режим **Tile** (расположение непересекающихся окон на экране «черепица») отображения окон.

После этого на экране будут видны и окно редактирования с текстом программы, и окно просмотра с текущими значениями избранных вами для наблюдения переменных.

Если вы ошибочно ввели неверный идентификатор переменной или желаете удалить некоторый идентификатор из окна просмотра, то нажмите клавишу **F6**. После этого из окна редактирования вы перейдете в окно просмотра. Клавишами со стрелками вверх-вниз установите курсор на удаляемом идентификаторе и нажмите клавишу **Del**.

Если вы хотите пополнить список переменных, чьи значения просматриваются в процессе отладки, то нажмите **Ins** и введите идентификатора или соответствующее выражение.

Для отладки программы в режиме пошагового прохода выберите режим **Step over** пункта **Run** главного меню интегрированной среды программирования.

Можно просто нажимать клавишу **F8**, при этом выполняется текущая подсвеченная строка и курсор перемещается к следующей строке, а вам остается наблюдать за изменениями значений переменных в окне просмотра.

Если эти изменения не соответствуют ожидаемым в соответствии с условием задачи, то займитесь коррекцией алгоритма и программы, после чего заново откомпилируйте программу и повторите проверку исполнения в пошаговом проходе. Если есть необходимость пройти по «шагам» через всю программу с пошаговой проверкой вызываемых процедур, функций, методов, объектов, то выберите режим **Trace into** пункта **Run** главного меню или нажмите **F7**. При этом программа в начале компилируется, а затем начинается ее пошаговое исполнение. Каждый шаг заключается в выполнении очередной строки при нажатии на клавишу **F7**.

Таким образом, используя возможность пошагового выполнения в сочетании с просмотром текущих значений переменных, можно провести достаточно подробный анализ работы программы.

Метод пошагового выполнения неудобен, если текст программы имеет достаточно большой объем, а программиста интересуют значения переменных только в определенных местах программы при выполнении некоторых условий. В этом случае используют команду **Breakpoints**, которая позволяет разместить в тексте программы «точки прерывания». Для каждой точки прерывания указывается номер строки, где она устанавливается, условие, при выполнении которого программа приостанавливается в указанном месте, а также количество проходов до ее «срабатывания». Установленные ранее точки можно удалить той же командой.

Завершить работу зациклившейся программы можно нажатием клавиш **Ctrl+Break**.

ЧАСТЬ 2
ЛАБОРАТОРНЫЕ РАБОТЫ ПО ОСНОВАМ ПРОГРАММИРОВАНИЯ
В СРЕДЕ TURBO PASCAL

Требования к выполнению лабораторных работ

Методические указания

Подготовка и выполнение программы осуществляется по инструкции к работе в интегрированной среде программирования Turbo Pascal 7.1.

Содержание лабораторной работы

- 1) Получить задание согласно номеру варианта.
- 2) Составить математическую модель решения задачи.
- 3) Составить алгоритм решения задачи в виде блок-схемы.
- 4) Составить текст программы.
- 5) Создать исходный файл в интегрированной среде программирования Turbo Pascal 7.1 в соответствии с пунктом 3.5. инструкции.
- 6) Выполнить и отладить программу в соответствии с пунктом 3.6. инструкции.
- 7) Получить листинг программы в соответствии с пунктом 3.11. инструкции.
- 8) Составить отчет.
- 9) Ответить на контрольные вопросы.

Содержание отчета

- 1) Тема, цель.
- 2) Текст задания.
- 3) Математическая модель решения задачи.
- 4) Блок-схема программы.
- 5) Текст программы.
- 6) Результат работы программы.

Лабораторная работа № 1

Разработка программы линейной конструкции с использованием стандартных функций.

(2 часа)

1. Цель работы:

- 1.1. Освоение простейшей структуры программы.
- 1.2. Получение навыков в организации ввода/вывода стандартных типов данных.
- 1.3. Приобретение навыков в записи выражений на языке Паскаль и использовании стандартных функций.
- 1.4. Получение практических навыков работы в диалоговом режиме.

2. Вопросы методики руководства

2.1. Вводное инструктирование:

- определение правил техники безопасности и организации рабочего места;
- определение цели работы (изложена выше);
- связь работы с изученной темой «Стандартные типы данных. Организация ввода/вывода. Вычисление выражений. Использование стандартных функций»;
- ознакомление студентов с порядком выполнения работы (постановка задачи, последовательное выполнение пунктов задания);
- указания по ведению записей, оформлению результатов и составлению отчета;
- выдача заданий.

2.2. Текущее инструктирование:

- оказание помощи студентам при создании программы;
- ответы на вопросы студентов.

2.3. Подведение итогов:

- устный опрос студентов по результатам, полученным в ходе наблюдений и экспериментов;
- контроль выполнения программы;
- ответы на контрольные вопросы;
- зачет выполнения работы.

3. Оборудование, материалы:

- 3.1. ПЭВМ, инструментальная среда программирования Turbo Pascal 7.1
- 3.2. Методические указания по выполнению работы

4. Теоретическая часть:

- 4.1. Стандартные типы данных языка программирования Turbo Pascal 7.1;
- 4.2. Структура программы на языке программирования Turbo Pascal 7.1;
- 4.3. Организация ввода/вывода;
- 4.4. Правила формирования выражений;
- 4.5. Стандартные функции;

5. Практическая часть:

- 5.1. Написать программу для вычисления значения выражения по формуле согласно конкретному варианту, используя стандартные типы данных и операторы ввода/вывода языка программирования Turbo Pascal 7.1;
- 5.2. Написать программу для решения задачи согласно конкретному варианту, используя стандартные типы данных и операторы ввода/вывода языка программирования Turbo Pascal 7.0;

6. Варианты заданий:

1 вариант

1.
$$\frac{b + \sqrt{b^2 + 4ac}}{2a} - a^3 c + b^{-2}$$

2. Вычислить периметр и площадь прямоугольного треугольника по длинам a и b двух катетов.

2 вариант

1. $\frac{a}{c} \cdot \frac{b}{d} - \frac{ab - c}{cd}$

2. Вычислить длину окружности и площадь круга одного и того же заданного радиуса R .

3 вариант

1. $\frac{\sin x + \cos y}{\cos x - \sin y} \operatorname{tg} xy$

2. Даны два числа. Найти среднее арифметическое кубов этих чисел.

4 вариант

1. $\frac{x + y}{y + 1} - \frac{xy - 12}{34 + x}$

2. Даны два числа. Найти среднее геометрическое модулей этих чисел.

5 вариант

1. $\frac{3 + e^{y-1}}{1 + x^2 |y - \operatorname{tg} x|}$

2. Вычислить расстояние между двумя точками с данными координатами (x_1, y_1) и (x_2, y_2) .

6 вариант

1. $x - \frac{x^3}{3} + \frac{x^5}{5}$

2. Даны два числа. Вычислить их сумму, разность, произведение и частное.

7 вариант

1. $\ln \left| \left(y - \sqrt{x} \right) \left(x - \frac{y}{x + \frac{x^2}{4}} \right) \right|$

2. Дана длина ребра куба. Найти площадь его грани.

8 вариант

1. $(1 - \operatorname{tg} x)^{\operatorname{ctg} x} + \cos(x - y)$

2. Дана длина ребра куба. Найти площадь полной поверхности куба.

9 вариант

1. $\frac{\ln|\cos x|}{\ln(1+x^2)}$

2. Дана длина ребра куба. Найти объем этого куба.

10 вариант

1. $\left(\frac{x+1}{x-1}\right)^x + 18xy^2$

2. Дана сторона равностороннего треугольника. Найти площадь этого треугольника.

11 вариант

1. $\left(1 + \frac{1}{x^2}\right)^x - 12x^2y$

2. Дана сторона равностороннего треугольника. Найти высоту этого треугольника.

12 вариант

1. $\frac{x^2 - 7x + 10}{x^2 - 8x + 12}$

2. Дана сторона равностороннего треугольника. Найти радиусы вписанной и описанной окружностей.

13 вариант

1. $\frac{\cos x}{\pi - 2x} + 16x \cos(xy) - 2$

2. Известна длина окружности. Найти площадь круга, ограниченного этой окружностью.

14 вариант

1. $2^{-x} - \cos x + \sin(2xy)$

2. Найти площадь равнобедренной трапеции с основаниями a и b и углом α при большем основании a .

15 вариант

1. $2 \operatorname{ctg}(3x) - \frac{1}{12x^2 + 7x - 5}$

2. Найти сумму членов арифметической прогрессии, если известны ее первый член, знаменатель и число членов прогрессии.

Контрольные вопросы:

1. Алфавит языка программирования Turbo Pascal.
2. Что представляет собой идентификатор?
3. Структура программы на языке Turbo Pascal.
4. Классификация типов данных Turbo Pascal.
5. Основные стандартные функции языка Turbo Pascal.
6. Операторы ввода/вывода языка Turbo Pascal.

Лабораторная работа № 2

Разработка программы разветвляющейся структуры

(2 часа)

1. Цель работы:

- 1.1. Получение навыков в использовании условного оператора в программе.
- 1.2. Знакомство с задачами, для решения которых используются условные операторы.

2. Вопросы методики руководства

2.1. Вводное инструктирование:

- определение правил техники безопасности и организации рабочего места;
- определение цели работы (изложена выше);
- связь работы с изученной темой «Оператор условного перехода»;
- ознакомление студентов с порядком выполнения работы (постановка задачи, последовательное выполнение пунктов задания);
- указания по ведению записей, оформлению результатов и составлению отчета;
- выдача заданий.

2.2. Текущее инструктирование:

- оказание помощи студентам при создании программы;
- ответы на вопросы студентов.

