

Документ подписан простой электронной подписью  
Информация о владельце:  
ФИО: Локтионова Оксана Геннадьевна  
Должность: проректор по учебной работе  
Дата подписания: 15.11.2022 09:29:47  
Уникальный программный ключ:  
0b817ca911e6668a912c4e7a1b0ca40c

## МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«Юго-Западный государственный университет»  
(ЮЗГУ)

Кафедра охраны труда и окружающей среды



## ОСНОВЫ АЛГОРИТМИЗАЦИИ

Методические указания к проведению  
лабораторных работ по дисциплине  
«Информационные технологии в сфере безопасности»  
для студентов направления подготовки 20.04.01  
«Техносферная безопасность»

Курск 2021

УДК 699.85

Составители: И.О. Кирильчук, А.В. Иорданова

**Рецензент**

Кандидат технических наук, доцент *Г.П. Тимофеев.*

**Основы алгоритмизации:** методические указания к проведению лабораторных работ по дисциплине «Информационные технологии в сфере безопасности» / Юго-Зап. гос. ун-т; сост.: И.О. Кирильчук, А.В. Иорданова. Курск, 2021. 7 с.

Описывается методика построения алгоритмов программ и записи их с помощью блок-схемы.

Методические указания предназначены для студентов направления подготовки 20.04.01 Техносферная безопасность.

Текст печатается в авторской редакции

Подписано в печать                      Формат 60x84 1/16.

Усл. печ. л. 0,4. Уч.-изд.л. 0,36. Тираж 30 экз. Заказ                      . Бесплатно.

Юго-Западный государственный университет.

305040, г. Курск, ул. 50 лет Октября, 94.

**Цель работы:**

– изучить методику построения алгоритмов программ и записи их с помощью блок-схемы.

**Общие положения**

**Алгоритм** – некоторая конечная последовательность правил (предписаний), определяющая процесс преобразования исходных и промежуточных данных в результат решения задачи.

Разрабатываемый алгоритм должен обладать следующими свойствами:

- *массовостью*, позволяющей решать не одну задачу, а целый класс задач;
- *детерминированностью*, однозначно определяющей выполняемые действия (промежуточные и окончательные результаты разных пользователей должны быть одинаковыми при одинаковых исходных данных);
- *результативностью*, позволяющей получить результат после конечного числа шагов.

По используемой структуре управления вычислительным процессом алгоритмы классифицируют следующим образом:

- линейной структуры;
- разветвляющейся структуры;
- циклической структуры;
- смешанной (комбинированной) структуры.

Алгоритм линейной структуры – алгоритм, в котором все действия выполняются последовательно друг за другом.

Алгоритм разветвляющейся структуры – алгоритм, в котором в зависимости от выполнения некоторого логического условия вычислительный процесс должен идти по одной или другой ветви.

Алгоритм циклической структуры – алгоритм, содержащий многократно выполняемые участки вычислительного процесса, называемые циклами. Если алгоритм содержит цикл, внутри которого размещен один или несколько других циклов, то такой алгоритм называется алгоритмом со структурой вложенных циклов.

Существует много способов записи алгоритмов, отличающихся друг от друга наглядностью, компактностью, степенью формализа-

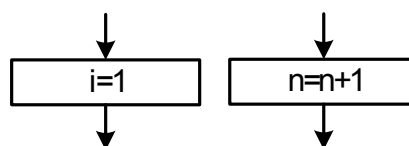
ции и другими показателями. Наибольшее распространение получил графический способ (блок-схем).

Схема алгоритма представляет собой последовательность блоков, предписывающих выполнение определенных действий, и связи между ними. Она может выполняться с разной степенью детализации.

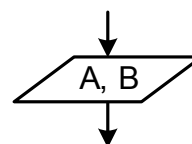
1 Начало и конец алгоритма (программы)



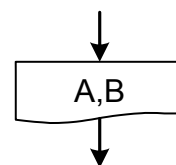
2 Блок присваивания



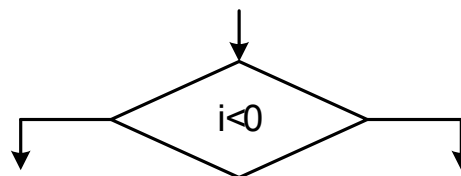
3 Ввод данных с клавиатуры



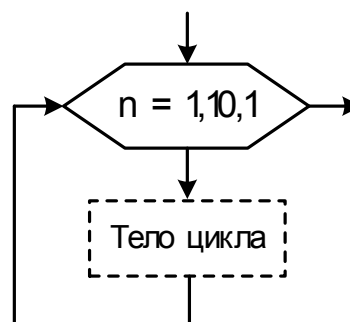
4 Вывод информации на экран (на печать)



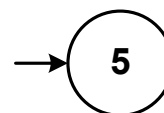
5 Блок ветвления (проверки условия)



6 Блок цикла с параметром



7 Нумерация блоков, значок перехода на блок с указанным номером



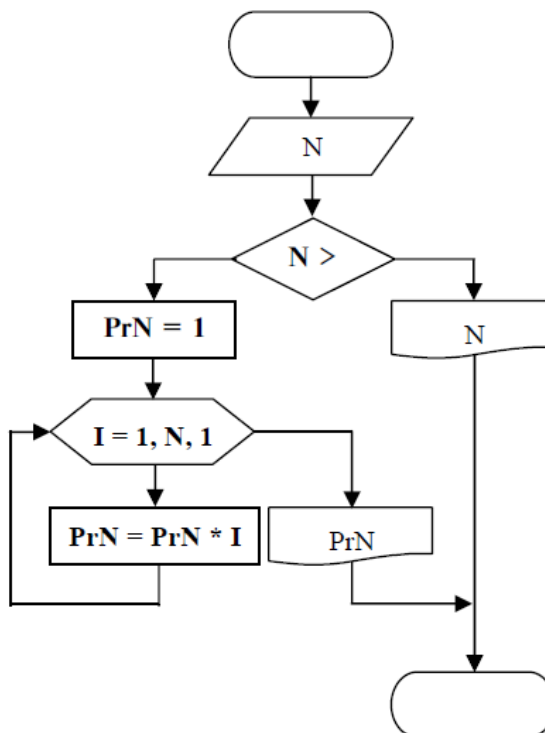
## Пример задания

Напишите блок-схему алгоритма программы, которая запрашивает с клавиатуры целое число  $N$  и если это число больше 10, то вычисляет и выводит на экран произведение всех целых чисел от 1 до  $N$ , иначе выводит на экран значение  $N$ .

Алгоритм решения задачи:

1. Ввод исходных данных ( $N$ ).
2. Проверка условия  $N > 10$
3. В случае выполнения условия вычисляется произведение всех целых чисел от 1 до  $N$ .
4. Вывод на экран полученного произведения и завершение выполнения программы.
5. В противном случае вывод на экран значения  $N$  и завершение выполнения программы.

Блок-схема:



## Задания для самостоятельной работы

Напишите блок-схему алгоритма программы, которая:

1) классифицирует компьютерную сеть. Программа запрашивает у пользователя число компьютеров в сети и в зависимости от введенного количества выводит класс сети (если число ЭВМ меньше 256 – то это сеть класса С, от 256 до 65535 – сеть класса В, свыше 65535 – сеть класса А);

2) запрашивает у пользователя номер одного из весенних месяцев, и выводит количество дней в этом месяце. Программа должна проверять, является ли введенный месяц весенним;

3) выводит на экран приглашение: «Который час?», вводит с клавиатуры число  $X$ , имеющее смысл времени суток, и печатает слова «Доброе утро», «Добрый день», «Добрый вечер» или «Доброй но-

чи» в зависимости от введенного времени. Программа должна реагировать на ввод неправильного времени: меньше 0 или больше 24;

4) запрашивает у пользователя размер хищений (р.), определяет и выводит на экран масштаб в соответствии с принятой классификацией (например, если размер хищений меньше 1000 р. – «мелкий», от 1000 до 10 000 – «крупный», свыше 100 000 – «особо крупный»);

5) запрашивает произвольное число  $N$ , вычисляет сумму всех целых чисел от 1 до  $N$ . Если полученная сумма больше 10, то выводит на экран ее значение, иначе выводит на экран сообщение «Сумма меньше 10»;

6) выводит на экран приглашение: «Введите месяц», вводит с клавиатуры число  $X$ , имеющее смысл месяца, и печатает слова «Зима», «Весна», «Лето» или «Осень» в зависимости от введенного месяца. Программа должна реагировать на ввод неправильного месяца: меньше 1 или больше 12;

7) запрашивает с клавиатуры два произвольных числа  $X$  и  $Y$ . Если  $X$  больше  $Y$ , то вычисляет и выводит на экран сумму всех целых чисел от 1 до  $X$ , иначе вычисляет и выводит на экран произведение всех целых чисел от  $X$  до  $Y$ ;

8) запрашивает у пользователя номер одного из летних месяцев, и выводит количество дней в этом месяце. Программа должна проверять, является ли введенный месяц летним;

9) вводит с клавиатуры 10 чисел и выводит на экран максимальное и минимальное из них;

10) запрашивает с клавиатуры два произвольных числа  $X$  и  $Y$ , вычисляет их разность. Если разность больше 10, то вычисляет и выводит на экран сумму всех целых чисел от  $Y$  до  $X$ , иначе выводит на экран значение разности;

11) выводит на экран приглашение: «Введите день недели», вводит с клавиатуры число  $X$ , имеющее смысл дня недели, и печатает слова «Рабочий день», «Короткий день», «Выходной» в зависимости от введенного дня. Программа должна реагировать на ввод неправильного дня недели: меньше 1 или больше 7;

12) запрашивает произвольное число  $N$ , вычисляет произведение всех целых чисел от 1 до  $N$ . Если полученное произведение больше 50, то выводит на экран ее значение, иначе выводит на экран сообщение «Произведение меньше 50»;

13) запрашивает у пользователя номер одного из осенних месяцев, и выводит количество дней в этом месяце. Программа должна проверять, является ли введенный месяц осенним;

14) запрашивает с клавиатуры два целых числа, их сумму и произведение и выводит на экран сообщение о правильности сделанных пользователем вычислений;

15) запрашивает с клавиатуры два произвольных числа  $X$  и  $Y$ . Если  $X$  больше  $Y$ , то вычисляет и выводит на экран произведение всех целых чисел от 1 до  $X$ , иначе вычисляет и выводит на экран сумму всех целых чисел от 1 до  $Y$ ;

16) запрашивает с клавиатуры два произвольных числа  $X$  и  $Y$ , вычисляет их разность. Если разность больше 10, то вычисляет и выводит на экран произведение всех целых чисел от  $Y$  до  $X$ , иначе выводит на экран значение разности;

17) запрашивает с клавиатуры два целых числа, их разность и частное (результат деления) и выводит на экран сообщение о правильности сделанных пользователем вычислений;

18) запрашивает у пользователя номер одного из зимних месяцев, и выводит количество дней в этом месяце. Программа должна проверять, является ли введенный месяц зимним;

19) запрашивает с клавиатуры число  $X$ . Если  $X$  меньше 10, то вычисляет и выводит на экран квадрат числа  $X$ , а если больше или равно, то вводилось новое число  $Y$ , а затем вычисляет и выводит на экран значение суммы  $X$  и  $Y$ ;

20) запрашивает с клавиатуры два произвольных числа  $X$  и  $Y$ . Если разность  $X$  и  $Y$  больше 0, то вычисляет и выводит на экран сумму всех целых чисел от  $Y$  до  $X$ , иначе вычисляет и выводит на экран разность всех целых чисел от  $Y$  до  $X$ .

### **Список рекомендуемой литературы:**

1. Голицына, О.Л. Основы алгоритмизации и программирования [Текст] / учеб. пособие. – 3-е изд., испр. и доп. / О.Л. Голицына, И.И. Попов. – М.: ФОРУМ, 2008. – 432 с.

2. Давыдов, В.Г. Программирование и основы алгоритмизации [Текст] / В.Г. Давыдов. – М.: Высшая школа, 2003. – 447 с.

## **МИНОБРНАУКИ РОССИИ**

Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«Юго-Западный государственный университет»  
(ЮЗГУ)

Кафедра охраны труда и окружающей среды



## **РАЗРАБОТКА АЛГОРИТМОВ И ПРОГРАММ В ИНТЕГРИРОВАННОЙ СРЕДЕ LAZARUS**

Методические указания к проведению  
лабораторных работ по дисциплине  
«Информационные технологии в сфере безопасности»  
для студентов направления подготовки 20.04.01  
«Техносферная безопасность»

Курск 2021



УДК 699.85

Составители: И.О. Кирильчук, А.В. Иорданова

**Рецензент**

Кандидат технических наук, доцент *Г.П. Тимофеев.*

**Разработка алгоритмов и программ в интегрированной среде Lazarus:** методические указания к проведению лабораторных работ по дисциплине «Информационные технологии в сфере безопасности» / Юго-Зап. гос. ун-т; сост.: И.О. Кирильчук, А.В.Иорданова. Курск, 2021. 25 с.

Описываются основные методы и приемы работы в интегрированной среде разработчика Lazarus, порядок разработки алгоритмов и программ линейной и разветвленной структуры, методика работы с массивами данных.

Методические указания предназначены для студентов для студентов направления подготовки 20.04.01 «Техносферная безопасность»

Текст печатается в авторской редакции

Подписано в печать                      Формат 60x84 1/16.Усл. печ. л.                      .

Уч.-изд.л.                      . Тираж 30 экз. Заказ                      . Бесплатно.

Юго-Западный государственный университет.

305040, г. Курск, ул. 50 лет Октября, 94.

**Цель работы:**

- познакомиться с основными элементами среды Lazarus, принципами её функционирования, а также с технологией создания приложений;
- освоить процесс построения структур данных, алгоритмов и программ линейной и разветвленной структуры;
- освоить методику работы с массивами данных.

**I. Основные положения****1. Элементы среды разработки Lazarus****1.1. Вход в Lazarus**

После запуска оболочки Windows на рабочем столе или в подменю «Программы» меню «Пуск» необходимо найти пиктограмму «Lazarus». Пиктограмма Lazarus – это кнопка для старта *интегрированной среды разработчика* (Integrated Development Environment – IDE).

IDE включает ряд окон и инструментальных средств, при помощи которых можно создавать прототип, проектировать, кодировать, транслировать, выполнять и отлаживать программы для Windows. Причина, по которой среда называется "интегрированной", заключается в том, что из неё не надо выходить в течение всего цикла реализации. Все эти задачи могут выполняться прямо из IDE.

Чтобы запустить среду разработки Lazarus, надо дважды щёлкнуть мышью на пиктограмме Lazarus. После запуска Lazarus на дисплее появится окно, приблизительно соответствующее виду, представленному на рисунке 2.

**1.2. Основные элементы интерфейса Lazarus**

Интегрированная среда состоит из следующих элементов:

- основное окно (Main);
- окно Инспектора объекта (Object Inspector);
- окно Дизайнера форм (Form Designer);
- окно Редактора кода (Code Editor);
- Менеджер проекта (Project Manager);
- окно Списка наблюдений (Watch List);
- окно Стека вызовов (Call Stack);
- Список контрольных точек (Breakpoint List);

– Просмотрщик объектов (Object Browser) и др.

Не все эти элементы видны сразу после запуска Lazarus. Наиболее часто используемые элементы: Инспектор объекта, Дизайнер форм, Редактор кода, Менеджер проекта.

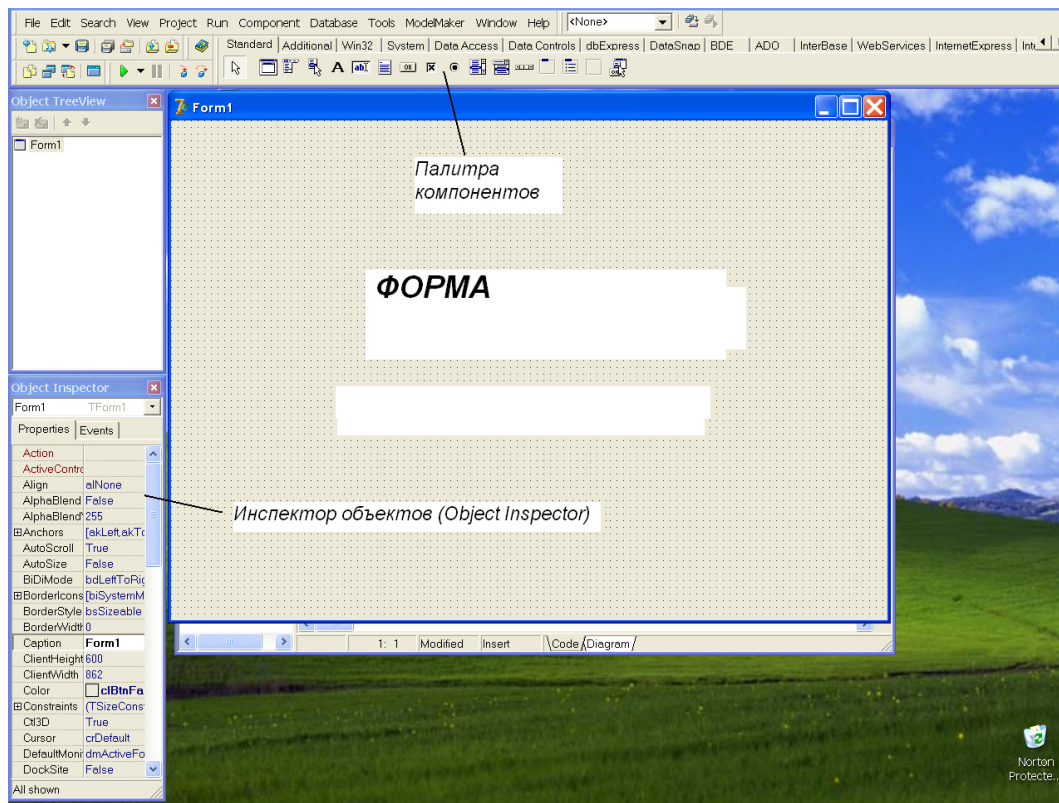


Рисунок 2 – Интегрированная среда разработчика Lazarus (IDE)

### 1.3. Основное окно

Основное окно помещено наверху экрана. Минимизация основного окна уменьшает все другие зависимые окна среды Lazarus. На рисунке 3 изображено основное окно Lazarus в конфигурации по умолчанию.

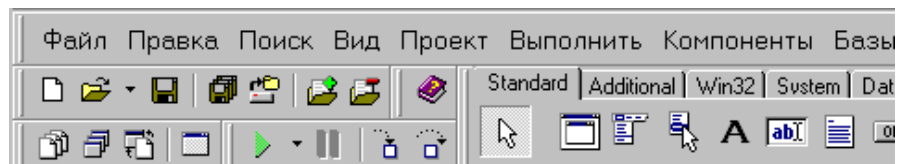


Рисунок 3 – Основное окно Lazarus

Основное окно содержит строку меню, Линейку инструментов и Палитру компонентов. Слева, под меню Lazarus, находится Линейка инструментов. Она содержит кнопки быстрого доступа (быстрые кнопки). Чтобы выполнить какую-либо команду, надо нажать кнопку быстрого доступа мышью (например, для того, чтобы дать указание

Lazarus открыть файл или скопировать некоторый выбранный текст в буфер обмена).

*Кнопка быстрого доступа* выглядит как маленькая кнопка с пиктограммой. Это удобное сокращение для доступа к часто используемым командам. Все действия, которые могут выполняться при помощи кнопки быстрого доступа, также могут быть выполнены через меню программы. Кнопки быстрого доступа просто дают оперативный доступ к этим действиям.

Серая область, на которой располагаются все кнопки быстрого доступа, называется *Линейкой инструментов*. Линейку инструментов можно настраивать и изменять её размеры. Для того, чтобы открыть редактор Линейки инструментов, надо щёлкнуть правой кнопкой мыши в области Линейки инструментов и выбрать из выпавшего меню опцию «Настроить» (Configure). Это позволит настраивать содержимое Линейки инструментов, перемещая в неё и из неё кнопки быстрого доступа.

Справа от Линейки инструментов находится серое поле с пиктограммами, которое называется *Палитрой компонентов*. При визуальном формировании программы надо выбрать пиктограмму, которая представляет определённый компонент в палитре, и поместить её в форму. Форма отображается как отдельное окно ниже Палитры компонентов. Палитра состоит из нескольких страниц компонентов, помеченных как Standard (стандартные), Additional (дополнительные), System (системные), Dialogs (диалоги) и т.д. Можно «пролистывать» страницы, нажимая мышью на одну из меток вверху строки пиктограмм компонентов.

#### **1.4. Окно Дизайнера форм**

Дизайнер форм – это пустое серое окно, озаглавленное Form1, покрытое регулярно размещёнными точками. После первого запуска Lazarus ниже Палитры компонентов появляется *Дизайнер форм* (Form Designer). Начальный внешний вид Дизайнера форм представлен на рисунке 4. Заголовок окна дизайнера форм (Form1) является стандартным заголовком новой формы, который Lazarus автоматически создаёт всякий раз, когда запрошен новый проект.

*Форма* – это термин, используемый для описания окна программы – области дисплея, где могут быть размещены различные

компоненты интерфейса пользователя, например: кнопки, метки, поля редактирования и др. Форма является каркасом, на котором размещаются компоненты, и обеспечивает их взаимодействие. Фактически каждая программа Lazarus состоит, по крайней мере, из одной формы.

Дизайнер форм – это место, где осуществляется визуальное программирование. Программа создаётся посредством размещения на форме компонентов, выбранных из палитры. Позднее к компонентам можно присоединить некоторый код (текст программы), чтобы позволить этим компонентам взаимодействовать друг с другом. Визуальное программирование заметно ускоряет весь процесс разработки. Термин «визуальное программирование» используется как раз для того, чтобы описать такой способ разработки программ.

Размеры окна Дизайнера форм можно изменять подобно любому другому стандартному окну Windows, перемещая рамку мышью. Установленные во время разработки размеры отразятся в конечной программе после её запуска. Это означает, что установленные во время разработки размеры формы определяют размер формы во время выполнения программы.

Даже простейшая пустая форма имеет ряд свойств, которые могут изменяться. Устанавливая значения этих свойств, можно настраивать вид формы и её поведение.

*Свойство* – атрибут данных компонента. Можно думать о свойстве как о ячейке, содержащей единственное значение. Например, свойство Width (ширина) каждой формы содержит горизонтальный размер формы в пикселах.

Можно устанавливать текущее значение свойства, если оно устраивает пользователя, или изменять его. Менять значение свойства можно как во время разработки новой формы, так и во время выполнения программы. Установка значений свойств во время разработки даёт им начальные значения.

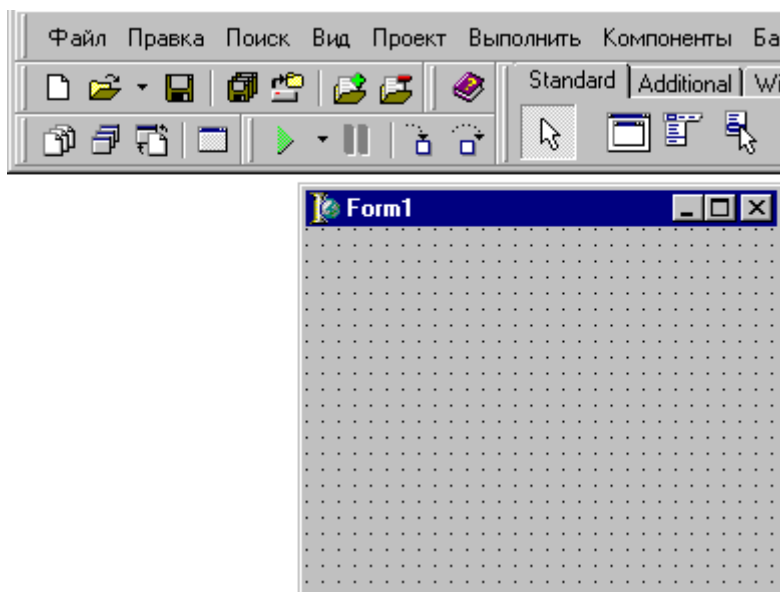


Рисунок 4 – Незаполненное окно Дизайнера форм

### 1.5. Окно Инспектора объекта

*Инспектор объекта* (Object Inspector) отображает своё окно по левому краю экрана. На рисунке 5 показано типовое окно Инспектора объекта.

Внутри Инспектора объекта находятся таблицы, расположенные на двух страницах. Между страницами можно переключаться, нажимая на одну из меток вблизи крайней верхней части окна. Эти метки названы «Свойства» (Properties) и «События» (Events).

Инспектор объекта тесно связан с Дизайнером форм. Управляющие элементы выбираются из окна Дизайнера форм, а их свойства контролируются Инспектором объекта.

*Управляющий элемент* – это визуальный компонент, который может быть помещён в форму, и является, как правило, видимым во время выполнения. Стандартные управляющие элементы Windows включают метки (статический текст), окошки редактирования, кнопки, контрольные индикаторы, радиокнопки, окошки списка, комбинированные окошки и линейки прокрутки.

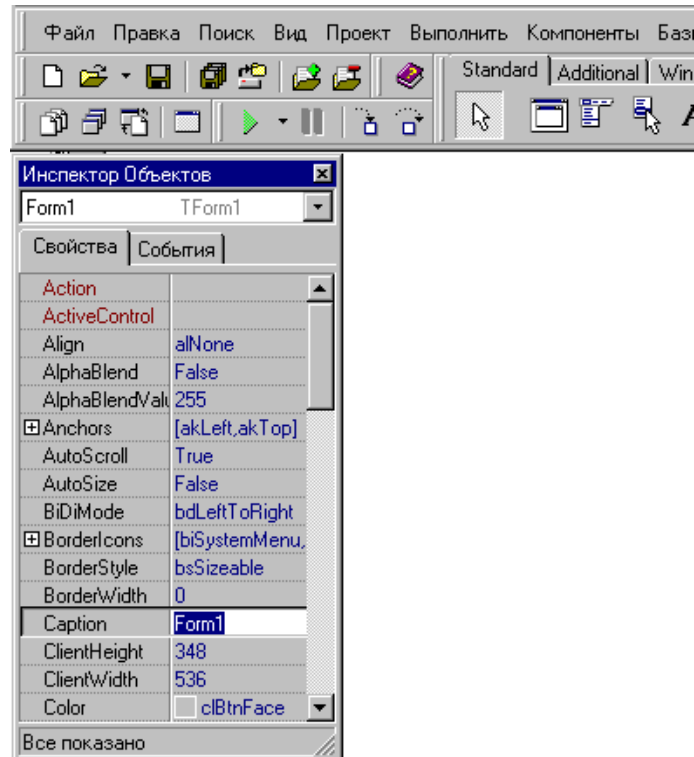


Рисунок 5 – Окно Инспектора объекта

Страница «События» показывает список событий, которые могли бы быть связаны с некоторым расположенным на форме объектом.

