

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ЛУГАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИМЕНИ ВЛАДИМИРА ДАЛЯ»

Стахановский инженерно-педагогический институт (филиал)  
федерального государственного бюджетного образовательного учреждения  
высшего образования «Луганский государственный университет имени  
Владимира Даля»

Кафедра информационных систем

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ЛАБОРАТОРНЫМ РАБОТАМ**  
по дисциплине  
**«УПРАВЛЕНИЕ ИНФОРМАЦИЕЙ И ИНТЕЛЛЕКТУАЛЬНЫЕ  
СИСТЕМЫ»**

для студентов направления подготовки  
Профессиональное обучение (по отраслям),  
профиль «Информационные технологии и системы»  
(в 2-х частях). Часть 2

Луганск 2023

*Рекомендовано к изданию Учебно-методическим советом  
ФГБОУ ВО «ЛГУ им. В. Даля»  
(протокол № от . .2023 г.)*

Методические указания к лабораторным работам по дисциплине **«Управление информацией и интеллектуальные системы»** для студентов направления подготовки **Профессиональное обучение (по отраслям)**, профиль **«Информационные технологии и системы»** (в 2-х частях). Часть / Сост.: Е.С. Чёрная. – Стаханов: ФГБОУ ВО «ЛГУ им. В. Даля», 2023. – 60 с.

Методические указания к лабораторным работам определяют планирование, организацию и проведение лабораторных работ по учебной дисциплине «Управление информацией и интеллектуальные системы». Лабораторные работы относятся к основным видам учебных занятий и составляют важную часть теоретической и профессиональной практической подготовки. Представленные лабораторные работы направлены на формирование умений применять полученные знания на практике, реализацию единства интеллектуальной и практической деятельности. В методических указаниях представлены семь лабораторных работ. В методических рекомендациях в лаконичной форме дана справочная информация по изучаемой теме, методические указания студентам по выполнению лабораторных работ, эталон решения задачи, контрольные задания в необходимом количестве вариантов, дающие возможность обеспечить индивидуальное выполнение задания студентом

Предназначены для студентов профиля «Информационные технологии и системы».

Составитель:	доц., Чёрная Е.С.
Ответственный за выпуск:	доц., Карчевский В.П.
Рецензент:	доц., Карчевская Н.В.

© Чёрная Е.С., 2023  
© ФГБОУ ВО «ЛГУ им. В. Даля», 2023

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	4
ЛАБОРАТОРНАЯ РАБОТА 1 .....	6
ЛАБОРАТОРНАЯ РАБОТА 2 .....	12
ЛАБОРАТОРНАЯ РАБОТА 3 .....	18
ЛАБОРАТОРНАЯ РАБОТА 4 .....	25
ЛАБОРАТОРНАЯ РАБОТА 5 .....	33
ЛАБОРАТОРНАЯ РАБОТА 6 .....	41
ЛАБОРАТОРНАЯ РАБОТА 7 .....	45

## ВВЕДЕНИЕ

Дисциплина «Управление информацией и интеллектуальные системы» входит в вариативную часть профессионального цикла дисциплин подготовки студентов по направлению подготовки 44.03.04 Профессиональное обучение (по отраслям).

Дисциплина реализуется кафедрой информационных систем.

Основывается на базе дисциплин: «Информатика и информационные технологии», «Программное обеспечение систем управления и обучения», «Интеллектуальные системы».

Является основой для изучения следующих дисциплин: «Программная инженерия», «Объектно-ориентированное программирование и технологии разработки программного обеспечения».

Цель изучения дисциплины: знакомство с разнообразными информационными технологиями управления базами данных информационных систем; рассмотрение основных подходов к проектированию информационных систем, концепции баз данных; получение студентами глубоких и систематизированных знаний о методологии проектирования баз данных, о проектировании автоматизированных систем управления.

Основными задачами изучения дисциплины «Управление информацией и интеллектуальные системы» являются:

получение студентами необходимых теоретических и практических знаний по работе с различными информационными технологиями управления базами данных информационных систем;

получение навыков ставить прикладные задачи в области интеллектуальных информационных систем, обосновывать их постановку;

получение умений выбрать форму представления знаний и инструментальное средство разработки ИИС для конкретной предметной области;

приобретение навыков в проектировании баз данных, их формализованном описании и наполнении.

Студенты, завершившие изучение дисциплины «Управление информацией и интеллектуальные системы», должны

**знать:**

- определение интеллектуальных систем, структуру статических и динамических экспертных систем;

- теоретические основы построения и функционирования прикладных информационных систем, ключевые направления применения новых

информационных систем при автоматизации процессов принятия управленческих решений;

- методы построения эксплуатации и разработки информационных систем;

- модели представления знаний;

- возможности интеллектуальных систем и имеющихся программных продуктов;

- основные источники научно-технической информации по основным направлениям, методам, моделям и инструментальным средствам конструирования информационных систем;

***уметь:***

- формулировать цели и задачи автоматизации обработки управленческой информации;

- применять интеллектуальные системы для решения задач оценки и прогнозирования состояния объектов;

- оперировать с объектами баз данных - таблицами, формами и запросами, отчетами, макросами;

- управлять базой данных с помощью различных элементов управления;

- создавать сложные формы с использованием различных элементов управления;

- настраивать интерфейс пользователя базы данных;

***владеть навыками:***

- терминологией, навыками поиска и использования научно-технической информации по профессиональной тематике;

- разработки концепции интеллектуальных систем;


- анализа информационных систем управления персоналом.

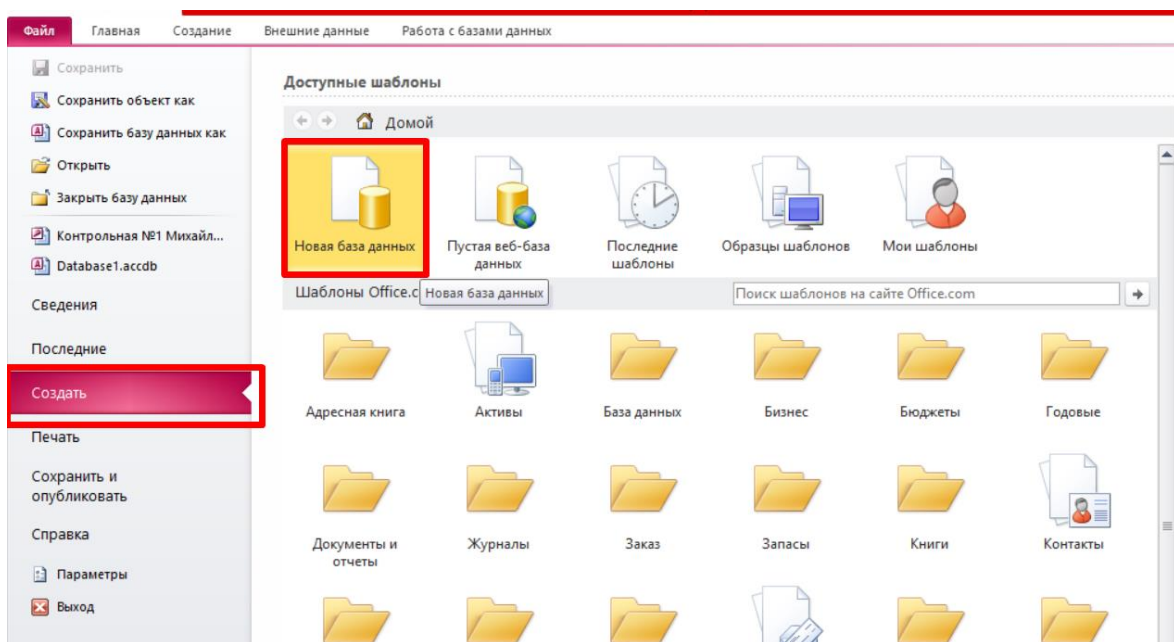
Дисциплина нацелена на формирование универсальных (УК-2, УК-5) и профессиональных компетенций (ПК-1) выпускника.

