



## **1 Методические указания по профессиональному модулю ПМ. 01. Разработка модулей программного обеспечения для компьютерных систем**

Профессиональный модуль ПМ. 01. «Разработка модулей программного обеспечения для компьютерных систем» изучается в третьем, четвертом и пятом семестрах. В процессе изучения профессионального модуля используются различные виды занятий: лекции, лабораторные и самостоятельные (индивидуальные) занятия, курсовое проектирование. На первом занятии по данному модулю необходимо ознакомить обучающихся с требованиями к его изучению.

В процессе проведения занятий используются следующие образовательные технологии:

- развивающее обучение;
- проблемное обучение;
- информационно-коммуникационные технологии и т.д.

Профессиональный модуль ПМ. 01. «Разработка модулей программного обеспечения для компьютерных систем» включает в себя:

- МДК.01.01 Разработка программных модулей;
- МДК.01.02 Поддержка и тестирование программных модулей;
- МДК.01.03 Разработка мобильных приложений;
- МДК.01.04 Системное программирование;
- УП. 01. Учебная практика;
- ПП.01. Производственная практика.

## **2 Методические рекомендации при работе над конспектом лекций во время проведения теоретических занятий**

В ходе учебных занятий необходимо вести конспектирование учебного материала. Обращать внимание на категории, формулировки, раскрывающие содержание тех или иных явлений и процессов, научные выводы и практические рекомендации, положительный опыт в изучении проблем логики. Желательно оставить в рабочих конспектах поля, на которых делать пометки из рекомендованной литературы, дополняющие материал, а также подчеркивающие особую важность тех или иных теоретических положений. Задавать преподавателю уточняющие вопросы с целью уяснения теоретического материала, разрешения спорных ситуаций.

## **3 Методические рекомендации при подготовке к практическим занятиям**

Составной частью учебного процесса в колледже являются практические занятия.

Практическое занятие - это занятие, проводимое под руководством преподавателя в учебной аудитории, направленное на углубление теоретических знаний и овладение практическим опытом. Перед практическим занятием следует изучить теоретический материал, обращая внимание на практическое их применение.

На практическом занятии главное уяснить связь решаемых ситуаций с теоретическими положениями. Для ведения записей на практических занятиях обычно ведется журнал практических работ. Логическая связь теоретических и практических занятий заключается в том, что информация, полученная на лекции, осмысливается и перерабатывается, при помощи преподавателя анализируется до мельчайших подробностей, после чего прочно усваивается.

Успешное освоение профессионального модуля требует регулярных, последовательных и систематических занятий.

Выполнение обучающимися практических работ направлено на:

- обобщение, систематизация, углубление, закрепление полученных теоретических знаний по конкретным темам междисциплинарных курсов профессионального модуля
- формирование умений применять полученные знания на практике, реализация единства интеллектуальной и практической деятельности;
- развитие личностных качеств, направленных на устойчивое стремление к самосовершенствованию: самопознанию, самоконтролю, самооценке, саморазвитию и саморегуляции;
- выработку таких профессионально значимых качеств, как самостоятельность, ответственность, точность, творческая инициатива.

Практические занятия включают следующие необходимые структурные элементы:

- инструктаж, проводимый преподавателем;
- самостоятельная деятельность обучающихся;
- обсуждение итогов выполнения работы (задания).

Перед выполнением практического задания проводится проверка знаний обучающихся – их теоретической готовности к выполнению задания. Количество, объем и содержание практических занятий определяются рабочей программой по междисциплинарным курсам профессионального модуля.

Практические занятия по междисциплинарным курсам профессионального модуля направлены на формирование у обучающихся практических и профессиональных умений при решении задач и при выполнении определенных заданий, необходимых в последующей профессиональной деятельности.

Наряду с формированием умений и овладением практического опыта в процессе практических занятий теоретические знания обобщаются, систематизируются, углубляются и конкретизируются.

Содержание практических занятий фиксируется в рабочей учебной программе междисциплинарных курсов профессионального модуля в разделе «Содержание учебной дисциплины» и планируется с расчетом, чтобы за отведенное время они могли быть выполнены большинством обучающихся.

При выполнении заданий обучающиеся имеют возможность пользоваться лекционным материалом, с разрешения преподавателя, осуществлять деловое общение с товарищами.

Оценка компетентности осуществляется следующим образом: по окончании выполнения задания обучающиеся оформляют журнал лабораторных работ, который затем выносится на завершающий этап формы изучения дисциплины.

**Методические указания к выполнению практических работ представлены в приложении.**

**– МДК.01.01 Разработка программных модулей;**

## **4 Методические рекомендации по выполнению курсовой работы**

Курсовая работа, предусмотренная рабочим учебным планом, является важным этапом в усвоении обучающимися изучаемого профессионального модуля. Выполнение курсовой работы рассматривается как вид учебной работы и реализуется в пределах времени, отведенного на его изучение. Процесс выполнения курсовой работы способствует формированию у обучающихся профессиональных и общих компетенций, аналитического мышления. В ходе работы над выполнением курсовой работы обучающийся учится грамотно и четко излагать мысли, что важно для выполнения им выпускной квалификационной работы.

Выполнение курсовой работы осуществляется под руководством преподавателя профессионального модуля. Результатом данной работы должен стать курсовая работа, выполненная и оформленная в соответствии с установленными требованиями. Курсовая работа подлежит обязательной защите. Методические указания к выполнению курсовой работы по представлены в виде отдельной методической разработки.

## 5 Методические рекомендации для самостоятельной работы

Самостоятельная работа - это планируемая работа обучающихся, выполняемая по заданию и при методическом руководстве преподавателя, но без его непосредственного участия. Междисциплинарный курс профессионального модуля предусматривает два вида самостоятельной работы:

- аудиторная;
- внеаудиторная.

Самостоятельная работа выполняет ряд функций, среди которых особенно выделяются:

- развивающая (повышение культуры умственного труда, приобщение к творческим видам деятельности, обогащение интеллектуальных способностей обучающихся);
- ориентирующая и стимулирующая (процессу обучения придается ускорение и мотивация);
- воспитательная (формируются и развиваются профессиональные качества специалиста);
- исследовательская (новый уровень профессионально-творческого мышления);
- информационно-обучающая (учебная деятельность обучающихся на аудиторных занятиях).

Целью самостоятельных занятий является самостоятельное более глубокое изучение обучающимися вопросов междисциплинарного курса с использованием рекомендуемой литературы и других информационных источников.

Задачами самостоятельной работы являются:

- систематизация и закрепление полученных теоретических знаний и практических умений обучающихся;
- углубление и расширение теоретических знаний;
- формирование умения использовать справочную литературу;
- развитие познавательных способностей и активности обучающихся: творческой инициативы, ответственности и организованности.

Внеаудиторная самостоятельная работа включает такие формы работы, как:

1) индивидуальные занятия (домашние занятия):

- изучение программного материала дисциплины (работа с учебником и конспектом лекции);
- изучение рекомендуемых литературных источников;
- конспектирование источников;
- работа с нормативными документами;
- работа с электронными информационными ресурсами и ресурсами Internet;
- составление схем, таблиц, для систематизации учебного материала;
- подготовка презентаций
- ответы на контрольные вопросы;
- написание рефератов;

2) групповая самостоятельная работа студентов:

- подготовка к занятиям, проводимым с использованием активных форм обучения (круглые столы, деловые игры и др.);
- анализ деловых ситуаций (мини-кейсов) и др.

3) получение консультаций для разъяснений по вопросам изучаемой дисциплины.

## **6 Методические рекомендации по учебной и производственной практикам**

Практика обучающихся является составной частью образовательного процесса и составной частью основной профессиональной образовательной программы (ОПОП) среднего профессионального образования (СПО), обеспечивающей реализацию Федерального государственного образовательного стандарта (ФГОС) СПО.

Практика имеет целью комплексного освоения обучающимися всех видов профессиональной деятельности по специальности среднего профессионального образования, формирование общих и профессиональных компетенций, а также приобретение необходимых умений и опыта практической работы по специальности 09.02.03 Программирование в компьютерных системах.

Видами практики обучающихся, осваивающих основную профессиональную образовательную программу (ОПОП) среднего профессионального образования (СПО), являются: учебная практика и производственная практика по профилю специальности.

Основной целью учебной практики является получение первичных профессиональных умений и навыков, формирование общих и профессиональных компетенций.

Производственная практика (по профилю специальности) направлена на формирование у обучающегося общих и профессиональных компетенций, приобретение практического опыта и реализуется в рамках профессиональных модулей ОПОП СПО по каждому из видов профессиональной деятельности, предусмотренных ФГОС СПО по специальности 09.02.03 Программирование в компьютерных системах.

Методические указания по выполнению учебной практики УП.02 и производственной практик ПП.02 представлены в отдельных методических материалах.

## **7 Методические рекомендации по работе с литературой**

Авиационный колледж обеспечивает учебно-методическую и материально -техническую базу для организации самостоятельной работы обучающихся.

Библиотека обеспечивает:

- учебный процесс необходимой литературой и информацией (комплектует библиотечный фонд учебной, методической, научной, справочной литературой в соответствии с учебными планами и программами, в том числе на электронных носителях);
- доступ к основным информационным образовательным ресурсам, информационной базе данных, в том числе возможность выхода в Интернет.

Колледж:

- обеспечивает доступность всего необходимого учебно-методического и справочного материала.
- разрабатывает: учебно-методические комплексы, программы, пособия, материалы по учебным дисциплинам в соответствии с ФГОС; методические указания по дисциплинам, методические рекомендации по планированию и организации самостоятельной работы обучающихся, методические рекомендации по выполнению ВКР и т.д

## **8 Описание мультимедийных средств**

В преподавании дисциплины новые возможности, предоставляемые мультимедийными средствами, нашли самое разнообразное применение.

На занятиях используются несложные мультимедийные документы, которые может создать сам преподаватель и обучающийся в программах Microsoft Word или Power Point и т.д.

Системный подход в мультимедийных проектах даёт возможность использовать средства визуальной информации не только в качестве иллюстративного материала, сопровождающего рассказ преподавателя, но и структурировать подготовленные материалы применительно к специализациям обучающихся. Все это положительно влияет на усвоение материалов и поднимает уровень и качество обученности.