Страница свойств Инспектора объекта содержит таблицу, состоящую из двух колонок с названием свойства в левой колонке и его значением в правой. Некоторые из этих значений первоначально могут оказаться пустыми. Каждый ряд этой таблицы представляет одно свойство компоненты, выбранной в данное время в Дизайнере форм. Если форма не заполнена, то единственные видимые в Инспекторе объекта свойства – это непосредственно свойства формы.

Можно получить детальную информацию по любому свойству, перечисленному в Инспекторе объекта, выбирая это свойство мышью и затем нажимая F1. Lazarus выведет страницу интерактивной справки конкретно по указанному свойству.

Наверху окна Инспектора объекта имеется выпадающий список компонентов. Чтобы показать его выпадающую часть, надо нажать стрелку справа от списка. После первоначального запуска Lazarus, при незаполненной форме, в список внесена только сама форма (Form1). Никакие другие компоненты в форме пока ещё не присутствуют.

## 1.6. Окно Редактора кода

Наряду с визуальным проектированием интерфейса пользователя программы (формы) обычно разрабатывается код, лежащий в основе программы. Язык программирования, на котором основана Lazarus, называется *Object Pascal*. Код пишется текстовым редактором. В принципе, не требуется, чтобы использовался какой-то определённый текстовый редактор. Можно использовать даже стандартный для Windows редактор Notepad (блокнот). Однако редактор, который предоставляет Lazarus, отлично подходит для этой цели благодаря интеграции с другими элементами среды Lazarus.

Интегрированный Редактор кода – это ещё и отладчик, позволяющий шаг за шагом проверять исходный код программы, изучать значения переменных и свойств во время выполнения, устанавливать контрольные точки и т. д.

Когда создаётся новый проект или Lazarus запускается в первый раз, в окне Редактора кода создаётся новый исходный файл Object Pascal. Этот исходный файл является модулем Object Pascal с именем по умолчанию unit1. Программы Object Pascal обычно построены из большого количества модулей.

Многие модули, необходимые для создания программ, уже существуют в Lazarus. Они называются стандартными, или встроенными, модулями и включают в себя Forms, Windows, SysUtils, Messages, Classes и др.

## 1.7. Менеджер проектов

Чтобы открыть окно Менеджера проектов, надо выбрать «Менеджер проектов» из окна «Вид» главного меню Lazarus. Появится список элементов проекта, активного в настоящее время. Эти элементы – модули и формы, соответствующие некоторым из этих модулей.

В Lazarus можно иметь только один открытый проект, но проект может содержать большое количество модулей и форм.

На рисунке 6 показан Менеджер проектов с открытым проектом, который включает в себя единственный модуль Object Pascal и соответствующую ему форму. Двойной щелчок мышью по модулю Unit1 открывает этот модуль в редакторе кода. Двойной щелчок по



форме Form1 открывает соответствующую форму для визуального проектирования.

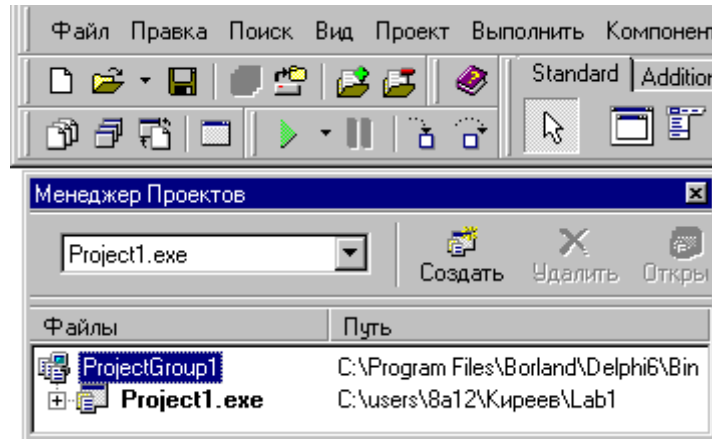


Рисунок 6 – Окно Менеджера проектов

Форма Lazarus всегда имеет соответствующий модуль Object Pascal, который её поддерживает. С другой стороны, исходный код паскалевского модуля не должен быть обязательно связан с какой-либо определённой формой; он может обеспечивать услуги, которые являются потенциально доступными и используются всеми формами.

### 1.8. Выход из Lazarus

Для того чтобы выйти из Lazarus, даже если имеется множество зависимых открытых окон, надо просто выбрать «Выход» (Exit) из меню «Файл» (File) главного окна Lazarus (оно находится вверху экрана вместе с кнопками быстрого доступа и Палитрой компонентов).

При попытке выйти из Lazarus после каких-либо изменений в проекте всегда задаётся вопрос, надо ли сохранить проект. На этот вопрос нельзя отвечать механически, не думая. Нужно быть внимательным. Это последняя возможность сохранить проект, который был модифицирован. Если не сохранить отредактированный проект, это означает, что пользователь хочет вернуться к последней сохранённой на диске версии.

## 2. Разработка алгоритмов и программ линейной и разветвляющейся структуры

Пример:

Разработать структуру данных, алгоритм и программу для вывода на экран слов приветствия и прощания по щелчку кнопки мыши.

Решение:

1. Создайте папку с именем Hello.
2. Запустите на выполнение Lazarus, дважды щелкнув на его пиктограмме на рабочем столе. Появится окно интегрированной среды (рисунок 7).

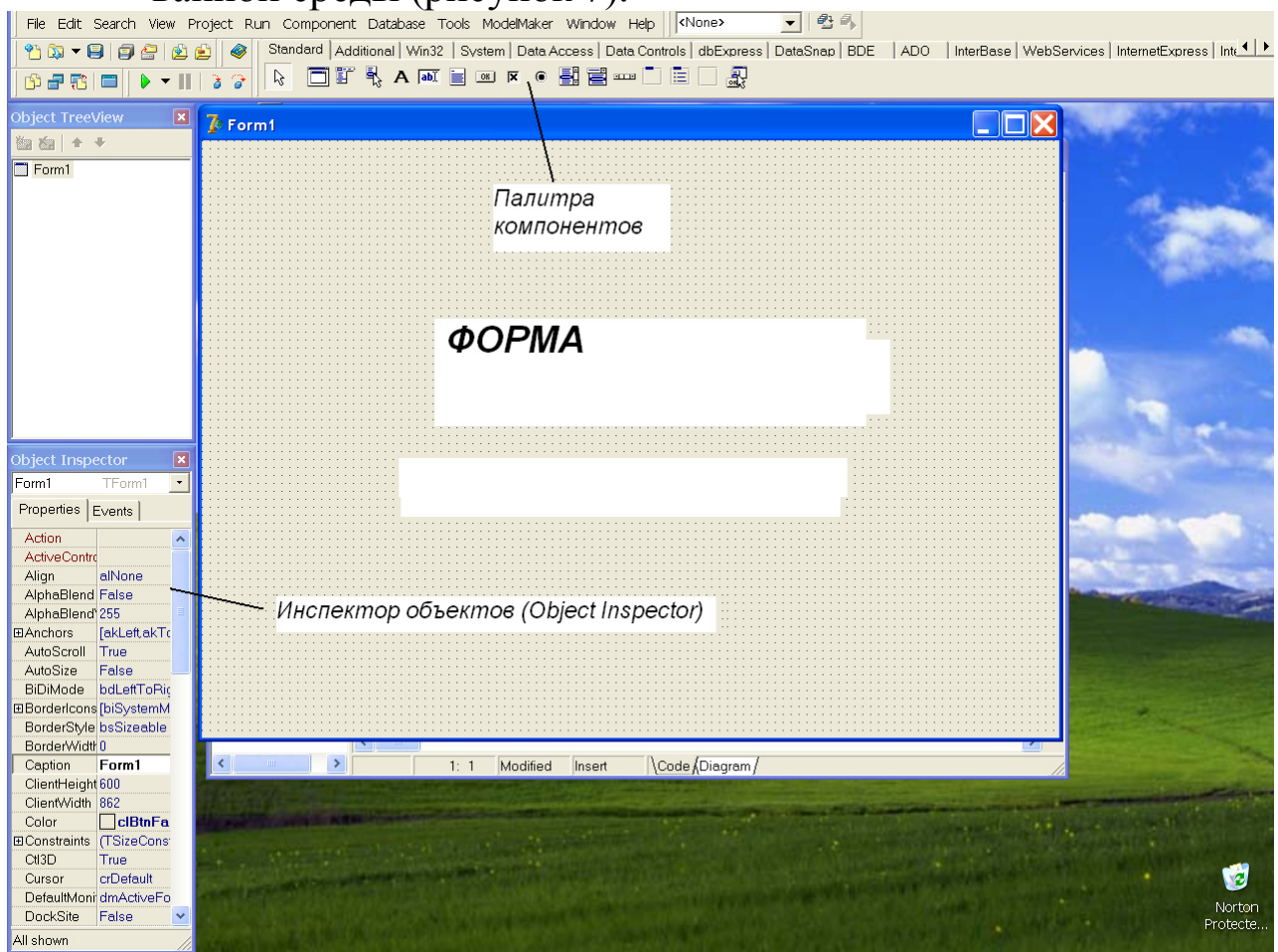
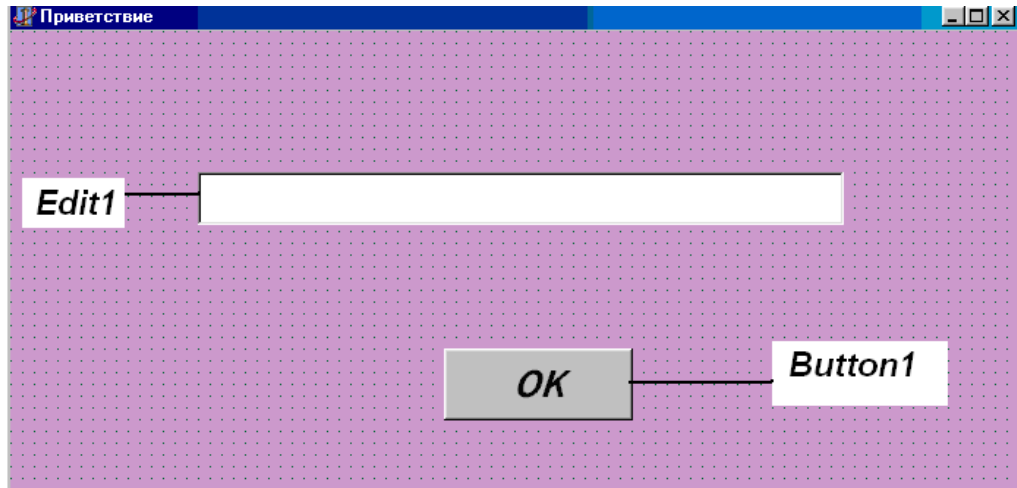


Рисунок 7 – Окно интегрированной среды разработчика Lazarus

3. Постройте форму с компонентами следующего вида:



Для того, чтобы разместить на форме компоненты Edit и Button, необходимо сначала щелкнуть по пиктограмме нужного компонента на палитре компонентов, а затем – в том месте формы, где должен быть расположен компонент (рисунок 8).

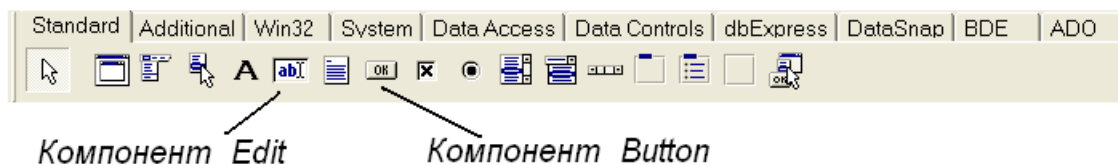


Рисунок 8 – Вид палитры компонентов Standard (Стандартная)

4. Сделайте активным компонент Edit, щелкнув на нем мышью. Появятся маркеры – признак того, что компонент активен.
5. Щелчком мыши перейдите в окно инспектора объектов (Object Inspector). Так как в данный момент активен компонент Edit1, в окне инспектора объектов отображаются его свойства.
6. Найдите в списке свойств строку Text и очистите ее содержимое.
7. Найдите свойство с названием Font (Шрифт), щелкните в поле справа от слова Font, а затем – на кнопке с изображением многоточия. Откроется диалоговое окно Выбор шрифта. Задайте вид шрифта *Arial*, размер шрифта *18 nm*, начертание *Полужирный курсив* и выберите понравившийся вам цвет шрифта. Подтвердите свой выбор, нажав кнопку ОК.

8. Сделайте активным компонент `Button1`. Найдите в окне инспектора объектов свойство `Caption` и замените предлагаемое системой значение на ОК.
9. Установите параметры свойства `Font`, аналогичные заданным для компонента `Edit`.
10. Щелкните в любом месте формы (вне компонентов). Форма становится активной, а в окне инспектора объектов теперь можно редактировать значения свойств формы.
11. Найдите свойство `Color` (Цвет) и выберите понравившееся вам значение цвета.
12. Измените значение свойства `Caption` со стандартного `Form1` на ПРИВЕТСТВИЕ.
13. Дважды щелкните на компоненте `Button.1`. Вы увидите открывшееся окно редактора кода с заготовкой для обработчика события нажатия клавиши (`Click`):

```
procedure TForm1.Button1Click(Sender: TObject);
begin

end;
```

14. Наберите после слова `begin` следующую строку:

```
Edit1.Text := 'Привет!';
```

Получится текст обработчика события нажатия командной кнопки следующего вида:

```
procedure TForm1.Button1Click(Sender: TObject);
begin
    Edit1.Text := 'Привет!';
end;
```

15. Сохраните проект в созданной ранее папке. Для этого: Выберите в главном меню Lazarus команду:  
*File* → *Save Project As*.

На экране появится окно сохранения файла программы. В открывшемся окне в поле Папка откройте папку *Hello* и нажмите кнопку **Сохранить**.

В следующем окне также нажмите кнопку **Сохранить**.

16. Запустите ваш проект на выполнение, нажав на панели инструментов кнопку с изображением зеленой стрелки. Щелкните по кнопке ОК и увидите в окне редактирования приветствие.
17. Внесите в полученную программу изменения таким образом, чтобы на форме появилась еще одна командная кнопка, при нажатии на которую выводилось бы сообщение «ДО СВИДАНИЯ!».
18. Закройте Lazarus. Если система будет предлагать сохранить изменения в проекте, ответьте утвердительно. Откройте папку HELLO и внимательно посмотрите на файлы, которые в ней появились.

Сохраните проект на портативном носителе информации. Помните, что каждый проект Lazarus должен храниться в отдельной папке! Поэтому, прежде чем копировать файлы проекта, создайте на ней папку с именем HELLO. После этого скопируйте на дискету все файлы, кроме файла с расширением имени EXE (Это готовая к выполнению программа, строится каждый раз при компиляции проекта) и файлов, в именах которых есть символ “~” - это предыдущие копии файлов (создаются при внесении в программу изменений).

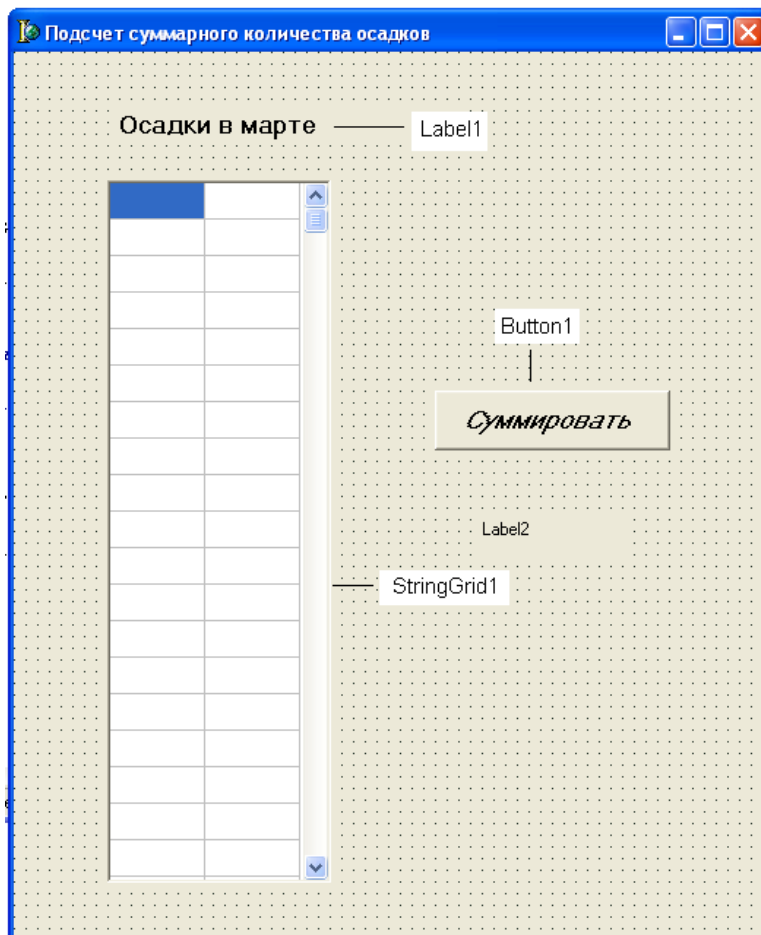
### 3. Работа с массивами

#### Пример

Разработать структуру данных, алгоритм и программу для решения следующей задачи. На метеостанции ведется ежедневное измерение количества выпавших осадков. Определить суммарное количество осадков, выпавших в течение марта.

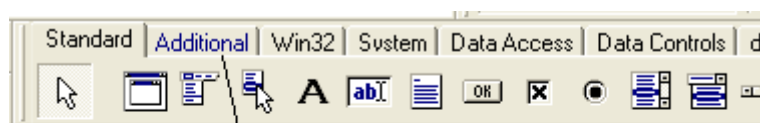
#### Решение

Форма программы имеет следующий вид.

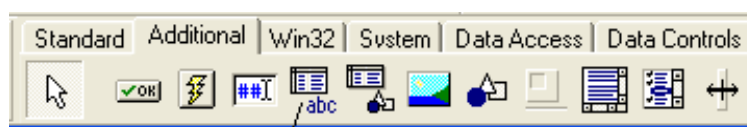


### Технология работы с компонентом StringGrid

1. Чтобы поместить на форме компонент StringGrid (Сетка), надо выбрать в палитре компонентов вкладку Additional (дополнительные).



Набор компонента Additional

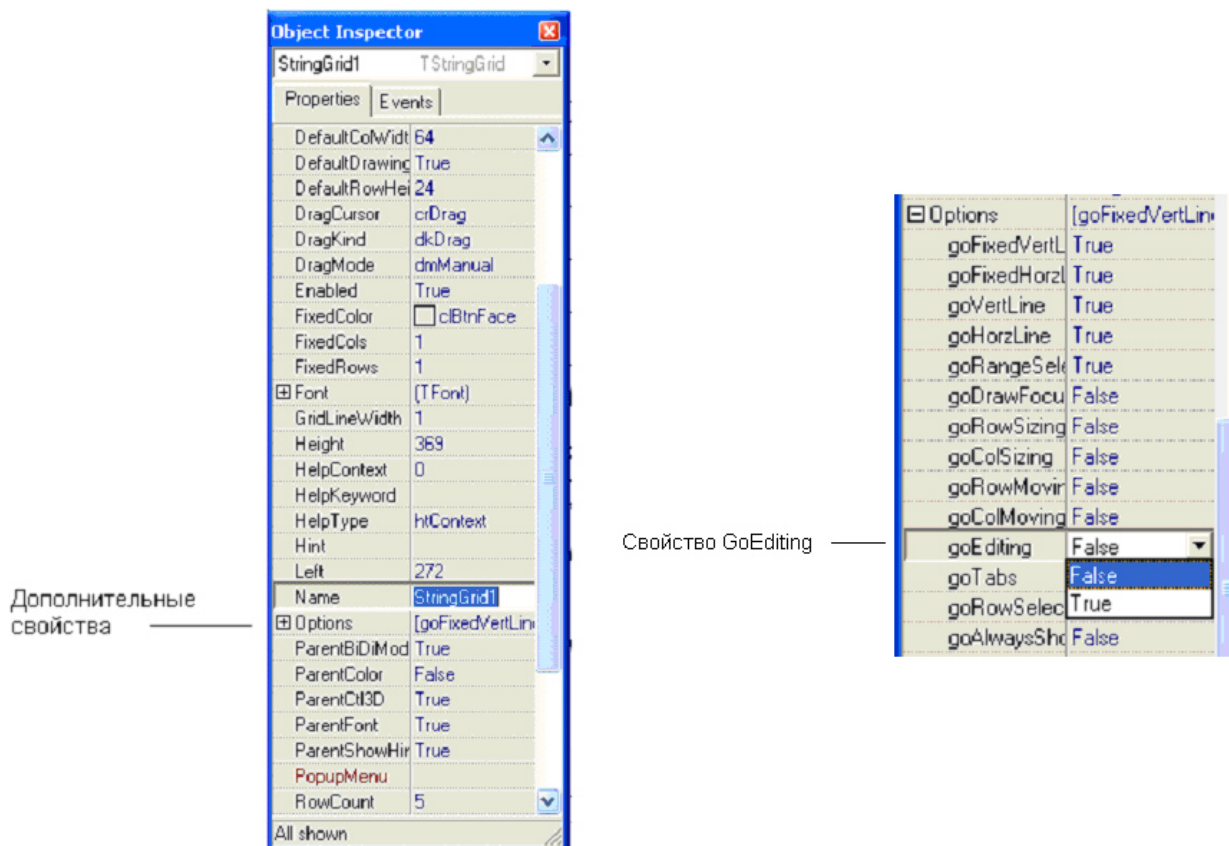


Компонент StringGrid

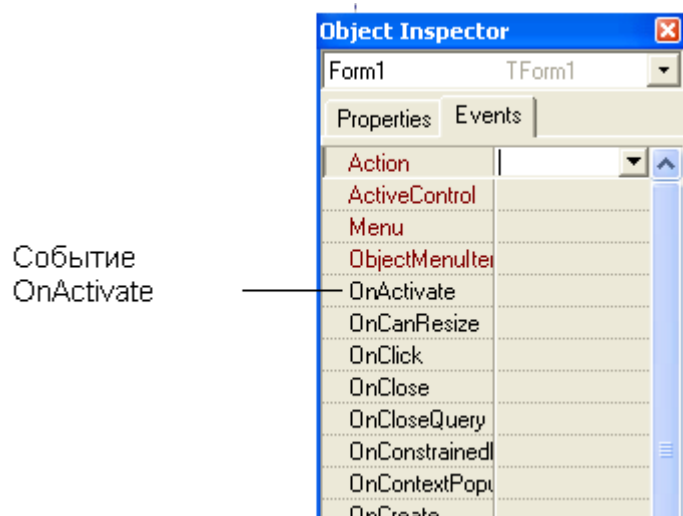
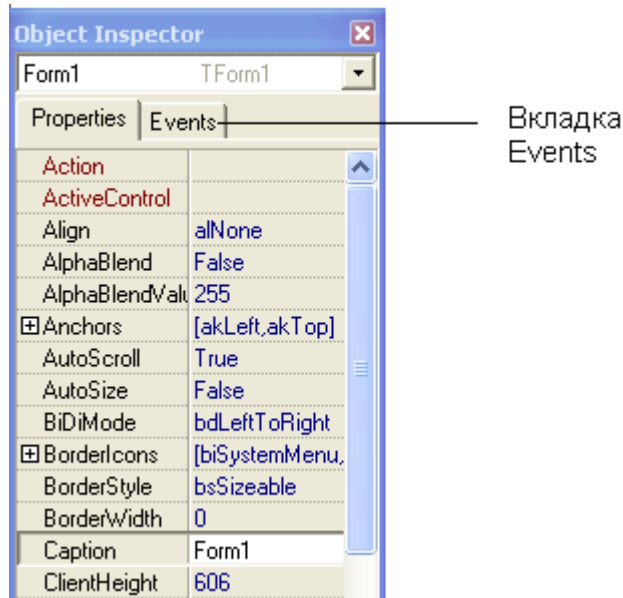
2. Сделать компонент активным, щелкнув по нему мышью, и установить следующие значения перечисленных свойств:

Свойство (Property)	Значение свойства
ColCount	2
RowCount	32
FixedCols	0
FixedRows	0

3. Чтобы сделать таблицу доступной для ввода в нее данных с клавиатуры, надо разрешить ее редактирование. Для этого в списке свойств надо найти строку Options и нажать на знак «+» слева от этого слова. Раскроется список дополнительных свойств компонента. В этом списке найти свойство GoEditing и установить его значение в True.



4. Сделать активной форму. В окне Object Inspector перейти на вкладку Events (События).