2.3. Подведение итогов:

- устный опрос студентов по результатам, полученным в ходе наблюдений и экспериментов;
- контроль выполнения программы;
- ответы на контрольные вопросы;
- зачет выполнения работы.

3. Оборудование, материалы:

- 3.1. ПЭВМ, инструментальная среда программирования Turbo Pascal 7.1

3.2. Методические указания по выполнению работы

4. Теоретическая часть:

4.1. Назначение оператора условного перехода системы программирования Turbo Pascal 7.0;

4.2. Полная и краткая формы оператора условного перехода;

4.3. Назначение составного оператора;

5. Практическая часть:

5.1. Составить программу для нахождения значения функции $y(x)$ согласно конкретному варианту, используя оператор условного перехода;

5.2. Составить программу для решения задачи согласно конкретному варианту, используя оператор условного перехода и составного оператора.

6. Варианты заданий:

1 вариант

$$1. f(x) = \begin{cases} -x^2 + x - 9, & \text{если } x \geq 8; \\ \frac{1}{x^4 - 6}, & \text{если } x < 8 \end{cases}$$

2. Даны числа a, b, c . Проверить, выполняется ли неравенство $a^3 \geq b^2 \geq c$.

2 вариант

$$1. f(x) = \begin{cases} 3x + 9, & \text{если } x > 3; \\ \frac{x^3}{x^2 + 8}, & \text{если } x \leq 3 \end{cases}$$

2. Даны числа a, b, c . Утроить эти числа, если $a \geq b \geq c$.

3 вариант

$$1. f(x) = \begin{cases} -x^3 + 9, & \text{если } x \leq 13; \\ -\frac{3}{x+1}, & \text{если } x > 13 \end{cases}$$

2. Дана точка на плоскости с координатами (x, y) . Определить, принадлежит ли она I четверти декартовой системы координат.

4 вариант

$$1. f(x) = \begin{cases} 45x^2 + 5, & \text{если } x > 3,6; \\ \frac{5x}{10x^2 + 1}, & \text{если } x \leq 3,6 \end{cases}$$

2. Дана точка на плоскости с координатами (x,y) . Определить, принадлежит ли она II четверти декартовой системы координат.

5 вариант

$$1. f(x) = \begin{cases} x^4 + 9, & \text{если } x < 3,2; \\ \frac{54x^4}{-5x^2 + 7}, & \text{если } x \geq 3,2 \end{cases}$$

2. Дана точка на плоскости с координатами (x,y) . Определить, принадлежит ли она III четверти декартовой системы координат.

6 вариант

$$1. f(x) = \begin{cases} 1,2x^2 - 3x - 9, & \text{если } x \geq 3; \\ \frac{12,1}{2x^2 + 1}, & \text{если } x < 3 \end{cases}$$

2. Дана точка на плоскости с координатами (x,y) . Определить, принадлежит ли она IV четверти декартовой системы координат.

7 вариант

$$1. f(x) = \begin{cases} x^2 + 3x + 9, & \text{если } x \leq 3; \\ \frac{\sin x}{x^2 - 9}, & \text{если } x > 3 \end{cases}$$

2. Определите номер минимального элемента из трех заданных x_1, x_2, x_3 .

8 вариант

$$1. f(x) = \begin{cases} \cos 2x + 9, & \text{если } x > -4; \\ -\frac{\cos x}{x - 9}, & \text{если } x \leq -4 \end{cases}$$

2. Даны три числа. Возведите в квадрат те из них, значения которых неотрицательны, и в четвертую степень – отрицательные.

9 вариант

$$1. f(x) = \begin{cases} \ln x + 9, & \text{если } x > 0; \\ -\frac{x}{x^2 - 7}, & \text{если } x \leq 0 \end{cases}$$

2. Даны две точки $A(x_1, y_1)$ и $B(x_2, y_2)$. Определить, какая из точек находится ближе к началу координат.

10 вариант

$$1. f(x) = \begin{cases} -x^2 - 1, 1x + 9, & \text{если } x \leq -3; \\ \frac{\ln(x + 3)}{x^2 + 9}, & \text{если } x > -3 \end{cases}$$

2. Подсчитать количество отрицательных чисел среди трех заданных.

11 вариант

$$1. f(x) = \begin{cases} 9 - x, & \text{если } x \geq -1, 1; \\ \frac{\sin 3x}{x^4 + 1}, & \text{если } x < -1, 1 \end{cases}$$

2. Подсчитать количество положительных чисел среди трех заданных.

12 вариант

$$1. f(x) = \begin{cases} 4x^2 + 2x - 19, & \text{если } x \geq -3, 5; \\ -\frac{2x}{-4x + 1}, & \text{если } x < -3, 5 \end{cases}$$

2. Подсчитать количество нулевых чисел среди трех заданных.

13 вариант

$$1. f(x) = \begin{cases} -x^2, & \text{если } x \geq 7; \\ \frac{2^{-x}}{x^2 - 9}, & \text{если } x < 7 \end{cases}$$

2. Подсчитать количество четных чисел среди трех заданных.

14 вариант

$$1. f(x) = \begin{cases} -x^2 + 3x + 9, & \text{если } x \leq 3; \\ \frac{x}{x^2 + 1}, & \text{если } x > 3 \end{cases}$$

2. Подсчитать количество нечетных чисел среди трех заданных.

15 вариант

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

2. Подсчитать количество целых чисел среди трех заданных.

Контрольные вопросы:

1. Как происходит работа условного оператора *If*?
2. В каких формах может быть использован условный оператор *If*?
3. Может ли встречаться символ «;» внутри любого структурного оператора и почему?
4. Что такое составной оператор и как он работает?

Лабораторная работа № 3

Разработка программ циклической структуры

(4 часа)

1. Цель работы:

- 1.1. Получение навыков в выборе и использовании операторов цикла.
- 1.2. Знакомство с итерационными процессами.

2. Вопросы методики руководства

2.1. Вводное инструктирование:

- определение правил техники безопасности и организации рабочего места;
- определение цели работы (изложена выше);
- связь работы с изученной темой «Операторы циклов»;
- ознакомление студентов с порядком выполнения работы (постановка задачи, последовательное выполнение пунктов задания);
- указания по ведению записей, оформлению результатов и составлению отчета;
- выдача заданий.

2.2. Текущее инструктирование:

- оказание помощи студентам при создании программы;
- ответы на вопросы студентов.

2.3. Подведение итогов:

- устный опрос студентов по результатам, полученным в ходе наблюдений и экспериментов;
- контроль выполнения программы;
- ответы на контрольные вопросы;
- зачет выполнения работы.

3. Оборудование, материалы:

- 3.1. ПЭВМ, инструментальная среда программирования Turbo Pascal 7.1
- 3.2. Методические указания по выполнению работы

4. Теоретическая часть:

- 4.1. Назначение операторов цикла;
- 4.2. Цикл с параметром *For* системы программирования Turbo Pascal 7.1;
- 4.3. Цикл с предусловием *While* системы программирования Turbo Pascal 7.1;
- 4.4. Цикл с постусловием *Repeat* системы программирования Turbo Pascal 7.1;
- 4.5. Различия в работе операторов цикла;

5. Практическая часть:

- 5.1. Составить программу с использованием оператора цикла **FOR** для вычисления значения функции $f(x)$ (см. задание 1) на отрезке $[a, b]$ с шагом h , где $h = \frac{b-a}{m}$, m – заданное число. Результат представить в виде таблицы, первый столбец которой – значения аргумента, второй – соответствующие значения функции.
- 5.2. Составить программу с использованием оператора цикла **WHILE ...DO** для решения предыдущей задачи.
- 5.3. Составить программу для решения задачи согласно конкретному варианту (см. задание 2), использовав оператор цикла **REPEAT ... UNTIL**.

6. Варианты заданий:

1 вариант

1. $f(x) = x - \sin(x)$, $a=0$, $b=\pi/2$, $m=10$
2. Найти сумму целых положительных чисел, кратных 4, меньших 50. Вывести эти числа на экран.

2 вариант

1. $f(x) = \sin(x)$, $a=\pi/4$, $b=\pi/2$, $m=15$
2. Найти сумму целых положительных четных чисел, меньших 100. Вывести эти числа на экран.

3 вариант

1. $f(x) = \cos(x)$, $a=\pi/3$, $b=2\pi/3$, $m=20$

2. Найти сумму целых положительных нечетных чисел, меньших 200. Вывести эти числа на экран.

4 вариант

1. $f(x) = \operatorname{tg}(x)$, $a=0$, $b=\pi/4$, $m=10$
2. Найти сумму целых положительных чисел, кратных 5 и меньших 200. Вывести эти числа на экран.

5 вариант

1. $f(x) = \operatorname{ctg}(x)$, $a=\pi/4$, $b=\pi/2$, $m=15$
2. Найти сумму целых положительных чисел, кратных 3, больших 20 и меньших 100. Вывести эти числа на экран.

6 вариант

1. $f(x) = \arcsin(x)$, $a=0$, $b=1$, $m=20$
2. Найти сумму целых отрицательных чисел, кратных 4, больших -20 . Вывести эти числа на экран.

7 вариант

1. $f(x) = \arccos(x)$, $a=0.5$, $b=1$, $m=10$
2. Найти сумму целых отрицательных четных чисел, больших -10 . Вывести эти числа на экран.

8 вариант

1. $f(x) = \operatorname{arctg}(x)$, $a=2$, $b=7$, $m=15$
2. Найти сумму целых отрицательных нечетных чисел, больших -20 . Вывести эти числа на экран.

9 вариант

1. $f(x) = \sin(x) - \cos(x)$, $a=0$, $b=\pi/2$, $m=20$
2. Найти сумму целых отрицательных чисел, кратных 5 и больших -200 . Вывести эти числа на экран.