# ЛАБОРАТОРНАЯ РАБОТА 1

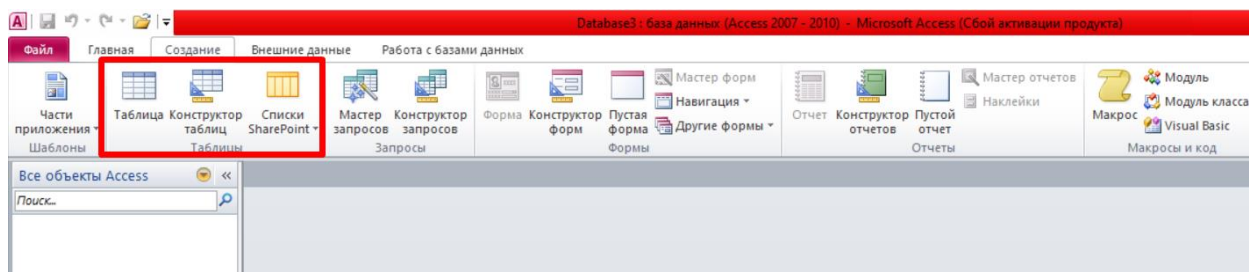
## ТЕМА: «НАЧАЛЬНЫЕ НАВЫКИ РАБОТЫ В СУБД ACCESS»

**Задание 1. Создайте новую базу данных и изготовьте структуру таблицы с информацией о студентах «Компьютерной школы».**

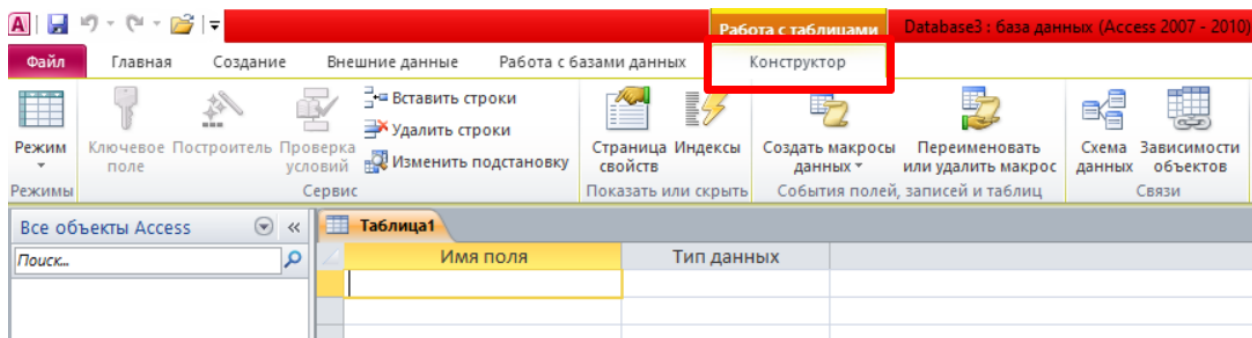
Запустите программу Access. Нажмите мышкой на кнопку  новая база данных. Появится окно, где в поле имя файла в качестве имени базы данных введите название базы данных.



Щелкните по кнопке создать или нажмите на клавишу [Enter]. В следующем окне выберите тип создаваемого документа. Вы создаете таблицу, поэтому выберите закладку таблица. Нажмите кнопку создать.




Переходим к работе со следующим окном: Новая таблица. Здесь несколько вариантов, но вы выберите конструктор и щелкните по кнопке ОК. Появится окно конструктора.



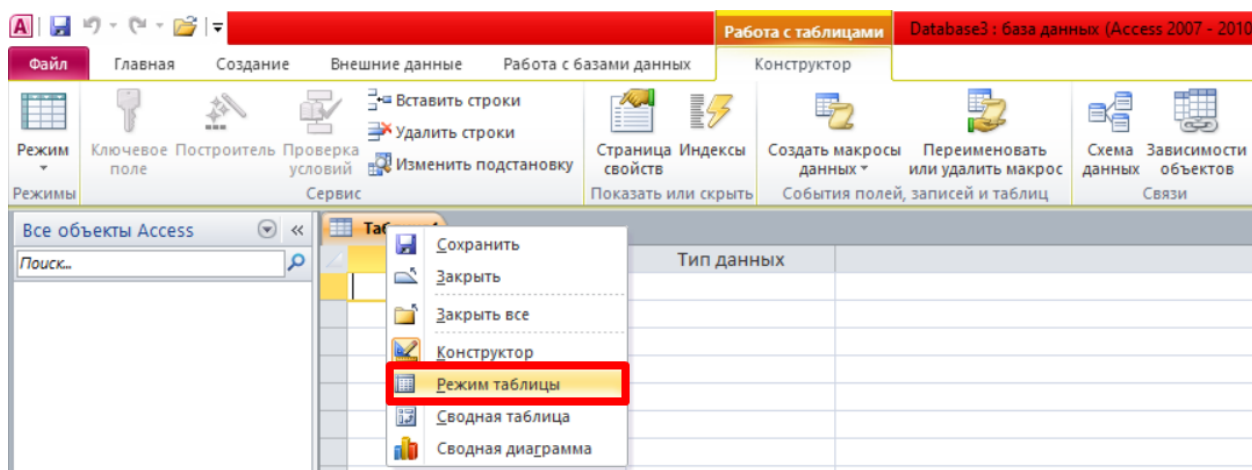
В левой верхней клетке введите имя поля и нажмите на клавишу [Enter]. В соседней клетке появится тип данных, по умолчанию он задается текстовый. Любой другой выбирается с помощью ниспадающего меню.

Заполните поля в конструкторе данными. Общие свойства поля оставляем по умолчанию.

	Имя поля	Тип данных
▶	Фамилия	Текстовый
	Имя	Текстовый
	Отчество	Текстовый
	Год рождения	Числовой
	Школа	Числовой
	Класс	Числовой

Сохраните таблицу, нажав на клавишу . В появившемся окне наберите имя таблицы «Список» и нажмите ОК. Появится запрос на создание ключевого поля – уникального поля записи, по которому удобно связывать таблицы. В данном варианте будут самостоятельно появляться числа – номера записей. Нажмите Да.


Перейдите в режим таблицы.



Ввод данных вы будете производить в этом режиме, заполняя клетки таблицы. Значения поля код будет меняться автоматически. Если закончить ввод в ячейку нажатием клавиши [Enter], то маркер перейдет в следующую ячейку.

Заполните базу данных значениями.

	Код	Фамилия	Имя	Отчество	Год рождения	Школа	Класс
	1	Иванова	Анна	Ивановна	1987	1	9
	2	Баранова	Ирина	Алексеевна	1986	3	10
	3	Корнилова	Ольга	Владимировна	1987	5	9
	4	Воробьев	Алексей	Петрович	1987	1	10
	5	Потылицин	Денис	Иванович	1987	3	9
	6	Коростелев	Иван	Григорьевич	1988	5	8
	7	Скоркин	Александр	Евгеньевич	1985	1	11
	8	Володина	Анна	Алексеевна	1987	3	9
	9	Новоселов	Алексей	Антонович	1986	5	10
▶	10	Александрова	Елена	Алексеевна	1987	1	9

Сохраните введенные данные, нажав на . В результате вы получите таблицу, с которой можно работать.

### Задание 2. Выполните редактирование ячеек.

Замените фамилию Иванова на Иванникова. Для этого выделите редактируемую ячейку и наберите новую фамилию. Замените год рождения на 1986. Для этого следует щелкнуть мышкой по нужной ячейке, и она откроется для редактирования. Удалите цифру 7 и введите 6.

### Задание 3. Отсортируйте значения таблицы.

*Замечание.* Чтобы произвести сортировку значений, нужно поставить курсор в любую ячейку сортируемого столбца и нажать на кнопку:






если нужно отсортировать по возрастанию.




если нужно отсортировать по убыванию.



Отсортируйте:

1. Фамилии – по алфавиту. Для этого поставьте маркер на любую фамилию в столбце с названием Фамилия и щелкните мышкой по кнопке .
2. Имя – по алфавиту. Для этого поставьте маркер на любую имя в столбце с названием имя и щелкните мышкой по кнопке .
3. Номер школы – по убыванию. Для этого поставьте маркер на любой номер школы в столбце с названием школа и щелкните мышкой по кнопке .




4. Класс – по возрастанию. Для этого поставьте маркер на любой класс в столбце с названием класс и щелкните мышкой по кнопке .

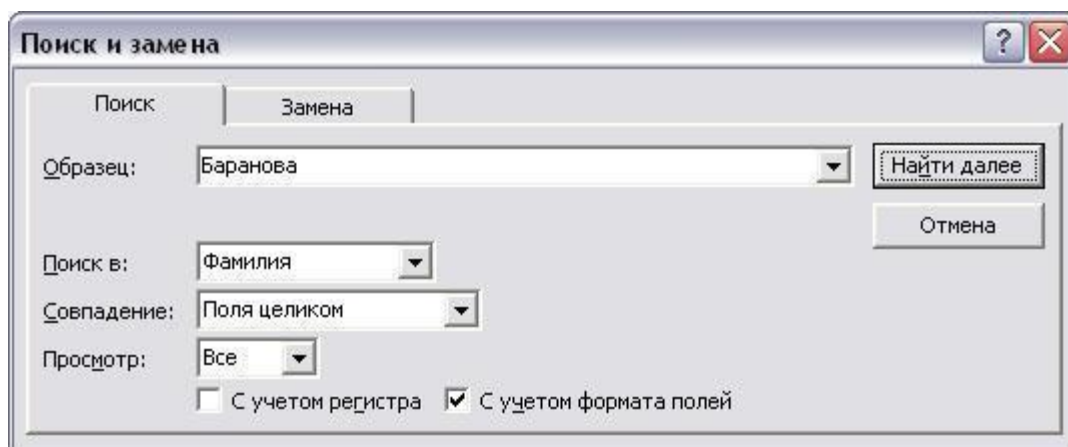
#### **Задание 4. Знакомство с функциями: сохранить, закрыть, открыть.**

Сохраните текущую таблицу, нажав на кнопку . Закройте таблицу, нажав на кнопку  в правом верхнем углу окна. Повторите аналогичную операцию еще раз, в результате чего вы закроете текущую базу данных, получив пустое окно для новой работы. В этом положении можно создать новую базу данных или открыть существующую для продолжения работы.