Дважды щелкнуть в пустой строке рядом с названием события. Появится шаблон процедуры

```
procedure TForm1.FormActivate(Sender: TObject);
begin

end;
```



и набрать следующий текст:

```

procedure TForm1.FormActivate(Sender: TObject);
var
  i: integer;
begin
  StringGrid1.Cells [0,0] := 'Число месяца';
  StringGrid1.Cells [1,0] := 'Количество осадков,
мм';
  For i:=1 to 31 do
    StringGrid1.Cells [0,i] := IntToStr(i);
end;

procedure TForm1.Button1Click(Sender: TObject);
var
  Osadki : array [1..31] of integer;
  i, Sum : integer;
begin
  { Ввод элементов массива }
  For i:=1 to 31 do
    Osadki [i] := StrToInt(StringGrid1.Cells[1,i]);
  { Суммирование значений элементов массива }
  Sum:=0;
  For i:=1 to 31 do Sum:=Sum + Osadki [i];
  { Вывод результата }
  Label2.Caption := 'Суммарное количество осадков в
марте'+ IntToStr (Sum);
end;

```

#### 4. Задания для самостоятельной работы.

##### 4.1. Разработка алгоритмов и программ линейной структуры

###### Вариант 1

###### *Задача 1*

Разработать структуру данных, алгоритм и программу на языке Lazarus для решения следующей задачи:

Определить площадь и периметр прямоугольника по заданным длинам сторон. Исходные данные вводят с клавиатуры.

###### *Задача 2*

Предложите пользователю ввести с клавиатуры 3 положительных числа. Напечатать сообщение об ошибке, если среди этих чисел будет хотя бы одно отрицательное. В противном случае вычислите сумму чисел.

### *Задача 3*

В машину вводятся 3 числа. Можно ли построить равнобедренный треугольник с такими длинами сторон?

### Вариант 2

#### *Задача 1*

Разработать структуру данных, алгоритм и программу на языке Lazarus для решения следующей задачи:

Определить площадь и периметр прямоугольного треугольника. Исходные данные вводят с клавиатуры.

#### *Задача 2*

Даны 3 числа. Если первые 2 из них больше 10, то вычислить квадрат третьего числа. В противном случае увеличить 2-е и 3-е числа на значение первого.

#### *Задача 3*

Вычислить  $y = x^2 + 5$ , если  $x < -20$ ;  
 $y = x^4 / 4$ , если  $-20 \leq x \leq 10$ ;  
 $y = 0$  в остальных случаях.

### Вариант 3

#### *Задача 1*

Разработать структуру данных, алгоритм и программу на языке Lazarus для решения следующей задачи:

Вычислить площадь круга и длину окружности заданного радиуса. Исходные данные вводят с клавиатуры.

#### *Задача 2*

Даны числа  $a, b, c$ . Проверить, сколько действительных корней имеет квадратное уравнение  $ax^2 + bx + c = 0$ .

#### *Задача 3*

С клавиатуры вводятся координаты 3-х точек. Подсчитать количество точек, попадающих в круг радиусом  $R$  с центром в начале координат.

### Вариант 4

#### *Задача 1*

Разработать структуру данных, алгоритм и программу на языке Lazarus для решения следующей задачи:

Вычислить площадь трапеции. Исходные данные вводят с клавиатуры.

*Задача 2*

С клавиатуры вводят 3 числа. Сколько из этих чисел не находятся в интервале  $[10,25]$ ?

*Задача 3*

Даны числа  $a, b, c, d$ . Если  $a > b > c > d$ , то вычислить квадрат наименьшего из них. В противном случае вычислить сумму чисел.

Вариант 5

*Задача 1*

Разработать структуру данных, алгоритм и программу на языке Lazarus для решения следующей задачи:

Вычислить объем конуса. Исходные данные вводят с клавиатуры.

*Задача 2*

Даны 3 числа. Вычислить сумму квадратов этих чисел, если все они меньше нуля. Если все числа больше нуля, то умножить каждое из них на 2. В остальных случаях напечатать сообщение о том, что числа имеют разные знаки.

*Задача 3*

В ЭВМ вводятся 3 числа, являющиеся оценками на вступительных экзаменах. Зачисляют людей с оценками «отлично» на всех экзаменах. Напечатать сообщение о том будет зачислен абитуриент или нет.

Вариант 6

*Задача 1*

Разработать структуру данных, алгоритм и программу на языке Lazarus для решения следующей задачи:

Вычислить объем шара. Исходные данные вводят с клавиатуры.

*Задача 2*

Даны числа с  $x_1, y_1$  и  $x_2, y_2$ . Верно ли, что через две точки с такими координатами можно провести окружность с центром в начале координат?

*Задача 3*

Даны 3 числа. Если первые два из них больше 10, то вычислите квадрат третьего числа. В противном случае увеличьте каждое из чисел на значение первого числа.

Вариант 7

*Задача 1*

Разработать структуру данных, алгоритм и программу на языке Lazarus для решения следующей задачи:

Вычислить объем куба. Исходные данные вводят с клавиатуры.

*Задача 2*

Даны координаты трех точек. Сколько из них попадают в круг радиусом  $R$  и с центром в начале координат?

*Задача 3*

Даны 3 оценки студента за сессию. Если все оценки «отлично», то напечатать сообщение «отличник». Если есть хотя бы одна «двойка» – «неуспевающий». В остальных случаях – «Сессия сдана успешно».

Вариант 8

*Задача 1*

Разработать структуру данных, алгоритм и программу на языке Lazarus для решения следующей задачи:

Вычислить объем параллелепипеда. Исходные данные вводят с клавиатуры.

*Задача 2*

Даны 3 числа. Могут ли эти числа быть сторонами прямоугольного треугольника?

*Задача 3*

С клавиатуры вводятся числа  $X$ ,  $Y$ ,  $Z$ . Если  $X < Y < Z$ , то возвести наименьшее из них в квадрат. В противном случае удвоить все числа.

Вариант 9

*Задача 1*

Разработать структуру данных, алгоритм и программу на языке Lazarus для решения следующей задачи:

Вычислить площадь параллелограмма. Исходные данные вводят с клавиатуры.

*Задача 2*

Дано 3 целых числа. Если все эти числа могут быть оценками на экзамене (т.е. находиться в интервале от 2 до 5), то напечатать об

этом сообщении. В противном случае присвоить каждому числу значение 4.

### *Задача 3*

Проверить, можно ли через 2 точки с заданными координатами провести окружность с центром в начале координат.

### Вариант 10

#### *Задача 1*

Разработать структуру данных, алгоритм и программу на языке Lazarus для решения следующей задачи:

Вычислить расстояние от точки на координатной плоскости до начала координат. Исходные данные вводят с клавиатуры.

#### *Задача 2*

Составить программу, которая запрашивает пароль на вход в программу. Если пароль введен верно, то предложить пользователю ввести 3 числа и возвести каждое из них в квадрат. При неверном пароле вывести соответствующее сообщение.

#### *Задача 3*

Даны 3 числа. Если эти числа могут быть длинами сторон треугольника, то вычислить его полупериметр. В противном случае найти максимальное из них.

### Вариант 11

#### *Задача 1*

Разработать структуру данных, алгоритм и программу на языке Lazarus для решения следующей задачи:

Вычислить длину гипотенузы прямоугольного треугольника. Исходные данные вводят с клавиатуры.

#### *Задача 2*

Если среди введенных с клавиатуры 3-х чисел есть хотя бы одно отрицательное число, то возвести в квадрат. Если нет, то напечатать об этом сообщение.

#### *Задача 3*

Даны 3 числа. Чему равно минимальное из них?

### Вариант 12

#### *Задача 1*

Разработать структуру данных, алгоритм и программу на языке Lazarus для решения следующей задачи:

Вычислить длину диагонали прямоугольника. Исходные данные вводят с клавиатуры.

*Задача 2*

Даны 3 числа. Вычислить сумму квадратов этих чисел, если все они меньше нуля. Если все числа больше нуля, то оставить числа без изменения. Если числа имеют разные знаки, то заменить их абсолютными значениями.

*Задача 3*

В ЭВМ вводятся координаты 3-х точек. Определить, сколько из них находятся в I-ой четверти координатной плоскости.

Вариант 13

*Задача 1*

Разработать структуру данных, алгоритм и программу на языке Lazarus для решения следующей задачи:

Вычислить радиус круга по заданному значению его площади. Исходные данные вводят с клавиатуры.

*Задача 2*

Даны числа a,b,c. Если эти числа могут быть длинами сторон прямоугольного треугольника, то вычислите его площадь.

*Задача 3*

Если все числа X, Y, Z >10, то найти среди них максимальное. В противном случае – найти минимальное число.

Вариант 14

*Задача 1*

Разработать структуру данных, алгоритм и программу на языке Lazarus для решения следующей задачи:

Вычислить радиус окружности по заданному значению её длины. Исходные данные вводят с клавиатуры.

*Задача 2*

Даны числа a,b,c. Проверить, сколько действительных корней имеет квадратное уравнение  $ax^2 + bx + c = 0$ .

*Задача 3*

С клавиатуры вводят число X. Вычислить Y:

$$y = x^2 + 10, \text{ если } x > -10;$$

$$y = x^3/3, \text{ если } -10 \leq x \leq 5;$$

$$y = 0 \text{ в остальных случаях.}$$

Вариант 15

*Задача 1*

Разработать структуру данных, алгоритм и программу на языке Lazarus для решения следующей задачи:

Определить площадь и периметр прямоугольника по заданным длинам сторон. Исходные данные вводят с клавиатуры.

*Задача 2*

Даны 3 оценки студента на экзамене. Проверьте, верно ли, что все оценки 4 и 5?

*Задача 3*

В ЭВМ вводятся координаты 3-х точек. Определить, сколько из них находятся в I-ой четверти координатной плоскости.

Вариант 16*Задача 1*

Разработать структуру данных, алгоритм и программу на языке Lazarus для решения следующей задачи:

Вычислить площадь круга и длину окружности заданного радиуса. Исходные данные вводят с клавиатуры.

*Задача 2*

Даны числа  $a, b, c, d$ . Если  $a > b > c > d$ , то заменить каждое из чисел значением наибольшего из них. В противном случае вычислить сумму чисел.

*Задача 3*

Даны 3 числа. Если эти числа могут быть длинами сторон треугольника, то вычислить его полупериметр. В противном случае найти максимальное из них.

**5. Контрольные вопросы**

1. Что такое IDE?
2. Как запустить Lazarus?
3. Что такое кнопка быстрого доступа?
4. Что такое Линейка инструментов?
5. Что такое Палитра компонентов?
6. Как выбрать нужную страницу Палитры компонентов?
7. Как можно настроить Линейку инструментов?
8. Что такое Дизайнер форм?
9. Что такое свойство?
10. Чем отличается Дизайнер форм от формы?

11. Что такое Инспектор объекта?
12. Как получить доступ к свойствам, расположенным на странице "События" (Events)?
13. Для чего нужно окно Редактора кода?
14. Как открыть окно Менеджера проектов?
15. Сколько одновременно открытых проектов может иметь Lazarus?



## **МИНОБРНАУКИ РОССИИ**

Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«Юго-Западный государственный университет»  
(ЮЗГУ)

Кафедра охраны труда и окружающей среды



### **РАСЧЕТ КОНЦЕНТРАЦИЙ ВРЕДНЫХ ВЕЩЕСТВ В ОКРУЖАЮЩЕЙ СРЕДЕ**

#### **С ИСПОЛЬЗОВАНИЕМ ПК «ПРИЗМА»**

Методические указания к проведению  
лабораторных работ по дисциплине

«Информационные технологии в сфере безопасности»  
для студентов направления подготовки 20.04.01  
«Техносферная безопасность»

Курск 2021

УДК 699.85

Составители: И.О. Кирильчук, В.В. Протасов

Рецензент

Кандидат технических наук, доцент *Г.П. Тимофеев.*

**Расчет концентраций вредных веществ в окружающей среде с использованием ПК «Призма»:** методические указания к проведению лабораторных работ по дисциплине «Информационные технологии в сфере безопасности» / Юго-Зап. гос. ун-т; сост.: И.О. Кирильчук, В.В. Протасов. Курск, 2021. 22 с.

Излагается методика использования программного комплекса «Призма» для расчета рассеивания в атмосфере выбросов от стационарных источников.

Методические указания предназначены для студентов направления подготовки 20.04.01 Техносферная безопасность.

Текст печатается в авторской редакции

Подписано в печать                      Формат 60x84 1/16.

Усл. печ. л. 1,28. Уч.-изд.л. 1,16. Тираж 30 экз. Заказ     . Бесплатно.

Юго-Западный государственный университет.

305040, г. Курск, ул. 50 лет Октября, 94.

**Цель работы:**

- изучение основных возможностей программного комплекса «Призма»;
- проведение на ЭВМ расчета концентраций вредных веществ в атмосфере от выбросов стационарных источников с использованием ПК «Призма».

**Общие сведения**

Программные комплексы (ПК) серии «Призма» на базе унифицированной программы для расчета загрязнения атмосферы (УПРЗА) предназначены для автоматизированной поддержки выработки и принятия управленческих, технологических и проектных решений по формированию комплекса воздухоохраных мероприятий на территории предприятия/города/региона.

ПК «Призма» позволяет:

- задавать в одном варианте расчета несколько промплощадок; задавать в одном варианте расчета несколько загрязняющих веществ и/или групп суммации загрязняющих веществ;
- задавать для каждого выброса источника свой уникальный коэффициент учета скорости оседания (F) по каждому выбрасываемому ЗВ;
- задавать в одном варианте расчета несколько участков местности и/или отдельных точек, для которых необходимо произвести расчет концентраций;
- проводить экспресс-расчет (до проведения основного расчета) максимальных концентраций, опасного расстояния и опасной скорости ветра по каждому источнику выбросов по каждому загрязняющему веществу
- проводить расчет полей приземных концентраций с перебором от одной до десяти заданных скоростей ветра или с перебором в заданном интервале с определенным шагом;
- проводить расчет полей приземных концентраций с перебором от  $1^\circ$  до  $360^\circ$  (для полного круга с шагом один градус) направлений ветра или с автоматическим определением направления ветра, при котором в расчетной точке достигается максимальная концентрация;

- проводить расчет полей приземных концентраций с учетом фоновых концентраций, при этом допускается задание фоновых концентраций не для всех веществ, участвующих в расчете;
- проводить расчет полей приземных концентраций по группам суммации, когда часть веществ, составляющих группу, находится в выбросах источников, а часть веществ из группы суммации в выбросах отсутствуют, но присутствуют в фоне;
- строить расчетные, нормативную и суммарную СЗЗ;
- решать задачу нормирования;
- сформировать все разделы тома ПДВ, включая шаблон пояснительной записки;
- выводить результаты расчета на экран монитора и принтер в форме графического и табличного представлений в произвольном (удобном) масштабе;
- сохранять результаты расчетов в файлах в графическом и текстовом виде;
- гибко работать с графическим представлением расчетного поля концентраций в виде изолиний в расчетном прямоугольнике, предоставляя широкие возможности в пластичной перерисовке картинки: добавление новых изолиний в режиме сканирования; удаление ненужных изолиний; изменение цвета объектов; прорисовка на картинке границ нормативной и расчетных СЗЗ.
- обеспечивать HELP-поддержку в любом режиме работы программы.

Количество источников выбросов, загрязняющих веществ и/или групп суммации, расчетных прямоугольников и/или точек, участвующих в расчете, не ограничено.

### **Порядок выполнения работы**

Работа состоит из следующих стадий:

- I. Подготовка исходных данных, на основании которых выполняются расчеты.
  1. Построение дерева «Административно - территориальной структуры».
  2. Внесение характеристик воздушного бассейна рассчитываемой территории (метеоусловий и фоновых концентраций ЗВ).

3. Внесение данных инвентаризации

II. Проведение расчета рассеяния.

4. Расчет, просмотр результатов.

**I. Подготовка исходных данных, на основании которых выполняются расчеты.**

**1. Построение дерева «Административно - территориальной структуры».**

1.1. В главном меню выбираем пункт «Исходные данные» с помощью мыши или клавиши «Alt». Появляется подменю из пяти пунктов:

**Административно - территориальная структура**

**Промплощадки – инвентаризация выбросов**

**Справочник загрязняющих веществ**

**Справочник групп суммации**

**Выход**

1.2. В подменю с помощью мыши или клавиш-«стрелок» выбираем пункт «Административно - территориальная структура». Нажимаем клавишу «Enter». Появляется древовидная форма БД «Административно - территориальная структура». Курсор (синяя полоска) вверху. Построим ветвь «дерева».

1.3. Выполним операцию добавления записи: а) нажмем кнопку «Добавить запись» или б) выберем пункт «Запись/Добавить сына» в меню БД или в) нажмем сочетание клавиш «Ctrl +». Появляется классификатор «Порядок».

1.4. В классификаторе «Порядок» выберем курсором тип объекта – «Область» (**обязательный**). Нажмем клавишу «Enter» (или кнопку «OK» на панели управления или дважды щелкнуть мышкой). Появится страничная форма «Область или город федерального подчинения».

1.5. Заполним поля формы (введем сведения):

1.5.1 Поле «Код территории по СОАТО» *можно не заполнять*.

1.5.2 В поле «Наименование территории» (**обязательное**) вводится наименование территории. Например: Московская обл. или г. Москва. Ввод или редактирование начинается с выбора курсором поля: выбранное поле станет синим. Затем а) нажатием клавиши «Enter» или б) любых букв переходим в режим ввода или редактирования поля БД. Окончание редактирования (ввода) подтверждаем

нажатием клавиши «*Enter*», отказ от редактирования - клавишей «*Esc*».

1.5.3 После окончания заполнения формы нажимаем кнопку «ОК».

1.6. В БД «Административно – территориальная структура» появится зеленый «листок» с названием территории. Перемещаем курсор (синюю полосу) на появившийся «листок».

1.7. Повторяем операцию добавления записи, описанную в п.1.3. Появляется классификатор «Порядок».

1.8. В классификаторе «Порядок» выбираем тип объекта: «Населенный пункт».

1.9. Выбран тип объекта «Населенный пункт» (**обязательный**). Нажмем клавишу «*Enter*» (или произведем равнозначные действия из п.1.4) - появится страничная форма «Административный округ /город /населенный пункт».

1.10. Повторяем п.1.5.,1.6.,1.7.

1.11. В классификаторе «Порядок» выбираем тип объекта: «Природопользователь» (**обязательный**). Появится страничная форма «Ввод общих сведений».

1.12. Заполняем поля формы: в поле «Природопользователь» (**обязательное**) вводится наименование предприятия - природопользователя.

1.13. После внесения общей информации о природопользователях можно переходить к следующему этапу. Для этого на панели управления древовидной формы БД нажимаем кнопку «ОК». Появляется табличная форма БД «Административно – территориальная структура».

1.14. Для завершения работы с табличной формой БД нажать кнопку «ОК» на панели управления БД.

**2. Внесение характеристик воздушного бассейна рассчитываемой территории (метеоусловий и фоновых концентраций ЗВ).**

2.1 В Главном меню выбираем пункт «Исходные данные» с помощью мыши или клавиши «*Alt*». Появляется подменю из пяти пунктов.

2.2 В подменю с помощью мыши или клавиш - «стрелок» выбираем пункт «Промплощадки – инвентаризация выбросов». На-

жимаем клавишу «Enter». Появляется древовидная форма БД «Инвентаризация выбросов по местоположению промплощадки»

2.3 Выбираем курсором «листок», которому соответствует населенный пункт (самый последний в «ветке»), нажимаем «Enter». Появляется страничная форма БД «**Характеристики воздушного бассейна**». Заполним поля БД.

2.4 Поле «Местоположение» является *автоматически заполняемым*, и поэтому если установить курсор на этом поле, оно будет черного цвета, без возможности редактирования.

2.5 Поле (*обязательное*) «Метеоусловия». Выбираем курсором поле и нажимаем «Enter». Открывается страничная форма подчиненной БД «Метеоусловия».

2.6 Заполняем поля БД «Метеоусловия»:

2.6.1 Поле (*черное, автоматически заполняемое*) «Наименование территории» не редактируется

2.6.2 Поле «Площадь города (км<sup>2</sup>)» заполнять *обязательно, только* если при расчете требуется учитывать изменения фоновой концентрации ЗВ на рассчитываемой территории (города) (тогда в п.2.8. вводят измерения фона ЗВ на нескольких постах).

2.6.3 Поле (*обязательное*) «Коэффициент А (безразмерный)» зависит от температурной стратификации атмосферы. Диапазон значений от 140 до 250.

2.6.4 В поле (*обязательное*) «Скорость ветра U\* (м/с)» вводится значение скорости ветра, превышаемое в данной местности в среднем многолетнем режиме в 5% случаев. Значение запрашивается в УГКС Госкомгидромета.

2.6.5 Поле (*обязательное*) «Температура зимой (градусы Цельсия)» используется для расчета  $\Delta T$ , как температура окружающего атмосферного воздуха для источников с признаком расчета «зима».

2.6.6 Поле (*обязательное*) «Температура летом (градусы Цельсия)» используется для расчета  $\Delta T$ , как температура окружающего атмосферного воздуха для источников с признаком расчета «лето».

2.6.7 Поля, объединенные заголовком «Роза ветров %», *обязательны только* при проведении расчета проектной СЗЗ с учетом розы ветров. В сумме должно быть или 0 или 100 %.

2.6.8 После того, как все поля заполнены, завершаем работу с БД «Метеоусловия» нажатием кнопки «ОК» на ее панели управления.

2.7 Вернулись в страничную форму БД «Характеристики воздушного бассейна».

2.8 Поле «Посты наблюдения за фоном» **обязательно только** для проведения расчета рассеяния с учетом фоновых концентраций. Выбираем поле и нажимаем «Enter». Появляется табличная форма БД «Посты наблюдения за фоном».

2.9 Заполняем ее поля:

2.9.1 В поле (**обязательное**) «Номер поста» вводится произвольное целое число.

2.9.2 В поле «Наименование поста» вводится любое наименование.

2.9.3 Поле (**обязательное**) «Активность поста» - если введено значение «ДА», то пост будет использоваться в расчетах, если значение пусто, то не будут копироваться значения фоновых концентраций данного поста в вариант расчета. Значение изменяется нажатием «Enter».

2.9.4 Поля «Координата X, Y поста в СК города» определяют местоположение поста на карте территории (города) и влияют на значение расчетного фона.

2.9.5 Поле (**обязательное**) «ЗВ в фоне атмосферного воздуха». После нажатия «Enter» появляется табличная форма БД «ЗВ в фоне атмосферного воздуха». Заполним ее поля.

2.9.6 Поле «Наименование ЗВ» указывает, какое ЗВ находится в атмосферном воздухе. Нажимаем «Enter», появляется окошко, в котором можно указать код ЗВ, если же код ЗВ неизвестен, то нажимаем «Enter» - появляется список «Справочник загрязняющих веществ в воздухе», в котором с помощью операций поиска отыскивают необходимое ЗВ. После окончания поиска нажать «Enter».

2.9.7 Поле «Режим задания фоновых значений» определяет, для скольких направления ветра (1, 5 или 9) будут вводиться значения фоновых концентраций вещества.

2.9.8 Поле «Фоновые концентрации» - после нажатия «Enter» появляется страничная форма БД «Фоновые концентрации». Заполним ее поля.



- Поле «Код» не редактируется.
- Поле «ПДК (с учетом x10)» не редактируется, а (x10) означает, что если в справочнике ЗВ не указано значение ПДКм.р., то в расчетах используется значение ПДКс.с., умноженное на 10, если не указано ПДКс.с., то используется ОБУВ (без умножения на 10).

- В поле (**обязательное**) «Значение концентраций (мг/м<sup>3</sup>)» вводятся значения концентраций, количество полей зависит от ввода в п.2.9.7. Работа с БД «Посты наблюдения за фоном» завершается нажатием кнопки «ОК» трижды.

2.10 Нажатием кнопки «ОК» на панели управления БД «Характеристики воздушного бассейна» вернемся в древовидную форму БД «Инвентаризация выбросов по местоположению промплощадки». Далее можно сразу перейти к выполнению п.3.3.

### **3. Внесение данных инвентаризации.**

3.1 В Главном меню выбираем пункт «Исходные данные» с помощью мыши или клавиши «Alt». Появляется подменю из пяти пунктов.

3.2 В подменю с помощью мыши или клавиш - «стрелок» выбираем пункт «Промплощадки – инвентаризация выбросов». Нажимаем клавишу «Enter». Появляется древовидная форма БД «Инвентаризация выбросов по местоположению промплощадки»

3.3 Выбираем курсором «листок», которому соответствует населенный пункт (самый последний в «ветке»)

3.4 Выполним операцию добавления записи: а) нажмем кнопку «Добавить запись» или б) выберем пункт меню «Запись/Добавить сына» или в) нажмем сочетание клавиш «Ctrl +». Появляется меню «Выбор БД для добавления».

3.5 Выбираем строку «Производственные площадки», нажимаем «Enter». Появляется табличная форма БД «**Производственные площадки**». Заполняем ее поля:

3.6 Поле (**обязательное**) «Предприятие». Нажимаем «Enter». Появляется древовидная форма БД «Административно - территориальное деление». Выбираем курсором «листок» «Предприятие» (самый последний уровень в «ветке» дерева). Нажимаем «Enter». Поле заполнилось.

3.7 Поле (**обязательное**) «Наименование промплощадки» заполняется *автоматически*, но его можно редактировать

3.8 Поле (*обязательное*) «Номер промплощадки» - указываем номер. После завершения ввода с помощью кнопки «ОК» возвращаемся в древовидную форму БД «Инвентаризация выбросов по местоположению промплощадки».

3.9 В древовидной форме БД «Инвентаризация выбросов по местоположению промплощадки» появится «листок» желтого цвета - промплощадка.

3.10 Выбираем промплощадку и нажимаем «*Enter*». Появляется страничная форма БД «**Производственная площадка**».

3.11 Все поля формы должны быть уже заполнены, кроме поля «Промплощадка как источник загрязнения атмосферы». Выбираем его и нажимаем «*Enter*». Появляется страничная форма БД «**Промплощадка (инвентаризация)**». Заполняем ее поля:

3.12 Первые три поля являются *автоматически* заполняемыми. Работа начинается с поля «Координаты, граница, застройка». Выбираем его и нажимаем «*Enter*». Появляется страничная форма БД «**Координаты и границы промплощадки в городе**». Заполняем ее поля:

3.12.1. Поля «Предприятие» и «Промплощадка» являются *автоматически* заполняемыми

3.12.2. В поле (*обязательное*) «Коэффициент учета влияния рельефа» вводится значение, определяемое по Методике расчета.