10 вариант

1. $f(x) = x \sin(x)$, $a=0$, $b=3\pi$, $m=10$

2. Найти сумму целых отрицательных чисел, кратных 3, больших -20. Вывести эти числа на экран.

11 вариант

1. $f(x) = \sin(1/x)$, $a = \pi/8$, $b = 2/\pi$, $m = 15$
2. Найти сумму целых чисел, кратных 4, в диапазоне от -20 до 20. Вывести эти числа на экран.

12 вариант

1. $f(x) = \cos(1/x)$, $a = \pi/4$, $b = 4/\pi$, $m = 20$
2. Найти сумму целых четных чисел, в диапазоне от -10 до 10. Вывести эти числа на экран.

13 вариант

1. $f(x) = \sin(x^2)$, $a = \pi/6$, $b = 2\pi/3$, $m = 10$
2. Найти сумму целых нечетных чисел, в диапазоне от -10 до 10. Вывести эти числа на экран.

14 вариант

1. $f(x) = \cos(x^2)$, $a = \pi/3$, $b = 3\pi/2$, $m = 15$
2. Найти сумму целых чисел, кратных 5, в диапазоне от -10 до 10. Вывести эти числа на экран.

15 вариант

1. $f(x) = \sin(x) + \operatorname{tg}(x)$, $a = 0$, $b = \pi/4$, $m = 20$
2. Найти сумму целых чисел, кратных 3, в диапазоне от -10 до 10. Вывести эти числа на экран.

Контрольные вопросы:

1. Какие управляющие конструкции повторения поддерживаются в языке Turbo Pascal?
2. Как работает оператор цикла *While*?
3. В чем заключается различие между операторами *Repeat* и *While*?
4. Какие ограничения накладываются на использование оператора *For*?

Лабораторная работа № 4

Разработка программ усложненной структуры.

(2 часа)

1. Цель работы:

- 1.1. Получение навыков в выборе и использовании операторов цикла.
- 1.2. Знакомство с итерационными процессами.

2. Вопросы методики руководства

2.1. Вводное инструктирование:

- определение правил техники безопасности и организации рабочего места;
- определение цели работы (изложена выше);
- связь работы с изученными темами «Оператор условного перехода», «Операторы циклов»;
- ознакомление студентов с порядком выполнения работы (постановка задачи, последовательное выполнение пунктов задания);
- указания по ведению записей, оформлению результатов и составлению отчета;
- выдача заданий.

2.2. Текущее инструктирование:

- оказание помощи студентам при создании программы;
- ответы на вопросы студентов.

2.3. Подведение итогов:

- устный опрос студентов по результатам, полученным в ходе наблюдений и экспериментов;
- контроль выполнения программы;
- ответы на контрольные вопросы;
- зачет выполнения работы.

3. Оборудование, материалы:

- 3.1. ПЭВМ, инструментальная среда программирования Turbo Pascal 7.1

3.2. Методические указания по выполнению работы

4. Теоретическая часть:

4.1. Назначение условного оператора;

4.2. Назначение операторов цикла;

5. Практическая часть:

Составить программу с использованием операторов условного перехода **IF** и одного из операторов для решения задачи согласно конкретному варианту.

6. Варианты заданий:

1 вариант

Найти сумму ряда с точностью $\varepsilon = 10^{-3}$, общий член которого $a_n = (-1)^{n-1} / n^n$

2 вариант

Найти сумму ряда с точностью $\varepsilon = 10^{-3}$, общий член которого $a_n = 1/2^n + 1/3^n$

3 вариант

Найти сумму ряда с точностью $\varepsilon = 10^{-3}$, общий член которого $a_n = (2n - 1) / 2^n$

4 вариант

Найти сумму ряда с точностью $\varepsilon = 10^{-3}$, общий член которого $a_n = 1 / ((3n - 2)(3n + 1))$

5 вариант

Найти сумму ряда с точностью $\varepsilon = 10^{-3}$, общий член которого $a_n = 10^n / n!$

6 вариант

Найти сумму ряда с точностью $\varepsilon = 10^{-3}$, общий член которого $a_n = (n!) / (2n!)$

7 вариант

Найти сумму ряда с точностью $\varepsilon = 10^{-2}$, общий член которого $a_n = (n!) / n^n$

8 вариант

Найти сумму ряда с точностью $\varepsilon = 10^{-3}$, общий член которого $a_n = (2n - 1) / 2^n$

9 вариант

Найти сумму ряда с точностью $\varepsilon = 10^{-3}$, общий член которого $a_n = 3^n \cdot n! / (3n!)$

10 вариант

Найти сумму ряда с точностью $\varepsilon = 10^{-3}$, общий член которого $a_n = n! / (3n!)$

11 вариант

Найти сумму ряда с точностью $\varepsilon = 10^{-3}$, общий член которого $a_n = (n!)^2 / (2^{n^2})!$

12 вариант

Найти сумму 10 членов ряда, в котором $a_n = \ln(n!)n^2$

13 вариант

Найти сумму 15 членов ряда, в котором $a_n = n^{\ln n} / (\ln n)^n$

14 вариант

Найти сумму 12 членов ряда, в котором $a_n = n! / n^{\sqrt{n}}$

15 вариант

Найти сумму 7 членов ряда, в котором $a_n = e^{-\sqrt[3]{n}}$

Контрольные вопросы:

1. Как происходит работа условного оператора *If*?
2. Какие управляющие конструкции повторения поддерживаются в языке Turbo Pascal?

Лабораторная работа № 5

Разработка программ с использованием перечислимых и ограниченных типов данных

(2 часа)

1. Цель работы:

- 1.1. Ознакомление с перечислимыми и ограниченными типами данных.
- 1.2. Ознакомление с оператором выбора.
- 1.3. Получение навыков в организации ввода/вывода значений переменных перечислимых типов данных.

2. Вопросы методики руководства

2.1. Вводное инструктирование:

- определение правил техники безопасности и организации рабочего места;
- определение цели работы (изложена выше);
- связь работы с изученной темой «Перечисляемые и ограниченные типы данных. Оператор выбора»;
- ознакомление студентов с порядком выполнения работы (постановка задачи, последовательное выполнение пунктов задания);
- указания по ведению записей, оформлению результатов и составлению отчета;
- выдача заданий.

2.2. Текущее инструктирование:

- оказание помощи студентам при создании программы;
- ответы на вопросы студентов.

2.3. Подведение итогов:

- устный опрос студентов по результатам, полученным в ходе наблюдений и экспериментов;
- контроль выполнения программы;
- ответы на контрольные вопросы;

– зачет выполнения работы.

3. Оборудование, материалы:

3.1. ПЭВМ, инструментальная среда программирования Turbo Pascal 7.1

3.2. Методические указания по выполнению работы

4. Теоретическая часть:

4.1. Перечисляемый тип данных системы программирования Turbo Pascal 7.1;

4.2. Ограниченный тип данных системы программирования Turbo Pascal 7.1;

4.3. Оператор выбора системы программирования Turbo Pascal 7.1;

5. Практическая часть:

5.1. По заданным номеру квартала, дате (включающую число и название месяца), дню недели конкретного дня, требуется определить дату и день недели следующего дня.

6. Варианты заданий:

| Номер варианта | Номер квартала | Дата и день недели | Номер варианта | Номер квартала | Дата и день недели |
|----------------|----------------|------------------------|----------------|----------------|------------------------|
| 1 | 1 | 27 февраля пятница | 16 | 2 | 1 июня понедельник |
| 2 | 2 | 12 апреля воскресенье | 17 | 3 | 13 августа четверг |
| 3 | 3 | 26 августа среда | 18 | 4 | 21 декабря понедельник |
| 4 | 4 | 30 ноября понедельник | 19 | 1 | 28 февраля суббота |
| 5 | 1 | 32 марта понедельник | 20 | 2 | 13 апреля понедельник |
| 6 | 2 | 9 мая суббота | 21 | 3 | 31 августа вторник |
| 7 | 3 | 1 сентября вторник | 22 | 1 | 29 марта воскресенье |
| 8 | 4 | 9 декабря среда | 23 | 4 | 31 декабря среда |
| 9 | 1 | 12 января понедельник | 24 | 2 | 30 апреля четверг |
| 10 | 2 | 1 мая пятница | 25 | 3 | 14 августа пятница |
| 11 | 3 | 30 августа воскресенье | 26 | 4 | 22 декабря вторник |
| 12 | 4 | 7 октября среда | 27 | 4 | 17 ноября вторник |
| 13 | 1 | 23 января пятница | 28 | 4 | 8 октября четверг |

| Номер варианта | Номер квартала | Дата и день недели | Номер варианта | Номер квартала | Дата и день недели |
|----------------|----------------|----------------------|----------------|----------------|--------------------|
| 14 | 2 | 11 мая понедельник | 29 | 2 | 2 июня вторник |
| 15 | 3 | 29 марта воскресенье | 30 | 2 | 6 мая пятница |

Контрольные вопросы:

1. Какой оператор позволяет выполнить одно из нескольких действий в зависимости от результата вычисления выражения?
2. Почему интервальный тип называется диапазоном?
3. Когда используется интервальный тип?
4. Можно ли к стандартные функции *ord*, *succ*, *pred* применять к переменным интервального типа?
5. как описываются переменные ограниченных типов?
6. Можно ли операцию, применимую к переменной некоторого типа, применять к переменной, относящейся к соответствующему диапазону?

Лабораторная работа № 6
Обработка одномерных массивов
(2 часа)

1. Цель работы:

1.1. Получение практических навыков в работе с одномерными массивами.