*Приложение*

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ**  
**по выполнению практической работы в среде**  
**объектно-ориентированного программирования**  
**«Разработка проектов»**

для студентов специальности 09.02.07

**СОДЕРЖАНИЕ**

Введение.....	3
1 Постановка задачи .....	4
1.1 Подготовка к выполнению задания .....	5
2 Теоретический материал.....	6
2.1 Описание компонента TStringGrid .....	6
2.2 Описание компонента Treeview .....	10
2.3 Описание компонента TChart .....	15
3 Пример программной реализации проекта.....	19
4 Требования к оформлению отчета.....	26
5 Список используемой литературы .....	27

**ВВЕДЕНИЕ**

Практические занятия, как вид учебных занятий, строятся в форме самостоятельного выполнения студентом определенных программой практических учебных задач.

Данные методические указания предназначены для обеспечения студентов 3-го курса методическими и теоретическими материалами, необходимыми при выполнении и оформлении практической работы «Разработка проектов в среде ООП Lazarus».

Разработка проекта позволяет студентам:

- закрепить теоретические знания, полученные на лекциях и уроках по изучению основ программирования в объектно-ориентированной среде ;
- изучить свойства и методы компонентов этой среды, приемы программирования задач, максимально приближенных к реальным прикладным задачам;
- улучшить навыки использования современных методов программирования и методов разработки эффективных алгоритмов решения прикладных задач.

**Цели практической работы:**

1. Создание алгоритма и программного средства (ПС) для реализации конкретной прикладной задачи. ПС должно быть разработано с учетом современных требований прикладного программирования с обеспечением удобного и понятного интерфейса.
2. Приобретение навыков самостоятельной работы по отладке и тестированию ПС.
3. Приобретение опыта оформления отчета по конкретному заданию.

Далее рассматривается конкретный пример разработки проекта и необходимый для этого теоретический материал.

**1 Постановка задачи**

**Разработать проект** с использованием следующих компонентов управления – меню, страниц-вкладок, переключателей и командных кнопок, моделирующего работу отдела госдепартамента по контролю торговых организаций. В столбцах исходной таблицы задаются следующие данные - дата проведения контроля, тип организации (магазин, склад, киоск), дата окончания лицензии, сумма штрафа, срок оплаты штрафа. Создать

программу контроля заполнения обязательных полей и дат. Сформировать на страницах-вкладках новые таблицы (или списки) со следующими данными:

- Отсортировать перечень проверенных организаций в алфавитном порядке.
- Данные организаций, у которых просрочены сроки оплаты штрафа на текущий день, их количество и общая сумма штрафов,
- Список магазинов, которые контролировались в отчетном месяце (номер отчетного месяца вводится в окно EDIT на форме),
- Список организаций, у которых заканчиваются лицензии в отчетном месяце, указав их наименования в алфавитном порядке.

Сформировать форму 2 «Оплата штрафа» для вывода счета для оплаты, с указанием суммы штрафа, вида нарушения и даты. Организация может выбрать вид оплаты- наличными, по кредитной карте или чеком, а если выбран «наличными», то в какой валюте- рублями, долларами или евро. Причем, если включена одна из радиокнопок - «в долларах» или «в евро», то должны быть активизированы надпись «Курс валюты» и окно для ввода этого курса. Иначе - эти объекты должны быть серыми (недоступны). Если выбрана оплата по кредитной карте, то проверить ее срок окончания относительно текущей даты и выводить сообщение, если она просрочена.

### 1.1 Подготовка к выполнению задания

1. На первом этапе выполнения работы необходимо четко распределить на какой форме и с помощью каких компонентов будет выполняться ввод исходных данных, контроль вводимой информации и какие данные будут выводиться на форму с комментариями.

2. Далее разрабатывается визуальная часть проекта, состоящая из нескольких форм, включая форму – заставку с названием проекта и справочными данными о разработчике, парольную форму и функциональные формы – приложения.

3. Создать меню для сохранения исходной таблицы в файле и считывания ее из файла.

3. Вся необходимая информация отображается на форме: исходные данные вводятся в компоненты ввода и редактирования данных, результаты решения выводятся в соответствующие компоненты для вывода данных. Элементы управления процессом вычисления выбирает студент по своему усмотрению - кнопки, меню, переключатели.

4. Необходимо продумать алгоритм программного контроля правильности ввода информации.

5. Студент должен предусмотреть кнопку «Статистика», по которой на форму выводятся диаграммы, отображающие соотношение данных в указанных в задании массивах.

6. Заполнить в исходных таблицах 7-10 строк с необходимыми данными и записать их в файл.

## 2 Теоретический материал

### 2.1 Описание компонента TStringGrid

Компонент TStringGrid предназначен для создания таблиц, в ячейках которых располагаются произвольные текстовые данные.

Данные таблицы доступны для чтения или редактирования. Таблица делится на две части – фиксированную и рабочую. Фиксированная часть служит для показа заголовков колонок и рядов (шапки), а также для ручного управления их размерами.

С помощью свойств FixedCols и FixedRows можно задать любое количество фиксированных колонок и рядов. Если эти свойства имеют значение 0, то таблица не содержит фиксированной зоны. Рабочая часть – это остальная часть таблицы. Она может содержать произвольное количество колонок и рядов, более того, эти величины могут изменяться программно. Таблица может иметь полосы прокрутки. При прокрутке рабочей области фиксированная область не исчезает, но меняется ее содержимое – заголовки колонок и рядов.

Компонент StringGrid предназначен в первую очередь для отображения таблиц, но следует помнить, что все данные записаны в символьном формате. На этапе визуального конструирования проекта необходимо задать несколько свойств компонента для удобного и правильного его использования. Все свойства и методы компонента **TStringGrid** задаются в окне Object Inspector на соответствующих вкладках.

#### Основные свойства компонента TStringGrid

Таблица 1

Свойство	Описание
Cells[Acol,Arow]	Содержимое ячейки. Конкретная ячейка определяется парой чисел – номером колонки и номером ряда, на пересечении которых она находится (нумерация начинается с нуля). Свойство Cells имеет тип String, поэтому программа

	может легко прочитать или записать содержимое нужной ячейки.
Cols[index]	Список строк, содержащихся в столбце с индексом index
Rows[index]	Список строк, содержащихся в строке с индексом index
ColCount	Число столбцов таблицы
RowCount	Число строк таблицы
FixedCols	Число фиксированных не прокручиваемых столбцов
FixedRows	Число фиксированных не прокручиваемых строк
FixedColor	Цвет фона фиксированных ячеек
Col	Номер колонки с ячейкой, имеющей фокус ввода
Row	Содержит номер ряда ячейки, имеющей фокус ввода

Назначение подсвойств свойства + TGridOptions

Таблица 2

Элемент	Описание
GoFixedVertLine	Колонки фиксированной зоны разделяются вертикальными линиями
GoFixedHorzLine	Ряды фиксированной зоны разделяются горизонтальными линиями
GoVertLine	Колонки рабочей зоны разделяются вертикальными линиями
GoHorzLine	Колонки рабочей зоны разделяются горизонтальными линиями
GoRangeSelect	Разрешено выделение нескольких ячеек. Игнорируется, если в множество включен элемент goEditing
GoDrawFocusSelected	Разрешено показывать ячейку, имеющую фокус ввода, так же, как выделенную ячейку
<b>GoEditing</b>	Разрешено редактирование ячейки. Игнорируется, если включен элемент GoRowSelect. Редактирование начинается после щелчка мышью или нажатой клавиши <F2> и завершается при щелчке на другой ячейку или нажатой клавиши <Enter>
<b>GoAlwaysShowEditor</b>	Разрешено редактировать ячейку, имеющую фокус ввода: редактирование возможно после выделения ячейки клавишей <Tab> (<Shift>+<Tab>). Игнорируется, если не включен элемент goEditing

Каждая ячейка таблицы имеет два индекса – номер строки и номер столбца, аналогично элементам обычного двумерного массива. Доступ к каждой ячейке таблицы выполняется командой - Cells [I,j].

Но в таблице можно работать не только с каждой ячейкой, но и со всей строкой или со всем столбцом, как с массивами.

Команды работы со всеми элементами строки:

```
buf1:=tab1.Rows[i].Text;
```

```
tab1.Rows[i].Text:=tab1.Rows[i+1].Text;
```

либо со всеми элементами столбца :

```
buf1:=tab1.Cols[i].Text;
```

```
tab1.Cols[i].Text:=tab1.Cols[i+1].Text;
```

Пример программы записи в таблицу данных так, чтобы программа проверяла номер последнего столбца по строке и переходила на новую строку автоматически.

Процедура T1KeyPress обрабатывает событие – нажатие клавиши клавиатуры при заполнении таблицы T1.

```
procedure TForm1.T1KeyPress(Sender: TObject; var Key: Char);
```

```
begin
```

```
  if Key = #13 then
```

```
    with T1 do
```

```
      if Col < ColCount - 1 then Col := Col + 1
```

```
      else if Row < RowCount - 1 then
```

```
        begin
```

```
          Row := Row + 1;
```

```
          Col := 0;
```

```
        end
```

```
      else
```

```
        begin
```

```
          T1.RowCount:=T1.RowCount + 1;
```

```
          Row := Row + 1;
```

```
          Col := 0;
```



```
end;
end;
```

## 2.2 Описание компонента Tree View

Компонент Tree View представляет собой дерево со своей структурой разветвлений, который применяют при работе с таблицами. Основной частью работы с TreeView является свойство Items типа TTreeNode.

При работе с компонентом, расположите его на форме своей программы. После этого, для настройки структуры дерева нажмите два раза по компоненту и откроется окно TreeView Items Editors, в котором вы можете легко настроить структуру дерева.

Описание свойств окна TreeView Items Editors.

### Кнопки:

New Item - создание раздела дерева,

New SubItem - создание узла дерева,

Delete - удаление узла \ раздела,

Load - загрузка дерева из файла.

Слева от кнопок располагается окно, в котором можно посмотреть построенное дерево.

### Поля ввода:

Text - название узла или раздела.

Остальные поля ввода, это настройка изображения узлов \ разделов.