3.12.3. Поле «Выбор СК промплощадки» при нажатии «*Enter*» появляется меню «Выбор СК задания границ промплощадки» из двух строк. Если выбрана первая строка, то все координаты вводятся в городской СК. Иначе координаты границ промплощадки и ИЗА задаются в координатах промплощадки, которые связаны с городской СК (см. п.3.12.4)

3.12.4. Поля «Координаты X /Y нуля СК промплощадки в городской СК (м)» и «Угол разворота СК промплощадки относительно СК города (градусы)» используются при расчете рассеяния. Заполнять *обязательно только*, когда в поле «Выбор СК промплощадки» из п. 3.12.3. выбрано «2 – промплощадки СК». Позволяют задавать координаты границ промплощадки и ИЗА в координатах промплощадки.

3.12.5. Поле «Санитарная классификация промплощадки» - при нажатии «*Enter*» появляется меню «Санитарная классификация

промплощадки», содержащее пять классов промплощадок. Выберем курсором строку меню, нажмем «*Enter*», при этом *автоматически* заполняется поле «Минимальный размер санзоны» (см.п.3.12.6) (в соответствии с выбранным классом промплощадки).

3.12.6. Поле «Минимальный размер санзоны» - значение заполняется *автоматически* (см.п.3.12.5), но можно отредактировать.

3.12.7. Поле «Границы промплощадки» - при нажатии «*Enter*» появляется табличная форма БД «Границы промплощадки». Заполняем ее поля:

- Поля «Координата вершины X, Y» задают местоположение вершин многоугольника, который ограничивает промплощадку. «ОК» завершение работы с БД.

3.12.8. Поле «Границы жилой зоны» - при нажатии «*Enter*» появляется табличная форма БД «Границы жилой зоны». Заполняем ее поля:

- Поля «Координата вершины X, Y» задают местоположение вершин ломаной линии, которая ограничивает жилую зону. «ОК» завершение работы с БД.

3.12.9. Поле «Застройка» - при нажатии «*Enter*» появляется табличная форма БД «Застройка», которая используется для учета влияния застройки при расчете полей рассеяния, а также для прорисовки строений застройки при просмотре графического представления результатов расчетов. В нее вводят номер строения, его высоту, ширину и координаты строения по длине. «ОК» завершение работы с БД.

3.13 Вернулись в БД «Промплощадка (по местоположению)». Поле (*обязательное*) «Режим работы (выбросов)» - при нажатии «*Enter*» появляется табличная форма БД «**Режимы выбросов всего предприятия (промплощадки) в целом**». Заполняем ее поля:

- Поле «Промплощадка» является *автоматически* заполняемым.

- Поле (*обязательное*) «Рабочее состояние» - при нажатии «*Enter*» появляется меню «Рабочее состояние», в котором делаем выбор. Выберем «Действующий».

- Поле (*обязательное*) «Тип режима выбросов» - при нажатии «*Enter*» появляется меню «Тип режима выбросов». Выберем «Штатный».

- В поле (*обязательное*) «Номер режима» вводим цифру.
- В поле (*обязательное*) «Наименование режима» - вводим любой текст.
- Введенной информации достаточно для работы «Призмы» и поле «Периоды времени реализации режима выбросов» *может быть пропущено*. «ОК» завершение работы с БД.

3.14 Вернулись в БД «Промплощадка (по местоположению)». Поле (*обязательное*) «ИНВЕНТАРИЗАЦИЯ (форма 1-воздух)» - нажимаем «Enter». Появляется табличная форма БД «**Инвентаризация (форма 1-воздух)**». Заполняем ее поля:

3.15 Поля «Промплощадка» и «Категория предприятия...» не редактируются.

3.16 Поле (*обязательное*) «Дата начала инвентаризации» - нажатием «Пробела» вводим текущую дату, но можно редактировать на любую.

3.17 Поле (*обязательное*) «ИЗА (инвентаризация)» - нажимаем «Enter» появляется табличная форма подчиненной БД «**ИЗА (инвентаризация)**». Заполняем ее поля:

3.18 В поле (*обязательное*) «Номер источника выбросов» вводим номер ИЗА. При этом в поле «Наименование источника выбросов» *автоматически* появляется номер как часть наименования.

3.19 Поле (*обязательное*) «Наименование источника выбросов». При вводе наименования в нем можно удалить номер, введенный в п.3.20 как часть наименования.

3.20 Поле (*обязательное*) «Активность ИЗА» - изменяет значение при нажатии «Enter», если указано «ДА», то источник используется для формирования варианта расчета. Если пусто, то ИЗА не обрабатывается.

3.21 Поле (*обязательное*) «Организованный/неорганизованный» - нажимаем «Enter» - появляется меню, в котором выбираем вид источника. Обычно, к «организованным» относятся трубы (точечный круглый, точечный прямоугольный), к «неорганизованным» - дороги и т.п. (площадной - пылящий). Значение поля используется при формировании тома ПДВ.

3.22 Поле (*обязательное*) «Тип источника» - нажимаем «Enter» - появляется меню «**Тип источника**», в котором необходимо выбрать один из 6 типов ИЗА. С типом связано задание параметров ис-

точника выбросов ИЗА (см.п.3.28). Тип «Точечный круглый» описывает обычные трубы, тип «Площадной - пылящий» описывает дорогу и т.п.

3.23 Поле (*обязательное*) «Характеристики функционирования (инвентаризация)» - при нажатии «*Enter*» появляется табличная форма подчиненной БД «Характеристики функционирования ИЗА (инвентаризация)». Заполним ее поля:

3.24 Поле (*обязательное*) «Режим работы предприятия» - нажимаем «*Enter*». Появляется меню «Режимы выбросов всего предприятия (промплощадки) в целом», содержащее перечень режимов работы предприятия, сформированных в п.3.15. Выбираем нужный и нажимаем «*Enter*».

3.25 Поле «Рабочее состояние режима» заполняется *автоматически*.

3.26 Поле (*обязательное*) «Параметры источника выбросов» нажимаем «*Enter*». Появляется страничная форма БД, название которой соответствует 3.24 (например «Точечный круглый источник выбросов»).

3.27 Заполним поля страничной формы (количество и состав полей зависят от 3.24). Три следующих поля относятся к любому типу источника:

- Поле «Высота» - при значениях  $< 2\text{м}$  в расчетах принимается  $=2\text{м}$

- Поле (*обязательное*) «Признак расчета по зимней температуре» - при нажатии «*Enter*» появляется меню из двух строк: «Зима» и «Лето».

- В поле «Коэффициент учета влияния рельефа» *автоматически* переносится значение из 3.14.2., но при необходимости оно может быть изменено, и при расчете будет использовано измененное значение.

3.28 Остальные поля зависят от выбора в п.3.24. Далее перечислены возможные варианты заполнения:

1) Тип «Точечный круглый». Необходимо заполнить поля «Диаметр», «Температура (С°)», «Скорость выхода ГВС (м/с)», «Координаты источника выбросов в СК предприятия (промплощадки) (м) X, Y» Вместо поля «Скорость выхода ГВС (м/с)» можно запол-

нить поле «Средний расход ГВС (м<sup>3</sup>/с)» (каждое из этих полей автоматически рассчитывается через другое с помощью диаметра).

2) Тип «Точечный прямоугольный». Необходимо заполнить поля «Длина», «Ширина», «Температура (С°)», «Скорость выхода ГВС (м/с)», «Координаты источника выбросов в СК предприятия (промплощадки) (м) X, Y» ». Вместо поля «Скорость выхода ГВС (м/с)» можно заполнить поле «Средний расход ГВС (м<sup>3</sup>/с)» (каждое из этих полей автоматически рассчитывается через другое с помощью длины и ширины).

3) Тип «Линейный - аэрационный фонарь». Необходимо заполнить поля «Температура (С°)», «Средний расход ГВС (м<sup>3</sup>/с)», «Средняя скорость выхода ГВС (м/с)», «Координаты концов в СК предприятия (промплощадки) (м) X1, Y1, X2, Y2».

4) Тип «Линейный – аппроксимация точечных ». Необходимо заполнить поля «Средний диаметр одиночного источника», «Температура (С°)», «Средняя скорость выхода ГВС (м/с)», «Координаты концов в СК предприятия (промплощадки) (м) X1, Y1, X2, Y2». Вместо поля «Средняя скорость выхода ГВС (м/с)» можно заполнить поле «Средний расход ГВС одиночного источника (м<sup>3</sup>/с)» (каждое из этих полей автоматически рассчитывается через другое с помощью среднего диаметра).

5) Тип «Площадной – пылящий». Необходимо заполнить поля «Координаты середин сторон источника в СК предприятия (промплощадки) (м) X1, Y1, X2, Y2», «Ширина площадного источника выбросов (м)». Площадной источник представляется в виде прямоугольника, расположенного вдоль линии, соединяющей середины сторон с заданной шириной.

6) Тип «Площадной – аппроксимация точечных». Необходимо заполнить поля «Средний диаметр одиночного источника (м)», «Температура (С°)», «Средняя скорость выхода ГВС (м/с)», «Координаты середин сторон источника в СК предприятия (промплощадки) (м) X1, Y1, X2, Y2», «Ширина площадного источника выбросов (м)». Вместо поля «Средняя скорость выхода ГВС (м/с)» можно заполнить поле «Средний расход ГВС одиночного источника (м<sup>3</sup>/с)» (каждое из этих полей автоматически рассчитывается через другое с помощью среднего диаметра). Площадной источник представляется

в виде прямоугольника, расположенного вдоль линии, соединяющей середины сторон с заданной шириной.

3.29 Поле (**обязательное**) «Выбросы ЗВ из ИЗА (инвентаризация)». При нажатии «Enter» появляется табличная форма БД «**Выбросы ЗВ из ИЗА (инвентаризация)**». Эта БД содержит следующие поля:

3.29.1. Поле «Режим функционирования» заполняется *автоматически*.

3.29.2. Поле «Загрязняющее вещество» - при нажатии «Enter» появляется возможность указать код ЗВ, если код неизвестен, повторным нажатием «Enter» вызывается справочник ЗВ, из которого можно выбрать нужное ЗВ.

3.29.3. Поля «Код» и «Объем ГВС м<sup>3</sup>/с» заполняются *автоматически*.

3.29.4. Поле «Концентрация ЗВ в ГВС, С<sub>мр</sub> (мг/м<sup>3</sup>)» - вводится, если известна концентрация ЗВ, в этом случае с использованием объема выброса *автоматически* рассчитывается поле из п.3.60.5.

3.29.5. Поле «Мощность выброса М<sub>мр</sub> (г/с)» - сколько выбрасывается из «Трубы» г/с.

3.29.6. Поле «Валовый выброс, М<sub>год</sub> (т/год)» - сколько выбрасывается из «Трубы» тонн/год.

3.29.7. Поле «Коэффициент F» *автоматически* рассчитывается. «ОК» трижды - завершение работы.

## **II. Проведение расчета рассеяния, просмотр результатов.**

### **4. Расчет.**

4.1 В Главном меню выбираем пункт «**Расчет**». Появляется подменю из трех пунктов.

#### **Создание варианта расчета и расчет полей рассеяния**

#### **Расчет нормативов выбросов**

#### **Формирование Проекта нормативов ПДВ**

4.2 В подменю с помощью мыши или клавиш - «стрелок» выбираем пункт «**Создание варианта расчета и расчет полей рассеяния**». Нажимаем клавишу «Enter». Появляется страничная форма «Вариант расчета рассеяния».

4.3 Для создания нового варианта используйте кнопку панели управления «Добавить запись». Заполним поля формы.

4.4 Поле (*обязательное*) «ВЫБОР МЕСТА РАСЧЕТА (город)» - при нажатии «*Enter*» появляется древовидная форма БД «Административно-территориальное деление». Выбираем «листок», соответствующий населенному пункту, нажимаем «*Enter*». Поле заполнилось.

4.5 Кнопка (*обязательная*) «Выбор» - операция «ВЫБОР ПРЕДПРИЯТИЙ» - при нажатии «*Enter*» на кнопке появляется древовидная форма БД «Выбор промплощадок в вариант расчета». Выбираем курсором желтые «листочки» - промплощадки и помечаем нажатием клавиши «Пробел». Завершаем работу кнопкой «ОК».

4.6 Поле (*обязательное*) «Выбор режима работы предприятий» - при нажатии «*Enter*» появляется табличная форма подчиненной БД «Предприятия (промплощадки)». Заполним ее поля:

4.6.1 Поле «Предприятие (промплощадка)» заполняется *автоматически*.

4.6.2 Поле (*обязательное*) «Режим работы предприятия» - при нажатии «*Enter*» появляется меню, состоящее из режимов выбросов промплощадки, нужно указать один

4.6.3 Поле (*обязательное*) «Дата начала инвентаризации» - при нажатии «*Enter*» появляется меню, состоящее из дат инвентаризаций. «ОК» завершение работы с БД.

4.7 В поле (*обязательное*) «Имя варианта расчета» вводим любую текстовую информацию.

4.8 Кнопка (*обязательная*) «+ИЗА» в разделе «КОПИРОВАТЬ». Операция «Источники загр. атм. (ИЗА)» - при нажатии «*Enter*» на кнопке заполняется починенная БД «Источники выбросов варианта расчета» из раздела «ПРАВКА» (см.п.4.9). В нее из БД «ИЗА (инвентаризация)» (см. п.3.45) копируются данные инвентаризации ИЗА по промплощадкам, выбранным в п.4.5.

4.9 В разделе «ПРАВКА» при нажатии «*Enter*» на первом поле открывается табличная форма починенной БД «Источники выбросов варианта расчета». Информация этой БД используется в качестве исходных данных для расчета рассеяния. Эту информацию можно отфильтровать с помощью следующих полей:

4.9.1 В поле «В расчете» (выбором из меню, которое появляется при нажатии «*Enter*» на этом поле) можно проставить «нет» для тех источников выбросов, которые требуется исключить из расчета. Все



источники, помеченные таким образом, в расчете не учитываются, и выбросы из этих источников не рассматриваются.

4.9.2 В поле «Отношение к фону» можно, в случае расчета с фоном, пометить источники выбросов, вклады от которых нужно исключать, добавлять или пересчитать в фоновых концентрациях, выбрав из меню одно из значений этого поля: “исключить из фона”, “добавить к фону”, “учитывается в расчетах”.

4.10 Кнопка (*обязательная*) «+ЗВ» в разделе «КОПИРОВАТЬ» - операция «Загрязняющие вещества (ЗВ)». При нажатии «*Enter*» на этой кнопке *автоматически* заполняется Подч.БД «Загрязняющие вещества», в разделе «ПРАВКА» (см.п.4.11.), содержащая перечень ЗВ, по которым будет производиться расчет. Перечень формируется на основе ИЗА, помеченных словом «есть» в БД «Источники выбросов варианта расчета» (см.п.4.9.1.), т.е. участвующих в расчете.

4.11 В разделе «ПРАВКА» при нажатии «*Enter*» на втором поле открывается *автоматически* заполняемая табличная форма Подч.БД «Загрязняющие вещества». Чтобы исключить отдельные ЗВ из расчета, можно просто удалить по ним записи в этой БД. Отдельные ЗВ можно удалить и потом, если расчет не был прерван, и если удаляемые вещества не входят в группы суммации, наличие которых уже определено. БД «Загрязняющие вещества» имеет свои *автоматически* заполняемые починенные БД:

4.11.1 Поле «Расчетные точки» - при нажатии «*Enter*» открывается табличная форма Подч.БД «Расчетные точки», которая заполняется *автоматически только* после проведения расчета в п.4.27. БД имеет *автоматически* заполняемую Подч.БД:

- Поле «Вкладчики» - при нажатии «*Enter*» открывается табличная форма БД «Вкладчики», которая заполняется *автоматически только* при ненулевом значении поля «Количество вкладчиков» (см.п.4.25.) формы «Вариант расчета рассеяния».

4.12 Поле «учет фона» в разделе «КОПИРОВАТЬ» позволяет включить в расчет фоновые концентрации ЗВ. При нажатии «*Enter*» появляется меню для выбора одного из трех вариантов:

1) Вариант «Без учета фона» означает, что фоновые концентрации в расчетах не будут учитываться.

2) Вариант «Фон однородный» означает, что фоновые концентрации какого-либо вещества или нескольких веществ (для группы

суммации) будут учитываться, как одинаковые во всех расчетных точках и должны быть заданы только для одного поста в выбранной местности проведения расчета (при этом для каждого вещества может быть задан свой пост). Координаты постов в этом режиме игнорируются.

3) Вариант «Фон расчетный» означает, что значение фоновых концентраций в расчетной точке будет определяться по значениям фона на постах наблюдения в выбранной местности проведения расчета. Задается несколькими постами в городе. Для этого режима в форме «Метеоусловия» необходимо ввести площадь города, в пределах которой по «Уточненной схеме интерполяции и экстраполяции фона», предложенной ГГО им. А.И.Воейкова, производится расчет фоновых концентраций, измеренных на постах. В случае расчетов по группам суммации на постах наблюдения могут быть заданы концентрации веществ, входящих в группы суммации, если даже эти вещества отсутствуют в выбросах рассчитываемых источников. При этом расчет фона ведется для всех веществ, входящих в группы суммации, независимо от выбросов источников. Концентрация в каждой расчетной точке представляет собой сумму фоновой концентрации и концентрации, создаваемой выбросами источников

4.13 Кнопка «+ГС» в разделе «КОПИРОВАТЬ» - операция «Группы суммации (ГС)» - позволяет включить в расчет группы суммации. При нажатии «*Enter*» заполняется Подч.БД «Группы суммации ЗВ» в разделе «ПРАВКА». Формирование этой БД производится на основе анализа Подч.БД «Загрязняющие вещества» (см п.4.11), а также ЗВ, присутствующих в фоне, если фон учитывается (по значению поля «учет фона» из п.4.12), т.е. при формировании групп суммации программа учтет и те группы, часть ЗВ которых присутствует только в фоне. Информация этой БД используется в качестве исходных данных для расчета.

4.14 В разделе «ПРАВКА» при нажатии «*Enter*» на третьем поле открывается *автоматически* заполняемая табличная форма Подч. БД «Группы суммации (ГС)», которая аналогична БД «Загрязняющие вещества» из п.4.11, но содержит информацию по группам суммации.

4.15 В поле «Расчет по ЗВ из ГС» в разделе «Дополнительные параметры» нажатием клавиши «*Enter*» можно указать «Да», если

требуется произвести расчет отдельно по каждому ЗВ, входящему в группу суммации.

4.16 Поле (*обязательное*) «Расчетные прямоугольники» - при нажатии «*Enter*» появляется табличная форма БД «Расчетные прямоугольники». Заполним ее поля:

4.16.1 Поле (*обязательное*) «Номер прямоугольника» - любая цифра.

4.16.2 Поля «Координаты X, У центра в городской СК (м)» определяют местоположение центра расчетного прямоугольника.

4.16.3 Поля (*обязательные*) «Длина (м)» и «Ширина (м)» определяют размеры расчетного прямоугольника.

4.16.4 Поля (*обязательные*) «Шаг сетки по X, У (м)» определяют количество узлов расчетной сетки. Отношение «Длина или Ширина/шаг» должно быть целым числом. «ОК» завершение работы.

4.17 Поле «и (или) ТОЧКИ» - при нажатии «*Enter*» появляется табличная форма БД «Фиксированные точки». Заполним ее поля:

4.17.1 В поле «Индекс точки» вводим номер точки.

4.17.2 Поля «Координаты X, У точки в СК города (м)» определяют местоположение точки.

4.18 В поле «Режим расчета» (*необязательное*) можно задать выбором из списка один из четырех режимов: “Эксперт”, “Контроль”, “Экстремум”, “Пользователь”. По умолчанию задан режим “Эксперт”

4.19 Поле «Параметры режима» доступно, если в 4.18. выбран режим “Пользователь”. При нажатии на нем клавиши «*Enter*» появляется форма для выбора режимов перебора скоростей (поле “Скорость ветра”) и направлений ветра (поле “Направление ветра”), которые учитываются при расчете. Заполним поля формы:

4.20 Поле «Скорость ветра» - при нажатии «*Enter*» появляется меню «Режим указания скорости ветра», в котором делаем выбор.

4.21 Поле «Скорость ветра» «Параметры» содержит информацию о скоростях ветра, используемых в расчете. При нажатии «*Enter*» появляется страничная форма, соответствующая режиму, выбранному в п.4.20.

4.22 Поле «Направление ветра» - при нажатии «*Enter*» появляется меню «Режим указания направления ветра», в котором делаем выбор.

4.23 Поле «Направление ветра» «Параметры» содержит информацию о направлениях ветра. При нажатии «Enter» появляется страничная форма, соответствующая режиму, выбранному в п.4.22.

4.24 В поле «Учет застройки» нажатием клавиши «Enter» необходимо указать “Да” для учета влияния застройки при выполнении расчета рассеяния. При этом будет задействован модуль “Застройка”, если он подключен к комплексу (т.е. если используется комплекс “Призма с застройкой”)

4.25 В поле «Количество вкладчиков» вводят целое число, указывающее, информацию о скольких вкладчиках в порядке убывания поместить в подчиненную БД «Вкладчики» по каждой расчетной точке, контрольным точкам и точкам максимальных концентраций.

4.26 Кнопка (**обязательная**) «Проведение расчета» - нажатие кнопки приводит в действие программу расчета. Время расчета определяется соотношением из п.4.16.4, количеством веществ, участвующих в расчете и т.п. Процесс расчета можно прервать, при повторном нажатии кнопки появится запрос на продолжение расчета.

4.27 Поле «Отдельно по ЗВ» в разделе «РЕЗУЛЬТАТЫ РАСЧЕТА»- при нажатии «Enter» появляется список ЗВ, по которым проводился расчет.

4.28 В списке «Выбор загрязняющего вещества» выбираем курсором ЗВ, нажимаем «Enter». Появляется страничная форма БД «Расчетные прямоугольники».

4.29 Поля «Наименование ЗВ» и «Расчетный прямоугольник» заполняются *автоматически*.

4.30 Кнопка (*с датой*) «ПОСТРОЕНИЕ ИЗОЛИНИЙ» - при нажатии «Enter» появляется графическая картинка изолиний, работа с которой осуществляется с помощью ее собственного меню. Далее следует описание пунктов этого меню.

## Индивидуальные задания

Вариант	1	2	3	4	5	6	7	8	9	10	
Среднегодовая роза ветров, %:											
С	9										
СВ	12										
В	14										
ЮВ	12										
Ю	10										
ЮЗ	15										
З	17										
СЗ	11										
Координаты поста наблюдения:								X		Y	
								14350		13200	
Загрязняющее вещество, выбрасываемое в атмосферу (код):	163	303	303	163	302	322	303	1555	620	3620	
Загрязняющие вещества в фоне:	-	303	303	-	302	322	303	1555	-	-	
Класс опасности:	2	4	4	2	2	2	4	3	2	1	
Способ задания фоновых значений:	Константа										
Концентрация в фоне (мг/м <sup>3</sup> ):	-	0,01	0,01	-	0,02	0,01	0,01	0,01	-	-	
Привязка системы координат предприятия к городской системе:	система координат предприятия совпадает с городской										
Границы промплощадки							№	X	Y		
							1	14180	11380		
							2	14050	12250		
							3	14600	12360		
							4	14570	11500		
Границы жилой зоны:							№	X	Y		
							1	13680	12780		
							2	15140	12500		
							3	14940	12220		
							4	14880	11710		
						5	15700	11380			
Источник загрязнения атмосферы: организованный/неорганизованный	орг	орг	орг	орг	орг	орг	орг	орг	орг	орг	
Количество часов работы в сутки:	8	8	8	8	8	8	8	8	12	24	
Концентрация ЗВ в ГВС, мг/м <sup>3</sup>	0,01	0,5	0,5	0,01	0,8	0,6	0,5	0,4	0,08	0,05	
Учет фона:	Без учета	Однородный	Однородный	Без учета	Однородный	Однородный	Без учета	Однородный	Без учета	Без учета	
Тип источника:	Точечный круглый										
Высота, м:	20										
Диаметр, м:	1,0										
Температура, °С	60										
Скорость выхода ГВС	2										
Число источников выбросов	1										
Координаты источника выбросов в СК предприятия								X		Y	
								14190		12038	
Количество расчетных прямоугольников:	1										
Координаты центра прямоугольника в городской системе координат:								X		Y	
								14180		11380	

Длина, м	3000		
Ширина, м	3000		
Шаг по длине, м	300		
Шаг по ширине, м	300		
Высота, м	0		
Количество расчетных точек:	5		
Координаты расчетных точек:	№	X	Y
	1	14090	12750
	2	14660	11620
	3	14180	11080
	4	13880	11380
5	14180	12580	
Режим расчета:	Пользователь		
Скорость ветра:	перебор с шагом: Начало 0.50 Конец 7.00 Шаг 0.50		
Направление ветра:	перебор с шагом: Начало 0 Конец 360 Шаг 10		

Примечание: Принять следующие параметры расчета: площадь города 190 км<sup>2</sup>; коэффициент, зависящий от стратификации атмосферы А 180; коэффициент рельефа местности η 1; средняя температура наружного воздуха самого жаркого месяца в 13 часов дня 24 °С; средняя температура наружного воздуха наиболее холодного месяца -9 °С; скорость ветра (U\*), повторяемость превышения которой составляет 5% 9 м/с 1. СК Промплощадки – Городская СК, Координата X нуля СК промплощадки в городской СК (м) – 14180, Координата Y нуля СК промплощадки в городской СК (м) – 11380.

## **МИНОБРНАУКИ РОССИИ**

Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«Юго-Западный государственный университет»  
(ЮЗГУ)

Кафедра охраны труда и окружающей среды



### **РАСЧЕТ РАССЕИВАНИЯ ВРЕДНЫХ ПРИМЕСЕЙ С ИСПОЛЬЗОВАНИЕМ ПРОГРАММНОГО КОМПЛЕКСА «ЭКОЛОГ»**

Методические указания к проведению  
лабораторных работ по дисциплине  
«Информационные технологии в сфере безопасности»  
для студентов направления подготовки 20.04.01  
«Техносферная безопасность»

Курск 2021

УДК 699.85

Составители: И.О. Кирильчук, В.В. Протасов, А.В. Иорданова

Рецензент

Кандидат технических наук, доцент *Г.П. Тимофеев.*

**Расчет рассеивания вредных примесей с использованием программного комплекса «Эколог»:** Методические указания к проведению лабораторных работ по дисциплине «Информационные технологии в сфере безопасности» / Юго-Зап. гос. ун-т; сост.: И.О. Кирильчук, В.В. Протасов, А.В. Иорданова. Курск, 2021. 13 с.