Откройте снова свою базу данных с помощью команд Файл – Открыть – Имя своей базы данных – Открыть. Откройте таблицу Список.

#### **Задание 5. Выполните поиск записей по образцу.**

Установите текстовый курсор в поле фамилия. Нажмите на кнопку , которая позволяет найти запись по введенному значению.



Наберите в поле образец фамилию Баранова и нажмите на кнопку найти далее.

#### **Задание 6. Завершите работу с Access.**

Выберете пункт меню Файл – Выход. Если вы производили какие-либо действия в базе данных, появится вопрос о сохранении изменений, сохраните свою базу данных.

#### **Задание 7. Выполните задание согласно варианту.**

ВАРИАНТ	БАЗА ДАННЫХ
1	Создать однотоабличную базу данных «Магазин продовольственных товаров». Предусмотреть наличие не менее 7 полей и 10 записей. Произведите сортировку по возрастанию одного из полей. Выполните поиск записей по нескольким параметрам. На проверку нужно прислать файл базы данных и отчет со скриншотами и полным описанием всех выполняемых действий. Выводы к работе обязательны.
2	Создать однотоабличную базу данных «Магазин игрушек». Предусмотреть наличие не менее 7 полей и 10 записей. Произведите сортировку по возрастанию одного из полей. Выполните поиск записей по нескольким параметрам. На проверку нужно прислать файл базы данных и отчет со скриншотами и полным описанием всех выполняемых действий. Выводы к работе обязательны.
3	Создать однотоабличную базу данных «Магазин электроники». Предусмотреть наличие не менее 7 полей и 10 записей. Произведите сортировку по возрастанию одного из полей. Выполните поиск записей по нескольким параметрам. На проверку нужно прислать файл базы данных и отчет со скриншотами и полным описанием всех выполняемых действий. Выводы к работе обязательны.
4	Создать однотоабличную базу данных «Магазин компьютерной техники». Предусмотреть наличие не менее 7 полей и 10 записей. Произведите сортировку по возрастанию одного из полей. Выполните поиск записей по нескольким параметрам. На проверку нужно прислать файл базы данных и отчет со скриншотами и полным описанием всех выполняемых действий. Выводы к работе обязательны.
5	Создать однотоабличную базу данных «Цветочный магазин». Предусмотреть наличие не менее 7 полей и 10 записей. Произведите сортировку по возрастанию одного из полей. Выполните поиск записей по нескольким параметрам. На проверку нужно прислать файл базы данных и отчет со скриншотами и полным описанием всех выполняемых действий. Выводы к работе обязательны.
6	Создать однотоабличную базу данных «Магазин кондитерских изделий». Предусмотреть наличие не менее 7 полей и 10


	<p>записей. Произведите сортировку по возрастанию одного из полей. Выполните поиск записей по нескольким параметрам. На проверку нужно прислать файл базы данных и отчет со скриншотами и полным описанием всех выполняемых действий. Выводы к работе обязательны.</p>
7	<p>Создать однотабличную базу данных «Магазин канцтоваров». Предусмотреть наличие не менее 7 полей и 10 записей. Произведите сортировку по возрастанию одного из полей. Выполните поиск записей по нескольким параметрам. На проверку нужно прислать файл базы данных и отчет со скриншотами и полным описанием всех выполняемых действий. Выводы к работе обязательны.</p>
8	<p>Создать однотабличную базу данных «Магазин одежды». Предусмотреть наличие не менее 7 полей и 10 записей. Произведите сортировку по возрастанию одного из полей. Выполните поиск записей по нескольким параметрам. На проверку нужно прислать файл базы данных и отчет со скриншотами и полным описанием всех выполняемых действий. Выводы к работе обязательны.</p>
9	<p>Создать однотабличную базу данных «Автоматический магазин». Предусмотреть наличие не менее 7 полей и 10 записей. Произведите сортировку по возрастанию одного из полей. Выполните поиск записей по нескольким параметрам. На проверку нужно прислать файл базы данных и отчет со скриншотами и полным описанием всех выполняемых действий. Выводы к работе обязательны.</p>
10	<p>Создать однотабличную базу данных «Книжный магазин». Предусмотреть наличие не менее 7 полей и 10 записей. Произведите сортировку по возрастанию одного из полей. Выполните поиск записей по нескольким параметрам. На проверку нужно прислать файл базы данных и отчет со скриншотами и полным описанием всех выполняемых действий. Выводы к работе обязательны.</p>

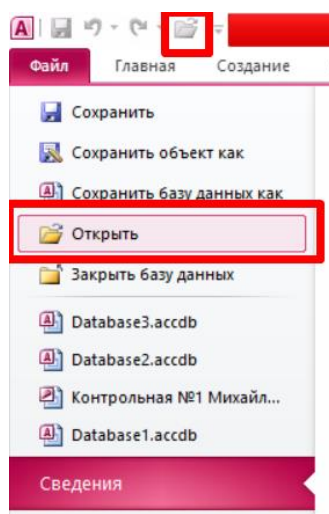
**Внимание!** Все 7 заданий обязательны к выполнению. На проверку нужно прислать файл базы данных и отчет со скриншотами и полным описанием всех выполняемых действий. Выводы к работе обязательны.

## ЛАБОРАТОРНАЯ РАБОТА 2

### ТЕМА: «ПРОЕКТИРОВАНИЕ МНОГО ТАБЛИЧНЫХ БАЗ ДАННЫХ С ПОМОЩЬЮ СУБД ACCESS»

**Задание 1. Откройте учебную базу данных, созданную в лабораторной работе 1.**

Запустите программу Microsoft Access. Нажмите на кнопку  открыть базу данных. Выделите ее из списка баз данных и нажмите ОК.



**Задание 2. Удалите таблицу Список.**

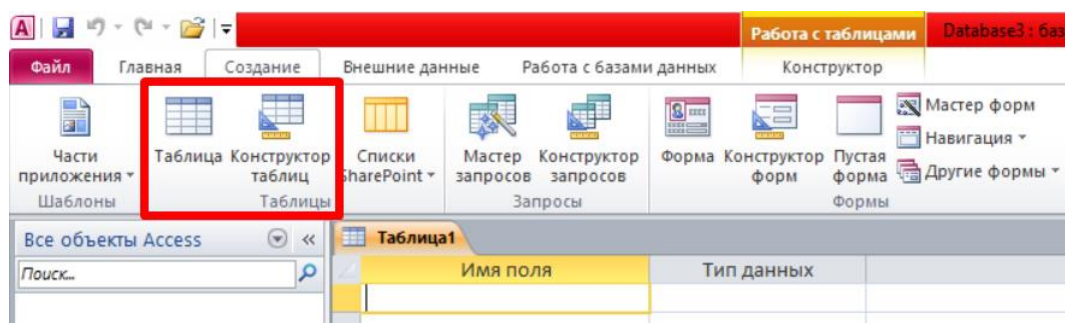
Выделите имя таблицы.

Нажмите клавишу [Delete].