### Список свойств типа TTreeNode.

property HasChildren: Boolean; - Равно True, если узел имеет дочерние узлы.

property Count: Integer; - Счетчик числа дочерних узлов данного узла.

property Item [Index: Integer] : TTreeNode; - Список дочерних узлов.

property Parent: TTreeNode; - Ссылка на объект — родительский узел (верхнего уровня).

property Level: Integer; - Уровень, на котором находится узел. Для корневого узла это свойство равно 0; его потомки имеют значение Level=1 и т. д.

property Text: string; - Текст узла.

property Data: Pointer; - Данные, связанные с узлом.

property TreeView: TCustomTreeView; - Ссылка на компонент TTreeView, в котором отображается данный узел

property Handle: HWND; - Дескриптор окна компонента TTreeView, в котором отображается данный узел

property Owner: TTreeNode; - Ссылка на компонент TTreeNode, которому принадлежит данный узел.

property Index: Longint; - Индекс узла в списке своего родителя.

property IsVisible: Boolean; - Равно True, если узел видим (все его родительские узлы развернуты).

property Selected: Boolean; - Равно True, если данный узел выбран пользователем.

property Focused: Boolean; - Равно True, если данный узел выбран пользователем для редактирования текста узла.

property Expanded: Boolean; - Равно True, если данный узел развернут (показываются его дочерние узлы).

Для программной работы с компонентом используются следующие методы:

Создание раздела : `TreeView1.Items.Add(nil,'7"A");`

Создаётся переменная с типом TTreeNode. Сначала в эту переменную заносим главную информацию об узле:

```
Node:=TreeView1.Items.Add(nil,'7"A");
```

И только после этого подключать к нему дочерние узлы.

```
TreeView1.Items.AddChild(node,'Иванов');
```

```
TreeView1.Items.AddChild(node,'Петрова');
```

### Методы позволяющие добавлять узлы и разделы в дерево TreeView и TTreeNode.

```
function Add (Node: TTreeNode; const S: string) : TTreeNode;
```

Узел добавляется последним в тот же список, что и узел Node.

```
function AddObject (Node: TTreeNode; const S: string; Ptr: Pointer) : TTreeNode;
```

То же, что и метод Add, но с узлом связываются данные из параметра Ptr:

```
function AddFirst (Node: TTreeNode; const S: string): TTreeNode;
```

Узел добавляется первым в тот же список, что и узел Node:

```
function AddObjectFirst (Node: TTreeNode; const S: string; Ptr: Pointer) : TTreeNode; - То же, что и метод AddFirst, но с узлом связываются данные из параметра Ptr:
```

```
function AddChildfNode: TTreeNode; const S: string): TTreeNode; - Узел добавляется последним в список дочерних узлов узла Node.
```

```
function AddChildObject (Node: TTreeNode; const S: string; Ptr: Pointer) : TTreeNode; - То же, что и метод AddChild, но с узлом связываются данные из параметра Ptr.
```

```
function AddChildFirst (Node: TTreeNode; const S: string): TTreeNode;
```

### Пример использования компонента **TreeView**

Форма программы.

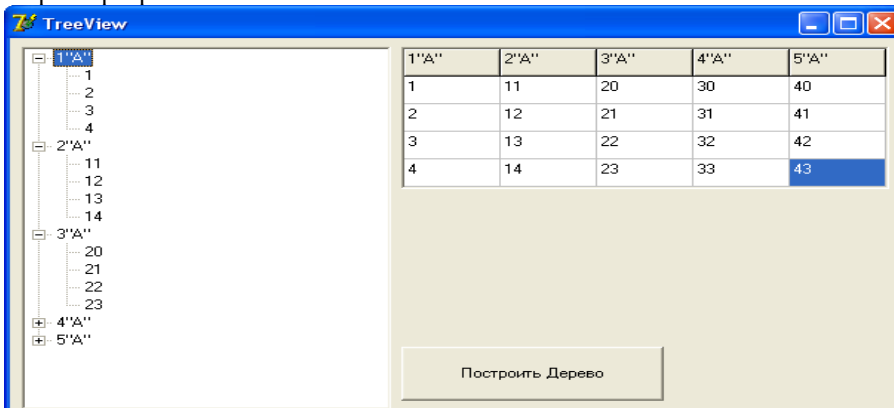


Рисунок 1 – Форма для демонстрации работы компонента **TreeView**

### Текст модуля программы.

```
unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Grids, ComCtrls;
type
  TForm1 = class(TForm)
    TV1: TTreeView;
    SG1: TStringGrid;
    Button1: TButton;
  procedure FormCreate(Sender: TObject);
  procedure Button1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form1: TForm1;
implementation
{$R *.dfm}
procedure TForm1.FormCreate(Sender: TObject); //Создание шапки таблицы
begin
  sg1.Cells[0,0]:='1''A'';
  sg1.Cells[1,0]:='2''A'';
  sg1.Cells[2,0]:='3''A'';
  sg1.Cells[3,0]:='4''A'';
  sg1.Cells[4,0]:='5''A'';
end;
procedure TForm1.Button1Click(Sender: TObject);
//Нажатие кнопки «Построить Дерево»
var
  Node:TTreeNode; //переменная узла
  i,j:byte;
begin
  For i:=0 to SG1.ColCount-1 do //цикл создания узла
  begin
    Node:=TV1.Items.Add(nil,sg1.Cells[i,0]); //Создание узла по шапке
    for j:=1 to SG1.RowCount-1 do //цикл создания дочерних узлов
      TV1.Items.AddChild(Node,sg1.Cells[i,j]);
    //создание узла по таблице с выводом на компонент
  end; end; end.
```

### 2.3 Описание компонента **TChart** для построения диаграмм.

Компонент **TChart** формирует в приложении различные диаграммы и графики. Для каждой диаграммы рядом располагается её легенда, надпись сверху диаграммы как заголовок. Компонент TChart находится на закладке



Additional, в виде круга с разноцветными частями.

Основные свойства компонента настраиваются с помощью окна Editing, для его вызова нужно два раза кликнуть по компоненту. Данное окно разделено на две основные закладки настроек - Chart и Series.

На закладке Chart настраиваются основные свойства диаграммы:

Series (создание диаграммы):

Add – создание диаграммы.

Delete – удаление диаграммы.

Title – название диаграммы.

Change – замена диаграммы.

Zoom – увеличение.

Allow Scroll – расположение ползунка.

Axis (оси диаграммы или графика):

Show Axis – показывать оси.

Legend – настройка легенды для диаграммы.

Panel – настройка панели отображения.

3D – настройка 3D отображения.

Назначение закладок Series:

Format – формат отображения диаграммы.

General – основные настройки диаграммы.

Marks – настройка маркеров диаграммы.

Data Source – настройка источника данных.

После выбора настраиваемой оси, рядом находятся закладки для изменения настроек, это: Scale (размер), Title (надписи), Labels (этикетки), Ticks (метки),

Minor (незначительные метки), Position (расположение).

Visible – видимость.

Title (надпись):

В ComboBox е выберите тип расположения надписи.

Title – можно использовать как название диаграммы, т.к. находится сверху.

Foot – можно использовать как какие-либо пояснения к диаграмме.

**Порядок работы с компонентом TChart:**

1) Для начала со страницы Additional возьмите компонент Chart и расположите его на форме. 2) Двойным кликом по компоненту открывается окно настройки (Editing Chart).

3) В закладке Chart-> Series, нажмите на кнопку Add для добавления новой диаграммы, выберите тот вид диаграммы, который вам нужен и нажмите ОК.

4) По кнопке Title задать название диаграммы.

5) Теперь для построения диаграммы программно нужно обращаться к названию диаграммы, используя команду (метод) Add.

Её синтаксис такой:

Add(группа данных для отображения, название, цвет);

Например: Add(mass, 'равные нулю', clBlue );

В основном в окне Editing Chart настраиваются все визуальные параметры компонента.

Для более удобного обращения к компоненту зададим краткое имя свойству Series. Для этого в окне Object Tree View (дерево проекта) выбрать компонент Chart1 и кликнуть по его плюсику, изменить имя Series1 на S. В дальнейшем в программе это имя будет использоваться для задания значений для диаграммы.

**Пример использования компонента TChart.**

Задание: Отобразить на диаграмме количество элементов списка ListBox1 равных нулю, меньше нуля, больше

**Исходный текст модуля Unit1.**

```
procedure TForm1.Button1Click(Sender: TObject); // кнопка подсчитать
```

```
var bol,men,raw,ch:integer; i:byte;
```

```
begin
```

```
raw:=0; men:=0; bol:=0;
```

```
for i:=0 to ListBox1.Items.Count-1 do
```

```
begin
```

```
ch:=StrToInt(Listbox1.Items.Strings[i]);
```

```
if ch=0 then inc(raw);
```

```

if ch>0 then inc(bol);
if ch<0 then inc(men);
end;
s.Add(raw,'Равных нулю',clBlue);
s.Add(bol,'Больше нуля',clRed);
s.Add(men,'Меньше нуля',clYellow);
end;
procedure TForm1.Button2Click(Sender: TObject); // кнопка ввод
var k,kk:byte;
begin
kk:=strtoint(InputBox('Ввод','Введите кол. чисел:'));
for k:=1 to kk do
ListBox1.Items.Add(InputBox('Ввод','Введите '+inttostr(k)+' число:'));
end;
procedure TForm1.Button3Click(Sender: TObject); // кнопка Очистка
begin
ListBox1.Clear;
end;
end.

```

### 3 Пример приложения проекта и программного кода для реализации поставленной задачи

В данной работе удобно использовать компонент TPageControl –закладки страниц, который позволяет расположить на форме несколько таблиц как бы «слоями». Каждая страница-вкладка имеет свой индекс-номер, начиная с нуля, то есть система формирует массив страниц-вкладок. Номер активной вкладки хранится в свойстве PageControl1.TabIndex, следовательно в программе можно организовать переход по этому индексу к нужной вкладке. Каждая вкладка имеет Name – TabSheet. Надпись выполняется в свойстве TabSheet1.Caption.