Излагается методика использования программного комплекса «Эколог» для расчета рассеивания в атмосфере вредных выбросов от стационарных источников промышленных предприятий.

Методические указания предназначены для студентов направления подготовки 20.04.01 Техносферная безопасность.

Текст печатается в авторской редакции

Подписано в печать                      Формат 60x84 1/16.

Усл. печ. л. 0,75. Уч.-изд.л. 0,68. Тираж 30 экз. Заказ     . Бесплатно.

Юго-Западный государственный университет.

305040, г. Курск, ул. 50 лет Октября, 94.



**Цель работы:**

- изучение основных возможностей программного комплекса «Эколог»;
- проведение на ЭВМ расчета концентраций вредных веществ в атмосфере, создаваемых выбросами промышленных предприятий, с использованием ПК «Эколог».

**Общие сведения**

Унифицированная программа расчета загрязнения атмосферы (УПРЗА) «Эколог» реализует положения «Методики расчета концентраций в атмосферном воздухе вредных веществ, содержащихся в выбросах предприятий». Программа позволяет по данным об источниках выброса веществ и условиях местности рассчитывать разовые (осредненные за 20 - 30 минутный интервал) концентрации веществ в приземном слое при неблагоприятных метеорологических условиях.

Рассчитываются приземные концентрации как отдельных веществ, так и групп веществ с суммирующимся вредным действием. Суммарное количество веществ и групп суммации в одном расчете не ограничено.

В расчетах могут быть учтены нагретые и холодные выбросы точечных, линейных и площадных источников. Площадные источники могут быть трех типов:

- с выбросом со сплошной поверхности, для которых нельзя указать полного набора характеристик газовой струи: скорости и объема выходящих газов, диаметра устья источника (например, пруды-испарители, пылящие поверхности и т.п.);
- с выбросом со сплошной поверхности, для которых выброс по каждому веществу может иметь несколько (до пяти) значений в зависимости от наблюдаемой скорости ветра;
- описывающие выбросы из многих мелких точечных источников (например, печных труб в поселке);
- описывающие выбросы от автомагистралей

Общее число источников выбросов практически не ограничено.

Каждый источник выбросов может иметь несколько вариантов исходных параметров.

Учитывается влияние рельефа на рассеивание веществ (с помощью введения поправок на рельеф для источников).

Учитывается фоновая концентрация веществ, дифференцированная по скоростям и направлениям ветра и по расположению постов наблюдений за фоном. При этом программа позволяет оценить фоновое загрязнение воздуха без учета вклада отдельных источников, что упрощает расчет загрязнения воздуха для реконструируемых предприятий.

Имеется возможность построения нормативных санитарно-защитных зон (СЗЗ) предприятия, а также задания охранных и производственных зон.

Встроенный редактор позволяет занести и редактировать карту-схему предприятия и местности, на которую будут нанесены результаты расчета рассеивания.

Расчет по предприятию может иметь несколько вариантов, существует возможность проведения расчета с минимальным заданием исходных данных.

Расчеты ведутся на задаваемом пользователем множестве точек на местности, которое может включать в себя:

- узлы прямоугольных сеток в нескольких прямоугольных областях;
- отдельно заданные точки и точки, описывающие СЗЗ предприятия, границы зданий и особых зон.










Общее количество расчетных областей практически не ограничено.

Результаты расчетов - выдаются значения приземных концентраций в расчетных точках в  $\text{мг}/\text{м}^3$  или в долях ПДК. Эти значения сведены в специальные таблицы. Выдаются карты изолиний приземных концентраций вредных веществ на местности в любом задаваемом пользователем масштабе. Масштаб вывода карт также может выбираться автоматически с учетом удобства пользования картой.

Программой могут быть автоматически определены точки с максимальной концентрацией загрязняющих веществ. Программа может найти источники, дающие наибольшие вклады в загрязнение атмосферы как в целом по предприятию, так и из задаваемого пользователем множества.

Печать отчетов производится как на принтер, так и в файл. Объем отчета регулируется пользователем.

Управление программой осуществляется пользователем при помощи стандартных для Windows-программ элементов интерфейса: пунктов меню, дублирующих их кнопок и функциональных клавиш. Ниже приведена сводка наиболее распространенных из них. Под «объектом» подразумевается город, район, предприятие, вариант исходных данных или расчета, вещество и т.д. и т.п. в зависимости от контекста.

-  **Новый** (*Ins*) - добавление нового объекта.
-  **Удалить** (*Ctrl+Del*) - удаление выбранного объекта.
-  **Копировать** (*Ctrl+F2*) - создание копии объекта.
-  **Поиск** (*Ctrl+F*) - поиск объекта.
-  **Записать** отредактированные данные.
-  **Отменить** редактирование (*Esc*)
-  **Применить** (*Ctrl+S*) - подтверждение и запись изменений (без возможности отмены).
-  **Откат** (*Ctrl+Z*) - отмена последнего внесенного в справочник изменения или всех изменений (отмена становится невозможной после выполнения команды **Применить**).
-  **Заккрыть** (*Esc*) - закрытие текущего окна.

### **Порядок выполнения работы**

Работа состоит из следующих стадий:

#### ***I. Подготовка программы к работе.***

- А. Настройка интерфейса программы.
- Б. Подготовка справочника веществ.

#### ***II. Подготовка ко вводу исходных данных.***

- А. Занесение или выбор города, района, предприятия.
- Б. Создание нового варианта исходных данных.

#### ***III. Ввод исходных данных.***

#### ***IV. Проведение расчета рассеивания.***

- А. Создание нового варианта расчета.
- Б. Подготовка и проведение расчета.

#### ***V. Анализ и оформление результатов расчета.***

- А. Просмотр результатов расчета.
- Б. Построение и печать отчета о результатах расчета.

В. Графическое отображение результатов расчета (карты рассеивания).

### **Порядок работы с программой:**

1. В меню Вид установите режимы **Показывать города** и **Показывать районы**.

2. Если в дереве данных еще нет города, в котором находится ваше предприятие, откройте меню

 **Новый**

и выберите из него команду

 **Город**

3. На экране появится окно, в которое вы занесете название и номер города. В этом окне поставьте галочку напротив текста **Сразу перейти к редактированию данных** и нажмите на кнопку **Создать**.

4. На экране появится окно **Данные о городе**, в котором можно занести метеорологические и геоинформационные параметры. Снимите в этом окне галочку напротив текста **Создать новое предприятие** и нажмите на кнопку **ОК**. Установите курсор мыши на вновь созданный город.

5. Если город, в котором находится создаваемое предприятие, имеет районное деление, откройте меню

 **Новый**

и выберите из него команду

 **Район**

Если районного деления нет или нужный район уже присутствует в дереве, перейдите к п. 7.

6. На экране появится окно, в которое вы занесете название и номер района. В этом окне снимите галочку напротив текста **Сразу перейти к редактированию данных**. Нажмите на кнопку **Создать**. Установите курсор мыши на вновь созданный район.

7. Откройте меню

 **Новый**

и выберите из него команду

 **Предприятие**

8. На экране появится окно, в которое вы занесете название и номер предприятия. В этом окне поставьте галочку напротив текста **Сразу перейти к редактированию данных**. Нажмите на кнопку **Создать**.

9. На экране появится окно **Данные о предприятии**, в котором можно занести общую информацию о предприятии, метеорологические параметры местности, указать организацию, проводящую расчет. Кроме того, из этого окна при помощи соответствующих кнопок можно вызвать окно **Площадки и цеха предприятия**, редактор топоосновы, **Справочник локальных систем координат**. Занеся необходимые данные, поставьте галочку напротив текста **Создать новый вариант исходных данных** и нажмите на кнопку **ОК**.

10. На экране появится окно **Создание нового варианта исходных данных**, в которое вы занесете название и номер варианта исходных данных. В этом окне поставьте галочку напротив текста **Сразу перейти к редактированию данных**. Нажмите на кнопку **Создать**.

11. Откроется окно **Вариант исходных данных**.

Окно варианта исходных данных состоит из горизонтального меню, панели инструментов, области вывода общей информации о варианте, центральной области, состоящей из нескольких закладок, и управляющих кнопок. Внизу окна имеется строка состояния, в которой индицируется текущий режим работы с таблицей и номер текущей записи.

Центральная часть окна может содержать следующие закладки (их набор зависит от того, какие данные вы занесли): **Источники**, **Застройка**, **Санитарно-защитные зоны**, **Промышленные зоны**, **Жилые зоны**, **Варианты расчета**.

При выборе в окне **Вариант исходных данных для предприятия** закладки **Источники** появляется таблица параметров источников, состоящая из следующих граф (табл. 1):

Таблица 1 Закладка Источники

Название графы	Описание	Тип	Способ редактирования
Номер источника		Число от 1 до 9999	Ввод с клавиатуры
Наименование источника		Текст	Ввод с клавиатуры

Продолжение таблицы 1

Тип источника		Текст	Выбирается из списка
Площадка/цех	Площадка и цех, к которым относится источник	Текст	Выбирается из справочника
Расположение источника: Площадка	Номер площадки, к которой принадлежит источник.	Число	Ввод с клавиатуры. При указании номера несуществующей площадки она может быть создана.
Расположение источника: Цех	Номер цеха, к которому принадлежит источник.	Число	Ввод с клавиатуры. При указании номера несуществующего цеха он может быть создан
Высота (м)	Высота источника над уровнем земли	Число	Ввод с клавиатуры
Диаметр устья (м)	Диаметр устья точечного источника или эффективный диаметр линейного источника.	Число	Ввод с клавиатуры или расчет программой по расходу и скорости ГВС для точечного источника, автоматический расчет программой для линейного
Объем ГВС (куб. м)	Расход газовой смеси на источнике.	Число	Ввод с клавиатуры или расчет программой по диаметру устья и скорости ГВС
Скорость выхода ГВС (м/с)	Линейная скорость выхода газовой смеси из источника.	Число	Ввод с клавиатуры или расчет программой по диаметру устья и расходу ГВС

Продолжение таблицы 1

Температура ГВС (град.)		Число	Ввод с клавиатуры
Система координат	Система координат, в которой занесена координаты источника	Текст	Выбирается из списка
Локальные координаты (м): X1, Y1, X2, Y2	Координаты источника в выбранной системе координат	Число	Ввод с клавиатуры
Ширина площадного источника (м)		Число	Ввод с клавиатуры
Радиус нормативной санзоны источника (м)	Используется программой для построение санитарно-защитной зоны предприятия от источников.	Число	Ввод с клавиатуры
Коэффициент рельефа		Число	Ввод с клавиатуры
Координаты в основной системе (м): X1, Y1, X2, Y2	Координаты источника в основной системе координат	Число	Ввод с клавиатуры
Расчет Cm/Xm/Um	Указывается способ учета расчетный параметров источника	Текст	Выбор из списка
Лето/Зима: Cm/Xm/Um	Расчетные параметры источника	Числа	Ввод с клавиатуры или расчет программой
Время начала выброса (час)	Час суток, когда источник начинает работу	Число	Ввод с клавиатуры

Продолжение таблицы 1

Время окончания выброса (час)	Час суток, когда источник оканчивает работу	Число	Ввод с клавиатуры
Сезон начала работы (месяц)	Номер месяца, когда источник начинает работу	Число	Ввод с клавиатуры
Сезон окончания работы (месяц)	Номер месяца, когда источник оканчивает работу	Число	Ввод с клавиатуры
Вариант	Номер варианта параметров источника. Графа введена для совместимости со старыми версиями программы.	Число	
Угол наклона (град.)	Угол наклона направления выброса ГВС относительно вертикали. В настоящее время в расчете не учитывается.	Число	Ввод с клавиатуры
Дата занесения	Дата занесения информации об источнике.	Текст	Ввод с клавиатуры или выбор из календаря

Закладки **Застройка**, **Санитарно-защитные зоны**, **Промышленные зоны**, **Жилые зоны** появляются в окне **Вариант исходных данных для предприятия** после занесения информации о застройке в графическом редакторе (вызывается кнопкой **Графика**).

Окно **варианта исходных данных** состоит из горизонтального меню, панели инструментов, панели кнопок, при помощи которых осуществляется выбор того или иного раздела окна и центральной области, состоящей из нескольких закладок. Внизу окна имеется строка состояния, в которой индицируется текущий режим работы с таблицей и номер текущей записи.



Нажимая на ту или иную кнопку в верхней части окна, вы можете увидеть следующие разделы окна: **Источники, вещества, фон, Константы и метеопараметры, Точки, площадки, вкладчики, Расчет, Результаты.**

Для проведения расчета необходимо открыть и просмотреть каждую закладку в разделе **Источники, вещества, фон.**

Таблица 2 Закладка Вещества

Название графы	Описание	Тип	Способ редактирования
Код вещества		Числовой	Не редактируется
Название вещества		Текстовый	Не редактируется
ПДК	Значение ПДК, которое будет использоваться только в данном варианте расчета.	Числовой	Ввод с клавиатуры
Расчет вещества	Включение/выключение данного вещества в расчет/из расчета	Логический	Ввод с клавиатуры или установка мышью
Учет фона	Включение/выключение учета фоновых концентраций для данного вещества	Логический	Ввод с клавиатуры или установка мышью
Интерполяция фона	Включение/выключение интерполяции фоновых концентраций для данного вещества	Логический	Ввод с клавиатуры или установка мышью
Коэффициент экологической ситуации	Коэффициент поправки на экологическую ситуацию	Числовой	Ввод с клавиатуры

При выборе раздела **Расчет** окна **Вариант расчета** в центральной части последнего появляются элементы управления, позволяю-

щие выбрать один из установленный на вашем компьютере расчетных модулей, указать сезон года, на который должен быть проведен расчет, и осуществить запуск расчетного модуля. Во время работы расчетного модуля на экране появляется окно, в котором наглядно отображается ход расчета и его промежуточные результаты.

При выборе раздела **Результаты** окна **Вариант расчета** в центральной части последнего появляется список результатов расчета. Вы можете просмотреть результаты расчета, распечатать их на принтере или записать в файл.

### Индивидуальные задания

Вариант	1	2	3	4	5	6	7	8	9	10
Номер источника:	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$
Загрязняющее вещество, выбрасываемое в атмосферу (код):	$\frac{621}{621}$	$\frac{1071}{1071}$	$\frac{342}{342}$	$\frac{333}{333}$	$\frac{1325}{1325}$	$\frac{349}{349}$	$\frac{301}{301}$	$\frac{330}{330}$	$\frac{330}{330}$	$\frac{334}{334}$
Выброс, г/с:	$\frac{0,1}{0,09}$	$\frac{0,01}{0,02}$	$\frac{0,5}{0,3}$	$\frac{0,3}{0,2}$	$\frac{0,3}{0,15}$	$\frac{0,4}{0,5}$	$\frac{0,25}{0,3}$	$\frac{8}{7}$	$\frac{10}{5}$	$\frac{0,25}{0,2}$
Загрязняющие вещества в фоне:	$\frac{621}{621}$	$\frac{1071}{1071}$	$\frac{342}{342}$	$\frac{333}{333}$	$\frac{1325}{1325}$	$\frac{349}{349}$	$\frac{301}{301}$	$\frac{330}{330}$	$\frac{330}{330}$	$\frac{334}{334}$
Концентрация ЗВ в фоне – штиль (указана в долях ПДК):	$\frac{0,1}{0,09}$	$\frac{0,1}{0,1}$	$\frac{0,2}{0,2}$	$\frac{0,2}{0,2}$	$\frac{0,2}{0,2}$	$\frac{0,02}{0,02}$	$\frac{0,2}{0,2}$	$\frac{0,2}{0,2}$	$\frac{0,2}{0,2}$	$\frac{0,2}{0,2}$
Концентрация ЗВ в фоне – Север, Юг, Запад, Восток (указана в долях ПДК):	$\frac{0,075}{0,07}$	$\frac{0,1}{0,1}$	$\frac{0,1}{0,1}$	$\frac{0,1}{0,1}$	$\frac{0,1}{0,1}$	$\frac{0,01}{0,01}$	$\frac{0,1}{0,1}$	$\frac{0,1}{0,1}$	$\frac{0,1}{0,1}$	$\frac{0,1}{0,1}$
Класс опасности:	$\frac{3}{3}$	$\frac{2}{2}$	$\frac{2}{2}$	$\frac{2}{2}$	$\frac{2}{2}$	$\frac{2}{2}$	$\frac{2}{2}$	$\frac{2}{2}$	$\frac{2}{2}$	$\frac{2}{3}$
Величина нормативной санзоны, м:	$\frac{1000}{1000}$	$\frac{1000}{1000}$	$\frac{1000}{1000}$	$\frac{1000}{1000}$	$\frac{1000}{1000}$	$\frac{1000}{1000}$	$\frac{1000}{1000}$	$\frac{1000}{1000}$	$\frac{1000}{1000}$	$\frac{1000}{1000}$
Высота источника, м:	$\frac{20}{25}$	$\frac{30}{25}$	$\frac{25}{30}$	$\frac{30}{20}$	$\frac{15}{25}$	$\frac{15}{15}$	$\frac{30}{20}$	$\frac{25}{25}$	$\frac{20}{30}$	$\frac{25}{20}$
Диаметр устья, м:	$\frac{1}{1}$	$\frac{1}{1}$	$\frac{1}{1}$	$\frac{1}{1}$	$\frac{1}{1}$	$\frac{1}{1}$	$\frac{1}{1}$	$\frac{1}{1}$	$\frac{1}{1}$	$\frac{1}{1}$
Расход ГВС, м <sup>3</sup> /с:	$\frac{5}{5}$	$\frac{5}{5}$	$\frac{5}{5}$	$\frac{5}{5}$	$\frac{5}{5}$	$\frac{5}{5}$	$\frac{5}{5}$	$\frac{5}{5}$	$\frac{5}{5}$	$\frac{5}{5}$
Температура ГВС:	$\frac{50}{60}$	$\frac{70}{80}$	$\frac{80}{60}$	$\frac{60}{70}$	$\frac{60}{70}$	$\frac{70}{70}$	$\frac{80}{80}$	$\frac{70}{70}$	$\frac{60}{70}$	$\frac{70}{70}$

Примечание: Город: Курск. Район: Центральный. Предприятие: Предприятие 1. Отрасль промышленности: в соответствии с производством. Минимальная температура (зима): -9 град. Максимальная температура (лето): 24 град. Коэффициент стратификации, А: 180. Максимальная скорость ветра: 9 м/с. Тип документа: Инвентаризация. Площадка (наименование): в соответствии с технологией производства. Цех (наименование): в соответствии с технологией производства. Источник1 (наименование): в соответствии с технологией производства. Источник2 (наименование): в соответствии с технологией производства. Тип источника: точечный (для обоих). Площадка/цех: в соответствии с технологией производства. Система координат: Городская. Локальные координаты: 1. X=14490; Y=11752. 2. X=14090; Y=12038. Коэффициент рельефа: 1. Координаты поста наблюдения за фоном: X=14180; Y=11560. Учет поста: да. Список учитываемых веществ: указать все вещества из индивидуального задания. Коэффициент экологической ситуации: 1,8. Расчетная площадка: Тип: Автомат. Ширина: 2000 м. Шаг по ширине и длине: 100 м. Высота: 2 м. Площадь города – 190 км<sup>2</sup>. Метеопараметры – уточненный перебор.

## **МИНОБРНАУКИ РОССИИ**

Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«Юго-Западный государственный университет»  
(ЮЗГУ)

Кафедра охраны труда и окружающей среды



## **ПРОЕКТИРОВАНИЕ СТРУКТУР БАЗ ДАННЫХ ИНФОРМАЦИОННЫХ СИСТЕМ**

Методические указания к проведению  
лабораторных работ по дисциплине  
«Информационные технологии в сфере безопасности»  
для студентов направления подготовки 20.04.01  
«Техносферная безопасность»

Курск 2021

УДК 371.64/.69:004

Составители: И.О. Кирильчук, А.В. Иорданова

**Рецензент**

Кандидат технических наук, доцент *Г.П. Тимофеев.*

**Проектирование структур баз данных информационных систем:** Методические указания к проведению лабораторных работ по дисциплине «Информационные технологии в сфере безопасности» / Юго-Зап. гос. ун-т; сост.: И.О. Кирильчук, А.В. Иорданова. Курск, 2021. 27с.

Излагаются основные положения теории реляционных баз данных, и даются навыки грамотного планирования баз данных систем обработки информации.

Методические указания предназначены для студентов направления подготовки 20.04.01 Техносферная безопасность.

Текст печатается в авторской редакции

Подписано в печать                      Формат 60x84 1/16.

Усл. печ. л. 1,56. Уч.-изд. л. 1,42. Тираж 30 экз. Заказ     . Бесплатно.

Юго-Западный государственный университет.

305040, г. Курск, ул. 50 лет Октября, 94.

**Цель работы:**

– изучение основных положений теории реляционных баз данных; получение навыков грамотного планирования баз данных информационных систем.

**1 Общие положения****1.1 Структурные элементы базы данных**

Целью любой информационной системы является хранение и обработка данных о каких-либо объектах. В широком смысле слова база данных – это совокупность сведений о конкретных объектах. При создании базы данных (БД) в основном преследуется цель упорядочить данные по различным признакам, чтобы иметь возможность быстро извлекать нужную информацию. В современной технологии баз данных предполагается, что создание БД, ее поддержка, управление, а также доступ пользователей к самим данным осуществляется с помощью специальных программных продуктов – *систем управления базами данных* (СУБД).

В терминологии баз данных описываемые объекты часто называют *сущностями* (entity), а сами данные – *атрибутами* (attribute). Объект (или сущность) – это нечто, существующее в реальном мире и различимое, что имеет название и имеется способ отличать один подобный объект от другого. Сущностью может являться ВУЗ, студент, ручка, аудитория и т.д. Кроме того, помимо физического предмета сущностью могут являться и абстрактные вещи, которые, тем не менее, могут быть описаны: лекция, телевизионная передача, правовые нормы, правила поведения и т.д. Группа всех подобных объектов образует *набор объектов*. Набором объектов для объекта "сотрудник" будут являться все сотрудники, работающие на предприятии.

*Атрибут* – это некоторый показатель, характеризующий объект и принимающий для каждого конкретного объекта индивидуальное значение (текстовое, числовое и т.п.). Если в качестве объекта рассмотреть сотрудника предприятия, то этот объект будет иметь атрибуты имени, фамилии и отчества, принимающие текстовые

значения, числовой атрибут номера подразделения, в котором он работает, атрибут даты рождения, имеющий тип, описывающий дату и время, и многие другие атрибуты. Атрибут может являться также и набором объектов. Например, атрибутом работника является номер подразделения, в котором он работает. С другой стороны, подразделение является экземпляром набора подразделений и описывается набором атрибутов: начальник подразделения, количество сотрудников, адрес, специализация и т.п. В свою очередь, некоторые из атрибутов подразделения сами могут являться экземплярами набора объектов.

Помимо понятий сущность и атрибут, при работе с базами данных часто используются понятия поле, запись, таблица. *Поле* (field) – это минимальная неделимая структура организации данных. Для каждого поля базы данных разработчик должен задать следующие значения:

- уникальное в пределах отношения *имя* (name);
- *тип данных*, которые будут храниться в поле (числовой, символьный, логический и т.д.);
- максимальная *длина* (или *размер*), который могут иметь данные, хранящиеся в поле;
- дополнительные характеристики (например, для числовых данных – точность, для символьных – формат, и т.д.).

Множество логически связанных полей образуют *запись* (record). Запись является ничем, как строкой таблицы. Экземпляр записи – это отдельная реализация записи, содержащая конкретные значения ее полей. Множество экземпляров записи одной структуры образуют *таблицу* (table). При описании таблицы указывается последовательность расположения полей и их основные характеристики (имя, длина и точность). Количество записей в таблице может меняться.

## 1.2 Реляционная модель данных

*Модель данных* (data model) включает в себя структуры данных, операции их обработки и ограничения целостности. За все время существования компьютеров для обработки данных было разработано

множество различных моделей данных. Наибольшее распространение получила реляционная модель данных, характеризующаяся простотой структуры данных, удобным для пользователя табличным представлением и возможностью использования формального аппарата алгебры отношений и реляционного исчисления для обработки данных.

В 1970 г. сотрудником фирмы IBM доктором Э. Коддом было доказано, что любой набор данных может быть представлен в виде простых двумерных таблиц, называемых в математике "*отношением*" ("relation"). Разработанная Коддом теория реляционных баз данных описывает правила эффективной организации данных и управления ими.

Отметим, что для описания одних и тех же объектов в теории баз данных вообще и в реляционной теории баз данных в частности, а также при практической работе с СУБД используются разные термины. В таблице 1 приведено соответствие описания объектов в терминах разных областей.

Таблица 1

Соответствие терминов, применяемых при проектировании БД

Теория БД	Реляционные БД	СУБД
Сущность (Entity)	Отношение (Relation)	Таблица (Table)
Кортеж (Tuple)	Запись (Record)	Строка (Row)
Атрибут (Attribute)	Поле (Field)	Столбец, колонка (Column)

Итак, каждая *реляционная таблица* представляет собой *двумерный массив* и обладает следующими свойствами:

- любой элемент таблицы является минимальным элементом данных;
- элементы в пределах одного столбца имеют одинаковый тип (числовой, символьный и т.д.);
- все столбцы одной таблицы должны иметь уникальное имя;
- в таблице отсутствуют совпадающие строки;



– порядок следования строк и столбцов в таблице может быть произвольным.

Пример реляционной таблицы, содержащей номер личного дела сотрудника, его имя, фамилию, отчество, дату рождения и идентификационный номер подразделения, в котором он работает, приведен в таблице 2.