2. Вопросы методики руководства

2.1. *Вводное инструктирование:*

- определение правил техники безопасности и организации рабочего места;
- определение цели работы (изложена выше);
- связь работы с изученной темой «Массивы»;
- ознакомление студентов с порядком выполнения работы (постановка задачи, последовательное выполнение пунктов задания);
- указания по ведению записей, оформлению результатов и составлению отчета;
- выдача заданий.

2.2. *Текущее инструктирование:*

- оказание помощи студентам при создании программы;
- ответы на вопросы студентов.

2.3. *Подведение итогов:*

- устный опрос студентов по результатам, полученным в ходе наблюдений и экспериментов;
- контроль выполнения программы;
- ответы на контрольные вопросы;
- зачет выполнения работы.

3. Оборудование, материалы:

3.1. ПЭВМ, инструментальная среда программирования Turbo Pascal 7.1

3.2. Методические указания по выполнению работы

4. Теоретическая часть:

- 4.1. Массив как структурированный тип данных;
- 4.2. Размерность массива;
- 4.3. Описание и использование массивов;
- 4.4. Сортировка массивов.

5. Практическая часть:

- 5.1. Для конкретного варианта составить программы с использованием одномерного массива, используя ручной способ ввода;
- 5.2. Для конкретного варианта составить программу сортировки одномерного массива указанным способом. Использовать автоматический ввод массива с помощью генератора случайных чисел

6. Варианты заданий:

1 вариант

1. Создать массив $A(n)$. Указать номер максимального элемента массива.
2. Отсортировать одномерный массив по возрастанию методом выбора.

2 вариант

1. Создать массив $A(n)$. Подсчитать количество нечетных значений элементов данного массива.
2. Отсортировать одномерный массив по убыванию методом вставки.

3 вариант

1. Создать массив $A(n)$. Подсчитать количество четных значений элементов данного массива.
2. Отсортировать одномерный массив по возрастанию методом обмена

4 вариант

1. Создать массив $A(n)$. Определить, сколько элементов имеют значения меньше 25.
2. Отсортировать одномерный массив по убыванию методом выбора.

5 вариант

1. Создать массив $A(n)$. Вычислить сумму значений элементов массива $A(n)$, имеющих четные индексы.
2. Отсортировать одномерный массив по возрастанию методом вставки.

6 вариант

1. Создать массив $A(n)$. Вычислить произведение значений элементов массива $A(n)$, имеющих нечетные индексы.
2. Отсортировать одномерный массив по убыванию методом обмена

7 вариант

1. Создать массив $A(n)$. Найти среднее арифметическое положительных элементов этого массива.
2. Отсортировать одномерный массив по возрастанию методом выбора

8 вариант

1. Имеется n итоговых оценок студента. Определить является ли студент отличником.
2. Отсортировать одномерный массив по убыванию методом вставки.

9 вариант

1. Имеется n итоговых оценок студента. Подсчитать количество двоек, троек, четверок и пятерок.
2. Отсортировать одномерный массив по возрастанию методом обмена.

10 вариант

1. Создать массив $A(n)$. Определить количество элементов массива, больших заданного числа.
2. Отсортировать одномерный массив по убыванию методом выбора.

11 вариант

1. Создать массив $A(n)$. Определить количество элементов массива, имеющих четные порядковые номера и являющихся нечетными числами.
2. Отсортировать одномерный массив по возрастанию методом вставки.

12 вариант

1. В соревнованиях по фигурному катанию n судей выставляют оценки. Напишите программу, которая выводит на печать номера судей, поставивших наименьшую и наибольшую оценки.
2. Отсортировать одномерный массив по убыванию методом обмена.

13 вариант

1. В соревнованиях по фигурному катанию n судей выставляют оценки. Наибольшая и наименьшая оценки отбрасываются, а по остальным выставляется средняя. Напишите программу, вычисляющую среднюю оценку.
2. Отсортировать одномерный массив по возрастанию методом выбора

14 вариант

1. Создать массив $A(n)$. Подсчитать сколько раз встречаются в нем числа, кратные 3.
2. Отсортировать одномерный массив по убыванию методом вставки

15 вариант

1. Создать массив $A(n)$. Определить номера элементов, значения которых кратны 3.
2. Отсортировать одномерный массив по возрастанию методом обмена.

Контрольные вопросы:

1. Какая форма организации данных называется массивом?
2. Для чего нужно описывать массив?
3. Что включают в себя понятия номер и значение элемента массива?
4. Сортировка одномерного массива методом выбора.
5. Сортировка одномерного массива методом вставки.
6. Сортировка одномерного массива методом обмена.

Лабораторная работа № 7

Обработка многомерных массивов

(2 часа)

1. Цель работы:

1.1. Получение практических навыков в работе с двумерными массивами.

2. Вопросы методики руководства

2.1. *Вводное инструктирование:*

- определение правил техники безопасности и организации рабочего места;
- определение цели работы (изложена выше);
- связь работы с изученной темой «Массивы»;
- ознакомление студентов с порядком выполнения работы (постановка задачи, последовательное выполнение пунктов задания);
- указания по ведению записей, оформлению результатов и составлению отчета;
- выдача заданий.

2.2. *Текущее инструктирование:*

- оказание помощи студентам при создании программы;
- ответы на вопросы студентов.

2.3. *Подведение итогов:*

- устный опрос студентов по результатам, полученным в ходе наблюдений и экспериментов;
- контроль выполнения программы;
- ответы на контрольные вопросы;
- зачет выполнения работы.

3. Оборудование, материалы:

3.1. ПЭВМ, инструментальная среда программирования Turbo Pascal 7.1

3.2. Методические указания по выполнению работы

4. Теоретическая часть:

- 4.1. Массив как структурированный тип данных;
- 4.2. Размерность массива;
- 4.3. Описание и использование массивов;

5. Практическая часть:

- 5.1. Для конкретного варианта составить программу с использованием двумерного массива. Организовать два способа ввода элементов массива: ручной и автоматический с помощью генератора случайных чисел.

6. Варианты заданий:

1 вариант

Создать массив $A(n \times m)$. Подсчитать количество четных элементов в данной матрице.

2 вариант

Создать массив $A(n \times n)$. Подсчитать количество четных чисел в каждой строке массива.

3 вариант

Создать массив $A(n \times n)$. Подсчитать количество отрицательных чисел в каждом столбце массива.

4 вариант

Создать массив $A(n \times n)$. Подсчитать количество элементов, расположенных на главной диагонали, значения которых кратны 3.

5 вариант

Создать массив $A(n \times n)$. Подсчитать сумму элементов, значения которых кратны 5 и расположенных на побочной диагонали.

6 вариант

Создать массив $A(2 \times n, 2 \times n)$. Подсчитать сумму значений элементов в каждой четверти матрицы.

7 вариант

Создать массив $A(n \times n)$. Сформировать одномерный массив из сумм каждого столбца матрицы.

8 вариант

Создать массив $A(n \times n)$. Сформировать одномерный массив из минимальных значений элементов каждой строки массива.

9 вариант

Создать массив $A(n \times n)$. Сформировать одномерный массив из значений элементов матрицы, расположенных на главной диагонали.

10 вариант

Создать массив $A(n \times n)$. Сформировать одномерный массив из элементов матрицы, расположенных по спирали, начиная с первого элемента матрицы.

11 вариант

Создать массив $A(n \times n)$. Найти среднее геометрическое значение элементов матрицы и выяснить, сколько элементов в матрице больших среднего геометрического.

12 вариант

Создать массив $A(n \times n)$. Переставить местами элементы главной и побочной диагонали.

13 вариант

Создать массив $A(n \times n)$. Поменять местами элементы симметрично главной диагонали.

14 вариант

Создать массив $A(n \times n)$. Найти максимальный по модулю элемент массива.

15 вариант

Создать массив $A(n \times n)$. Сформировать одномерный массив, переслав в него по строкам верхний треугольник матрицы, включая элементы главной диагонали.

Контрольные вопросы:

1. Какая форма организации данных называется массивом?
2. Для чего нужно описывать массив?
3. Что включают в себя понятия номер и значение элемента массива?
4. Как организовать ввод двумерного массива по строкам, по столбцам?

Лабораторная работа № 8

Разработка программ с использованием строковых типов данных

(2 часа)

1. Цель работы:

1.1. Получение навыков работы со строчными типами данных.

2. Вопросы методики руководства

2.1. *Вводное инструктирование:*

- определение правил техники безопасности и организации рабочего места;
- определение цели работы (изложена выше);
- связь работы с изученной темой «Строковый тип данных»;
- ознакомление студентов с порядком выполнения работы (постановка задачи, последовательное выполнение пунктов задания);
- указания по ведению записей, оформлению результатов и составлению отчета;
- выдача заданий.

2.2. *Текущее инструктирование:*

- оказание помощи студентам при создании программы;
- ответы на вопросы студентов.

2.3. *Подведение итогов:*

- устный опрос студентов по результатам, полученным в ходе наблюдений и экспериментов;
- контроль выполнения программы;
- ответы на контрольные вопросы;
- зачет выполнения работы.

3. Оборудование, материалы:

3.1. ПЭВМ, инструментальная среда программирования Turbo Pascal 7.1

3.2. Методические указания по выполнению работы

4. Теоретическая часть:

- 4.1. Строка как структурированный тип данных;
- 4.2. Стандартные процедуры и функции для работы со строками;

5. Практическая часть:

- 5.1. Составить программы для решения задач согласно конкретному варианту, используя стандартные процедуры и функции для работы со строками.

6. Варианты заданий:

1 вариант

1. Вводится строка. Определить на какую букву оканчивается третье слово.
2. Проверить, одинаковое ли количество число открывающихся и закрывающихся скобок в данной строке.