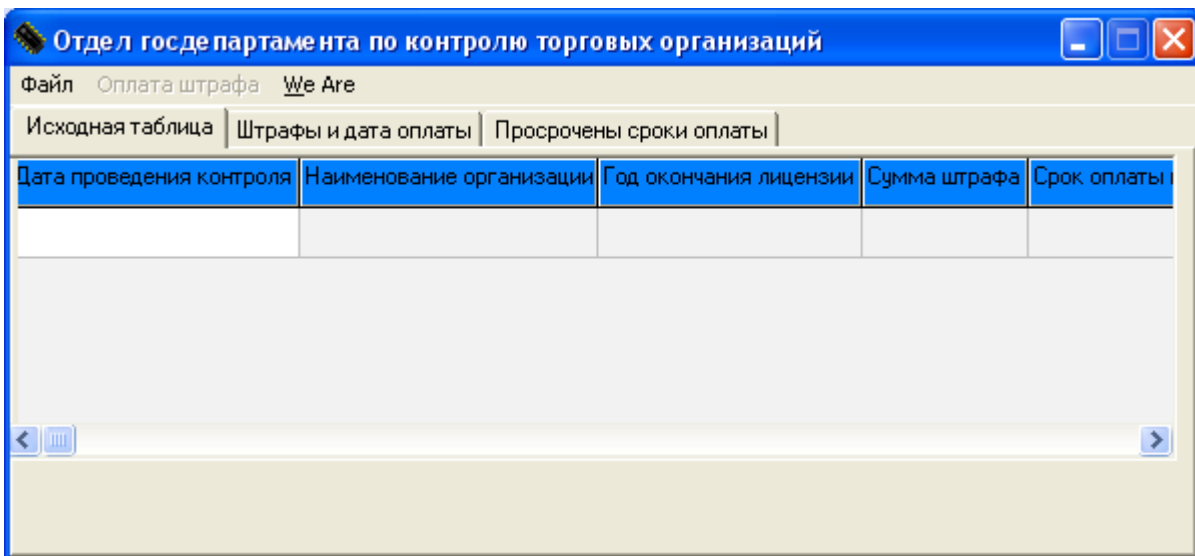


Рисунок 4- На форме использованы компоненты управления расчетом – Меню и 3 вкладки, на каждой из них устанавливается таблица – компонент **StringGrid**.

#### Исходный код программы:

```

unit Unit1;
var
    // объявление глобальных переменных
    Form1 :TForm1;
    ks :byte;
    bs :byte;
implementation
procedure TForm1.N5Click(Sender: TObject); //выход
begin
close;
end;
procedure TForm1.N7Click(Sender: TObject); //Активизация соответствующих вкладок
begin
Pagecontrol1.ActivePage:=TabSheet1;
end;
end.

```

```

end;
procedure TForm1.N8Click(Sender: TObject);
begin
    Pagecontrol1.ActivePage:=TabSheet2;
end;
procedure TForm1.N9Click(Sender: TObject);
begin
    Pagecontrol1.ActivePage:=TabSheet3;
end;
procedure TForm1.FormCreate(Sender: TObject); // Заполнение строки-шапки исходной таблицы
begin
    T1.Cells[0,0]:= ' дата проведения контроля';
    T1.Cells[1,0]:= ' Наименование ';
    T1.Cells[2,0]:= ' год окончания лицензии';
    T1.Cells[3,0]:= ' Дата окончания лицензии';
    T1.Cells[4,0]:= ' сумма штрафа';
    T2.Cells[0,0]:= ' срок оплаты';
end;
procedure TForm1.CloseALL1Click(Sender: TObject);
begin
    close;
end;
procedure TForm1.FormActivate(Sender: TObject); // активизация первой вкладки
begin
    PageControl1.ActivePage:=TabSheet1; end;
procedure TForm1.N11Click(Sender: TObject);
begin
    Form3 := TForm3.Create(Application);
    Form3.show;
end;
procedure TForm1.T1KeyPress(Sender: TObject; var Key: Char);
// заполнение исходной таблицы
begin
    if Key = #13 then
        with T1 do
            if Col < ColCount - 1 then
                Col := Col + 1
            else if Row < RowCount - 1 then
                begin
                    Row := Row + 1;
                    Col := 0; end
            else begin
                T1.RowCount:=T1.RowCount + 1;
                Row := Row + 1;
                Col := 0; end;
end;
procedure TForm1.N13Click(Sender: TObject);
var i,j: Integer;
begin
    j:=T1.Row;
    T1.Rows[j].Clear;
    for i:=j to T1.RowCount-2 do
        T1.Rows[i].Assign(T1.Rows[i+1]);
        T1.RowCount:=T1.RowCount-1;
end;
procedure TForm1.PageControl1Change(Sender: TObject);
// переходы по вкладкам
var i,j,k,h :Byte;
min,max :Integer;
begin
    Case PageControl1.TabIndex of // Оператор выбора по индексу
                                вкладки
        0:Begin

```

```

    N11.Enabled:=False;
End;
1:begin
k:=1;
N11.Enabled:=True;
for i:=1 to T1.RowCount -1 do
begin
if T1.Cells[3,i]<>" Then
If StrToDate(T1.Cells[3,i])<Date Then
Begin
T2.Cells[0,k]:=T1.Cells[0,i];
T2.Cells[1,k]:=T1.Cells[3,i];
T2.Cells[2,k]:=T1.Cells[4,i];
k:=k+1;
h:=h+1;
label4.Caption:=intToStr(h);
T2.RowCount:=i+1;
end; end; End;
2:begin
max:=StrToInt(T1.Cells[2,1]);
N11.Enabled:=False;
for j:=2 to T1.RowCount-1 do
if StrToInt(T1.Cells[2,j])>max then
max:=StrToInt(T1.Cells[2,j]);
Edit1.Text:=IntToStr(max);
min:=StrToInt(T1.Cells[2,1]);
for j:=2 to T1.RowCount-1 do
if StrToInt(T1.Cells[2,j])<min then
min:=StrToInt(T1.Cells[2,j]);
Edit2.Text:=IntToStr(min); end;
end; end;
procedure TForm1.N10Click(Sender: TObject); // очистка компонентов
begin
Edit1.Text:="";
Edit2.Text:="";
T1.RowCount:=2;
T1.Rows[1].Clear;
end;
// процедура открытия файла и считывания данных в таблицу

procedure TForm2.N2Click(Sender: TObject);
var b:string; F:TextFile; m:String; n:String; i,j:Integer;
begin
if OpenFileDialog1.Execute then
begin
b:=(OpenDialog1.FileName);
AssignFile(f,b);
Reset(f);
with tabl do
begin
readln(f,m);
readln(f,n);
RowCount:=StrToInt(m); // кол строк считывается из 1-ой строки файла
ColCount:=StrToInt(n); // кол столбцов считывается из 2-ой строки файла
for i:=0 to ColCount-1 do
for j:=1 to RowCount-1 do
begin
readln(f,m); // считываются данные из таблицы
cells[i,j]:=m;
end; end;
end; // конец with
CloseFile(F);
end; // конец процедуры

```

Комплекс практических заданий позволяет закрепить и углубить теоретические знания, полученные студентами при изучении учебного материала по прикладному программированию в среде объектно-ориентированного программирования ООП Lazarus.

В процессе выполнения практических заданий студентам предлагается ряд задач прикладного характера, для решения которых необходимо хорошо знать системные средства компьютеров и возможности современных языков программирования, ориентированных на использование этих средств.

ООП Lazarus.— одна из самых мощных систем программирования высокого уровня, позволяющих на самом современном уровне создавать прикладные программы. Мощность и гибкость языка программирования ООП Lazarus выгодно отличающее эту среду от других инструментов RAD (класс инструментальных средств ускоренной разработки программ). Ядром этой среды ООП является язык Object Pascal.

Для проведения практических занятий разработан комплекс практических работ по всем темам, изучаемым в теоретической части курса. Каждая работа включает до 30 вариантов разной степени сложности. Практические работы должны проводиться в компьютерном классе с использованием ПК, что определяет обязательное выполнение студентами индивидуальных заданий различного уровня сложности. Такой режим работы позволяет преподавателю-руководителю практики контролировать выполнение работ и консультировать в случае необходимости индивидуально каждого студента.

Таким образом, практические работы позволяют комплексно подойти к решению поставленной задачи - систематизировать знания студентов, развить и закрепить навыки самостоятельной работы на ПК, подготовить студентов к процессу курсового и дипломного проектирования.

Постоянное развитие средств вычислительной техники диктует необходимость периодических внесений изменений в содержание программного материала.

Приведенное в программе распределение часов по темам является рекомендательным и может изменяться в пределах общего объема часов по практике.

#### Практические работы №1, №2 и №3

Тема: Программирование приложения в **Lazarus** с использованием всех ранее изученных элементов управления расчетами. Алгоритмы прикладных приложений в Lazarus с использованием компонентов ListBox. Обеспечить контроль вводимой информации

#### Варианты заданий

##### **Вариант 1.**

Известен список студентов группы (StringGrid) и средний балл студента за весь период обучения (ListBox2). Выполнить следующие задания:

- Отсортировать список студентов;
- Дописать новых студентов;
- Выбрать всех студентов у которых средний балл выше или равен 4.
- Удалить студентов, у которых средний балл равен 2.

##### **Вариант 2.**

Известен список студентов группы (StringGrid) и средний балл студента за весь период обучения (ListBox2). Выполнить следующие задания:

- Отсортировать список студентов;
- Дописать новых студентов;
- Выбрать всех студентов у которых средний балл выше или равен 3;
- Удалить отдельных студентов из списка.

##### **Вариант 3.**

Известен список студентов группы (StringGrid) и средний балл студента за весь период обучения (ListBox2). Выполнить следующие задания:

- Отсортировать список студентов по баллу;
- Дописать новых студентов;
- Выбрать всех студентов, у которых средний балл выше или равен 3.5;
- Удалить из списка студентов, средний балл у которых меньше 2.5.

##### **Вариант 4.**

Известен список студентов группы (StringGrid) и средний балл студента за весь период обучения (ListBox2). Выполнить следующие задания:

- Дописать новых студентов;
- Отсортировать список студентов по среднему баллу;
- Дописать новых студентов;
- Выбрать всех студентов, у которых средний балл выше или равен 4.5.

**Вариант 5.**

Известно фамилия, имя, отчество абонента (ListBox.1), номер телефона (ListBox.2).

Выполнить следующие задания:

- Дописать в неупорядоченный телефонный справочник новых абонентов;
- Упорядочить абонентов по телефону;
- Дописать в упорядоченный телефонный справочник новых абонентов
- Выбрать абонентов с фамилией на букву «К»

**Вариант 6.**

Известно фамилия, имя, отчество абонента (ListBox.1), номер телефона (ListBox.2).

Выполнить следующие задания:

- Дописать в неупорядоченный телефонный справочник новых абонентов;
- Упорядочить абонентов по № телефона;
- Дописать в упорядоченный телефонный справочник новых абонентов, сохранив упорядоченность;
- Предусмотреть выборку по телефону или по фамилии.