Таблица 2

Пример реляционной таблицы

№ дела	Фамилия	Имя	Отчество	Дата рожд.	№ подр.
19200	Иванов	Петр	Сергеевич	14.02.70	27
19133	Кузнецов	Владимир	Павлович	23.11.65	01
20457	Иванов	Игорь	Сергеевич	02.06.80	05
20458	Сидорова	Галина	Алексеевна	06.08.74	27
19987	Жуков	Петр	Николаевич	22.03.69	14

Как видно, каждый из атрибутов расположен в отдельном столбце, а каждая строка содержит набор атрибутов, описывающих конкретный экземпляр объекта. Одним из принципов реляционной теории является требование к минимальности элементов таблицы. Это означает, что каждое поле таблицы должно являться отдельным атомарным значением для конкретной предметной области. Под атомарным значением понимается значение, неделимое на более простые составляющие.

Однако следует отметить, что в некоторых случаях в качестве атомарного значения могут рассматриваться агрегированные объекты. Например, в одной предметной области адрес может являться атомарным значением и не разбиваться на более мелкие составляющие, тогда как в другой предметной области адрес будет представлен набором более конкретных значений: страна, район, город, улица, дом и квартира. Если в первом случае для адреса отводится единственный столбец таблицы, то во втором случае будет отведено шесть столбцов.

Доступ к конкретному элементу отношения может быть получен с указанием адреса этого элемента в формате  $A[i, j]$ , где  $A$  – элемент данных,  $i$  – номер строки,  $j$  – номер столбца отношения.

Количество атрибутов в отношении определяет *порядок* (или *степень*) этого отношения. Порядок отношения, приведенного в таблице 2, равен 6.

Множество всех значений  $A[i, j]$  при постоянном  $i$  и всех возможных  $j$  образует *кортеж* или, попросту, строку таблицы.

Как уже говорилось, реляционные таблицы связываются друг с другом. Чтобы иметь возможность связать таблицы, необходимо иметь какой-то идентификатор, который позволял бы уникально идентифицировать любую строку таблицы. Таким идентификатором является *ключ* (key). Ключом называется множество атрибутов, задание значений которых позволяет однозначно определить значения остальных атрибутов таблицы. В качестве ключа для отношения, приведенного в таблице 2, может выступать атрибут "№ дела" или набор атрибутов "Фамилия", "Имя" и "Отчество". В первом случае в качестве ключа выступает единственный атрибут – "№ дела" (номер личного дела у каждого работника свой). Во втором случае ключ является сложным и состоит из трех атрибутов. Если в качестве ключа выбрать только фамилию, то возможна ситуация, когда по одной фамилии нельзя будет однозначно идентифицировать человека – это бывает в случае, когда на предприятии работают однофамильцы или родственники. Чтобы добиться уникальности, в ключ входит еще имя и отчество. Но и здесь не всегда можно гарантировать, что ключ будет уникален. Поэтому рекомендуется в качестве ключа выбирать заведомо неповторяющиеся атрибуты, например номер паспорта.

Множество атрибутов отношения является *возможным ключом* отношения тогда и только тогда, когда выполняются два независимых от времени условия:

– **Уникальность.** В любой момент времени никакие два кортежа не имеют одинаковых значений для всех атрибутов, входящих в возможный ключ. Остальные атрибуты могут иметь произвольные значения. Например, если в качестве возможного ключа выбран номер

личного дела, то необходимо гарантировать, что в отношении никогда не появится двух личных дел с одинаковыми номерами.

– **Минимальность.** Ни один из атрибутов не может быть исключен из ключа без нарушения уникальности возможного ключа. Это означает, что в ключ разрешается вносить только минимальный набор атрибутов, который позволяет идентифицировать каждую строку отношения. То есть нет необходимости помещать в ключ и номер личного дела, и номер паспорта. При создании возможного ключа на основе атрибутов "Фамилия", "Имя" и "Отчество" исключение любого из атрибутов резко снижает уникальность ключа. Кроме того, не стоит вносить в возможный ключ повторяющийся атрибут, не увеличивающий уникальность ключа. То есть нельзя включать дополнительно к номеру личного дела еще и фамилию служащего.

Один из возможных ключей отношения может быть выбран в качестве *первичного ключа*. Остальные возможные ключи, если оно есть, принимаются за *альтернативные*.

### 1.3 Проектирование реляционных баз данных

*Реляционная база данных* – это совокупность отношений, в которых хранится вся информация баз данных. Для пользователя такая база данных представляется набором двумерных таблиц, что облегчает понимание структуры данных и управление ими. Таблицы реляционной базы данных связаны между собой *отношениями*.

**Замечание.** Не нужно путать *отношения* (relationship) между таблицами, определяющие правила связывания данных, с термином *отношение* (relation), обозначающим таблицу. Если последний используется в теории реляционных баз данных при описании сущности, то первый – при практическом проектировании баз данных. Оба термина (relationship и relation) переводятся с английского одинаково – *отношение*.

Требования к проектированию реляционных баз данных в самом общем виде можно свести к нескольким правилам. Рассмотрим их.

– Каждая таблица в базе данных имеет уникальное имя в пределах этой базы данных. Это позволяет уникально идентифицировать данные в таблице.

– Все строки в таблице однотипны, т.е. количество, набор и последовательность полей в каждой строке одной таблицы одинаковы. Кроме того, количество значений в строке также фиксировано. Иначе говоря, в каждой позиции таблицы на пересечении любых строки и столбца имеется только одно значение (в т.ч. и пустое). Множественные поля и группы недопустимы.

– Строки таблицы обязательно отличаются друг от друга хотя бы единственным значением. То есть в любой момент времени в таблице не присутствует двух одинаковых строк, что позволяет однозначно идентифицировать каждую из них.

– Каждому столбцу таблицы присваивается уникальное в пределах таблицы имя. Любой из столбцов служит для хранения данных строго определенного типа (даты, числового, символьного, денежного и т.п.). Кроме этого столбец несет еще и смысловую нагрузку. Предположим, что имеется столбец, в котором указывается фамилия сотрудника. Хотя она и используется для хранения символьных данных, все же нельзя хранить в этом столбце имя или отчество.

– При обращении к данным можно свободно обращаться к любой строке или столбцу таблицы. Сами данные не налагают никаких ограничений на последовательность обращений к ним. Кроме того, ни одна строка таблицы не зависит от значений, хранимых в другой строке.

Описание столбцов таблицы называется макетом таблицы. Простейшее представление макета является ничем иным, как обычной двумерной таблицей, в которой каждая строка соответствует столбцу описываемой таблицы, а в каждой из колонок макета указываются характеристики описываемой таблицы. В таблице 3 представлен макет сущности, приведенной в таблице 2.

Макет таблицы позволяет наглядно представить, какое количество столбцов имеется в таблице и какие данные должны храниться в

каждом столбце. Заметим, что набор колонок в макете таблицы зависит от конкретной системы, в которой создается таблица. Кроме того, даже в пределах одной системы не всегда необходимо указывать все параметры.

Таблица 3

Макет таблицы

Имя поля (Field name)	Тип данных (data type)	Размер (size)	Ключ (key)	Пустое (Null)	По умолчанию (default)	Индекс (index)
№ дела	Целое	4	Да	Нет		Да
Фамилия	Текстовый	30	Нет	Нет		Да
Имя	Текстовый	30	Нет	Да		Нет
Отчество	Текстовый	30	Нет	Да		Нет
Дата рожд.	Дата/Время	8	Нет	Да	01.01.1900	Нет
№ подр.	Целое	4	Нет	Нет	42	Нет

### 1.4 Нормализация данных

Как было уже сказано, основная цель проектирования баз данных – это сокращение избыточности хранимых данных. Грамотно спланированная база данных обеспечивает оптимальное использование оперативной и дисковой памяти, предоставляет удобный механизм изменения данных и обеспечивает высокую их целостность. В неудачно спланированной базе данных может иметь место дублирование данных. Пользователи должны следить за всеми копиями данных, т.е. при изменении одной копии необходимо исправить и другие копии. В противном случае целостность данных нарушается. Например, если сотрудники предприятия учитываются в профсоюзе, отделе кадров и в отделе, где они работают, то необходимо следить, чтобы изменения отображались во всех местах. Если сотрудница выходит замуж и меняет фамилию, а информация изменяется лишь в одном месте, то получится, что в организации работает два разных человека. Эту и многие другие проблемы при проектировании базы данных можно решить с помощью нормализации.

*Нормализация* – это процесс приведения структур данных в состояние, обеспечивающее лучшие условия выборки, включения, изменения и удаления данных. Это достигается разбиением одной большой таблицы на две более мелкие таблицы. Конечной целью нормализации является получение такого проекта базы данных, в котором *каждый факт появляется лишь в одном месте*, т.е. исключена избыточность информации. Это делается не столько с целью экономии памяти, сколько для исключения возможной противоречивости хранимых данных.

Прежде чем приступить к описанию нормализации, нужно сказать об универсальном отношении. Таблица, в которую включены все интересующие атрибуты, называется *универсальным отношением*. При использовании универсального отношения база данных будет состоять из единственной таблицы, в которой станет располагаться вся информация. Количество атрибутов в таком отношении может быть очень большим. В таблице 4 приведен пример универсального отношения, содержащего информацию об итогах сдачи сессии студентами.

Таблица 4

## Пример универсального отношения

Предмет	Преподаватель	Оценка	Фамилия	Имя	Отчество	ДатаРождения	МестоРождения	Группа
ГИС	Кирильчук	4	Иванов	Петр	Ильич	02.03.95	Курск	ТБ-11
ГИС	Кирильчук	3	Рогова	Ирина	Алексеевна	12.08.94	Льгов	ТБ-11
БЖД	Барков	4	Иванов	Петр	Ильич	02.03.95	Курск	ТБ-11
Экология	Юшин	5	Рогова	Наталья	Алексеевна	06.05.95	Льгов	ТБ-11
Экология	Юшин	4	Зуев	Игорь	Иванович	30.07.95	Курск	ТБ-21

Начинающий разработчик может использовать универсальное отношение, приведенное в таблице 4, в качестве завершённой БД. Действительно, зачем разбивать таблицу на несколько мелких, если и в

одной таблице можно с успехом хранить всю необходимую информацию. Однако применение универсального отношения связано с рядом проблем:

– **Избыточность.** Информация во многих столбцах многократно повторяется. Чем больше данных будет храниться в базе данных, тем больше информации дублируется и тем выше непроизводительные затраты.

– **Потенциальная противоречивость** (аномалии обновления). Вследствие наличия множества копий одних и тех же данных возможна ситуация, когда одна часть избыточных копий данных будет изменена, а другая – нет. Например, если для какой-либо дисциплины меняется преподаватель, то необходимо изменить информацию во всех строках, относящихся к этой дисциплине. Иначе дело будет выглядеть так, что один предмет ведут два преподавателя.

– **Аномалии включения.** В базу данных нельзя включить студента, если он не изучал ни одну из дисциплин (например, при переводе из другого ВУЗа). Однако часто бывает, что необходимо учесть все данные, которые могут потенциально использоваться.

– **Аномалии удаления.** При удалении дисциплины будет потеряна информация о преподавателе и студентах. Однако эти данные могут понадобиться в будущем, из-за чего потребуются их повторный ввод.

Большая часть проблем исчезнет, если данные из универсального отношения разнести в несколько мелких таблиц. Именно эту задачу и решает нормализация. Процесс нормализации разбивается на несколько этапов. На каждом из этапов структура данных должна удовлетворять определенным требованиям. Таблица считается *нормализованной на определенном уровне*, если она удовлетворяет требованиям, выдвигаемым соответствующей *формой нормализации* (*нормальной формы*).

При описании нормальных форм существуют несколько понятий:

– *Функциональной зависимостью* между полями  $X$  и  $Y$  называется зависимость, при которой каждому значению  $X$  в любой момент времени соответствует *единственное* значение  $Y$  из всех

возможных. Примером функциональной зависимости может служить связь города и страны, т.к. любой город находится в единственной стране и с течением времени эта связь не меняется.

– *Полной функциональной зависимостью* между составным полем  $X$  и полем  $Y$  называется зависимость, при которой поле  $Y$  зависит функционально от поля  $X$  и не зависит функционально от любого подмножества поля  $X$ .

– *Многозначная функциональная зависимость*. Поле  $X$  однозначно определяет поле  $Y$ , если для каждого значения поля  $X$  существует хорошо определенное множество соответствующих значений поля  $Y$ . Например, если рассматривать таблицу с описанием сотрудников организации и поле "Образование", то это поле имеет хорошо определенное множество допустимых значений (высшее, среднее, незаконченное среднее и т.д.).

– *Транзитивная функциональная зависимость* между полями  $X$  и  $Z$  наблюдается в том случае, если поле  $Y$  функционально зависит от поля  $X$  и поле  $Z$  функционально зависит от поля  $Y$ . В то же время не существует функциональной зависимости поля  $X$  от поля  $Y$ .

– Несколько полей *взаимно независимы*, если ни одно из них не является функционально зависимым от другого поля.

– *Неключевым* полем таблицы называется каждое поле, не входящее в состав первичного ключа.

Нормализация представляет собой последовательное изменение структуры данных и таблиц в соответствии с требованиями нормальных форм. Всего существуют шесть нормальных форм, однако на практике чаще всего применяются только три первых формы.

**Первая нормальная форма.** Считается, что таблица находится в первой нормальной форме, если ни одно поле строки не содержит более одного значения и любое ключевое поле не пусто. То есть ни один элемент таблицы не является, в свою очередь, таблицей и не содержит сложных данных. Не трудно заметить, что это требование перекликается с определением отношения, в котором говорится, что любой столбец состоит из атомарных значений, которые не могут быть разложены на более мелкие составляющие. То есть любая таблица в



реляционной базе данных автоматически находится в первой нормальной форме.

**Вторая нормальная форма.** Таблица находится во второй нормальной форме тогда и только тогда, когда она удовлетворяет требованиям первой нормальной формы и все ее поля, не входящие в первичный ключ, связаны полной функциональной зависимостью с первичным ключом. То есть необходимо, чтобы только первичный ключ однозначно идентифицировал значения в любом столбце, и в то же время значения в столбцах не зависели ни от какой части составного ключа. Если первичный ключ состоит из одного столбца, то это требование удовлетворяется автоматически. Если же первичный ключ состоит из двух и более столбцов, то таблица обязательно будет находиться во второй нормальной форме. В этом случае таблица должна быть разбита на две или более таблиц таким образом, чтобы первичный ключ однозначно идентифицировал значение в любом столбце. Бывают ситуации, когда в таблице имеется поле, не зависящее от первичного ключа. В этом случае необходимо добавить в первичный ключ дополнительный столбец.

**Третья нормальная форма.** Таблица находится в третьей нормальной форме тогда и только тогда, когда она удовлетворяет требованиям второй нормальной формы и ни одно из ее неключевых полей не зависит функционально от любого неключевого поля. Любое неключевое поле должно зависеть только от значения первичного ключа и не зависеть от любого другого неключевого атрибута. То есть каждое неключевое поле нетранзитивно зависит от первичного ключа. Для устранения транзитивной зависимости между неключевыми полями выполняется расщепление исходной таблицы. В результате часть полей удаляется из исходной таблицы и включается в состав других (возможно вновь созданных) таблиц.

Структура данных и таблиц после проведения нормализации универсального отношения "ОценкиСтудентов", приведенного в таблице 4, показана на рисунке 1.



Рисунок 1 – Нормализованное универсальное отношение

## 1.5 Связывание таблиц

После нормализации таблиц может быть получен набор из множества таблиц. Данные, принадлежащие единственной *логической записи*, могут находиться в нескольких таблицах. Когда данные хранились в универсальном отношении, то можно было легко сразу получить всю необходимую информацию. После проведения нормализации доступ к данным несколько усложнился. Чтобы выбрать ту ли иную информацию, необходимо просмотреть данные в нескольких таблицах. При этом в качестве механизма, обеспечивающего связывание данных в разных таблицах, выступают ключи.

### 1.5.1 Первичный и внешний ключи

Связывание строк таблиц реляционной базы данных выполняется с помощью *первичного* (primary) и *внешнего* (foreign) ключей. Разработчик базы данных должен определить правила связывания

данных в разных таблицах, выделив в них одну или более колонок в качестве первичного или внешнего ключа. Напомним, что первичный ключ позволяет однозначно идентифицировать любую строку таблицы. При выборе столбцов, которые будут входить в состав первичного ключа, необходимо следовать требованиям *уникальности* и *минимальности*, описанным ранее. Также необходимо учитывать, что в первичный ключ не могут входить столбцы, для которых разрешено хранение пустых значений (Null).

В качестве первичного ключа могут быть использованы как уже имеющиеся столбцы таблицы, так и новые столбцы, специально созданные для этих целей. Хотя в качестве первичного ключа можно выбрать один или более атрибутов описываемого в таблице объекта, все же на практике часто создают специально выделенный новый столбец. Использование существующих атрибутов связано с определенными сложностями. Например, на первый взгляд кажется, что номер паспорта – хороший пример первичного ключа. Однако при связывании строк в разных таблицах необходимо будет каждый раз указывать этот номер. При этом длина ключа может быть сравнительно большой. Кроме того, человек может поменять паспорт, и тогда необходимо будет изменить данные во всех связанных таблицах. Использование специальных столбцов снимает эти проблемы.

Создание первичного ключа – лишь половина дела. Следующий шаг в связывании таблиц – определение *внешнего ключа*. Внешний ключ создается в таблице, поля которой ссылаются на строки главной таблицы. Для каждой строки зависимой таблицы необходимо, чтобы значению внешнего ключа было сопоставлено значение первичного ключа. То есть нельзя вставлять в зависимую таблицу строки со значением внешнего ключа, не определенного в главной таблице. Однако допускается, что значение внешнего ключа в зависимой таблице будет не определено, т.е. внешний ключ будет хранить значение Null. Впоследствии это значение может быть изменено на корректное значение, соответствующее значению внешнего ключа в главной таблице.

В отличие от первичного ключа, внешний ключ не должен быть уникальным. То есть в зависимой таблице может существовать

множество строк, имеющих одинаковые значения для полей, сконфигурированных в качестве внешнего ключа.

После того, как первичный и внешний ключи будут связаны, на данные в зависимой таблице будут наложены ограничения на значения полей, определенных в качестве внешнего ключа. При этом возникнет необходимость как-то согласовывать изменения ключевых полей, осуществляемые в главной таблице, со значениями в зависимой таблице. Если не выполнять никаких дополнительных действий, то возможно нарушение целостности данных. Например, если в качестве первичного ключа был выбран номер паспорта, и с ним было связано несколько внешних ключей разных таблиц, то при изменении у человека номера паспорта только в главной таблице связь станет нарушенной. Строки в зависимых таблицах окажутся *потерянными*, т.е. для них не будет сопоставлена ни одна строка главной таблицы. Это и есть нарушение целостности.

Во избежание подобных проблем в СУБД реализованы специальные механизмы, обеспечивающие автоматическую поддержку целостности данных. При попытке изменения (командой UPDATE) значения первичного ключа в главной таблице СУБД может вести себя следующим образом:

– **Установление (Relation)**. Когда значение первичного ключа главной таблицы изменяется, то автоматически устанавливаются значения внешних ключей во всех связанных строках в неопределенное значение (Null). При этом теряется информация о том, с какой строкой главной таблицы были связаны строки зависимой таблицы. При изменении в главной таблице более одной строки в зависимых таблицах может образоваться несколько наборов строк с неопределенным значением внешнего ключа. Определить, какая строка зависимой таблицы с какой строкой главной таблицы была связана, станет невозможно.

– **Ограничение (Restrict)**. В этом режиме будут отвергаться изменения значения первичного ключа, если в зависимой таблице имеется хоть одна строка, связанная с изменяемой строкой главной таблицы. Изменение разрешается только в том случае, если ни в одной

зависимой таблице не имеется ни одной строки, значение внешнего ключа которой совпадает со значением изменяемого первичного ключа. В общем случае, чтобы изменить значение первичного ключа, пользователь должен сам позаботиться о предварительном изменении значений связанных внешних ключей.

– **Каскадирование (Cascading)**. Это самый удобный и гибкий режим, обеспечивающий автоматическое соблюдение целостности данных. При изменении значения первичного ключа в главной таблице СУБД будет автоматически изменять значения всех связанных внешних ключей во всех строках зависимых таблиц.

Рассмотрим также возможность удаления строк главной таблицы. Если в зависимых таблицах с первичным ключом удаляемой строки не была связана ни одна строка, то проблем нет. Если же такие строки в зависимых таблицах существуют, то необходимо выполнить удаление таким образом, чтобы обеспечить целостность данных. Возможны следующие варианты поведения СУБД при удалении строк из главной таблицы:

– **Установление (Relation)**. При удалении первичного ключа для всех связанных внешних ключей будет автоматически устанавливаться неопределенное значение. Впоследствии такие строки могут быть удалены вручную или связаны с другим первичным ключом.

– **Ограничение (Restrict)**. Перед тем, как станет возможным удаление строки в главной таблице, ни в одной зависимой таблице не должно быть строки, имеющей то же значение внешнего ключа, что и первичный ключ удаляемой строки. Пользователь обязан либо удалить такие строки из зависимой таблицы, либо установить для них значение внешнего ключа в неопределенное значение, либо связать его с любым другим первичным ключом главной таблицы.

– **Каскадирование (Cascading)**. В этом режиме система станет автоматически удалять все связанные строки из зависимых таблиц.

Необходимо заметить, что режим Cascading следует использовать осторожно, т.е. случайное удаление строки главной таблицы может повлечь за собой удаление цепочки строк в зависимых таблицах. Более безопасным методом удаления является использование режима Restrict.

### 1.5.2 Типы связей между таблицами

При связывании строк главной и зависимой таблицы возможны самые разные сочетания. Тип связи определяет количественные правила сопоставления строк главной и зависимой таблицы. Существуют следующие виды связей.

**Один-к-одному (One-to-one).** При установке отношения "один-к-одному" (1:1) каждой строке главной таблицы соответствует единственная (или ни одной) строка зависимой таблицы. С другой стороны, каждая строка зависимой таблицы должна быть связана только с одной строкой главной таблицы. Этот тип связи используют не очень часто, поскольку такие данные могут быть помещены в одну таблицу. Связь с отношением "один-к-одному" используют для разделения очень широких таблиц, для отделения части таблицы по соображениям защиты, а также для сохранения сведений, относящихся к подмножеству записей в главной таблице. Например, такой тип связей между таблицами подходит для сохранения сведений об участии студентов в спортивных мероприятиях (рисунок 2).

**Один-ко-многим (One-to-many).** При использовании связи "один-ко-многим" (1:n) каждой строке главной таблицы соответствует ноль, одна или более строк зависимой таблицы. С другой стороны, каждая строка зависимой таблицы должна быть связана только с одной строкой главной таблицы (см. рисунки 3, 4). Примером такой связи может являться связь человека с его детьми. Частным случаем связи "один-ко-многим" является связь "один-к-одному".