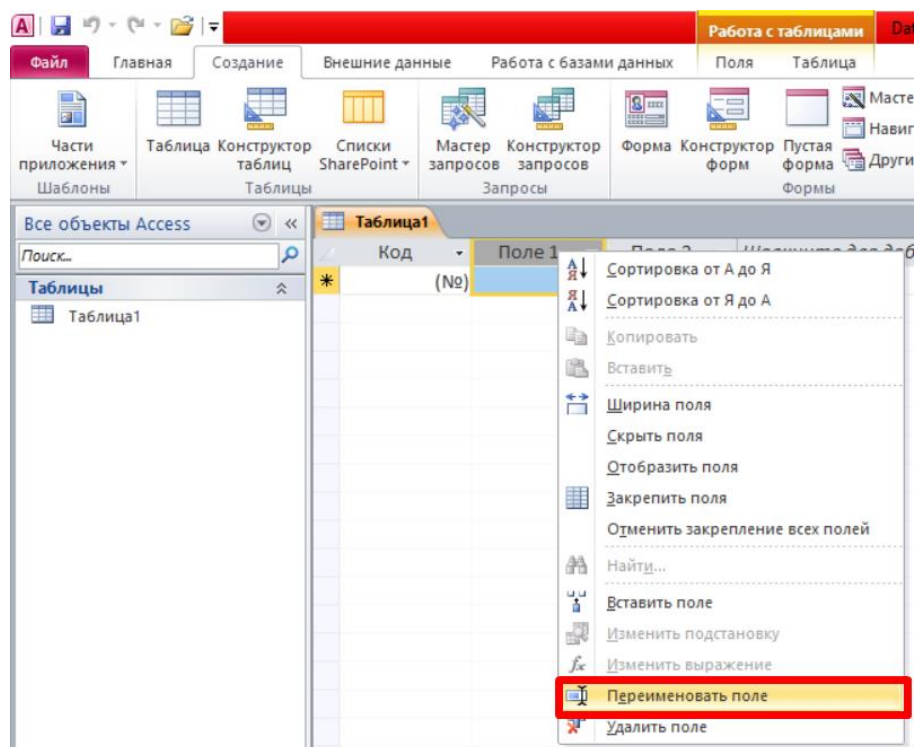
На вопрос о подтверждении удаления таблицы ответьте да.

**Задание 3. Создайте таблицу группы.**




Выберите закладку таблица, если находитесь в другом окне. Нажмите на кнопку создать.



Выберите режим таблицы и нажмите на ОК. Появится пустая таблица, поля которой не определены и не имеют названия. Тип поля будет выбран автоматически в зависимости от введенной информации. Переименуйте Поле 1. Для этого поставьте курсор в любую ячейку столбца поле 1 и выполните команду формат – переименовать столбец. Ячейка имени столбца окажется выделенной. Введите название поля Учебная группа и нажмите клавишу [Enter].



Переименуйте Поле 2. Для этого поставьте курсор в любую ячейку столбца поле 2 и выполните команду формат – переименовать столбец. Ячейка имени столбца окажется выделенной. Введите название поля преподаватель и нажмите клавишу [Enter].

Сохраните таблицу с именем группы, нажав на кнопку  сохранить. На вопрос о создании ключевого поля ответьте отрицательно. Перейдите в режим конструктор, нажав на кнопку , и посмотрите, как заданы поля. Сделайте поле учебная группа ключевым, поместив курсор на имя этого поля и нажав на кнопку  ключевое поле. Тип данных поля учебная группа задайте числовым, выбрав его мышкой из ниспадающего списка.




Сохраните изменения, нажав на кнопку . Закройте таблицу.


#### Задание 4. Создайте таблицу Список.

Выберите закладку таблица, если находитесь в другом окне. Нажмите на кнопку создать.

Выберите режим таблицы и нажмите на ОК. Появится пустая таблица, поля которой не определены и не имеют названия. Переименуйте поля по аналогии с предыдущим заданием, выполнив команду формат – переименовать поле.

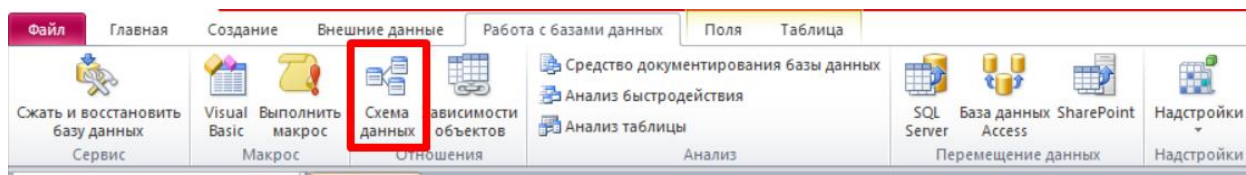
	Код	Фамилия	Имя	Отчество	Год рождения	Школа	Класс	Учебная группа

Сохраните таблицу с именем Список, нажав на кнопку сохранить . На вопрос о создании ключевого поля ответьте отрицательно. Перейдите в режим конструктор, нажав на кнопку , и посмотрите, как заданы поля. Сделайте поле код ключевым, поместив курсор на имя этого поля и нажав на кнопку  ключевое поле. Тип данных поля код задайте счетчик, полей фамилия, имя, отчество – текстовые, полей год рождения, школа, класс и учебная группа – числовые. Значения поля учебная группа надо не набивать вручную, а выбирать из списка, содержащегося в таблице группы. Для этого в свойствах поля следует указать, что здесь имеет место подстановка по следующей схеме: выберите закладку подстановка, тип элемента управления – поле со списком, источник строк – группы.

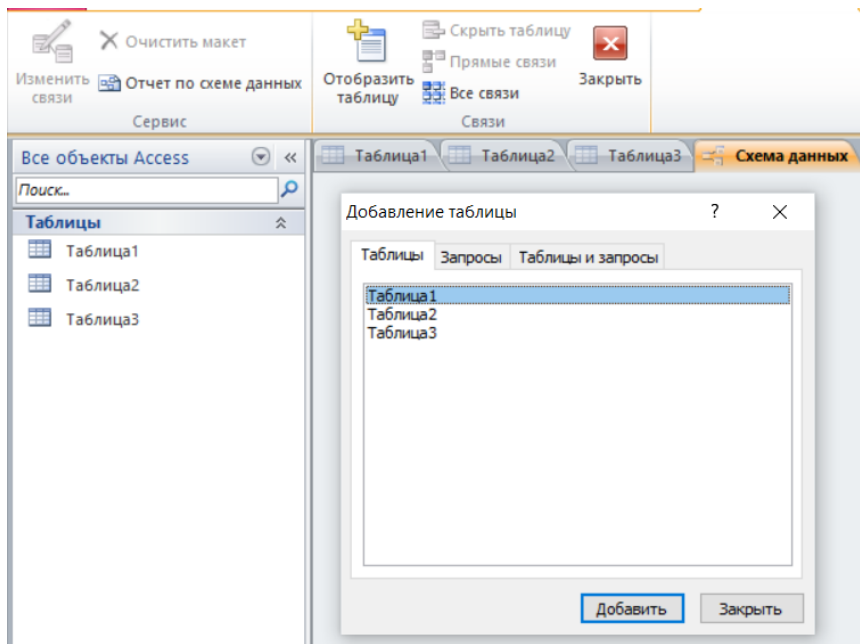
Сохраните изменения, нажав на кнопку . Закройте таблицу.

### Задание 5. Создайте схему данных.

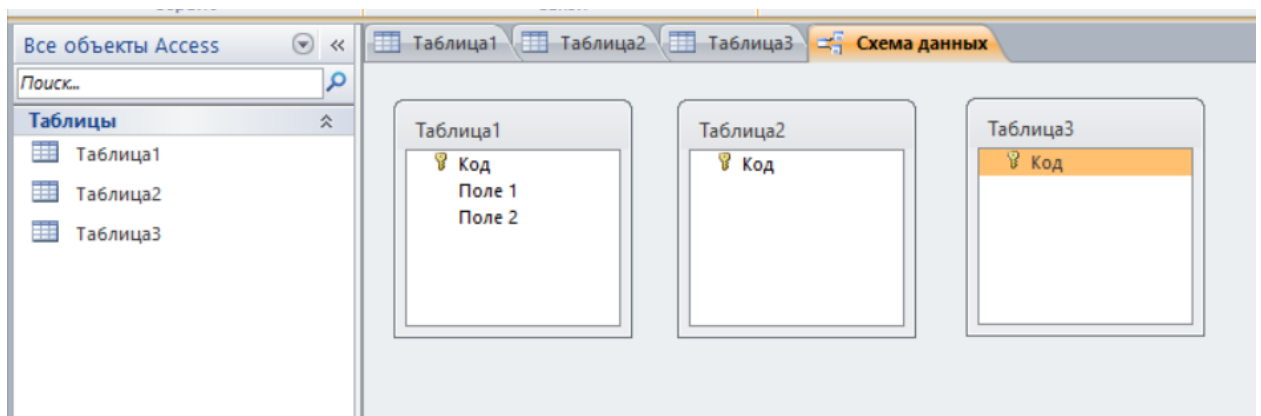
Нажмите на кнопку  схема данных.



В появившемся окне добавление таблицы выделите таблицу группы и нажмите на кнопку добавить.



Выделите таблицу список и щелкните по кнопке добавить. В окне схема данных появится условный вид этих таблиц.