**Вариант 6-2.**

По ГИБДД известны марка автомобиля (StringGrid), номер машины (ListBox2), номер двигателя (ListBox3), фамилия владельца (ListBox4), год производства (ListBox5).

Выполнить следующие задания:

- Дописать новые машины;
- Отсортировать данные по году производства;
- Дописать новые машины, сохранив упорядоченность;
- Выбрать машины определенной марки;
- Выбрать машины по фамилии владельца;

**Вариант 7.**

Известно фамилия, имя, отчество(ФИО) абонента (ListBox.1), номер телефона (ListBox.2).

Выполнить следующие задания:

- Дописать в неупорядоченный телефонный справочник новых абонентов;
- Упорядочить абонентов по телефону;
- Дописать в упорядоченный телефонный справочник новых абонентов, сохранив упорядоченность по ФИО;
- Предусмотреть выборку по телефону или по фамилии;
- Удалить отдельных абонентов.

**Вариант 8.**

Известно фамилия, имя, отчество (ФИО) абонента (ListBox.1), номер телефона (ListBox.2), дата рождения (ListBox3).

Выполнить следующие задания:

- Дописать в неупорядоченный телефонный справочник новых абонентов;
- Выбрать абонентов старше 25 лет ;
- Дописать в упорядоченный телефонный справочник новых абонентов, сохранив упорядоченность по ФИО ;
- Предусмотреть выборку по телефону или по фамилии.

**Вариант 9.**

По библиотеке известны код литературы (StringGrid), автор (ListBox2), название (ListBox3), год издания (ListBox4).

Выполнить следующие задания:

- Дописать новые книги;
- Отсортировать данные по году издания;
- Дописать новые книги, сохранив упорядоченность;
- Выбрать книги данного года издания;
- Удалить книги определенного автора;

**Вариант 10.**

По библиотеке известны код литературы (StringGrid), автор (ListBox2), название (ListBox3), год издания (ListBox4).

Выполнить следующие задания:

- Дописать новые книги;
- Отсортировать данные по коду литературы;
- Дописать новые книги, сохранив упорядоченность;
- Выбрать книги данного автора;
- Выбрать книги данного кода;

**Вариант 11.**

Известно фамилия, имя, отчество(ФИО) абонента (ListBox.1), номер телефона (ListBox.2), дата установки телефона (ListBox3).

Выполнить следующие задания:

- Дописать в неупорядоченный телефонный справочник новых абонентов;
- Упорядочить абонентов по абоненту;
- Выбрать абонентов, которым установлен телефон весной текущего года;
- Предусмотреть выборку по телефону или по фамилии;
- Предусмотреть выборку по дате установки телефона.

**Вариант 12.**

Известен список студентов группы) (StringGrid), число рождения (ListBox2) дата рождения (ListBox3), год поступления в колледж (ListBox4).. Выполнить следующие задания:



- Отсортировать список студентов;
- Дописать новых студентов, сохранив упорядоченность;
- Выбрать студентов 2 курса ;
- Выбрать всех студентов, у которых день рождения в мае месяце..

**Вариант 13.**

Известен список студентов группы (StringGrid), дата рождения (ListBox2), группа (ListBox3). Выполнить следующие задания:

- Выбрать студентов указанной группы;
- Дописать новых студентов, сохранив упорядоченность;
- Отсортировать студентов по году рождения;
- Выбрать всех студентов, кому меньше 16 лет, равно 16 лет, больше 16 лет.

**Вариант 14.**

Известны фамилии сотрудников (StringGrid), образование (ListBox2), стаж работы (ListBox3).

Выполнить следующие задания:

- Дописать новых сотрудников;
- Отсортировать сотрудников по фамилии;
- Дописать новых сотрудников, сохранив упорядоченность;
- Выбрать сотрудников по стажу работы – больше 10 лет;

**Вариант 15.**

Известны фамилии сотрудников (StringGrid), образование (ListBox2), зарплата (ListBox3).

Выполнить следующие задания:

- Дописать новых сотрудников;
- Отсортировать сотрудников по фамилии;
- Выбрать сотрудников с максимальной и минимальной зарплатой;
- Отсортировать сотрудников по зарплате;

**Вариант 16.**

Известны фамилии сотрудников (StringGrid), образование (ListBox2), стаж работы (ListBox3).

Выполнить следующие задания:

- Дописать новых сотрудников;
- Отсортировать сотрудников по фамилии;
- Отсортировать сотрудников по стажу работы;
- Выбрать с макс.стажем и миним. стажем;
- Выбрать сотрудников с указанным стажем работы.

**Вариант 17.**

По библиотеке известны код литературы (StringGrid), автор (ListBox2), название (ListBox3), год издания (ListBox4).

Выполнить следующие задания:

- Дописать новые книги;
- Отсортировать данные по автору;
- Дописать новые книги, сохранив упорядоченность;
- Выбрать книги данного автора;
- Выбрать книги данного года издания;
- Удалить книги определенного кода.

**Вариант 18.**

По магазину известны наименование игрушек (StringGrid), цена (ListBox2), возрастная группа (ListBox3), для которой данная игрушка рекомендована.

Выполнить следующие задания:

- Дописать новые игрушки;
- Отсортировать данные по наименованию игрушек;
- Дописать новые игрушки, сохранив упорядоченность;
- Выбрать игрушки по возрастной группе;
- Выбрать игрушки по наименованию.

**Вариант 19.**

По магазину известны наименование игрушек (StringGrid), цена (ListBox2), возрастная группа (ListBox3), для которой данная игрушка рекомендована.

Выполнить следующие задания:

- Дописать новые игрушки;
- Отсортировать данные по цене игрушек;
- Дописать новые игрушки, сохранив упорядоченность;
- Выбрать игрушки по возрастной группе;
- Выбрать игрушки по цене.

**Вариант 20.**

По магазину известны наименование игрушек (StringGrid), цена (ListBox2), возрастная группа (ListBox3), для которой данная игрушка рекомендована.

Выполнить следующие задания:

- Дописать новые игрушки;
- Отсортировать данные по возрастной группе;
- Дописать новые игрушки, сохранив упорядоченность;

- Выбрать игрушки по возрастной группе;
- Выбрать игрушки по наименованию;
- Выбрать игрушки по цене.

**Вариант 21.**

По библиотеке известны код литературы (StringGrid), автор (ListBox2), название (ListBox3), год издания (ListBox4).  
Выполнить следующие задания:

- Дописать новые книги;
- Отсортировать данные по коду литературы;
- Дописать новые книги, сохранив упорядоченность;
- Выбрать книги данного автора;
- Выбрать книги данного кода;
- Удалить книги определенного года.

**Вариант 22.**

Известно фамилия, имя, отчество (ФИО) абонента (ListBox.1), номер телефона (ListBox.2), дата рождения (ListBox3). Выполнить следующие задания:

- Дописать в неупорядоченный телефонный справочник новых абонентов;
- Упорядочить абонентов по абоненту;
- Дописать в упорядоченный телефонный справочник новых абонентов, сохранив упорядоченность по(ФИО);
- Предусмотреть выборку по телефону или по фамилии.
- Упорядочить абонентов по году рождения.

**Вариант 23.**

По ГИБДД известны марка автомобиля (StringGrid), номер машины (ListBox2), номер двигателя (ListBox3), фамилия владельца (ListBox4), год производства (ListBox5).

Выполнить следующие задания:

- Дописать новые машины;
- Отсортировать данные по маркам машин;
- Дописать новые машины, сохранив упорядоченность;
- Выбрать машины определенной марки;
- Выбрать машины данного года;
- Удалить машины, которые находятся в эксплуатации более 20 лет.

**Вариант 24.**

По ГИБДД известны марка автомобиля (StringGrid), номер машины (ListBox2), номер двигателя (ListBox3), фамилия владельца (ListBox4), год производства (ListBox5).

Выполнить следующие задания:

- Дописать новые машины;
- Отсортировать данные по фамилиям владельцев;
- Дописать новые машины, сохранив упорядоченность;
- Выбрать машины определенной марки;
- Выбрать машины по фамилии владельца;
- Удалить машины, которые находятся в эксплуатации более 15 лет.

**Вариант 25.**

По ГИБДД известны марка автомобиля (StringGrid), номер машины (ListBox2), номер двигателя (ListBox3), фамилия владельца (ListBox4), год производства (ListBox5).

Выполнить следующие задания:

- Дописать новые машины;
- Отсортировать данные по году производства;
- Дописать новые машины, сохранив упорядоченность;
- Выбрать машины определенной марки;
- Выбрать машины по фамилии владельца;
- Выбрать машины некоторого года;
- Выбрать машины, которые находятся в эксплуатации более 15 лет.

**Вариант 26.**

По ГИБДД известны марка автомобиля (StringGrid), номер машины (ListBox2), фамилия владельца (ListBox3), год производства (ListBox4).

Выполнить следующие задания:

- Дописать новые машины;
- Отсортировать данные по году производства;
- Дописать новые машины, сохранив упорядоченность;
- Выбрать машины определенной марки;
- Выбрать машины по фамилии владельца;
- Выбрать машины некоторого года;
- Выбрать машины, которые находятся в эксплуатации от 10 до 15 лет;

Практическая работа №4.

ТЕМА: Работа с несколькими таблицами. Компоненты TStringGrid. Выбор данных по заданным критериям.

**ЦЕЛЬ:** Закрепить полученные навыки в разработке объектно-ориентированных программных модулей с использованием визуальных компонентов среды OOLazarus для ввода и отображения числовой и символьной информации.

Методические указания:

1. Вся необходимая информация отображается на форме: исходные данные вводятся в компоненты ввода и редактирования данных, результаты решения выводятся в соответствующие компоненты для вывода данных. Элементы управления процессом вычисления выбирает студент по своему усмотрению.
2. Организовать программный контроль правильности ввода информации.