2 вариант

1. Вводится строка. Определить на какую букву начинается третье слово.
2. Удалить часть строки, заключенную в скобки (вместе со скобками).

3 вариант

1. Определить слова, которые оканчиваются на «ова».
2. Из заданной строки получить новую, выбросив из нее все точки.

4 вариант

1. Дана строка из слов. Найти самое длинное слово в строке.
2. Дана строка. Подсчитать, сколько в ней букв *r, k, l*.

5 вариант

1. Определите количество слов, имеющих количество символов первого слова.
2. Дана строка. Найти длину самого длинного слова.

6 вариант

1. Дан текст из слов. Подсчитать количество букв «d» в последнем слове.
2. Дана строка. Вывести слова, состоящие из трех букв.

7 вариант

1. Найти слова, начинающиеся на букву «л».
2. Дана строка. Преобразовать ее, удалив из нее все цифры.

8 вариант

1. Найти первое слово в строке, имеющее заданную длину.
2. Дана строка. Преобразовать ее, заменив буквы «а» буквами «б».

9 вариант

1. Определить, сколько слов A содержится в заданной строке. A – заданное слово.
2. Дана строка. Преобразовать ее, удалив каждый символ «*» и повторив каждый символ, отличный от «*».

10 вариант

1. Определить, является ли строка палиндромом по звучанию.
2. Дана строка. Подсчитать сколько различных символов встречается в ней. Вывести их на печать.

11 вариант

1. Дана строка из слов. Найти слово, которое имеет четную длину и начинается с заданного символа.
2. Дана строка. Подсчитать самую длинную последовательность подряд идущих букв A .

12 вариант

1. Дана строка. Найти слова, которые начинаются и заканчиваются одной и той же буквой.
2. Дана строка символов, среди которых есть одна открывающаяся и одна закрывающаяся скобка. Вывести на экран все символы, расположенные внутри этих скобок.

13 вариант

1. Найти в строке слово минимальной длины.
2. Дан набор слов, разделенных точкой с запятой (;). Набор заканчивается двоеточием (:). Определить, сколько в нем слов, заканчивающихся буквой A .

14 вариант

1. Дана строка из слов. Удалить из нее слова, начинающиеся с заданной буквы.
2. Дана строка. Указать те слова, которые содержат хотя бы одну букву L .

15 вариант

1. Написать программу, которая выдает номер позиции заданной строки, в которой первый раз встречается буква «л». Если такой буквы нет, то выдать об этом сообщение.
2. В строке заменить все двоеточия на точку с запятой. Подсчитать количество замен.

Контрольные вопросы:

1. Определение строкового типа данных.
2. Какие способы задания текста Вы знаете?
3. Какие операции определены для строковых данных?
4. Как определить длину строки?
5. Какие стандартные функции определены для строковых переменных?
6. Как вы думаете, в чем сходство способов обработки текстов и обработки массивов?

Лабораторная работа № 9

Разработка программ с использованием множественных типов данных

(2 часа)

1. Цель работы:

- 1.1. Получение навыков в задании переменных множественного типа и выполнении простейших операций над ними.
- 1.2. Получение навыков в организации ввода/вывода значений множественных типов.
- 1.3. Получение практических навыков в выполнении операций над множествами.
- 1.4. Знакомство с задачами, в которых целесообразно использовать переменные множественных типов.

2. Вопросы методики руководства

2.1. Вводное инструктирование:

- определение правил техники безопасности и организации рабочего места;
- определение цели работы (изложена выше);
- связь работы с изученной темой «Множества»;
- ознакомление студентов с порядком выполнения работы (постановка задачи, последовательное выполнение пунктов задания);
- указания по ведению записей, оформлению результатов и составлению отчета;
- выдача заданий.

2.2. Текущее инструктирование:

- оказание помощи студентам при создании программы;
- ответы на вопросы студентов.

2.3. Подведение итогов:

- устный опрос студентов по результатам, полученным в ходе наблюдений и экспериментов;

- контроль выполнения программы;
- ответы на контрольные вопросы;
- зачет выполнения работы.

3. Оборудование, материалы:

- 3.1. ПЭВМ, инструментальная среда программирования Turbo Pascal 7.1
- 3.2. Методические указания по выполнению работы

4. Теоретическая часть:

- 4.1. Множественный тип данных;
- 4.2. Организация ввода/вывода значений множественных типов;
- 4.3. Операции над множествами;

5. Практическая часть:

- 5.1. Для конкретного варианта составить программу с использованием множества.

6. Варианты заданий:

1 вариант Известны сорта роз, выращиваемые тремя цветоводами: «Анжелика», «Виктория», «Гагарин», «Ave Maria», «Катарина», «Юбилейная». Определить те сорта, которые имеются у каждого из цветоводов, которые есть хотя бы у одного из цветоводов, которых нет ни у одного из цветоводов.

2 вариант Заданы имена девочек. Определить, какие из этих имен встречаются во всех классах данной параллели, которые есть только в некоторых классах и какие из этих имен не встречаются ни в одном классе.

3 вариант Задан некоторый набор товаров. Определить для каждого из товаров, какие из них имеются в каждом из n магазинов, какие товары есть хотя бы в одном магазине и каких товаров нет ни в одном магазине.

4 вариант Имеется список класса (все имена различны). Определить, есть ли в классе человек, который побывал в гостях у всех. (Для каждого ученика класса составить множество побывавших у него в гостях друзей, сам ученик в это множество не входит).

- 5 вариант** Имеется множество, содержащее натуральные числа из некоторого диапазона. Сформировать два множества, первое из которых содержит все простые числа из данного множества, а второй – все составные.
- 6 вариант** На трех участках возделываются сельскохозяйственные культуры. Известны виды культур, выращиваемых на каждом из участков. Определить виды тех культур, которые возделывают на каждом из участков; возделывают хотя бы на одном из участков; не возделывают ни на одном участке.
- 7 вариант** Известны марки машин, изготавливаемых в данной стране и импортируемых за рубеж. Даны некоторые n стран. Определить для каждой из марок, какие из них были доставлены во все страны; доставлены в некоторые из стран; не доставлены ни в одну страну.
- 8 вариант** В озере водится несколько видов рыб. Три рыбака поймали рыб, представляющих некоторые из имеющихся видов. Определить: какие виды рыб есть у каждого из рыбаков; какие рыба есть в озере, но нет ни одного из рыбаков.
- 9 вариант** В n колхозах выращивают некоторые сельскохозяйственные культуры из имеющегося перечня. Определить культуры, возделываемые во всех колхозах; возделываемые только в некоторых колхозах; не возделываемые ни в одном из колхозов.
- 10 вариант** Есть список игрушек, некоторые из которых имеются в N детских садах. Определить игрушки из списка, которых нет ни в одном из садов; которые есть в каждом из детсадов.
- 11 вариант** Задано множество M и множество T того же типа. Подсчитать, сколько элементов из множеств T и M совпадает.
- 12 вариант** Дана непустая последовательность символов. Требуется построить и напечатать множество, элементами которого являются встречающиеся в последовательности цифры от '0' до '9'.
- 13 вариант** Дана непустая последовательность символов. Требуется построить и напечатать множество, элементами которого являются встречающиеся в последовательности буквы от 'A' до 'F' и от 'X' до 'Z'.

14 вариант Дана непустая последовательность символов. Требуется построить и напечатать множество, элементами которого являются встречающиеся в последовательности буквы от 'G' до 'N' и цифры от '0' до '9'.

15 вариант Дана непустая последовательность символов. Требуется построить и напечатать множество, элементами которого являются встречающиеся в последовательности буквы от 'A' до 'Z' и цифры от '0' до '5'.

Контрольные вопросы:

1. Дайте определение множества.
2. Что называется базовым типом множества?
3. Какой объем памяти занимает один элемент множества?
4. Что называется мощностью множества?
5. Какие операции допустимы над множествами?

Лабораторные работы № 10-12

Разработка программ с использованием процедур и функций

(6 часов)

1. Цель работы:

- 1.1. Получение навыков в написании программ с использованием функций.
- 1.2. Получение навыков в написании программ с использованием процедур.
- 1.3. Изучение механизма передачи параметров процедурам и функциям.

2. Вопросы методики руководства

2.1. Вводное инструктирование:

- определение правил техники безопасности и организации рабочего места;
- определение цели работы (изложена выше);
- связь работы с изученной темой «Процедуры и функции»;
- ознакомление студентов с порядком выполнения работы (постановка задачи, последовательное выполнение пунктов задания);
- указания по ведению записей, оформлению результатов и составлению отчета;
- выдача заданий.

2.2. Текущее инструктирование:

- оказание помощи студентам при создании программы;
- ответы на вопросы студентов.

2.3. Подведение итогов:

- устный опрос студентов по результатам, полученным в ходе наблюдений и экспериментов;
- контроль выполнения программы;
- ответы на контрольные вопросы;
- зачет выполнения работы.

3. Оборудование, материалы:

- 3.1. ПЭВМ, инструментальная среда программирования Turbo Pascal 7.1

3.2. Методические указания по выполнению работы

4. Теоретическая часть:

- 4.1. Назначение процедур и функций;
- 4.2. Отличие функций от процедур;
- 4.3. Область действия и время жизни идентификатора;
- 4.4. Методы передачи параметров процедурам и функциям: передача параметров-значений и параметров-переменных;

5. Практическая часть:

5.1. Составить программу для решения поставленной задачи согласно конкретного варианта с использованием:

- функции;
- процедуры.

6. Варианты заданий:

1 вариант Даны координаты трех вершин треугольника. Найти длины всех его сторон. Использовать подпрограмму для вычисления длины одной стороны.