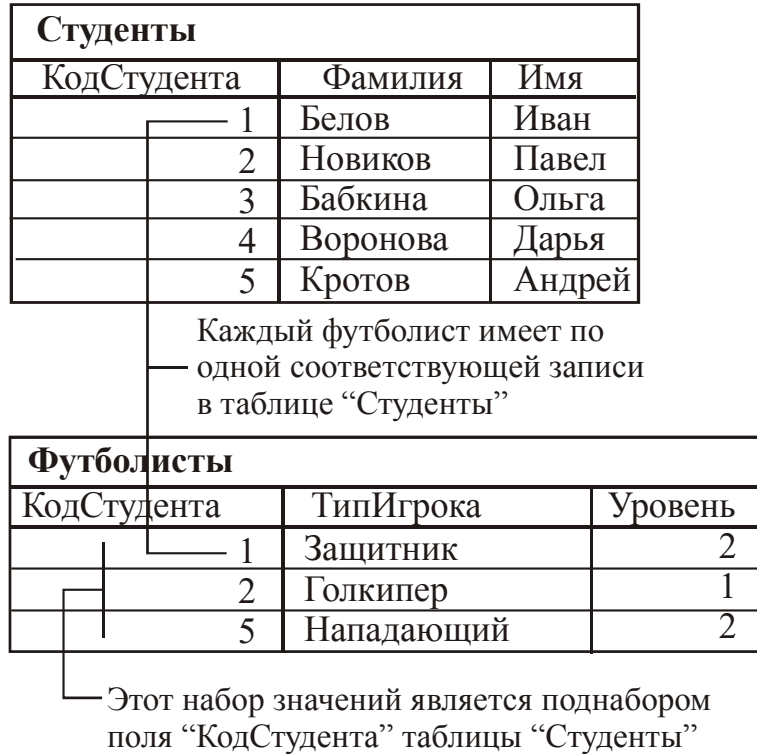


Рисунок 2 – Связь с отношением "один-к-одному"

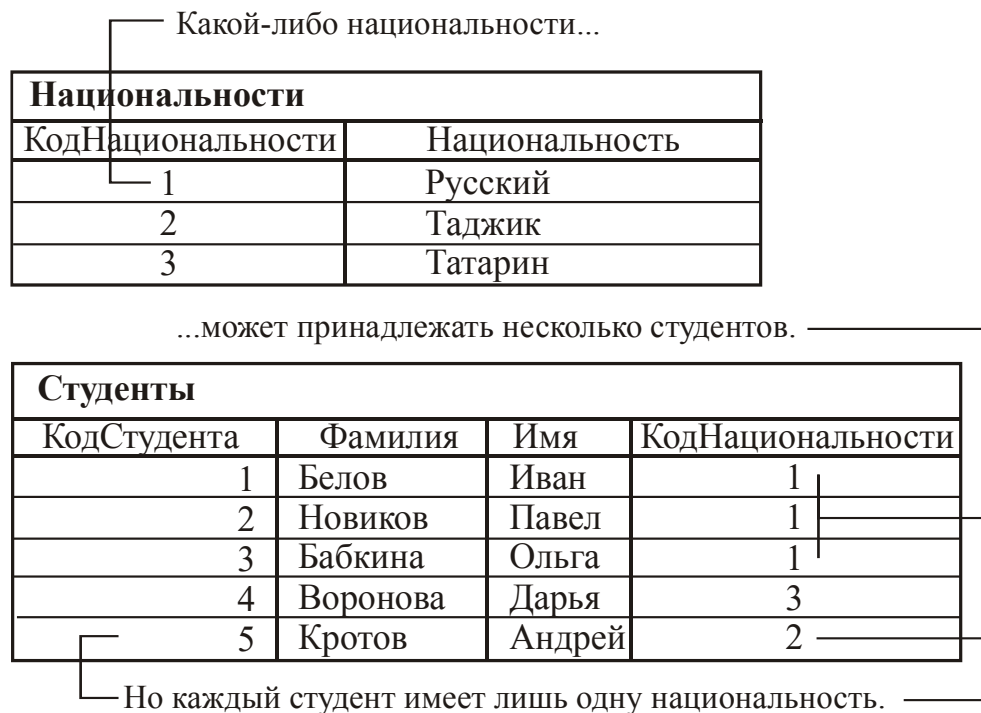


Рисунок 3 – Связь с отношением "один-ко-многим"

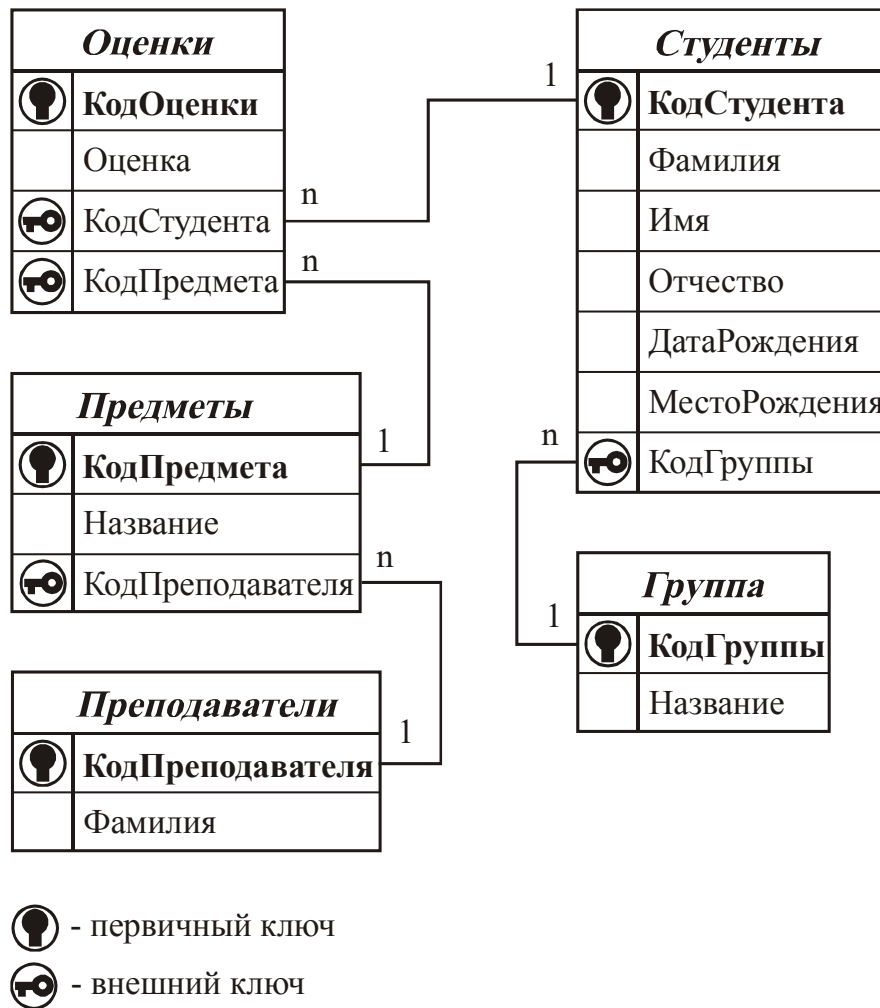


Рисунок 4 – Связи между таблицами для отношения "ОценкиСтудентов"

**Многие-ко-многим** (Many-to-Many). Этот тип связи предполагает, что любой строке главной таблицы может соответствовать ноль, одна или множество строк зависимой таблицы. При этом каждая строка зависимой таблицы может быть связана с одной или более строк главной таблицы. Примером такой связи может являться использование файлов сотрудниками.

При отношении "многие-ко-многим" одной записи в таблице А могут соответствовать несколько записей в таблице В, а одной записи в таблице В несколько записей в таблице А. Такая схема реализуется



только с помощью третьей (связующей) таблицы, ключ которой состоит из по крайней мере двух полей, которые являются полями внешнего ключа в таблицах А и В. Например, между таблицами "Предметы" и "Аудитории" имеется отношение "многие-ко-многим", которое определяется путем создания двух связей с отношением "один-ко-многим" для таблицы "Расписание" (рисунок 5).

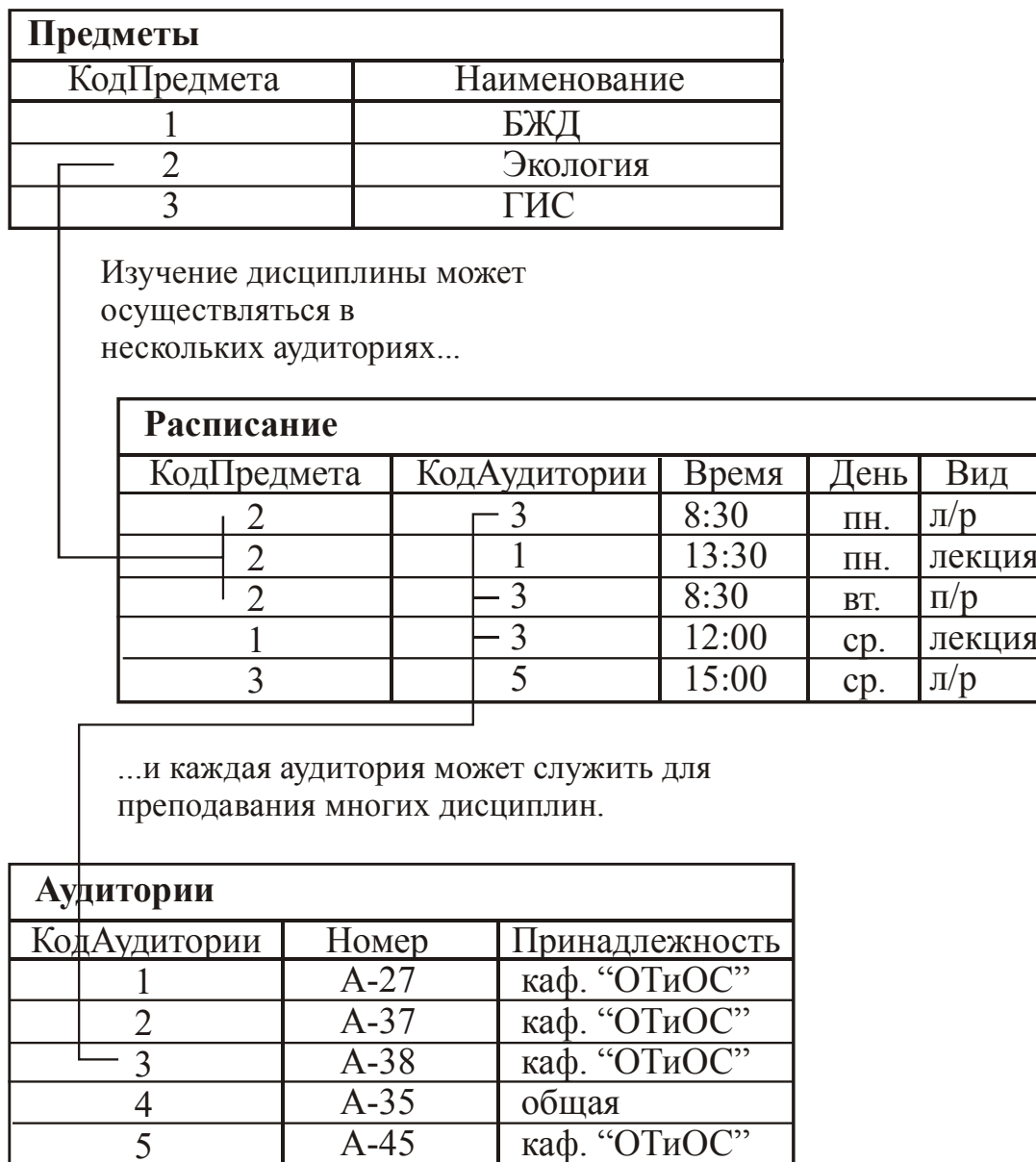


Рисунок 5 – Связь с отношением "многие-ко-многим"

### 1.5.3 Связывание таблиц в СУБД MS ACCESS

Для определения связей между таблицами необходимо выполнить следующие действия:

1. Закройте все открытые таблицы. Создавать или изменять связи между открытыми таблицами нельзя.

2. Переключитесь в окно базы данных.

3. Нажмите кнопку **Схема данных** на панели инструментов.

4. Если в базе данных не определено никаких связей, то на экран автоматически будет выведено окно **Добавление таблицы**. Если необходимо добавить таблицы до определения связей, а диалогового окна **Добавление таблицы** на экране нет, нажмите кнопку **Добавить таблицу** на панели инструментов. Если же таблицы, которые необходимо связать, отображены на экране, перейдите к шагу 6.

5. Дважды щелкните на именах таблиц, для которых требуется определить связи. Затем закройте диалоговое окно **Добавление таблицы**.

6. Для связывания полей выберите поле в одной таблице и перетащите его на соответствующее поле во второй таблице (для связывания сразу нескольких полей переместите их при нажатой клавише CTRL).

В большинстве случаев связывают поле первичного ключа (представленное в списке полей полужирным шрифтом) одной таблицы с соответствующим ему полем (часто имеющим то же имя) внешнего ключа во второй таблице (рисунок 6). Связанные поля не обязательно должны иметь одинаковые имена, но они должны иметь одинаковые типы данных и иметь содержимое одного типа. Кроме того, связываемые поля числового типа должны иметь одинаковые значения свойства **Размер поля (FieldSize)**. Необходимо заметить, что поле счетчика можно связывать с числовым полем, если в последнем в свойстве **Размер поля** задано значение "Длинное целое".

7. В диалоговом окне **Схема данных**, отображенном на экране, проверьте имена полей, представленные в двух колонках. Если необходимо, установите параметры связи (рисунок 7).

8. Для создания связи нажмите кнопку **Создать**.

9. Для каждой пары таблиц, которые необходимо связать, выполните шаги 5-8.

При закрытии окна схемы данных на экран будет выведено сообщение, нужно ли сохранять макет. Не зависимо от того, будет он сохранен или нет, связи, созданные в базе данных, будут сохранены.

**Примечание.** Для связывания таблицы самой с собой или для связывания поля таблицы с другим полем той же таблицы следует дважды добавить таблицу. Такая ситуация возникает при определении поля с подстановкой значений из той же таблицы.

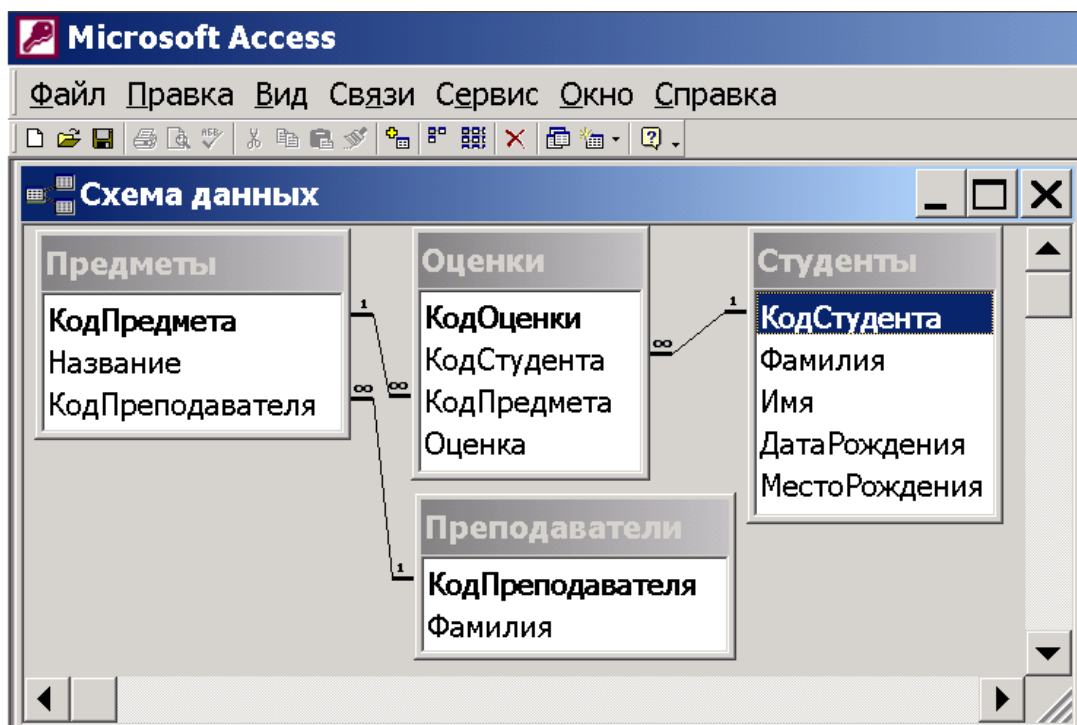


Рисунок 6 – Схема данных

**Изменение связей** [?] [X]

Таблица/запрос:  Связанная таблица/запрос:

КодПредмета	КодПредмета

Обеспечение целостности данных  
 каскадное обновление связанных полей  
 каскадное удаление связанных записей

Тип отношения:

Рисунок 7 – Установка параметров связи

## 2 Порядок выполнения работы

Согласно выданному преподавателем варианту:

1. Определить предварительную структуру базы данных, провести ее нормализацию до третьей нормальной формы и составить макеты таблиц.

2. Определить правила связывания данных в разных таблицах, для чего выделить поля, которые будут выступать в качестве первичных и внешних ключей. Задать типы связей между таблицами.

Проектирование структуры базы данных осуществляется в соответствии с приведенной ниже информацией.

**Вариант №1** – база данных по учету товаров в продуктовом магазине. Каждый товар имеет название и относится к определенному типу (кондитерские, мясные, рыбные и т.п. изделия). Товары одного названия могут присутствовать на складе по разной цене и в любом количестве. Товары поставляются несколькими поставщиками из разных городов. На склад товары поступают группой, при этом

учитывается день поставки. Товары, входящие в группу, могут отличаться по типу, наименованию, цене и количеству.

**Вариант №2** – база данных по учету результатов игр в КВН. В базе должна учитываться информация о типе игры (финал, полуфинал и т.д.), названии конкурсов, а также храниться фамилии судей, названия команд и город, откуда они приехали. При проведении игры в базу заносится день, в который она прошла, имена участвовавших команд, названия конкурсов, в которых они участвовали, полученные за конкурс оценки и фамилии поставивших их судей.

**Вариант №3** – база данных по учету вредных выбросов промышленных предприятий. В базе хранятся названия предприятий, названия городских округов, список вредных веществ и величины соответствующих ПДК, наименования систем человека (нервная, сердечно-сосудистая и др.). Учтите, что на какую-либо систему отрицательное воздействие могут оказывать несколько вредных веществ, и, в то же время, одно вещество может влиять на несколько систем человека. Данные о выбросах для каждого предприятия заносятся в базу ежемесячно, при этом указывается наименование вещества и величина соответствующего выброса.

**Вариант №4** – база данных по учету книг в библиотеке. В базе хранится информация по тематике книг (математика, физика, история, поэзия и т.д.), виду издания (монография, учебник, справочник), о фамилиях авторов. Принять, что автор у книги единственный, однако один автор может написать несколько различных книг. Каждому названию книги соответствует некоторый автор, тематика и вид издания, год выпуска, общее количество экземпляров, количество экземпляров, находящихся в фонде (остальные "на руках"), число читавших данную книгу.

**Вариант №5** – база данных по учету больных в поликлинике. В базе хранится информация о поле посетителей ("мужской", "женский"), обслуживаемых поликлиникой улицах, данные о человеке. Данные о человеке слагаются из фамилии, имени, года рождения, пола, названия улицы и номера дома. Должен присутствовать список специалистов (окулист, хирург, терапевт и т.п.). При учете фиксируются данные о

больном, месяц и год посещения, к какому специалисту было обращение.

### **Содержание отчета:**

1. Наименование работы.
2. Цель работы.
3. Структура разработанной базы данных, представленная посредством схемы базы данных и макетов таблиц.

### **Контрольные вопросы:**

1. Дайте определение понятий "сущность", "атрибут".
2. Перечислите требования, которые должны быть соблюдены при проектировании базы данных.
3. Из каких соображений выбирается ключевое поле таблицы?
4. С какой целью проводится процесс нормализации?
5. В чем заключается сущность нормальных форм?
6. Какими механизмами поддерживается целостность данных?
7. Какие существуют типы связей между таблицами?

## **МИНОБРНАУКИ РОССИИ**

Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«Юго-Западный государственный университет»  
(ЮЗГУ)

Кафедра охраны труда и окружающей среды



## **РЕАЛИЗАЦИЯ СТРУКТУР БАЗ ДАННЫХ В СИСТЕМАХ УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ**

Методические указания к проведению  
лабораторных работ по дисциплине  
«Информационные технологии в сфере безопасности»  
для студентов направления подготовки 20.04.01  
«Техносферная безопасность»

Курск 2021

УДК 699.85

Составители: И.О. Кирильчук, А.В. Иорданова

**Рецензент**

Кандидат технических наук, доцент *Г.П. Тимофеев.*

**Реализация структур баз данных в системах управления базами данных:** методические указания к проведению лабораторных работ по дисциплине «Информационные технологии в сфере безопасности» / Юго-Зап. гос. ун-т; сост.: И.О. Кирильчук, А.В. Иорданова. Курск, 2021. 19 с.

Излагается информация по созданию баз данных в геоинформационной системе MapInfo и системе управления базами данных Microsoft Access.

Методические указания предназначены для студентов направления подготовки 20.04.01 Техносферная безопасность.

Текст печатается в авторской редакции

Подписано в печать                      Формат 60x84 1/16.

Усл. печ. л. 1,1. Уч.-изд. л. 1,0. Тираж 30 экз. Заказ     . Бесплатно.

Юго-Западный государственный университет.

305040, г. Курск, ул. 50 лет Октября, 94.



**Цель работы:**

- получение навыков реализации структур баз данных в системе управления базами данных Microsoft Access и геоинформационной системе MapInfo.

**Общие положения****1. Создание нового файла базы данных****1.1. Microsoft Access**

Для создания нового файла базы данных необходимо запустить программу Microsoft Access и в разделе **Создание базы данных** (Create a new database using) диалогового окна Microsoft Access выбрать переключатель **Новая база данных** (Blank Access database).

В диалоговом окне **Файл новой базы данных** (File new database) укажите папку, в которой будет храниться создаваемая база данных, имя файла и нажмите кнопку **Создать** (Create). После этого в главном окне Microsoft Access появится окно новой базы данных.

**1.2. MapInfo**

В отличие от Microsoft Access, в котором все объекты (таблицы, запросы, индексы и т.д.) хранятся в единственном файле сложной структуры, в ГИС MapInfo хранение данных организовано по-другому.

В MapInfo картографическая информация выводится посредством наложения друг на друга отдельных слоев, каждый из которых описывает положение в пространстве и характеристики отдельного класса географических объектов. Например, итоговая карта города может слагаться из слоев, описывающих положение улиц, водных объектов, лесных массивов, железнодорожных путей, промышленных предприятий, школ и т.д. Каждому слою *логически* соответствует таблица с определенным именем, которая *физически* хранится в виде набора файлов с одинаковыми именами (совпадающими с именем логической таблицы), но разными расширениями. Например, слою, описывающему положение улиц города, может соответствовать таблица "улицы", хранящаяся на жестком диске компьютера

посредством файлов "улицы.tab", "улицы.id", "улицы.ind", "улицы.dat" и "улицы.map"). Главным из этих файлов является файл с расширением ".tab", который хранит настройки слоя и связывает в единую структуру все остальные файлы. Необходимо заметить, что как раз список файлов с расширением ".tab" появляется при открытии таблицы через меню **Файл/Открыть таблицу**.

Слои, определяющие вид итоговой карты, образуют *Рабочий Набор*. В *Рабочем Наборе* хранятся настройки, характеризующие множество слоев в целом. Таким образом, приблизительно *Рабочий Набор* представляет собой аналог файла базы данных Microsoft Access. Создать прямо *Рабочий Набор* нельзя, необходимо сначала открыть (или создать новые) таблицы слоев карты, а затем через меню **Файл/Сохранить Рабочий Набор** записать настройки карты в файл с расширением ".wor".

Для удобства работы с отдельными картами, принято помещать *Рабочий Набор* карты и все файлы, относящиеся к ней, в отдельную папку.

## 2. Создание таблиц

### 2.1. Microsoft Access

После создания нового файла базы данных выберите группу **Таблицы** (Tables). В появившемся списке присутствуют три ярлыка, соответствующие трем разным способам создания таблиц: путем ввода данных, с помощью *Конструктора таблиц* и с помощью *Мастера создания таблиц*. Рассмотрим второй способ.

Для открытия окна *Конструктора таблиц* необходимо дважды щелкнуть левой кнопкой мыши на ярлыке **Создание таблицы с помощью конструктора** (Create table in Design View).

В режиме *Конструктора таблицы* создаются путем задания имен полей, их типов и свойств посредством следующих действий (рисунок 1):

1. В столбец **Имя поля** (Field Name) вводятся имена полей создаваемой таблицы.

2. В столбце **Тип данных** (Data Type) для каждого поля выбирается из раскрывающегося списка тип данных, которые будут содержаться в поле.
3. В столбце **Описание** (Description) можно ввести описание данного поля (необязательно).
4. В нижней части *Конструктора таблиц* на вкладках **Общие** (General) и **Подстановка** (Lookup) вводятся свойства каждого поля или оставляются значения свойств по умолчанию.
5. Указываются ключевые поля, для чего выделяется соответствующая строка таблицы и нажимается кнопка **Ключевое поле** (Primary Key) на панели инструментов **Конструктор таблиц** (Table Design). Панель инструментов обычно расположена под основным меню.
6. После описания всех полей таблицы нажимается кнопка **Закреть** (Close) и в диалоговом окне **Сохранить как** (Save As) вводится имя создаваемой таблицы.

Для описания таблиц можно использовать как символы английского, так и русского алфавитов. При этом следует придерживаться правил именования пользовательских объектов Access:

- Имена полей в таблице не должны повторяться, т.е. должны быть уникальными.
- Имена могут содержать не более 64 символов, включая пробелы.
- Желательно избегать употребления имен, совпадающих с именами встроенных функций или свойств Microsoft Access (например, Name).
- Имя поля не должно начинаться с пробела или управляющего символа (коды ASCII 00-31).
- Имена полей могут содержать любые символы, включая буквы, цифры, пробелы, специальные символы, за исключением точки (.), восклицательного знака (!), надстрочного символа (^) и прямых скобок ([ ]).

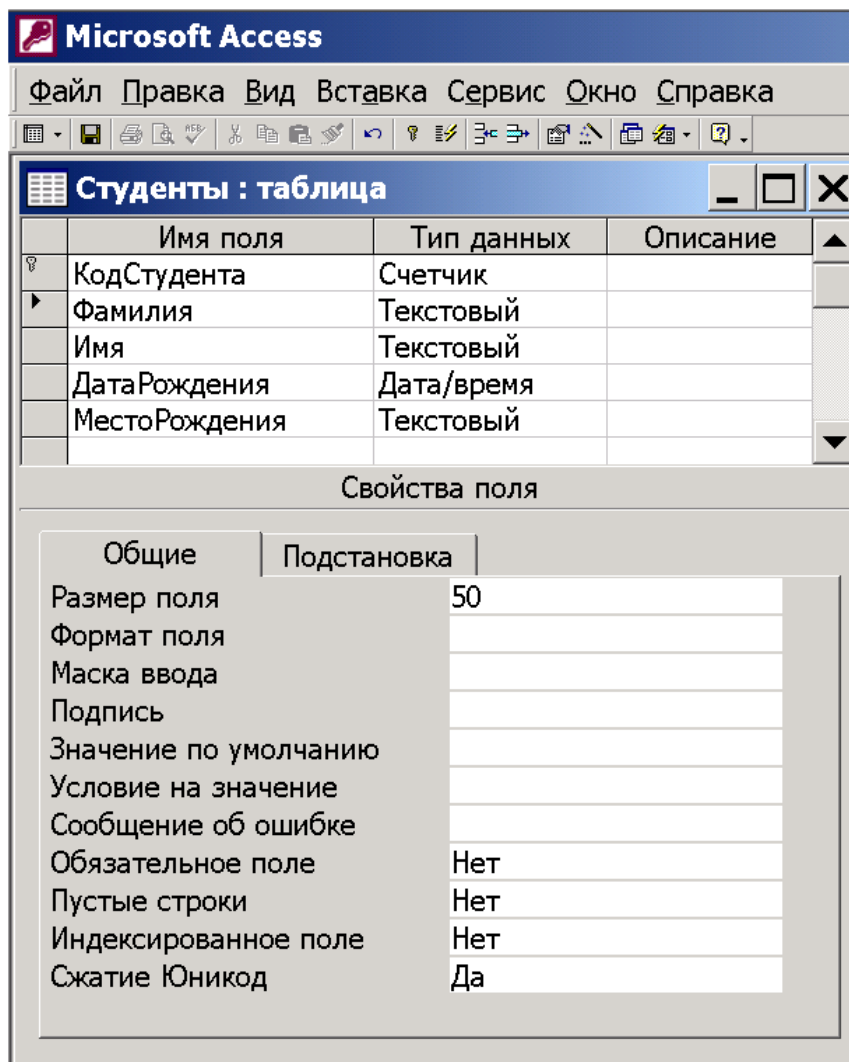


Рисунок 1 – Окно конструктора таблицы

Хотя при именовании полей допустимо использование пробелов и отличных от букв и цифр символов, рекомендуется этого избегать. Это обусловлено тем, что если хранящиеся в базе данные будут в дальнейшем обрабатываться в СУБД, отличной от Microsoft Access (например, MapInfo), то велика вероятность возникновения ошибок при подключении базы данных. Поэтому на практике часто используется следующий стиль написания имен полей: все слова пишутся слитно, начиная каждое с заглавной буквы (например, КодСтудента, ГодРождения, СемейноеПоложение).