### Задания

1. Расположить на форме компонент **Вкладки** и расположить на них 3 таблицы- одну с исходными данными о студенте: ФИО, отделение, Негруппы, стипендия. А во вторую выбирать студентов группы, указанной в окне Edit, в третью выбрать тех студентов, которые получают стипендию.
2. Расположить на форме компонент **Вкладки** и расположить на них 3 таблицы- одну с исходными данными о работнике: ФИО, должность, телефон, зарплата. Во вторую выбирать работников с зарплатой, указанной в окне Edit, в третью выбрать тех работников, ФИО которых начинается на букву «И».
3. Расположить на форме компонент **Вкладки** и расположить на них 3 таблицы- одну с исходными данными о студенте: ФИО, отделение, Негруппы, стипендия. А во вторую выбирать студентов отделения, указанного в окне Edit, в третью выбрать тех студентов, которые не получают стипендию.
4. Расположить на форме компонент **Вкладки** и расположить на них 3 таблицы- одну с исходными данными о работнике: ФИО, должность, телефон, зарплата. Во вторую выбирать работников с зарплатой, меньшей чем указана на форме в окне Edit, в третью выбрать тех работников, ФИО которых начинается на буквы «Пет».
5. Расположить на форме компонент **Вкладки** и расположить на них 3 таблицы- одну с исходными данными о студенте: ФИО, отделение, город проживания, пол. А во вторую выбирать студентов указанного отделения в окне Edit и живущих в указанном городе, в третью выбрать студентов, мужского пола.
6. Расположить на форме компонент **Вкладки** и расположить на них 3 таблицы- одну с исходными данными о товаре: наименование, стоимость, дата выпуска. Во вторую выбирать товары со стоимостью, указанной в окне Edit, в третью выбрать те товары, наименование которых начинается на букву «К».
7. Расположить на форме компонент **Вкладки** и расположить на них 3 таблицы- одну с исходными данными о студенте: ФИО, отделение, курс, стипендия. А во вторую выбирать студентов курса, указанного в окне Edit, в третью выбрать тех студентов, которые получают стипендию на втором курсе.
8. Расположить на форме компонент **Вкладки** и расположить на них 3 таблицы- одну с исходными данными о товаре: наименование, стоимость, дата выпуска. Во вторую выбирать товары с максимальной стоимостью, в третью выбрать те товары, наименование которых начинается на буквы «Ст».
9. Расположить на форме компонент **Вкладки** и расположить на них 3 таблицы- одну с исходными данными о работнике: ФИО, должность, телефон, зарплата. Во вторую выбирать работников с минимальной зарплатой, в третью выбрать тех работников, ФИО которых начинается на букву «И» из продавцов.
10. Расположить на форме компонент **Вкладки** и расположить на них 3 таблицы- одну с исходными данными о работнике: ФИО, дата рождения, телефон, зарплата. Во вторую выбирать работников родившихся летом, в третью выбрать тех работников, ФИО которых начинается на букву «И».
11. Расположить на форме компонент **Вкладки** и расположить на них 3 таблицы- одну с исходными данными о студенте: ФИО, отделение, Негруппы, стипендия. А во вторую выбирать студентов группы, указанной в окне Edit, в третью выбрать тех студентов, которые получают стипендию на отд ПВМ.
12. Расположить на форме компонент **Вкладки** и расположить на них 3 таблицы- одну с исходными данными о работнике: ФИО, должность, телефон, зарплата. Во вторую выбирать работников с телефоном, указанным в окне Edit, в третью выбрать тех работников, ФИО которых начинается на букву «К» с з/платой больше 5000.
13. Расположить на форме компонент **Вкладки** и расположить на них 3 таблицы- одну с исходными данными о работнике: ФИО, должность, дата рождения, зарплата. Во вторую выбирать работников, родившихся завтра, в третью выбрать тех работников, ФИО которых начинается на букву «М» с з/платой меньше 8000 из менеджеров.
14. Расположить на форме компонент **Вкладки** и расположить на них 3 таблицы- одну с исходными данными о работнике: ФИО, должность, дата рождения, зарплата. Во вторую выбирать работников, родившихся сегодня, в третью выбрать тех работников, ФИО которых начинается на буквы «Ст» с з/платой больше 7000 из бухгалтеров.
15. Расположить на форме компонент **Вкладки** и расположить на них 3 таблицы- одну с исходными данными о товаре: наименование, стоимость, дата окончания годности. Во вторую выбирать товары с максимальной стоимостью, в третью выбрать те товары, наименование которых начинается на буквы «С» и которые уже просрочены.
16. Расположить на форме компонент **Вкладки** и расположить на них 3 таблицы- одну с исходными данными о студенте: ФИО, отделение, дата рождения, стипендия. А во вторую выбирать студентов призывников отд ПВМ, в третью выбрать тех студентов, которые получают стипендию на 3 курсе.
17. Расположить на форме компонент **Вкладки** и расположить на них 3 таблицы- одну с исходными данными о товаре: наименование, стоимость, дата выпуска. Во вторую выбирать товары с максимальной стоимостью, в третью выбрать те товары, наименование которых начинается на буквы «Н», выпущенные в этом году в тек месяце.
18. Расположить на форме компонент **Вкладки** и расположить на них 3 таблицы- одну с исходными данными о работнике: ФИО, должность, телефон, зарплата. Во вторую выбирать работников с телефоном, указанным в окне Edit, в третью выбрать тех работников, ФИО которых начинается на букву «К» с з/платой больше 5000.
19. Расположить на форме компонент **Вкладки** и расположить на них 3 таблицы- одну с исходными данными о работнике: ФИО, должность, дата рождения, зарплата. Во вторую выбирать работников, родившихся завтра, в третью выбрать тех работников, ФИО которых начинается на букву «М» с з/платой больше 7000 из менеджеров.

20. Расположить на форме компонент **Вкладки** и расположить на них 3 таблицы- одну с исходными данными о работнике: ФИО, должность, дата рождения, зарплата. Во вторую выбирать работников , родившихся в мае, в третью выбрать тех работников, ФИО которых начинается на буквы «Ива» с з/платой больше 10000.
21. Расположить на форме компонент **Вкладки** и расположить на них 3 таблицы- одну с исходными данными о товаре: наименование, стоимость, дата окончания годности. Во вторую выбирать товары с мин стоимостью, в третью выбрать те товары, наименование которых начинается на буквы «М» и которые уже просрочены.
22. Расположить на форме компонент **Вкладки** и расположить на них 3 таблицы- одну с исходными данными о студенте: ФИО, отделение, дата рождения, стипендия. А во вторую выбирать студентов призывников отд ПВМ, в третью выбрать тех студентов, которые получают стипендию на 3 курсе.
23. Расположить на форме компонент **Вкладки** и расположить на них 3 таблицы- одну с исходными данными о товаре: наименование, стоимость, дата выпуска. Во вторую выбирать товары с мах стоимостью, в третью выбрать те товары, наименование которых начинается на буквы «Н», выпущенные в этом году в тек месяце .
24. Расположить на форме компонент **Вкладки** и расположить на них 3 таблицы- одну с исходными данными о работнике: ФИО, должность, дата рождения, зарплата. Во вторую выбирать работников , родившихся завтра, в третью выбрать тех работников, ФИО которых начинается на букву «М» с з/платой меньше 8000 из менеджеров.
25. Расположить на форме компонент **Вкладки** и расположить на них 3 таблицы- одну с исходными данными о работнике: ФИО, должность, телефон, зарплата. Во вторую выбирать работников с зарплатой, указанной в окне Edit, в третью выбрать тех работников, ФИО которых начинается на букву «И».
26. Расположить на форме компонент **Вкладки** и расположить на них 3 таблицы- одну с исходными данными о студенте: ФИО, отделение, Негруппы, стипендия. А во вторую выбирать студентов отделения, указанного в окне Edit, в третью выбрать тех студентов, которые не получают стипендию

#### Практические работы №5-6

Тема: Программирование приложения с использованием компонента TStringGrid для обработки элементов двумерного массива. Преобразование матрицы.

- Вариант1. Дана матрица размера  $M \times N$  и целые числа  $K1$  и  $K2$   
Поменять местами строки матрицы с номерами  $K1$  и  $K2$ .
- Вариант2. Дана матрица размера  $M \times N$  и целые числа  $K1$  и  $K2$   
Поменять местами столбцы матрицы с номерами  $K1$  и  $K2$ .
- Вариант3. Дана матрица размера  $M \times N$ . Преобразовать матрицу, поменяв местами минимальный и максимальный элемент в каждой строке.
- Вариант4. Дана матрица размера  $M \times N$ . Преобразовать матрицу, поменяв местами минимальный и максимальный элемент в каждом столбце.
- Вариант5. Дана матрица размера  $M \times N$ . Поменять местами строки, содержащие минимальный и максимальный элементы матрицы.
- Вариант6. Дана матрица размера  $M \times N$ . Поменять местами столбцы, содержащие минимальный и максимальный элементы матрицы.
- Вариант7 . Дана матрица размера  $M \times N$ . Поменять местами столбец с номером 1 и последний из столбцов, содержащих только положительные элементы. Если требуемых столбцов нет, то вывести матрицу без изменений.
- Вариант8. Дана матрица размера  $M \times N$ . Поменять местами столбец с номером  $N$  и первый из столбцов, содержащих только отрицательные элементы. Если требуемых столбцов нет, то вывести матрицу без изменений.
- Вариант9. Дана матрица размера  $M \times N$  ( $M$  — четное число). Поменять местами верхнюю и нижнюю половины матрицы.
- Вариант10. Дана матрица размера  $M \times N$  ( $N$  — четное число). Поменять местами левую и правую половины матрицы.
- Вариант11. Дана матрица размера  $M \times N$  ( $M$  и  $N$  — четные числа). Поменять местами левую верхнюю и правую нижнюю четверти матрицы.
- Вариант12. Дана матрица размера  $M \times N$  ( $M$  и  $N$  — четные числа). Поменять местами левую нижнюю и правую верхнюю четверти матрицы.
- Вариант13. Дана матрица размера  $M \times N$ . Зеркально отразить ее элементы относительно горизонтальной оси симметрии матрицы (при этом поменяются местами строки с номерами 1 и  $M$ , 2 и  $M - 1$  и т. д.).
- Вариант14. Дана матрица размера  $M \times N$ . Зеркально отразить ее элементы относительно вертикальной оси симметрии матрицы (при этом поменяются местами столбцы с номерами 1 и  $N$ , 2 и  $N - 1$  и т. д.).
- Вариант15 . Дана матрица размера  $M \times N$ . Поменять местами столбец с номером 1 и последний из столбцов, содержащих только положительные элементы. Если требуемых столбцов нет, то вывести матрицу без изменений.
- Вариант16. Дана матрица размера  $M \times N$ . Поменять местами столбец с номером  $N$  и первый из столбцов, содержащих только отрицательные элементы. Если требуемых столбцов нет, то вывести матрицу без изменений.
- Вариант17. Дана матрица размера  $M \times N$  и целые числа  $K1$  и  $K2$   
Поменять местами столбцы матрицы с номерами  $K1$  и  $K2$ .
- Вариант18. Дана матрица размера  $M \times N$ . Преобразовать матрицу, поменяв

местами минимальный и максимальный элемент в каждой строке.