Закройте окно добавление таблицы. Увеличьте окно таблицы список так, чтобы были видны все поля. Поставьте мышку на имя поля учебные группы в таблице группы и, не отпуская кнопку мышки, перетащите ее на поле учебные группы в таблице список. Отпустите мышку. Появится окно связи.

Включите обеспечение целостности данных. Это невозможно будет сделать, если типы обоих полей заданы не одинаково. Включите каскадное обновление связанных полей. Это приведет к тому, что при изменении номера группы в таблице группы автоматически изменится соответствующие номера групп.

Включите каскадное удаление связанных полей. Это приведет к тому, что при удалении записи с номером группы в таблице группы будут удалены

все записи из таблицы список, в которой стояли соответствующие номера групп.

Нажмите на кнопку создать. Появится связь один-ко-многим. Закройте схему данных, нажав на кнопку  в верхнем правом углу окна и ответив утвердительно на вопрос о сохранении схемы данных.

#### **Задание 6. Заполните таблицу группы значениями.**

Откройте таблицу группы в режиме таблицы. Заполните ее записями. Сохраните таблицу и закройте ее.

#### **Задание 7. Откорректируйте данные в таблице группы.**

Откройте таблицу группы, выделив ее и нажав на кнопку открыть. Добавьте недостающие записи, которые вы удалили в предыдущей работе и исправьте существующие.

Сохраните и закройте таблицу.

#### **Задание 8. Удалите все записи таблицы список, оставив ее структуру.**

Откройте таблицу Список.

Выделите все записи.

Нажмите клавишу [Delete]. Нажмите на кнопку ДА в вопросе о подтверждении удаления. (Можно выполнить команду правка – удалить запись).

Закройте таблицу, сохранив ее.

*Замечание.* Если теперь вводить данные в эту таблицу снова, то счетчик будет меняться с того номера, который был присвоен последней записи.

Чтобы нумерация снова начиналась с 1, выполните команду сервис – служебные программы – сжать базу данных. Подождите некоторое время, чтобы программа отработала.

#### **Задание 9. Используя мастер таблиц, создайте таблицу личные данные с ключевым полем.**

Выберите закладку таблица, если находитесь в другом окне. Нажмите на кнопку создать. В результате перейдем к работе со следующим



окном: новая таблица. Здесь несколько вариантов, но вы выберите мастер таблиц и нажмите на кнопку ОК.

В этом окне следует выбрать: в поле образцы таблиц – поле студенты; в поле образцы полей – поля КодСтудента, Адрес, Номертелефона, нажимая после каждого выбора на кнопку . Эти поля попадут в поля новой таблицы. Нажмите на кнопку Далее.

В появившемся окне задайте имя новой таблицы личные данные. Оставьте автоматический выбор ключа. Нажмите на кнопку далее. Замечание. Access проверит связи данной таблицы с другими таблицами. Так как вы еще не устанавливали связи, то они не будут найдены автоматически. В этот момент можно установить новые связи, но мы пока этого делать не будем.

Нажмите на кнопку Далее.

После появления вопроса о действиях после создания таблицы автоматически выбирайте ввод в режиме таблицы. Нажмите на кнопку готово.

Добавьте в таблицу личные данные еще три поля Word, Excel, Access, в которых будут находиться семестровые оценки по этим предметам. Выберите конструктор.

Закройте таблицу, предварительно сохранив ее. В результате вы получите три таблицы, две из которых связаны, а третья нет.

### **Задание 10. Исправьте схему данных.**

Нажмите на кнопку – схема данных. Нажмите на кнопку - добавить таблицу. В появившемся окне выделите таблицу личные данные и нажмите на кнопку добавить, а затем на кнопку закрыть окна добавления таблицы.

Поставьте мышку на имя поля КодСтудента в таблице личные данные и, не отпуская кнопку мышки, перетащите ее на поле Код в таблице список. Отпустите кнопку мышки. Появится окно связи. Включите обеспечение целостности данных. Это невозможно сделать, если типы обеих полей заданы неодинаково.

Нажмите на кнопку создать. Появится связь один-к-одному. Это значит, что одной записи в таблице список соответствует одна запись в таблице личные данные. Закройте схему данных, сохранив ее.

**Пояснение.** Теперь встает задача ввода записей одновременно в разные таблицы: список и личные данные. ФИО мы храним в одной таблице, а адрес и номер телефона – в другой. Можно попробовать ввести значения в каждую

таблицу отдельно, но удобнее видеть клетки обеих таблиц для ввода данных одновременно.

**Задание 11.** В базу данных, созданную в задании 7 лабораторной работе 1, добавьте по 2 таблицы, каждая из которых содержит не менее 5 полей и 10 записей. Произведите связывание таблиц в базе данных.

На проверку нужно прислать файл базы данных и отчет со скриншотами и полным описанием всех выполняемых действий. Выводы к работе обязательны.

**Внимание!** Все 11 заданий обязательны к выполнению. На проверку нужно прислать файл базы данных и отчет со скриншотами и полным описанием всех выполняемых действий. Выводы к работе обязательны.

## ЛАБОРАТОРНАЯ РАБОТА 3

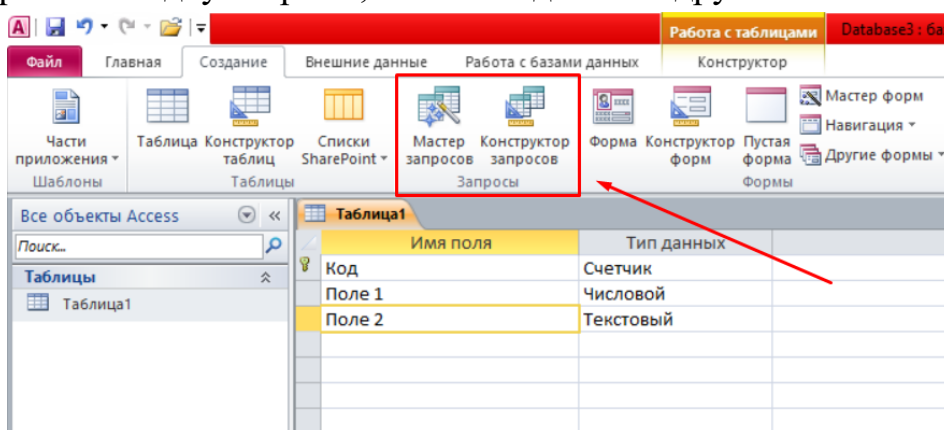
### ТЕМА: «ИСПОЛЬЗОВАНИЕ ЗАПРОСОВ В СУБД ACCESS»

**Задание 1.** Откройте базу данных, созданную в рамках предыдущей лабораторной работы.

Запустите программу Microsoft Access.

Нажмите на кнопку открыть базу данных. Выделите из списка баз данных имя вашей базы данных и нажмите на кнопку ОК.

**Задание 2.** Создайте запрос на выборку с именем Номера телефонов. Выберите закладку запросы, если находитесь в другом окне.

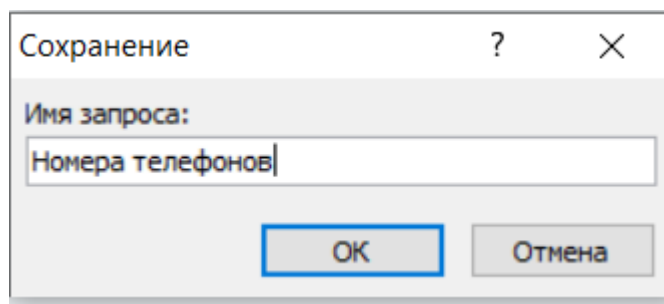


В появившемся окне выберите Конструктор, нажмите на кнопку ОК. Добавьте нужные таблицы (Личные данные и Список), выбирая их и щелкая по кнопке Добавить. Так как таблицы связаны, то между ними появится линия, обозначенная цифрами 1 (один-к-одному).

Закончите выбор, нажав на кнопку закрыть. Появляется возможность выбора полей из разных таблиц.

Выберите поля Фамилия, Имя и Отчество из таблицы Список и НомерТелефона – из таблицы Личные данные. Для этого достаточно сделать двойное нажатие мышкой по имени поля. Второй вариант – перетащить мышкой название поля в клетки запроса.