## 2.2. MapInfo

Для создания новой таблицы необходимо щелкнуть левой кнопкой мыши на пункте меню **Файл/Новая таблица**, что вызовет появление на экране диалога **Новая таблица** (рисунок 2). Это диалог позволяет выбрать, в каком окне будет показана новая таблица после ее создания: в окне *Списка*, в новом окне *Карты* или в виде нового слоя в уже открытом окне *Карты*:

- **Показать списком.** Если флажок выбран, то MapInfo откроет пустое окно *Списка*, когда формирование новой таблицы и задание для нее имени будет закончено.
- **Показать Картой.** Если флажок выбран, то MapInfo откроет окно *Карты*, когда формирование новой таблицы и задание для нее имени будет закончено (необходимо задать проекцию таблицы в диалоге "Создать структуру таблицы").
- **Добавить к Карте.** Если флажок выбран, то MapInfo добавляет слой в активное окно *Карты*. Таблица должна иметь проекцию карты.

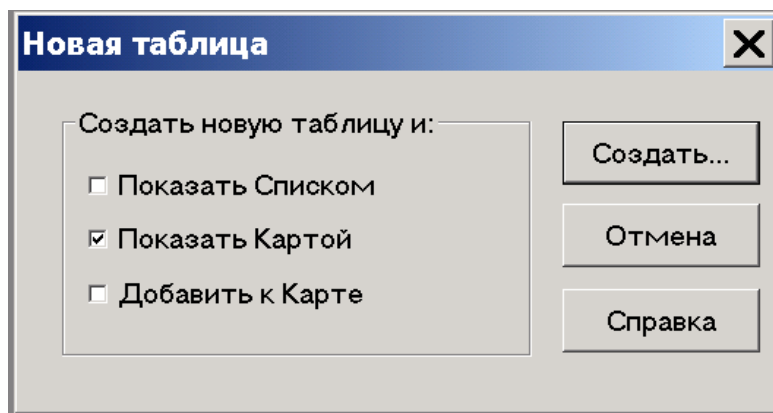


Рисунок 2 – Создание новой таблицы

После отметки нужных опций нажимается кнопка **Создать**, вызывающая появление диалога **Создать структуру таблицы** (рисунок 3).

В этом диалоге создается структура таблицы, позволяющая иметь графические объекты. Такая структура позволяет отображать таблицу в виде карты или плана. Каждому объекту в таблице соответствует запись с табличными (неграфическими данными). Данные таблицы,

имеющей структуру без графики, могут быть показаны только в окне *Списка*.

**Замечание:** Для использования команды **Запрос/Найти** таблица должна иметь индексированные поля и возможность иметь графические объекты.

Вверху диалога расположено окошко со списком, верхняя строчка которого соответствует первому полю таблицы, вторая строчка - второму полю и так далее. Для изменения поля, необходимо выбрать соответствующую строчку в списке.

Поля	Тип	Индекс
Поле1	Символьное(100)	<input checked="" type="checkbox"/>
Улица	Символьное(20)	<input checked="" type="checkbox"/>
Дом	Короткое целое	<input type="checkbox"/>
Квартира	Целое	<input type="checkbox"/>
Площадь	Вещественное	<input type="checkbox"/>
Высота	Десятичное(20.3)	<input type="checkbox"/>
Год	Дата	<input type="checkbox"/>
<b>Жилой</b>	<b>Логическое</b>	<input type="checkbox"/>

Можно присоединять географические объекты

Описание поля

Имя:

Тип:

Buttons: Вверх, Вниз, Добавить поле, Удалить поле, Проекция..., Create..., Отмена, Справка

Рисунок 3 – Определение структуры таблицы

Таблица должна иметь хотя бы одно поле. Если при создании таблицы заранее неизвестно, какие поля она должна иметь, то создается одно поле, а следующие добавляются уже потом.

Диалог **Создать структуру таблицы** состоит из следующих элементов:

- **Поля.** Слово расположено над окошком со списком полей и является заголовком колонки с именами полей (колонок) таблицы.
- **Тип.** Слово расположено над окошком со списком полей и является заголовком колонки с типами, назначенными полям таблицы. Некоторые типы включают в скобках длину поля (если поле фиксированной длины).
- **Индекс.** Слово расположено над окошком со списком полей и является заголовком колонки, в которой в строке индексированного поля ставится галочка. Если поле не индексировано, то в этой колонке будет пустое место. Индексировать можно любое количество столько полей. Индексирование не изменяет порядок записей в таблице. Индексирование необходимо для:
  - выполнения команды **Запрос/Найти** (поиск осуществляется только по индексированным колонкам);
  - ускорения выполнения запросов с использованием **Числового** или **алфавитного** сравнения;
  - ускорение процесса объединения.
- **Вверх, Вниз.** Изменяют порядок полей в таблице. Выберите поле в списке и нажмите на кнопку **Вверх**, если хотите уменьшить его порядковый номер, или на кнопку **Вниз**, если хотите увеличить. Соответственно меняется структура таблицы. От порядка расположения полей таблицы напрямую зависит расположение колонок в открываемом окне *Списка*. Первое поле соответствует самой левой колонке *Списка*, второе поле - второй колонке от левого края и т. д.
- **Добавить поле.** Добавляет в конец списка новое поле. Новому полю присваивается стандартное имя Поле1, Поле2, Поле3 и т.

- д. Номер зависит от порядкового номера, который присваивается новому полю при создании.
- **Удалить поле.** Удаляет поле из таблицы.
  - **Сопоставить таблице графические объекты.** Если флажок установить, то структура таблицы разрешает создание в таблице графических объектов.
  - **Проекция.** Вызывает диалог **Выбор проекции** для изменения проекции таблицы.
  - **Описание поля.** В этой части диалога производится настройка поля, выбранного в верхнем списке. Изменение в описанных ниже окошках отражается в колонках списка полей.
  - **Имя.** Введите имя поля. При создании поля ему присваивается стандартное имя Поле1, Поле2, Поле3 и т. д. Вы можете изменить имя поля на любое не более 31 символа длиной. Для имени могут использоваться буквы, цифры и символ подчеркивания. Пробелы не используются, в место них рекомендуем между словами использовать символ подчеркивания (\_). В именах можно использовать заглавные и строчные буквы, но следует помнить, что MapInfo их не различает.
  - **Тип.** Список содержит все типы данных, которые может содержать поле таблицы. Выберите один из них.
  - **Знаков.** Для символьного или десятичного типа. Максимальное значение для символьного типа – 250 знаков, а для десятичного – 19.
  - **Знаков после запятой.** Задайте число знаков для десятичного типа.
  - **Индекс.** Установите флажок для создания индекса для выбранного поля.
  - После нажатия кнопки **Создать (Create)** создается таблица с заданной структурой. Для задания имени файлов таблицы, а также диска и каталога будет открыт диалог **Создать новую таблицу** (рисунок 4). Новая таблица показывается или не



показывается в одном из окон MapInfo в соответствии с установкой в диалоге **Новая таблица**.

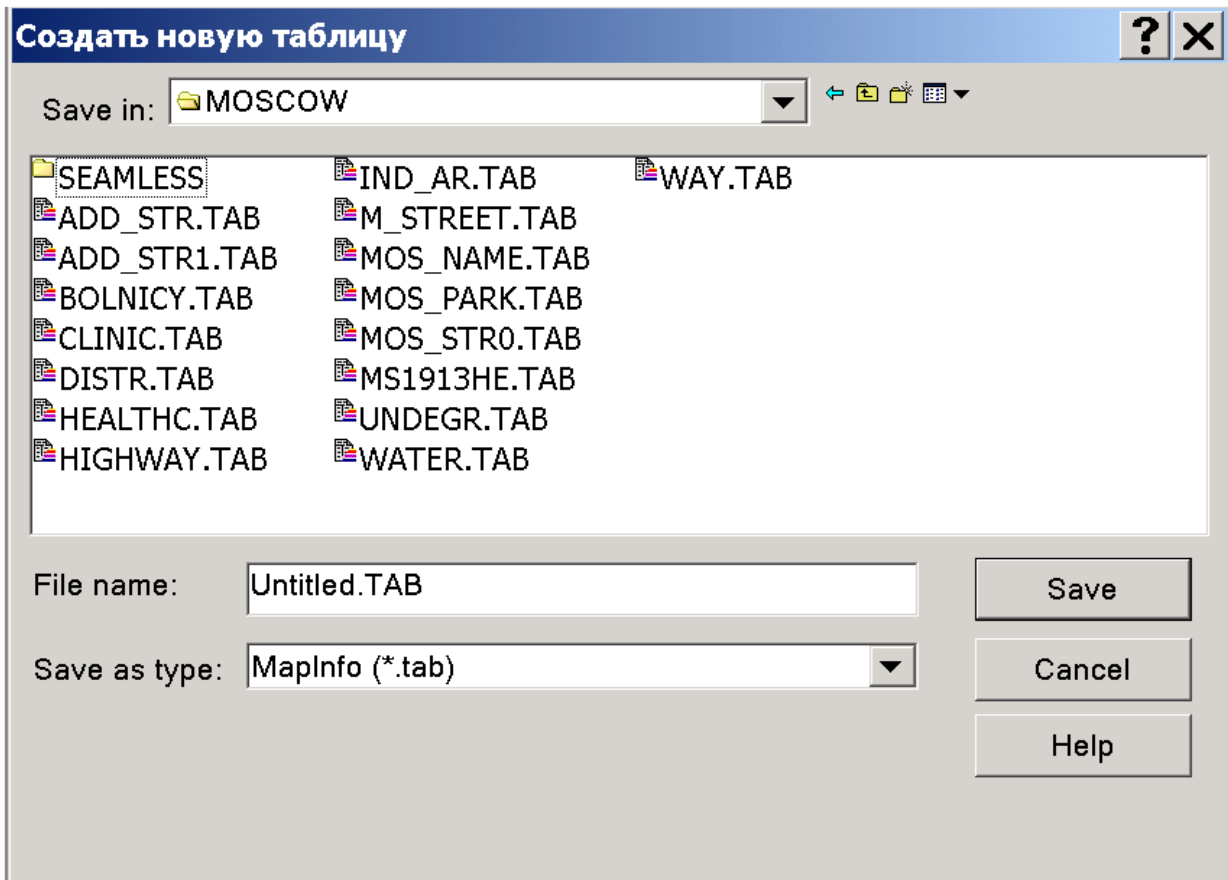


Рисунок 4 – Диалог сохранения таблицы

### 3. Задание типа данных

#### 3.1. Microsoft Access

Тип данных определяет вид информации, которая будет храниться в поле, и выбирается в столбце **Тип данных** (Data Type) *Конструктора таблиц* (рисунок 5). Описание типов данных приведено в таблице 1.

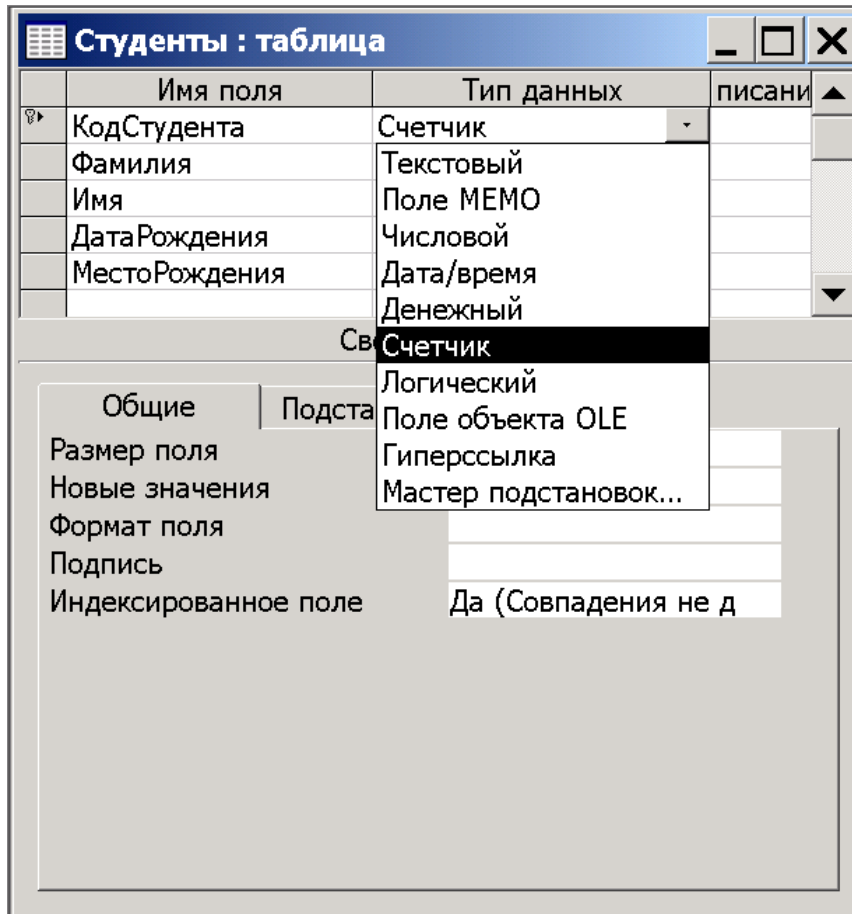


Рисунок 5 – Выбор типа данных поля таблицы

После определения типа данных, которые будут храниться в поле таблицы, в нижней части *Конструктора таблиц* на вкладке **Общие** (General) вводятся свойства каждого поля или оставляются значения свойств по умолчанию. Набор свойств поля зависит от выбранного типа данных. В таблице 2 дано краткое описание свойств полей.

Таблица 1

## Типы данных Microsoft Access

Тип данных	Описание
<b>Текстовый</b> (Text)	Символьные или числовые данные, не требующие вычислений. Поле данного типа может содержать до 255 ( $2^8-1$ ) символов. Размер текстового поля задается с

Тип данных	Описание
	помощью свойства <b>Размер Поля</b> (FieldSize). В нем указывается максимальное количество символов, которые могут быть введены в данное поле.
<b>Поле МЕМО</b> (Memo)	Поле МЕМО предназначено для ввода текстовой информации, по объему превышающей 255 символов. Это поле может содержать до 65535 ( $2^{16}-1$ ) символов. Этот тип данных отличается от типа <b>Текстовый</b> (Text) тем, что в таблице хранятся не сами данные, а ссылки на блоки данных, которые хранятся отдельно. За счет этого ускоряется обработка таблиц (сортировка, поиск и т.п.). Поле типа МЕМО не может быть ключевым или проиндексированным.
<b>Числовой</b> (Number)	Числовой тип используется для хранения числовых данных, используемых в математических расчетах. Имеет много подтипов. От выбора подтипа (размера) данных числового типа зависит точность вычислений. Для установки подтипа числовых данных служит свойство <b>Размер Поля</b> (FieldSize). Обычно по умолчанию используется подтип <b>Длинное целое</b> (Long Integer), которое занимает 4 байта и представляет собой число в пределах от $-2^{31}$ до $2^{31}-1$ .
<b>Дата/Время</b> (Data/Time)	Тип для представления даты и времени. Позволяет вводить даты с 100 по 9999 год. Размер поля 8 байт. Даты и время хранятся в специальном фиксированном числовом формате. Дата является целой частью значения поля типа <b>Дата/Время</b> , а время его дробной частью.
<b>Денежный</b> (Currency)	Тип данных, предназначенный для хранения данных, точность представления которых колеблется от 1 до 4 знаков после десятичной точки. Целая часть данного типа может содержать до 15 десятичных знаков.
<b>Счетчик</b>	Поле содержит 4-байтный уникальный номер,

Тип данных	Описание
(AutoNumber)	определяемый Microsoft Access для каждой новой записи автоматически путем увеличения предыдущего значения на 1 или случайным образом. Значения полей типа счетчика обновлять (вносить изменения) нельзя. Максимальное число записей в таблице с полем счетчика не должно превышать 2 миллиарда ( $2^{31}$ ).
<b>Логический</b> (Yes/No)	Логическое поле, которое может содержать только два значения, интерпретируемых как <b>Да/Нет</b> , <b>Истина/Ложь</b> , <b>Включено/Выключено</b> . Access использует величину $-1$ для представления значения <b>Да</b> и величину $0$ для <b>Нет</b> . Поля логического типа не могут быть ключевыми, но их можно индексировать.
<b>Поле объекта OLE</b> (OLE object)	Содержит ссылку на OLE-объект (лист Microsoft Excel, документы Microsoft Word, MathCAD, звук, изображение и т.п.). Объем объекта ограничивается имеющимся в наличии дисковым пространством.

Таблица 2

## Описание свойств полей таблицы

Свойство	Описание
<b>Размер поля</b> (FieldSize)	Определяет максимальную длину текстового или числового поля (учтите, что при создании полей слишком большого размера неэкономно расходуется память и замедляется работа СУБД, а полей маленького размера – искажается содержимое поля)
<b>Формат поля</b> (Format)	Устанавливает формат отображения данных в форме и запросе
<b>Число десятичных знаков</b> (DecimalPlaces)	Определяет количество разрядов в дробной части десятичного числа
<b>Маска ввода</b>	Определяет маску данных при вводе

Свойство	Описание
(InputMask)	
<b>Подпись</b> (Caption)	Содержит надпись, которая выводится рядом с полем в форме или отчете
<b>Значение по умолчанию</b> (DefaultValue)	Содержит значение, устанавливаемое в соответствующем поле таблицы по умолчанию при создании новой записи (например, для таблицы на рис.5 значение по умолчанию для поля <i>МестоРождения</i> можно задать как "Курск")
<b>Условие на значение</b> (ValidationRule)	Определяет множество значений, которые пользователь может вводить в это поле при заполнении таблицы (для поля <i>ДатаРождения</i> на рис.5 вполне можно задать условие: >1.1.1970)
<b>Сообщение об ошибке</b> (ValidationText)	Определяет сообщение, которое появляется на экране при вводе недопустимого значения
<b>Обязательное поле</b> (Required)	Данная опция определяет, обязательно ли нужно при добавлении новой записи сразу же заносить в поле какое-либо значение, или можно ли это сделать позже
<b>Пустые строки</b> (AllowZeroLength)	Данная опция определяет, допускается ли ввод в данное поле пустых строк ("")
<b>Индексированное поле</b> (Indexed)	Определяет простые индексы для ускорения поиска записей (поле первичного ключа индексируется автоматически)
<b>Сжатие Юникод</b> (Unicode Compression)	Указывает, используется ли сжатие символов в кодировке Unicode

Размер поля устанавливается с помощью параметра **Размер поля** и зависит от значений, которые предполагается вводить в данное поле. В таблице 3 и на рисунке 6 приведены значения, которые может иметь параметр **Размер поля** для числовых полей.

Характеристики подтипов для типа данных **Числовой**

Размер поля	Описание
<b>Байт</b> (Byte)	Хранятся целые числа из диапазона от 0 до 255 ( $2^8-1$ ). Поле занимает 1 байт.
<b>Целое</b> (Integer)	Хранятся целые числа из диапазона от $-32768$ до $32767$ ( $-2^{15} \dots 2^{15}-1$ ). Поле занимает 2 байта.
<b>Длинное целое</b> (Long Integer)	Хранятся целые числа из диапазона от $-2147483648$ до $2147483647$ ( $-2^{31} \dots 2^{31}-1$ ). Поле занимает 4 байта. Значение <b>Длинное целое</b> устанавливается по умолчанию.
<b>Одинарное с плавающей точкой</b> (Single)	Хранятся дробные числа с шестью знаками после запятой из диапазона от $-3.402823 \cdot 10^{38}$ до $3.402823 \cdot 10^{38}$ . Поле занимает 4 байта.
<b>Двойное с плавающей точкой</b> (Double)	Хранятся дробные числа с пятнадцатью знаками после запятой из диапазона от $-1.79769313486232 \cdot 10^{308}$ до $1.79769313486232 \cdot 10^{308}$ . Поле занимает 8 байт.
<b>Действительное</b> (Decimal)	Хранятся любые числа от $-10^{28}-1$ до $10^{28}-1$ , записанные максимум 28 цифрами. Поле занимает 12 байт.

Для хранения информации полей типа **Текстовый**, **МЕМО** в Access используется стандарт кодировки текста Unicode. В Unicode каждый символ представляется двумя байтами, что дает возможность поддерживать до 65536 символов (при однобайтовой кодировке обеспечивается доступ только к одной кодовой странице, состоящей из 256 символов). Следовательно, поля с кодировкой Unicode требуют больше места для хранения данных. Установленное в значение **Да** свойство поля **Сжатие Юникод** приводит к тому, что все символы, первый байт которых равен 0, будут сжиматься при сохранении и восстанавливаться при выборке.

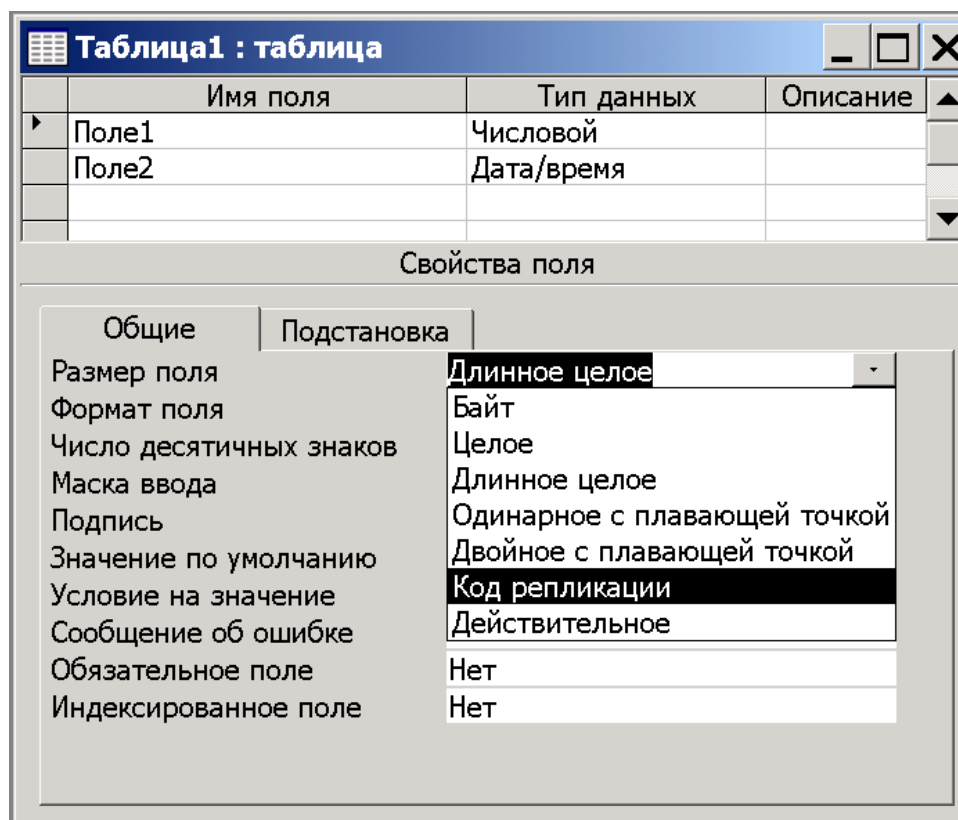


Рисунок 6 – Выбор размера поля таблицы

### 3.2. MapInfo

В MapInfo поле таблицы может содержать данные следующих типов:

- **Символьное** (Character). Строки до 254 символов длиной. Данный тип аналогичен типу **Текстовый** Microsoft Access.
- **Целое** (Integer). Целое число в пределах от  $-2^{31}$  до  $2^{31}-1$ . Данный тип аналогичен подтипу **Длинное целое** (Long Integer) Microsoft Access.
- **Короткое целое** (Small Integer). Целое число от  $-32768$  до  $32767$  ( $-2^{15}..2^{15}-1$ ). Данному типу соответствует подтип **Целое** (Integer) Microsoft Access.
- **Вещественное** (Float). Десятичные числа с плавающей точкой, соответствуют подтипу **Двойное с плавающей точкой** Microsoft Access.

- **Десятичное** (Decimal). Десятичные числа с фиксированной точкой. Данный тип аналогичен подтипу **Действительное** (Decimal) Microsoft Access.
- **Дата** (Date). Календарная дата. Данному типу соответствует тип **Дата/Время** Microsoft Access.
- **Логическое** (Logical). Значения, интерпретируемые как **Истина/Ложь**. В поле такого типа появляется либо символ "T" (TRUE) в случае значения **Истина**, либо литера "F" (FALSE) в случае значения **Ложь**. Данному типу соответствует тип **Логический** Microsoft Access.

### **Порядок выполнения работы:**

Согласно выданному преподавателем варианту реализовать в Microsoft Access структуру базы данных, разработанную в ходе проведения лабораторной работы "Проектирование структур баз данных информационных систем". Для достижения этой цели необходимо:

1. В режиме *Конструктора таблиц* Microsoft Access создать набор таблиц.
2. В режиме *Схемы данных* осуществить связывание данных в разных таблицах. В качестве ограничений целостности данных использовать каскадное обновление связанных полей.
3. Ввести данные в базу.

### **Содержание отчета:**

1. Наименование работы.
2. Цель работы.
3. Структура разработанной базы данных, представленная в виде схемы базы данных Microsoft Access и описания полей таблиц с указанием их типов.



**Контрольные вопросы:**

1. В чем состоит отличие баз данных ГИС MapInfo и Microsoft Access?
2. Какова процедура создания таблиц в Microsoft Access?
3. Перечислите общие и отличительные черты процессов создания таблиц в Microsoft Access и MapInfo.
4. Какие типы данных могут храниться в таблицах?