2 вариант Дано натуральное число. Найти все его делители. Подсчитать их число. Использовать подпрограмму, которая определяет, делится ли заданное число на другое число.

3 вариант Составить программу для вычисления $c = \frac{n!}{k!(n-k)}$. Использовать

подпрограмму для вычисления факториала числа. N, k – заданные числа.

4 вариант Определить, в каком из двух данных чисел больше цифр. Использовать подпрограмму, подсчитывающую количество цифр числа.

5 вариант Найти сумму цифр числа. Использовать подпрограмму, определяющую одну из цифр числа.

6 вариант Найти первую цифру числа. Использовать подпрограмму, определяющую одну из цифр числа.

- 7 вариант** Найти сумму всех делителей числа. Использовать подпрограмму для определения делителя числа.
- 8 вариант** Определить является ли заданное число простым. Использовать подпрограмму, которая определяет остаток деления.
- 9 вариант** Определить, является ли число *аморфным*, т.е. квадрат этого числа заканчивается этим же числом, например, число 6, так как его квадрат 36 заканчивается на 6. Использовать подпрограмму для выделения цифр из числа.
- 10 вариант** Среди чисел из интервала от А до В найти все простые числа. Использовать подпрограмму для определения остатка от деления.
- 11 вариант** Найти все аморфные числа из промежутка от А до В. Использовать подпрограмму определения аморфности числа. Число называется *аморфным*, если квадрат этого числа заканчивается этим же числом, например, число 6, так как его квадрат 36 заканчивается на 6.
- 12 вариант** Составить программу для нахождения *НОД* трех чисел. Использовать подпрограмму нахождения *НОД* двух чисел.
- 13 вариант** Треугольник задан координатами своих вершин. Составить программу для вычисления его площади. Использовать подпрограмму вычисления длины одной из сторон треугольника.
- 14 вариант** Написать программу для нахождения суммы большего и меньшего из трех чисел. Использовать подпрограмму для нахождения наибольшего и наименьшего среди двух чисел.
- 15 вариант** Вычислить площадь правильного шестиугольника со стороной А, используя подпрограмму для вычисления площади треугольника.

Контрольные вопросы:

1. Дайте определение программы.
2. Как выглядят заголовки процедуры и функции?
3. Чем параметры-значения отличаются от параметров-переменных?
4. Чем глобальные переменные отличаются от локальных переменных?
5. Чем фактические переменные отличаются от формальных переменных?
6. В чем отличие процедур от функций?

Лабораторная работа № 13-14

Разработка программ с использованием массива записей.

(4 часа)

1. Цель работы:

1.1. Получение навыков в организации ввода/вывода значений комбинированных типов данных.

1.2. Получение практических навыков программирования задач с использованием записей.

2. Вопросы методики руководства

2.1. Вводное инструктирование:

- определение правил техники безопасности и организации рабочего места;
- определение цели работы (изложена выше);
- связь работы с изученной темой «Записи»;
- ознакомление студентов с порядком выполнения работы (постановка задачи, последовательное выполнение пунктов задания);
- указания по ведению записей, оформлению результатов и составлению отчета;
- выдача заданий.

2.2. Текущее инструктирование:

- оказание помощи студентам при создании программы;
- ответы на вопросы студентов.

2.3. Подведение итогов:

- устный опрос студентов по результатам, полученным в ходе наблюдений и экспериментов;
- контроль выполнения программы;
- ответы на контрольные вопросы;
- зачет выполнения работы.

3. Оборудование, материалы:

- 3.1. ПЭВМ, инструментальная среда программирования Turbo Pascal 7.1
- 3.2. Методические указания по выполнению работы

4. Теоретическая часть:

- 4.1. Запись как структурированный тип данных;
- 4.2. Описание записи;
- 4.3. Обращение к полю записи;
- 4.4. Вложенные записи;
- 4.5. Оператор присоединения;
- 4.6. Массивы записей;

5. Практическая часть:

- 5.1. Составить программу для решения задачи согласно конкретному варианту с использованием массива записей. Вывести на экран всю базу данных.

6. Варианты заданий:

- 1 вариант** Распечатать анкетные данные учеников, участвовавших в олимпиаде по информатике и заработавших более 30 баллов.
- 2 вариант** В таблице хранятся следующие данные об учениках: фамилия, имя, отчество, рост, масса. Вычислить средний рост учеников.
- 3 вариант** В таблице хранятся следующие данные об учениках: фамилия, имя, отчество, рост, масса. Вычислить рост самого низкого ученика.
- 4 вариант** В таблице хранятся следующие данные об учениках: фамилия, имя, отчество, рост, масса. Вычислить рост самого высокого ученика.
- 5 вариант** В таблице хранятся следующие данные об учениках: фамилия, имя, отчество, рост, масса. Сколько учеников может заниматься в баскетбольной секции, если рост баскетболиста должен быть больше 170 см?
- 6 вариант** На аптечном складе хранятся лекарства. Сведения о лекарствах хранятся в специальной ведомости: наименование лекарства; количество; цена; срок хранения (в месяцах). Выяснить, сколько стоит самый дешевый препарат?

- 7 вариант** На аптечном складе хранятся лекарства. Сведения о лекарствах хранятся в специальной ведомости: наименование лекарства; количество; цена; срок хранения (в месяцах). Выяснить, сколько стоит самый дорогой препарат?
- 8 вариант** На аптечном складе хранятся лекарства. Сведения о лекарствах хранятся в специальной ведомости: наименование лекарства; количество; цена; срок хранения (в месяцах). Какие препараты имеют срок более 3 месяцев?
- 9 вариант** На аптечном складе хранятся лекарства. Сведения о лекарствах хранятся в специальной ведомости: наименование лекарства; количество; цена; срок хранения (в месяцах). Сколько стоят все препараты, хранящиеся на складе?
- 10 вариант** При поступлении в университет лица, получившие оценку «неудовлетворительно» на первом экзамене, ко второму экзамену не допускаются. Считая фамилии абитуриентов и их оценки после первого экзамена исходными данными, составить список абитуриентов, допущенных ко второму экзамену.
- 11 вариант** Даны результаты переписи населения, которые хранятся в памяти ЭВМ. Напечатать фамилии, имена и подсчитать общее число жителей, родившихся после 1990 года.
- 12 вариант** Дан массив, содержащий сведения об учениках школы. Заполнить второй массив данными об учениках только 9-х классах.
- 13 вариант** Дан массив, содержащий сведения об учениках школы. Выяснить, на сколько человек в восьмых классах больше, чем в девятых.
- 14 вариант** Багаж пассажира характеризуется количеством вещей и общим весом вещей. Дан массив, содержащий сведения о багаже нескольких пассажиров. Сведения о багаже пассажира представляют собой запись с двумя полями: количество вещей и вес в кг. Найти багаж, средний вес одной вещи в котором отличается не более чем на 0,3 кг от общего среднего веса одной вещи.
- 15 вариант** Багаж пассажира характеризуется количеством вещей и общим весом вещей. Дан массив, содержащий сведения о багаже нескольких пассажиров. Сведения о багаже пассажира представляют собой запись с двумя полями: количество вещей и вес в кг. Найти число пассажиров, имеющих более двух

вещей и число пассажиров, количество вещей которых превосходит среднее число вещей.

Контрольные вопросы:

1. Дайте определение записи.
2. Могут ли записи иметь вложенную структуру?
3. Дайте определение составного имени. Из каких частей оно состоит и как записывается?
4. Чем запись отличается от массива?
5. Какой оператор используется для упрощения доступа к полям записи

Лабораторная работа № 15

Разработка программ с использованием текстовых файлов

(2 часа)

1. Цель работы:

1.1. Ознакомление с возможностями организации файлов на внешних носителях.

1.2. Получение практических навыков работы с текстовыми и типизированными файлами.

2. Вопросы методики руководства

2.1. *Вводное инструктирование:*

- определение правил техники безопасности и организации рабочего места;
- определение цели работы (изложена выше);
- связь работы с изученными темами «Файлы»;
- ознакомление студентов с порядком выполнения работы (постановка задачи, последовательное выполнение пунктов задания);
- указания по ведению записей, оформлению результатов и составлению отчета;
- выдача заданий.

2.2. *Текущее инструктирование:*

- оказание помощи студентам при создании программы;
- ответы на вопросы студентов.

2.3. *Подведение итогов:*

- устный опрос студентов по результатам, полученным в ходе наблюдений и экспериментов;
- контроль выполнения программы;
- ответы на контрольные вопросы;
- зачет выполнения работы.

3. Оборудование, материалы:

- 3.1. ПЭВМ, инструментальная среда программирования Turbo Pascal 7.1
- 3.2. Методические указания по выполнению работы

4. Теоретическая часть:

- 4.1. Файловый тип данных;
- 4.2. Структура текстовых файлов;
- 4.3. Стандартные процедуры и функции для работы с файлами;
- 4.4. Файлы записей;

5. Практическая часть:

- 5.1. Составить программу с использованием текстового файла для решения задачи согласно конкретного варианта. Текстовый файл создается заранее с помощью текстового редактора.

6. Варианты заданий:

1 вариант

Напишите программу, которая считывает текстовый файл и определяет, содержатся ли в нем строчные буквы латинского алфавита. Записать эти буквы во второй текстовый файл.

2 вариант

Напишите программу, которая считывает текстовый файл и определяет количество содержащихся в нем n -символьных слов. Найденные слова записать во второй текстовый файл.

3 вариант

Напишите программу, которая считывает текстовый файл, содержащий русские фамилии, и записывает во второй текстовый файл те фамилии, которые начинаются на буквы *A* и *B*.

4 вариант

Напишите программу, которая считывает из текстового файла слова и записывает в новый текстовый файл только те из них, которые начинаются с указанной буквы.