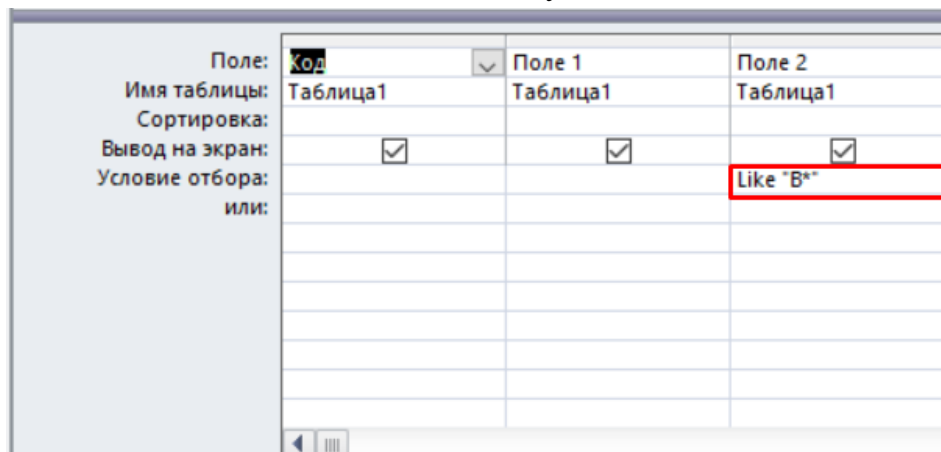
Сохраните запрос, нажав на кнопку сохранить. Введите имя запроса Номера телефонов и нажмите ОК.



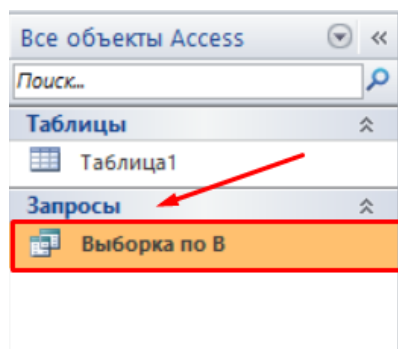
Нажмите на кнопку для представления запроса. Это самый простой вид запроса на выборку. В результате вы получаете новую таблицу с другим набором полей.

Перейдем в режим конструктор.

Сделайте запрос отображения всех фамилий, начинающихся с буквы В. Это можно сделать с помощью включения условия Like В\*”.



Нажмите на кнопку для представления запроса. Измените имя запроса, выбрав в меню пункт Файл – Сохранить как. В появившемся окне наберите новое имя запроса: Выборка по В. Теперь в меню базы данных в окне Запросы будет показано два запроса.



**Самостоятельное задание:**

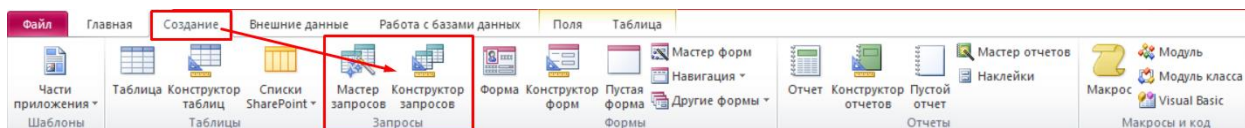
1. Составьте запрос на адреса девочек, имя которых «Анна». Сохраните запрос с именем Анна.

2. Составьте запрос на телефоны учащихся, отчество которых начинается на букву «А». Сохраните запрос с именем Выборка по А.

**Задание 3.** Составьте запрос с использованием логических операций в условии отбора.

*Предположим, что вам нужно составить ведомость для выплаты стипендии всем учащимся, которые учатся без троек. Для этого нужно выбрать записи, в которых оценки по предметам 4 ИЛИ 5.*

Выберите закладку Запрос, если находитесь в другом окне. Нажмите на кнопку Создать.



В появившемся окне выберите Конструктор, нажмите ОК. Добавьте нужные таблицы (Личные данные и Список), выбирая их и нажимая кнопку Добавить.

Закончите выбор, нажав на кнопку закрыть. Появляется возможность выбора полей из разных таблиц.

Выберите поля Фамилия, Имя и Отчество из таблицы Список и Word, Excel и Access – из таблицы Личные данные. Для этого достаточно сделать двойное нажатие мышкой по имени поля. Второй вариант – перетащить мышкой название поля в клетки запроса.

В строке Условие отбора под полями Word, Excel и Access поставьте 4 Or 5.

Нажмите на кнопку для представления запроса. Сохраните запрос, нажав на кнопку сохранить. Введите имя запроса Успеваемость1 и нажмите ОК.

### Самостоятельное задание:

1. Составьте запрос на учеников группы 101, у которых оценка по «Access» 4 или 5, сохраните запрос с именем Успеваемость2.
2. Составьте запрос на учеников групп 102 и 103, которые имеют оценку по «Word» и «Excel» 4 или 5, сохраните запрос с именем Успеваемость3.

**Задание 4.** Составьте запрос на выборку всех записей, кроме тех, в которых указана фамилия Баранова с использованием Построителя выражений.

Выберите закладку Запрос, если находитесь в другом окне.

Выделите запрос Номера телефонов.

Нажмите на кнопку Конструктор.

Удалите поле НомерТелефона.

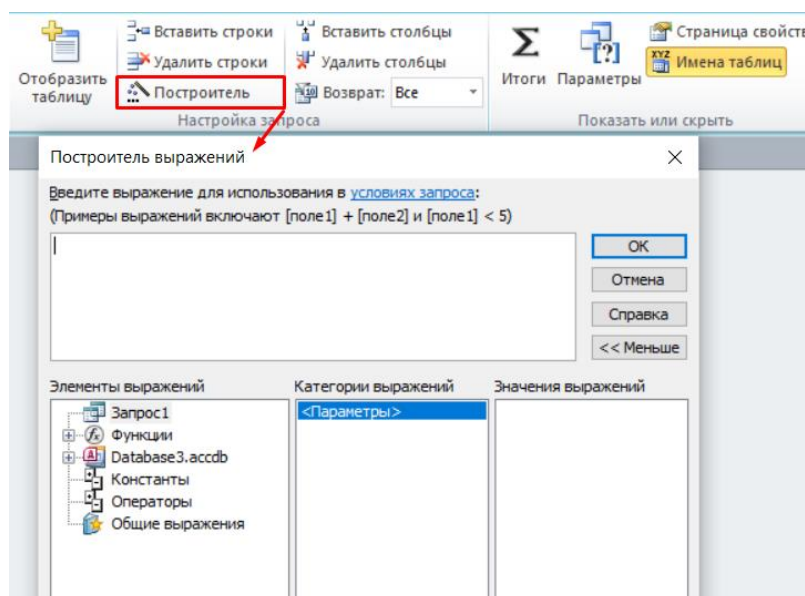
Добавьте поле Адрес.

Сохраните запрос с именем Адрес, выполнив команду Файл – Сохранить как.

Поставьте курсор в ячейку Условие отбора в столбце Фамилия.

Удалите надпись в этой ячейке.

Нажмите на кнопку - Построить. Появится окно, в котором можно строить сложные запросы.



Нажмите на кнопку Not, это слово появится в верхнем поле. Фамилию Баранова в кавычках наберите в ручную.

Нажмите на кнопку ОК.

Нажмите на кнопку для представления запроса.

Закройте запрос, сохранив его с именем не Баранова, выполнив команду  
Файл – Сохранить как.

### Задание 5. Создайте вычисляемые поля.

Выберите закладку Запрос, если находитесь в другом окне.

Нажмите на кнопку Создать.

В появившемся окне выберите Конструктор, нажмите ОК.

Добавьте нужные таблицы (Личные данные и Список), выбирая их и нажимая кнопку Добавить.

Закончите выбор, нажав на кнопку закрыть.

Выберите поля Фамилия и Имя из таблицы Список и поля Word и Excel – из таблицы Личные данные.

Поставьте курсор на клетку правее Excel на линии Поле.

Нажмите на кнопку - Построить.

В появившемся окне напечатайте выражение, представленное ниже. Нажмите ОК.

Среднее:([Word]+[Excel])/2

Это выражение подставится в новое поле. Нажмите клавишу [Enter].

Сохраните запрос с именем Среднее.

Нажмите на кнопку для представления запроса. Новое поле будет иметь имя Среднее.

Закройте запрос.

*Самостоятельное задание. Создайте новую таблицу с названием Новая группа и текстовыми полями Фамилия, Имя, Отчество и числовым полем Учебная группа с полем подстановок из таблицы Группы.*

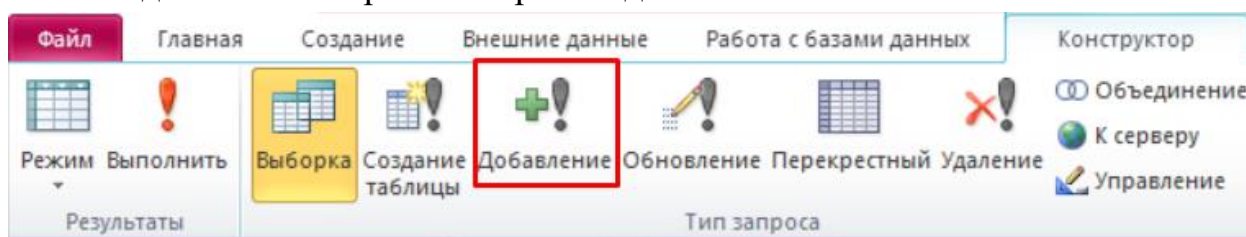
Для этого:

Задайте ключевое поле Код.