Вариант19. Дана матрица размера  $M \times N$ . Преобразовать матрицу, поменяв местами минимальный и максимальный элемент в каждом столбце.

Вариант20. Дана матрица размера  $M \times N$ . Зеркально отразить ее элементы относительно горизонтальной оси симметрии матрицы (при этом поменяются местами строки с номерами 1 и  $M$ , 2 и  $M-1$  и т. д.).

Вариант21. Дана матрица размера  $M \times N$ . Зеркально отразить ее элементы относительно вертикальной оси симметрии матрицы (при этом поменяются местами столбцы с номерами 1 и  $N$ , 2 и  $N-1$  и т. д.).

Вариант22. Дана матрица размера  $M \times N$ . Поменять местами столбец с номером 1 и последний из столбцов, содержащих только положительные элементы. Если требуемых столбцов нет, то вывести матрицу без изменений.

Вариант23. Дана матрица размера  $M \times N$ . Поменять местами столбец с номером  $N$  и первый из столбцов, содержащих только отрицательные элементы. Если требуемых столбцов нет, то вывести матрицу без изменений.

Вариант24. Дана матрица размера  $M \times N$  ( $M$  — четное число). Поменять местами верхнюю и нижнюю половины матрицы.

Вариант25. Дана матрица размера  $M \times N$  ( $N$  — четное число). Поменять местами левую и правую половины матрицы.

#### Практические работы №7-8.

Тема: Разработка приложений по индивидуальным заданиям, с реализацией алгоритмов анализа данных в таблицах

##### Задания

1. Ввести в компонент **StringGrid1** исходные числа, используя функцию `InputBox()`.

Перенести в компонент **StringGrid2** в соответствующие клетки только четные числа из исходной матрицы, а в остальные — нули. При включенных флажках вывести на форму `MAX` и `MIN` элементы  $k$ -го столбца исходной матрицы. Ввод, формирование и очистка матриц выполняется в соответствии с пунктами меню.

2. Ввести в компонент **StringGrid1** исходные числа, используя функцию `InputBox()`.

В компонент **StringGrid2** в соответствующие клетки занести нечетные числа, если таких нет в числе, то — ноль. Вывести в последнюю строку количество нулей по столбцам в компоненте `StringGrid2` и отсортировать эту строку по возрастанию. Ввод, формирование и очистка матриц выполняется в соответствии с пунктами меню. По включенным флажкам вывести на форму `MAX` и `MIN` элементы  $K$ -го столбца исходной матрицы.

3. Ввести в компонент **StringGrid1** исходные числа, используя функцию `InputBox()`.

В компонент **StringGrid2** в соответствующие клетки занести количество «1», в исходных числах, если таких нет в числе, то — ноль. Вывести в последнюю строку количество нулей по столбцам в компоненте `StringGrid2` и отсортировать ее по убыванию. По включенным флажкам вывести на форму `MAX` и `MIN` элементы  $K$ -го столбца исходной матрицы.

4. Ввести в компонент **StringGrid1** исходные числа, используя функцию `InputBox()`.

В компонент **StringGrid2** в соответствующие клетки занести размерности каждого числа. Вывести в последнюю строку количество нулей по столбцам в компоненте `StringGrid2` и отсортировать эту строку по возрастанию. Ввод, формирование и очистка матриц выполняется в соответствии с пунктами меню. По включенным флажкам вывести на форму `MAX` и `MIN` элементы  $K$ -ой строки исходной матрицы.

5. Ввести в компонент **StringGrid1** исходные числа, используя функцию `InputBox()`.

В компонент **StringGrid2** в соответствующие клетки занести отрицательные числа, если таких нет в числе, то — ноль. Ввод, формирование и очистка матриц выполняется в соответствии с пунктами меню. По включенным радиокнопкам вывести на форму `MAX` и `MIN` элементы  $K$ -ой строки исходной матрицы.

6. Ввести в компонент **StringGrid1** исходные числа, используя функцию `InputBox()`.

Сформировать в компонент **StringGrid2** новую таблицу, заменив числа исходной таблицы, равные  $2^{i+j}$  «1», а остальные числа — нулем. Вывести в последнюю строку количество нулей по столбцам в компоненте `StringGrid2` и отсортировать эту строку по убыванию. Ввод, формирование и очистка матриц выполняется в соответствии с пунктами меню. По включенным радиокнопкам вывести на форму `MAX` и `MIN` элементы  $K$ -ой строки исходной матрицы.

7. Ввести в компоненты **StringGrid1** и **StringGrid2** исходные числа, используя функцию `InputBox()`. В компонент **StringGrid3** в соответствующие клетки занести сумму элементов двух таблиц. Вывести в последнюю строку `StringGrid3` количество простых чисел по столбцам и отсортировать эту строку по возрастанию. Ввод, формирование и очистка матриц выполняется в соответствии с пунктами меню. По включенным флажкам вывести на форму `MAX` и `MIN` элементы  $K$ -ой строки исходных матриц.

8. Ввести в компоненты **StringGrid1** и **StringGrid2** исходные числа, используя функцию `InputBox()`. В компонент **StringGrid3** в соответствующие клетки занести сумму первых цифр элементов двух таблиц. Вывести в последний столбец `StringGrid3` количество четных чисел по строкам и отсортировать эту строку по возрастанию. Ввод, формирование и очистка матриц выполняется в соответствии с пунктами меню. По включенным радиокнопкам выводить на форму `MAX` и `MIN` элементы  $K$ -ой строки исходных матриц.

9. Ввести в компоненты **StringGrid1** и **StringGrid2** исходные числа, используя функцию `InputBox()`. В компонент **StringGrid3** в соответствующие клетки занести произведение четных элементов двух таблиц, в противном случае заносить их сумму. Вывести в последний столбец `StringGrid3` количество простых чисел по строкам и отсортировать эту строку по возрастанию. Ввод, формирование и очистка матриц выполняется в соответствии с пунктами меню. По включенным радиокнопкам выводить на форму `MAX` и `MIN` элементы  $K$ -ой строки исходных матриц.



количество четных чисел по строкам и отсортировать эту строку по возрастанию. Ввод, формирование и очистка матриц выполняется в соответствии с пунктами меню. По включенным радиокнопкам выводить на форму MAX и MIN элементы K-ой строки исходных матриц.

24. Ввести в компоненты **StringGrid1** и **StringGrid2** исходные числа, используя функцию `InputBox()`. В компонент **StringGrid3** в соответствующие клетки занести разность четных элементов двух таблиц, в противном случае заносить их сумму. Вывести в последний столбец **StringGrid3** количество простых чисел по строкам и отсортировать эту строку по возрастанию. Ввод, формирование и очистка матриц выполняется в соответствии с пунктами меню. По включенным радиокнопкам выводить на форму MAX и MIN элементы K-ой строки исходных матриц.

25. Ввести в компоненты **StringGrid1** и **StringGrid2** исходные числа, используя функцию `InputBox()`. В компонент **StringGrid3** в соответствующие клетки занести сумму четных элементов двух таблиц, в противном случае заносить их сумму. Вывести в последний столбец **StringGrid3** количество простых чисел по строкам и отсортировать эту строку по возрастанию. Ввод, формирование и очистка матриц выполняется в соответствии с пунктами меню. По включенным радиокнопкам выводить на форму MAX и MIN элементы K-ой строки исходных матриц.

#### Практическая работа № 9

Тема: Формирование диаграмм и графиков .

##### Методические указания:

1. Вся необходимая информация отображается на форме: исходные данные вводятся в компоненты ввода и редактирования данных, результаты решения выводятся в соответствующие компоненты для вывода данных. Элементы управления процессом вычисления выбирает студент по своему усмотрению.
  2. Организовать программный контроль правильности ввода информации.
1. В компонент `StringGrid` на форме вводится массив целых чисел. Определить процент чисел с первой цифрой =1, =2 и =5. Построить круговую диаграмму соответственно этим процентам.
  2. В компонент `StringGrid` на форме вводится целочисленный массив. Найти процент отрицательных, положительных и равных нулю чисел. Создать круговую диаграмму, отражающую соотношение указанных чисел в массиве.
  3. В компонент `StringGrid` на форме вводится массив целых чисел. Определить количество чисел с первой цифрой =4, =7 и =1. Построить круговую диаграмму соответственно этим количествам.
  4. В компонент `StringGrid` на форме вводится массив целых чисел. Определить количество чисел с первой цифрой кратной 2 и кратной 3. Построить круговую диаграмму соответственно этим количествам.
  5. В компонент `ListBox` на форме вводится целочисленный массив координат точек –вершин многоугольника  $X_i$  и  $Y_i$ . Вычислить длины его сторон и создать прямоугольную диаграмму – сколько из них больше среднего арифметического длин отрезков, равных ему и меньше среднего.
  6. В компонент `StringGrid` на форме вводится массив целых чисел. Найти максимальный элемент массива. Определить количество и процент чисел меньших MAX и равных ему. Построить круговые диаграммы соответственно этим количествам и процентам.
  7. В компонент `StringGrid` на форме вводится целочисленный массив. Найти процент нечетных отрицательных, положительных и равных нулю чисел. Создать круговую диаграмму, отражающую ситуацию.
  8. В компонент `StringGrid` на форме вводится массив целых чисел. Определить процент чисел с последней цифрой =1, =2 и =5. Построить круговую диаграмму соответственно этим процентам.
  9. В компонент `StringGrid` на форме вводится массив целых чисел. Определить процент чисел с четной и нечетной суммой цифрой. Построить круговую диаграмму соответственно этим процентам.
  10. В компонент `StringGrid` на форме вводится массив целых чисел. Определить процент чисел, записанных только цифрами 1, 2 или 3. Построить круговую диаграмму соответственно этим процентам.
  11. В компонент `StringGrid` на форме вводится массив целых чисел. Определить процент чисел с четным и нечетным произведением цифр. Построить круговую диаграмму соответственно этим процентам.
  12. В компонент `StringGrid` на форме вводится массив целых чисел. Определить процент чисел, в записи которых используется подряд две цифры 1, 3 или 5. Построить круговую диаграмму соответственно этим процентам.
  13. В компонент `StringGrid` на форме вводится массив целых чисел. Определить процент положительных четных, отрицательных нечетных чисел и равных 0. Построить круговую диаграмму соответственно этим процентам.
  14. В компонент `StringGrid` на форме вводится массив целых чисел. Определить количество чисел до первого максимального, после последнего максимального и равных максимальному. Построить круговую диаграмму соответственно этим количествам.
  15. В компонент `StringGrid` на форме вводится массив целых чисел. Определить количество четных чисел до первого максимального, после последнего максимального и чисел массива равных максимальному. Построить круговую диаграмму соответственно этим количествам.
  16. В компонент `StringGrid` на форме вводится массив целых чисел. Определить количество чисел до первого минимального, после последнего минимального и равных минимальному. Построить круговую диаграмму соответственно этим количествам.
  17. В компонент `StringGrid` на форме вводится массив целых чисел. Определить процент чисел записанных только цифрами 1, 2 и 5. Построить круговую диаграмму соответственно этим процентам.
  18. компонент `StringGrid` на форме вводится массив целых чисел. Определить процент чисел записанных только четными цифрами и только нечетными. Построить круговую диаграмму соответственно этим процентам.
  19. В компонент `StringGrid` на форме вводится массив целых чисел. Определить процент чисел записанных только цифрами 3, только цифрами 5, только цифрами 7. Построить круговую диаграмму соответственно этим процентам.
  20. В компонент `StringGrid` на форме вводится массив целых чисел. Определить процент чисел записанных только цифрами больших 5 и только цифрами меньших 5. Построить круговую диаграмму соответственно этим процентам.
  21. В компонент `StringGrid` на форме вводится массив целых чисел. Определить процент чисел, цифры которых упорядочены по возрастанию, по убыванию и не упорядочены. Построить круговую диаграмму соответственно этим процентам.