5 вариант

Дан текстовый файл, содержащий строки. Найти число строк.

6 вариант

Дан текстовый файл, содержащий строки. Найти число строк, начинающихся с заданной буквы.

7 вариант

Дан текстовый файл, содержащий строки. Найти число строк, заканчивающихся заданной буквой.

8 вариант

Дан текстовый файл, содержащий строки. Найти самую короткую строку.

9 вариант

Дан текстовый файл. Вставить в начало каждой строки ее номер.

10 вариант

Дан текстовый файл. Дописать в его конце информацию о количестве строк в данном файле.

11 вариант

Дан текстовый файл. Дописать в конец информацию о числе символов в каждой строке.

12 вариант

Дан текстовый файл. Дописать в конец информацию о числе элементов в каждой строке.

13 вариант

Дан текстовый файл. Дописать в конец информацию о числе элементов во всем файле.

14 вариант

Дан текстовый файл. Подсчитайте количество пустых строк.

15 вариант

Дан текстовый файл. Перепишите его во второй файл сохраняя его строчную структуру и удаляя пустые строки.

Контрольные вопросы:

1. Какие типы файлов вы можете перечислить? В чем их отличия?
2. Сколько компонентов может содержать файл?
3. Нужно ли при определении файла указывать его длину?
4. Куда помещается при записи очередной компонент файла?
5. Какой файл можно открыть одновременно для чтения и записи?
6. Какие стандартные процедуры и функции используются при работе с текстовыми файлами?

Лабораторная работа № 16

Разработка программ с использованием типизированных файлов

(2 часа)

1. Цель работы:

1.1. Ознакомление с возможностями организации файлов на внешних носителях.

1.2. Получение практических навыков работы с текстовыми и типизированными файлами.

2. Вопросы методики руководства

2.1. *Вводное инструктирование:*

- определение правил техники безопасности и организации рабочего места;
- определение цели работы (изложена выше);
- связь работы с изученными темами «Файлы»;
- ознакомление студентов с порядком выполнения работы (постановка задачи, последовательное выполнение пунктов задания);
- указания по ведению записей, оформлению результатов и составлению отчета;
- выдача заданий.

2.2. *Текущее инструктирование:*

- оказание помощи студентам при создании программы;
- ответы на вопросы студентов.

2.3. *Подведение итогов:*

- устный опрос студентов по результатам, полученным в ходе наблюдений и экспериментов;
- контроль выполнения программы;
- ответы на контрольные вопросы;
- зачет выполнения работы.

3. Оборудование, материалы:

- 3.1. ПЭВМ, инструментальная среда программирования Turbo Pascal 7.1
- 3.2. Методические указания по выполнению работы

4. Теоретическая часть:

- 4.1. Файловый тип данных;
- 4.2. Структура текстовых файлов;
- 4.3. Стандартные процедуры и функции для работы с файлами;
- 4.4. Файлы записей;

5. Практическая часть:

- 5.1. Составить программу с использованием типизированного файла для решения задачи согласно конкретного варианта.

6. Варианты заданий:

1 вариант

1. Заполнить файл прямого доступа F целыми числами, полученными с помощью генератора случайных чисел. Получить в файле G те компоненты файла F, которые являются четными.
2. Даны два файла чисел. Найти сумму из элементов первого файла, меньших «5», и элементов второго файла, меньших «3,7». Подсчет суммы оформить в виде подпрограммы.

2 вариант

1. Записать в файл последовательного доступа N действительных чисел. Вычислить произведение компонентов файла и вывести на печать.
2. Даны два файла чисел. Найти сумму из элементов первого файла, больших «4,3», и элементов второго файла, больших «8». Подсчет суммы оформить в виде подпрограммы.

3 вариант

1. Заполнить файл прямого доступа F целыми числами, полученными с помощью генератора случайных чисел. Получить в файле G все компоненты файла f, которые делятся на m и не делятся на n.

2. Даны два файла чисел. Найти сумму из элементов первого файла, меньших «5», и элементов второго файла, больших «9,7». Подсчет суммы оформить в виде подпрограммы.

4 вариант

1. Заполнить файл прямого доступа *f* целыми числами, полученными с помощью генератора случайных чисел. Из файла *f* получить файл *g*, исключив повторные вхождения чисел. Вывести файлы *f* и *g* на печать.
2. Даны два файла чисел. Найти сумму из элементов первого файла, больших «5,9», и элементов второго файла, меньших «0,7». Подсчет суммы оформить в виде подпрограммы.

5 вариант

1. Записать в файл прямого доступа *N* натуральных чисел. Переписать в другой файл прямого доступа те элементы, которые кратны *k*. Вывести полученный файл на печать.
2. Даны два файла чисел. Найти сумму из элементов первого файла, меньших «-9,6», и элементов второго файла, больших «8». Подсчет суммы оформить в виде подпрограммы.

6 вариант

1. Заполнить файл прямого доступа *N* целыми числами, полученными с помощью генератора случайных чисел. Найти сумму минимального и максимального элементов файла.
2. Даны два файла чисел. Найти сумму из элементов первого файла, меньших «24», и элементов второго файла, меньших «-0,8». Подсчет суммы оформить в виде подпрограммы

7 вариант

1. Записать в файл *f* прямого доступа *n* натуральных чисел с помощью генератора случайных чисел. Получить в другом файле прямого доступа *g* все компоненты файла *f*, кроме тех, которые кратны *k*. Вывести полученный файл на печать.

2. Даны два файла чисел. Найти сумму из элементов первого файла, меньших «9», и элементов второго файла, меньших «4,4». Подсчет суммы оформить в виде подпрограммы.

8 вариант

1. Заполнить файл прямого доступа f целыми числами, полученными с помощью генератора случайных чисел. Найти количество удвоенных нечетных чисел среди компонентов файла.
2. Даны два файла чисел. Найти сумму из элементов первого файла, меньших «5,8», и элементов второго файла, больших «8,6». Подсчет суммы оформить в виде подпрограммы

9 вариант

1. Заполнить файл прямого доступа f целыми числами с помощью генератора случайных чисел. Найти количество квадратов нечетных чисел среди компонентов.
2. Даны два файла чисел. Найти сумму из элементов первого файла, больших «4,5», и элементов второго файла, меньших «3». Подсчет суммы оформить в виде подпрограммы.

10 вариант

1. Записать в файл прямого доступа f целые числа, полученные с помощью генератора случайных чисел. Найти наибольшее из значений модулей компонентов с нечетными номерами.
2. Даны два символьных файла. Конкатенировать файлы в третий файл и вывести результат на экран.

11 вариант

1. Записать в файл прямого доступа f целые числа, полученные с помощью генератора случайных чисел. Заполнить файл g числами, которые являются произведениями соседних компонентов файла f.
2. Даны два символьных файла. Конкатенировать их в первый файл и вывести результат на экран.

12 вариант

1. Записать в файл прямого доступа n элементов последовательности

$$b_n = 1 - \frac{1}{2!} + \frac{1}{3!} - \frac{1}{4!} + \dots + (-1)^{n-1} \cdot \frac{1}{n!}.$$

Вывести на печать те компоненты файла, для которых выполняется $|b_n| > \varepsilon$, где ε - заданное число.

2. Построить конкатенацию символьного файла самого с собой, только записанного задом наперед. Записать результат во второй файл и вывести результат на экран.

13 вариант

1. Записать в файл прямого доступа f n действительных чисел a_1, a_2, \dots, a_n .

Организовать новый файл прямого доступа, элементы которого вычисляются

$$b_k = \frac{\sum_{k=1}^i a_k}{i}.$$

Вывести полученный файл на печать.

2. Задан файл. Добавить к каждому элементу файла символ «!» и результат записать во второй файл. Вывести результат на экран.

14 вариант

1. Записать в файл прямого доступа n натуральных чисел: a_1, a_2, \dots, a_n (числа

получить с помощью генератора случайных чисел). Сформировать новый

файл прямого доступа, элементами которого являются числа

$$a_1, a_1 \cdot a_2, a_1 \cdot a_2 \cdot a_3, \dots, a_1 \cdot a_2 \cdot a_3 \cdot \dots \cdot a_n.$$

2. Даны два файла чисел. Найти сумму из элементов первого файла, больших «5,9», и элементов второго файла, больших «-3,7». Подсчет суммы оформить в виде подпрограммы..

15 вариант

1. Записать в файл прямого доступа N целых чисел, полученных с помощью генератора случайных чисел. Подсчитать количество пар противоположных чисел среди компонентов этого файла.

2. Даны два символьных файла. Конкатенировать файлы в третий файл и вывести результат на экран.

Контрольные вопросы:

1. Какие типы файлов вы можете перечислить? В чем их отличия?
2. Сколько компонентов может содержать файл?
3. Нужно ли при определении файла указывать его длину?
4. Куда помещается при записи очередной компонент файла?
5. Какой файл можно открыть одновременно для чтения и записи?
6. Какие стандартные процедуры и функции используются при работе с типизированными файлами?

Лабораторная работа № 17

Разработка программ с использованием файлов записей.

(2 часа)

1. Цель работы:

1.1. Ознакомление с возможностями организации файлов на внешних носителях.

1.2. Получение практических навыков работы с типизированными файлами и записями.

2. Вопросы методики руководства

2.1. Вводное инструктирование:

- определение правил техники безопасности и организации рабочего места;
- определение цели работы (изложена выше);
- связь работы с изученными темами «Файлы», «Записи»;
- ознакомление студентов с порядком выполнения работы (постановка задачи, последовательное выполнение пунктов задания);
- указания по ведению записей, оформлению результатов и составлению отчета;
- выдача заданий.