Заполните значениями: Сидорова Анна Ивановна, Петрова Инна Сергеевна, Сергеева Ирина Петровна, Куликова Ольга Дмитриевна. Номер группы 101.

Закройте таблицу, предварительно сохранив ее.

### Задание 6. Постройте запрос на добавление.



Выберите закладку Запрос, если находитесь в другом окне. Нажмите на кнопку Создать.

В появившемся окне выберите Конструктор, нажмите на кнопку ОК. Добавьте таблицу Новая группа, выбрав ее из списка и нажав на кнопку Добавить.

Закончите выбор, нажав на кнопку Закрывать.

Выберите те поля, которые нужно добавить: Фамилия, Имя, Отчество, Учебная группа. Если у какого-либо поля поставить условие отбора, то добавятся только записи, удовлетворяющие этому условию.

Нажмите на стрелку рядом с кнопкой Тип запроса на панели инструментов и выберите команду Добавление.

В появившемся окне выберите имя таблицы Список, в которую будут добавляться данные. Эта таблица может быть из текущей базы данных или из любой другой. В данном случае у нас одна база данных, поэтому нажмите на кнопку ОК.

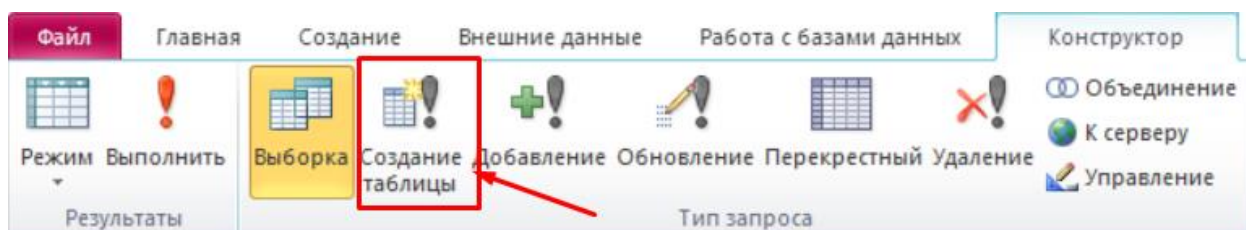
Нажмите на кнопку для выполнения запроса. Подтвердите выполнение запроса.

Закройте запрос, сохранив его с именем Добавление. Откройте таблицу Список. Просмотрите результат применения запроса.

### Задание 7. Постройте запрос на создание таблицы.

*Пояснение. Бывают ситуации, когда из старых таблиц нужно составить новые с другим набором полей. Предположим, что мы хотим создать таблицы успеваемости для учеников разных групп, поместив Список – Личные данные. Снова создайте связь Список – Личные данные.*

Выберите закладку Запрос, если находитесь в другом окне. Нажмите на кнопку Создать.



В появившемся окне выберите Конструктор, нажмите на кнопку ОК. Добавьте таблицы Список и Личные данные, выбрав их из списка и нажав на кнопку Добавить.

Закончите выбор, нажав на кнопку **Заккрыть**. Нажмите на стрелку рядом с кнопкой **Тип запроса** на панели инструментов и выберите команду **Создание таблицы**....

Напечатайте имя таблицы **Успеваемость** и нажмите на кнопку **ОК**. Выберите поля **Фамилия**, **Имя**, **Отчество** и **Учебная группа** из таблицы **Список** и поля **Word**, **Excel**, **Access** из таблицы **Личные данные**.

Нажмите на кнопку для выполнения запроса. Закройте запрос, сохранив его с именем **Новая таблица**. Откройте таблицу **Успеваемость**. Убедитесь, что данные были добавлены.

### **Задание 8.** Создайте перекрестный запрос.

*Допустим, вам нужно подсчитать для экзаменационной ведомости, сколько в группе, занимающейся изучением программы Word, получено троек, четверок и пятерок. Для подобных целей используется Перекрестный запрос.*

Выберите закладку **Запрос**, если находитесь в другом окне. Нажмите на кнопку **Создать**.

В появившемся окне выберите **Перекрестный запрос**, нажмите на кнопку **ОК**.

В окне создание перекрестных запросов выделите таблицу **Успеваемость** и нажмите **Далее**.

Выберите поле, значение которого будут использоваться в качестве заголовков столбцов, например **Word**. Нажмите **Далее**.

Выберите функцию, по которой будут вычисляться значения ячеек на пересечении столбцов и строк (в данном случае **Count** - количество). Здесь вы можете добавить итоговое поле для каждой строки (в данном случае это поле покажет общее количество оценок по каждой группе). Нажмите на кнопку **Далее**.

Задайте имя запроса **Word** и нажмите кнопку **Готово**. Завершите работу с программой **Access**.

**Внимание!** Все 8 заданий обязательны к выполнению. На проверку нужно прислать файл базы данных и отчет со скриншотами и полным описанием всех выполняемых действий. Выводы к работе обязательны.



## ЛАБОРАТОРНАЯ РАБОТА 4

### ТЕМА: «РАБОТА С ФОРМАМИ В СРЕДЕ СУБД ACCESS»

Главным преимуществом в работе с базами данных можно назвать удобство представления и эффективность организации данных. Access позволяет сконструировать форму наиболее удобным для пользователя способом, включив в нее записи различных связанных таблиц.

Для отображения информации сразу из двух таблиц или из таблицы и запроса создается составная форма, которая состоит из основной и находящейся внутри нее подчиненной формы. Это позволяет объединять информацию, не переключаясь на другие таблицы или формы. Как правило, подчиненная форма связана с основной. Сохраняется подчиненная форма в базе данных как отдельная форма. Основным способом создания формы является использование Мастера составной формы. Мастер может создать и связать обе формы в случае, если основная форма базируется на таблице, а подчиненная форма базируется на связанной с ней другой таблице или создана на базе таблицы и запроса, содержащего поле с тем же именем и типом данных, что и ключевое поле таблицы основной формы.

Форма - это объект Access, без которого можно обойтись. Он создан специально для пользователя. Экран принимает удобный для обработки информации вид с учетом того, что пользователь может работать в БД несколько часов. Оттого, каким будет экран компьютера все это время, зависит настроение и здоровье. Часто формы приближают к виду уже известных бумажных документов. Такой подход позволяет специалистам других отраслей легко перейти на электронные документы.

Через формы можно загружать данные в таблицы, просматривать и корректировать их. Работая с формой, пользователь может добавлять и удалять записи в таблицах, изменять значения в полях, получать расчетные данные. В форме можно контролировать вводимые данные, устанавливать ограничения на доступ к информации, выводить необходимые сообщения.

Источником данных для создания форм являются таблицы или запросы. Форму можно создать с помощью мастера или конструктора.

Таким образом, можно сделать вывод, что формы в Access являются главным средством разработки пользовательского интерфейса. В зависимости от того, с какой целью создается форма, разработчик определяет параметры ее внешнего вида.

#### **Создание формы с помощью Мастера форм.**

Мастер создает форму за четыре шага:

1. В главном окне базы данных, нужно выбрать объект Формы и нажать кнопку Создать. Появится окно Новая форма.

В качестве источника данных в окне надо выбрать таблицу или запрос и выделить фразу Мастер форм или требуемую автоформу, после чего нажать ОК. При выборе автоформы требуемая форма появится сразу. При выборе мастера форм последовательно откроются четыре окна Создание форм.

2. В первом окне будут представлены доступные поля выбранной таблицы или запроса. Надо перенести те из них, которые требуются для формы, в область Выбранные поля и нажать кнопку Далее.

В это окно можно попасть, пропустив предыдущий шаг и два раза щелкнув по надписи Создать форму с помощью мастера. При этом теряется возможность выбора автоформы. Так как пропущено предыдущее окно, здесь повторно дается возможность выбора таблицы или запроса в строке Таблицы и запросы.