22. В компонент StringGrid на форме вводится массив целых чисел. Определить процент чисел, четные цифры которых упорядочены по возрастанию, по убыванию и не упорядочены. Построить круговую диаграмму соответственно этим процентам
23. В компонент StringGrid на форме вводится массив целых чисел. Определить процент чисел, в которых чередуются цифры 3 и 5, и числа с чередующимися цифрами 6 и 8. Построить круговую диаграмму соответственно

#### Практические работы №10-11

Тема: Функции работы с файлами в Lazarus. Запись и считывание данных из таблиц TStringGrid. Разработка приложения для анализа динамики изменений состояния таблиц и визуализации этих изменений.

Методические рекомендации по выполнению практической работы.

Ход выполнения практической работы.

##### Постановка задачи:

Задана матрица  $A (m \times n)$ , задан также диапазон значений (целые числа). Проверить «правильность» матрицы по следующим критериям:

- если все элементы матрицы входят в заданный диапазон, то матрица 100% правильна. Этому соответствует определенный рисунок (изображение);
- если элементы матрицы входят в диапазон только на 75%, то выводится сообщение и другой рисунок, характеризующий небольшое ухудшение ситуации;
- если элементы матрицы входят в диапазон только на 50%, то выводится сообщение и следующий рисунок, характеризующий еще более ухудшение ситуации;
- если элементы матрицы не входят в диапазон, то выводится сообщение и другой рисунок, характеризующий наибольшее ухудшение ситуации.

Таким образом, в зависимости от состояния элементов матрицы на экран выводится сообщение и соответствующий рисунок (изображение).

В соответствии с заданием разработать проект в среде ООП с использованием компонентов управления расчетом – кнопок, меню, переключателей и компонентом отображения таблицы – TStringGrid.

Рисунки (изображения) студент выбирает самостоятельно.

#### Практическая работа №12

Тема: Функции кодирования и раскодирования текста. Хранение и шифрование паролей.

1. Дан текст. Зашифровать его, заменив каждую букву следующей за ней по алфавиту.
2. Дан текст и ввести число  $N$ . Зашифровать его, заменив каждую букву буквой, следующей за ней через  $N$  букв.
4. Дан текст. Зашифровать его, заменив каждую букву порядковым номером, который она имеет в алфавите, отделяя их точками. Знаки препинания не шифровать, а удалить.
5. Дан текст. Зашифровать его в виде последовательности чисел, разделенных точкой, каждый символ заменяется кодом символа.
6. Дан текст. Зашифровать его в виде последовательности чисел, разделенных точкой с запятой, каждый символ заменяется кодом следующего символа.
7. Дан текст. Шифруем текст с помощью самого текста: каждый символ заменяется символом, стоящим через  $N$  символов.
8. Дан текст. Каждое слово зашифровать след. образом:
  - \*Если слово состоит из 1 буквы, то его так и оставить;
  - \*Если в слове больше букв, то первая меняется местами с последней.
9. Дан текст, каждое слово зашифровать след. образом: к первой букве слова прибавлять последующие, то справа, то слева.
10. Дана последовательность, каждое слово зашифровать след. образом:
  - \*Если слово состоит из 1 буквы, то символ шифруется символом, у которого код на 1 меньше;
  - \*Если в слове больше 2-х букв, то 2 рядом стоящие согласные меняем местами;
  - \*Если в слове 2 буквы, то они меняются местами.
11. Дана последовательность, каждое слово зашифровать след. образом:
  - \*Если количество символов в слове нечетное, то первая буква меняется местами с последней, вторая с последней и т.д. Символ, стоящий посередине, заменяем символом, у которого код на 1 больше.
  - \*Если количество символов четное, то меняются левая и правая части.
12. Дана последовательность, каждое слово зашифровать след. образом:
  - \*Если количество символов в слове четное, то меняются местами 1-я буква со 2-й, 3-я с 4-й и т.д.
  - \*Если нечетное, то меняются местами 2-я с 3-й, 4-я с 5-й и т.д., а 1-я остается на своем месте.
13. Дана последовательность, каждое слово зашифровать след. образом:
  - \*Если слово состоит из 1 буквы, то символ шифруется символом, у которого код на 8 единиц меньше;
  - \*Если слово состоит из 2-х или 3-х букв, то каждая буква заменяется символом, у которого код на 4 единицы больше;
  - \*Если в слове более 3-х букв, то порядок меняется на противоположный, после чего 2-й и 3-й символ меняются местами, а также последний и предпоследний символы.
14. Дана последовательность, каждое слово зашифровать след. образом:
  - \*Если слово состоит из 1 буквы, то так и оставить;
  - \*Если в слове 2 буквы, то слово заменить кодами букв;



\*Если в слове больше двух букв, то поменять первую с последней, вторую с предпоследней и т.д.

15. Дана последовательность, каждое слово зашифровать след. образом:

\*Сжать текст, удалив разделители;

\*Все "О" и "о" заменить на "%";

\*Все "А" и "а" заменить на "?";

\*Из полученного текста получить перевертыши.

16. Дана последовательность, каждое слово зашифровать след. образом:

\*Если слово состоит из 1 буквы, то так и оставить.

\*Если в слове 2 буквы, то слово заменяется кодами этих букв. \*Если в слове больше 2-х букв, то поменять 1-ю с последней, 2-ю с предпоследней и т.д.

17. Дана последовательность, каждое слово зашифровать след. образом:

\*Если слово состоит из 1 буквы, то символ шифруется символом, у которого код на 1 меньше.

\*Если слово состоит из 2 букв, то первая буква заменяется символом, у которого код на 1 меньше, а вторая - у которого код на 1 больше.

\*Если в слове больше 2-х букв, то каждый слог нового слова есть сочетание первой и последней, 2-й и предпоследней буквы и т.д.

18. Дана последовательность, каждое слово зашифровать след. образом:

\*Если слово состоит из 1 буквы, то оно остается без изменений.

\*Если в слове 2 буквы, то они меняются местами и шифруются сл.обр.:

\*первая буква заменяется буквой, у которой код на 2 меньше, а 2-я остается без изменения.

\*Если в слове более 2-х букв, то 2-я и предпоследняя буквы меняются местами, первая половина слова шифруется кодом на 1 больше, вторая половина - на 1 меньше.

19. Дана последовательность, поместить последовательность в квадратную матрицу. Зашифрованный текст: матрица, прочитанная по спирали.

20. Дана последовательность, поместить последовательность в квадратную матрицу. Зашифрованный текст: Строки меняются со столбцами. Матрица читается по строкам.

21. Дана последовательность, поместить последовательность в квадратную матрицу.

Зашифрованный текст: Элементы выше главной диагонали меняются с элементами ниже главной диагонали. Матрица читается по строкам.

22. Дана последовательность, поместить последовательность в квадратную матрицу.

Зашифрованный текст: Элементы выше побочной диагонали меняются с элементами ниже главной диагонали. Читаем нечетные строки с начала, четные с конца.

23. Дана последовательность, поместить последовательность в квадратную матрицу.

Зашифрованный текст: Строки переворачиваются и меняются местами со столбцами. Матрица читается по строкам.

24. Дана последовательность, поместить последовательность в квадратную матрицу. Зашифрованный текст: Столбцы переворачиваются и меняются местами со строками.

25. Дана последовательность, поместить последовательность в квадратную матрицу. Зашифрованный текст: Строки меняются местами: 1-я со 2-й, 3-я с 4-й,...

#### Практическая работа №13

Тема: Парольные формы для приложений в Lazarus. Программы контроля дифференцированного входа

Методические указания: студенты разрабатывают парольную форму для разработанного приложения. Необходимо обеспечить проверку дифференцированного входа, т.е. в зависимости от логина и пароля пользователь получает определенные раннее права на выполнение некоторых функций.

#### Практическая работа №14

Тема: Разработка анимации в среде Lazarus. Компонент Timer.

Методические указания: студенты самостоятельно реализуют анимационную программу на выбранную тему.

#### Практические работы №15-16

Тема: Разработка приложения по индивидуальным заданиям.

Методические указания: студенты самостоятельно реализуют прикладную программу на выбранную тему.

#### Практические работы №17-18

Тема: Режимы отладки в среде ООП Lazarus. Отладка разработанного модуля Методики и этапы тестирования. Сценарии тестирования готовых модулей. Подготовка протоколов тестирования ля.

#### Практические работы №19-20

Тема: Программы автоматизации создания документации по программным средствам. Разработка документации по выполненным практическим работам.