2.2. Текущее инструктирование:

- оказание помощи студентам при создании программы;
- ответы на вопросы студентов.

2.3. Подведение итогов:

- устный опрос студентов по результатам, полученным в ходе наблюдений и экспериментов;
- контроль выполнения программы;
- ответы на контрольные вопросы;
- зачет выполнения работы.

3. Оборудование, материалы:

- 3.1. ПЭВМ, инструментальная среда программирования Turbo Pascal 7.1
- 3.2. Методические указания по выполнению работы

4. Теоретическая часть:

- 4.1. Файловый тип данных;
- 4.2. Стандартные процедуры и функции для работы с файлами;
- 4.3. Файлы записей;

5. Практическая часть:

- 5.1. Составить программу с использованием файла записей для решения задачи согласно конкретного варианта.

6. Варианты заданий: см. лабораторную работу № 13-14.

Контрольные вопросы:

1. Какие стандартные процедуры и функции используются при работе с файлами записей?
2. Какова структура файла записей?
3. В чем отличие файла записей от массива записей?

Лабораторная работа № 18-20

Графические возможности Turbo Pascal

(6 часов)

1. Цель работы:

1.1. Получение навыков в написании программ с использованием графических процедур и функций.

1.2. Получение навыков в написании программ с использованием видеостраниц.

2. Вопросы методики руководства

2.1. *Вводное инструктирование:*

- определение правил техники безопасности и организации рабочего места;
- определение цели работы (изложена выше);
- связь работы с изученной темой «Графические возможности Turbo Pascal 7.0»;
- ознакомление студентов с порядком выполнения работы (постановка задачи, последовательное выполнение пунктов задания);
- указания по ведению записей, оформлению результатов и составлению отчета;
- выдача заданий.

2.2. *Текущее инструктирование:*

- оказание помощи студентам при создании программы;
- ответы на вопросы студентов.

2.3. *Подведение итогов:*

- устный опрос студентов по результатам, полученным в ходе наблюдений и экспериментов;
- контроль выполнения программы;
- ответы на контрольные вопросы;
- зачет выполнения работы.

3. Оборудование, материалы:

- 3.1. ПЭВМ, инструментальная среда программирования Turbo Pascal 7.0
- 3.2. Методические указания по выполнению работы

4. Теоретическая часть:

- 4.1. Инициализация графического режима экрана;
- 4.2. Графические процедуры и функции модуля *Graph*.
- 4.3. Анимационные эффекты с использованием видеостраниц;

5. Практическая часть:

- 5.1. Изобразить на экране рисунок, используя процедуры модуля *Graph*;
- 5.2. Создать эффект движения созданного рисунка, используя динамическую память.

Контрольные вопросы:

1. Какие константы используются для задания логических операций при работе с изображениями с помощью процедуры *PutImage*?
2. С помощью каких процедур выполняется вывод текста на экран в графическом режиме?
3. Какой процедурой выполняется установка типа линий?
4. Какой вид имеет система координат в графическом режиме?
5. Какие процедуры выполняют перемещение указателя в графическом режиме?
6. Какой процедурой выполняется установка типа заливки?
7. Какая процедура предназначена для инициализации графического режима и какие параметры она имеет?
8. Какие процедуры выполняют рисование линии в графическом режиме?

Лабораторная работа № 21

Разработка программ с использованием рекурсии.

(2 часа)

1. Цель работы:

- 1.1. Ознакомление с динамической структурой данных – однонаправленным списком.
- 1.2. Получение навыков работы с переменными ссылочного типа.

2. Вопросы методики руководства

2.1. Вводное инструктирование:

- определение правил техники безопасности и организации рабочего места;
- определение цели работы (изложена выше);
- связь работы с изученной темой «Динамические типы данных»;
- ознакомление студентов с порядком выполнения работы (постановка задачи, последовательное выполнение пунктов задания);
- указания по ведению записей, оформлению результатов и составлению отчета;
- выдача заданий.

2.2. Текущее инструктирование:

- оказание помощи студентам при создании программы;
- ответы на вопросы студентов.

2.3. Подведение итогов:

- устный опрос студентов по результатам, полученным в ходе наблюдений и экспериментов;
- контроль выполнения программы;
- ответы на контрольные вопросы;
- зачет выполнения работы.

3. Оборудование, материалы:

- 3.1. ПЭВМ, инструментальная среда программирования Turbo Pascal 7.0

3.2. Методические указания по выполнению работы

4. Теоретическая часть:

4.1. Статическая и динамическая память;

4.2. Динамические переменные и указатели;

4.3. Стандартные процедуры для работы с динамическими данными и указателями;

4.4. Динамические структуры данных;

5. Практическая часть:

5.1. Для конкретного варианта создать входную информацию в виде списка данных. Разработать программу для решения поставленной задачи. По окончании работы вывести результативный список на экран.

6. Варианты заданий:

1 вариант Составить программу, которая вставляет в список L новый элемент F за каждым вхождением элемента E .

2 вариант Составить программу, которая вставляет в список L новый элемент F перед первым вхождением элемента E .

3 вариант Составить программу, которая вставляет в непустой список L , элементы которого упорядочены по неубыванию, новый элемент E так, чтобы сохранилась упорядоченность.

4 вариант Составить программу, которая удаляет из списка L все элементы E , если таковые имеются.

5 вариант Составить программу, которая удаляет из списка L за каждым вхождением элемента E один элемент, если таковой имеется и он отличен от E .

6 вариант Составить программу, которая удаляет из списка L все отрицательные элементы.

7 вариант Составить программу, которая переносит в конец непустого списка L его первый элемент.

8 вариант Составить программу, которая формирует список L , включив в него по одному разу элементы, которые входят одновременно в оба списка $L1$ и $L2$.

- 9 вариант** Составить программу, которая формирует список L , включив в него по одному разу элементы, которые входят в список $L1$, но не входят в список $L2$.
- 10 вариант** Многочлен $P(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$ с целыми коэффициентами можно представить в виде списка. Проверить на равенство многочлены P и Q .
- 11 вариант** Многочлен $P(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$ с целыми коэффициентами можно представить в виде списка. Вычислить значение многочлена P в целочисленной точке x .
- 12 вариант** Многочлен $P(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$ с целыми коэффициентами можно представить в виде списка. Построить многочлен Q – производную многочлена P .
- 13 вариант** Многочлен $P(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$ с целыми коэффициентами можно представить в виде списка. Построить многочлен R – сумму многочленов P и Q .
- 14 вариант** Дан список, содержащий целые числа. Удалить те его элементы, которые кратны заданному числу k .
- 15 вариант** Дан список, содержащий целые числа. Определить количество различных элементов этого списка.
- 16 вариант** Составить программу, которая должна находить среднее арифметическое элементов списка вещественных чисел и заменять все вхождения числа x на число y .
- 17 вариант** Дан список вещественных чисел. Проверить наличие в нем двух одинаковых чисел.
- 18 вариант** Дан список вещественных чисел. Перенести в начало списка его последний элемент.
- 19 вариант** Дан список вещественных чисел. Скопировать в конец списка его первый элемент.
- 20 вариант** Дан список вещественных чисел. Вставит список в себя вслед за последним элементом.

- 21 вариант** Дан список вещественных чисел. Удалить из списка второй элемент, если такой есть.
- 22 вариант** Дан список вещественных чисел. Удалить из списка первый отрицательный элемент, если такой есть.
- 23 вариант** Дан список вещественных чисел. Удалить из списка все нечетные числа.
- 24 вариант** Многочлен $P(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$ с целыми коэффициентами можно представить в виде списка. Вычислить значение многочлена P в целочисленной точке x .
- 25 вариант** Многочлен $P(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$ с целыми коэффициентами можно представить в виде списка. Построить многочлен Q – производную многочлена P .
- 26 вариант** Многочлен $P(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$ с целыми коэффициентами можно представить в виде списка. Построить многочлен R – сумму многочленов P и Q .
- 27 вариант** Составить программу, которая вставляет в список L новый элемент F за каждым вхождением элемента E .
- 28 вариант** Составить программу, которая вставляет в список L новый элемент F перед первым вхождением элемента E .
- 29 вариант** Составить программу, которая вставляет в непустой список L , элементы которого упорядочены по неубыванию, новый элемент E так, чтобы сохранилась упорядоченность.
- 30 вариант** Дан список вещественных чисел. Перенести в начало списка его последний элемент.

Контрольные вопросы:

1. В чем отличие статических переменных от динамических?
2. Что называется указателем?
3. Для чего необходимо значение *nil* при работе с указателями?
4. На каком этапе (компиляции, выполнения) происходит выделение памяти под динамические структуры данных?

5. Что описывается в первую очередь – тип указателя или тип данных, на который он указывает?
6. Каково назначение процедуры *NEW*?
7. Укажите причины использования динамической памяти?
8. Поясните роль базового типа при работе с указателями.
9. Каково назначение процедуры *DISPOSE*?

СОДЕРЖАНИЕ

1. Марченко А.И., Марченко Л.А. Программирование в среде Turbo Pascal 7.0 - К.: ВЕК+, 2000.- 464 с.
2. Семакин И.Г., Шестаков А.П. Основы программирования: учебник. – М.: Мастерство, 2002.- 432 с.
3. Васюкова Н.Д., Тюляева В.В. Практикум по основам программирования. Язык Паскаль. – М.: «Высшая школа», 1990 - 160 с.
4. Энго Франк. Как программировать на Delphi.- К.: «ДиаСофт», -1997- 320с.
5. Программирование на языке Паскаль: задачник / под ред. Усковой О.Ф. – СПб.: Питер, 2003. – 336 с.
6. Немнюгин С.А. Turbo Pascal: Практикум. – СПб.: Питер, 2002. – 256 с.