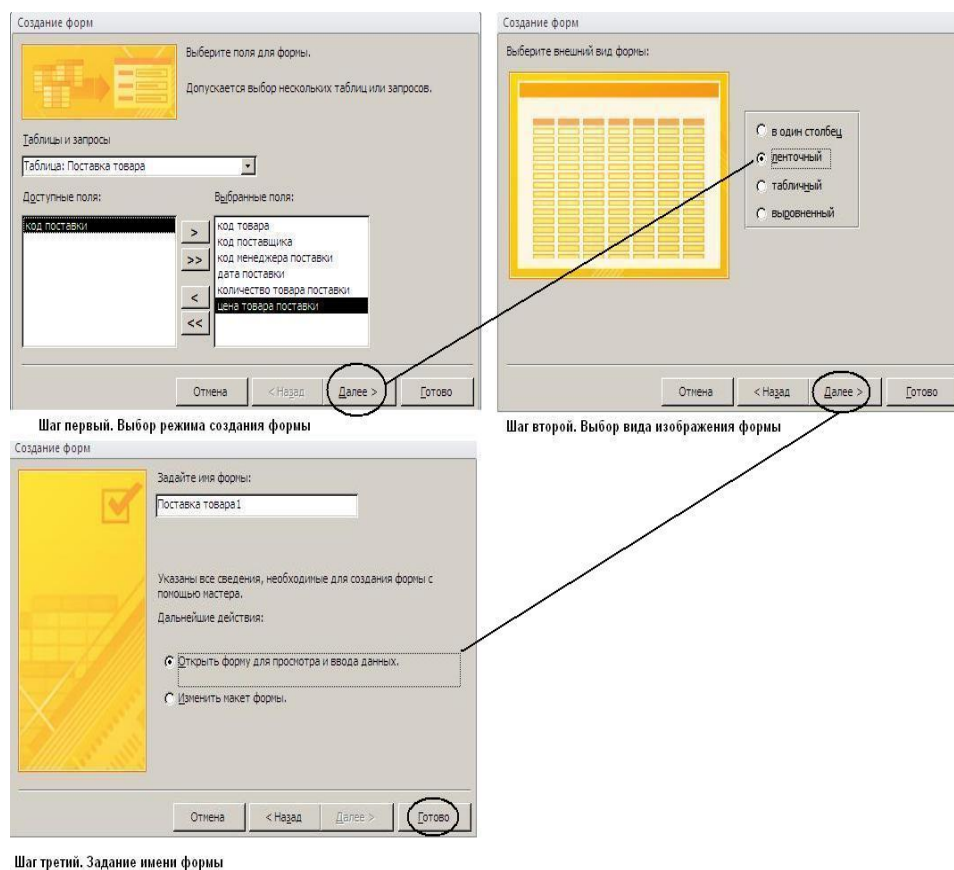


Рисунок 4.1 – Мастер создания форм

3. Во втором окне необходимо выбрать внешний вид формы и опять нажать кнопку Далее.

4. В третьем окне надо выбрать стиль оформления и снова нажать кнопку Далее.
5. В четвертом, последнем, окне задается имя формы, после чего необходимо нажать кнопку Готово.
6. На экране должна появиться созданная форма. При необходимости ее можно доработать в режиме Конструктор.

### Создание формы с помощью конструктора

Конструктор поможет создать форму в более свободном стиле. Начало будет таким же, как и в мастере.

1. В главном окне базы данных следует выбрать объект Формы и нажать кнопку Создать. Появится окно Новая форма. Здесь надо выбрать в качестве источника данных таблицу или запрос, однако выделить теперь необходимо фразу Конструктор, после чего нажать ОК. На экране должны появиться окна: главное окно Форма1: форма и вспомогательные окна Список полей выбранной таблицы или запроса и Панель элементов.

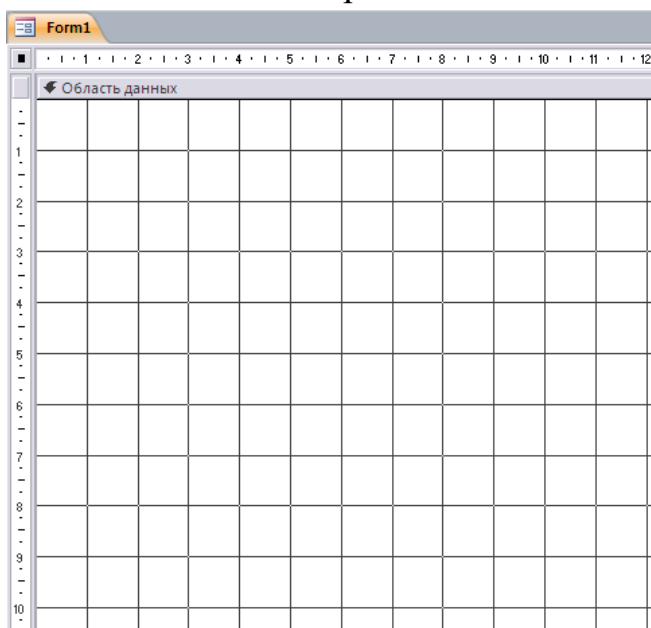


Рисунок 4.2 – Окно Форма1:форма.

2. Из меню Вид надо выбрать пункт Заголовок/примечание формы. Форма поделится на три части, которые разделены поперечными разделителями с названиями Заголовок формы, Область данных и Примечание формы. Размер каждой части можно регулировать передвижением разделителей вверх или вниз. В зоне Заголовок формы нужно вписать название формы.

3. В окне Список полей поочередно необходимо выделить поля и, удерживая их левой кнопкой мыши, перетащить из таблицы в зону Область данных. Здесь поля превратятся в элементы управления,

состоящие из двух рамок с одинаковым названием. Поле в белой рамке – это связанное поле, то есть надпись при выходе в режим формы заменится на данные из отобранных таблиц или запросов, поэтому текст в связанном поле изменять нельзя, его можно только форматировать: изменить размер, шрифт, цвет. Поле в серой рамке – это присоединенная надпись (то есть поясняющая). При переходе в режим Форма она будет располагаться рядом с данными таблицы как пояснение. Присоединенную надпись можно полностью изменять для более привычного обозначения. Эту часть также можно форматировать. Форматирование каждой части производится автономно. Элементы управления образуют единое целое и переносятся в конструкторе совместно, если указатель мыши примет форму раскрытой ладони. При изменении указателя на руку с вытянутым указательным пальцем в левом верхнем углу выделенной рамки (прямоугольник выделенной рамки левого верхнего угла имеет больший размер) каждый из элементов управления можно будет перенести отдельно.

### **Подчиненная форма**

В каждой базе данных есть одна или несколько основных форм. В них часто добавляют подчиненную форму, или подформу, то есть одну форму вкладывают в другую.

Подчиненную форму можно использовать в следующих случаях:

- для сбора в одной форме данных из нескольких таблиц с целью показа дополнительной (справочной) информации;
- для получения доступа к записям из различных таблиц в одной форме.

Обычно подформа работает синхронно с основной формой: при выборе одного их полей в основной форме информация в подчиненной форме автоматически изменяется, так как она дополняет информацию выбранного поля. Поле для синхронизации в основной форме называется основным, а в подформе – подчиненным. Для создания таких форм используются специальные запросы.

### **Задания.**

1. Откройте учебную базу данных, созданную в лабораторной работе 2.
2. Добавьте таблицу, которая содержит оценки учеников по 10 предметам. Задайте название таблицы – «Ведомость оценок».

3. Добавьте таблицу «Кружки», в которой будет информация об увлечениях ученика.
4. Добавьте таблицу «Учителя», в которой будет представлена информация об учителях и предметах, которые читает учитель.
5. Задайте связи для таблиц.
6. Создайте составную форму для ввода информации об оценках ученика.
7. Создайте составную форму для ввода информации о кружках, которые посещает ученик.
8. Создайте составную форму для ввода информации о предметах, которые преподает учитель.

### **Технология выполнения работы.**

1. Создать составную форму для ввода информации об оценках ученика. Для создания формы надо выбрать в окне База данных вкладку Формы, а затем выбрать Создание формы с помощью мастера. Далее осуществить следующие действия:

- в появившемся окне Создание форм в подокне Таблицы и запросы выбрать в качестве источника данных таблицу Ученик. В списке полей таблицы Ученик выбрать из подокна Доступные поля все имеющиеся поля в качестве полей основной формы, нажимая после выбора каждого поля кнопку «>». При этом выделенное поле из Доступного поля будет переноситься в подокно Выбранные поля.
- из подокна Таблицы и запросы выбрать таблицу Ведомость оценок, содержащую данные для подчиненной формы.
- в списке полей таблицы Ведомость оценок выбрать все поля в качестве полей подчиненной формы, кроме поля № личного дела, а затем нажать кнопку «Далее». Выбрать Подчиненные формы и нажать кнопку «Далее». Выбрать вид Табличный и нажать кнопку «Далее».
- выбрать Стиль и нажать кнопку «Далее».

2. Ввести заголовок новой формы Ввод оценок и нажать кнопку «Готово». На экране появится окно с выводом данных из таблиц в виде формы, представленной на рисунке

Название предмета	Оценка 1 четверть	Оценка 2 четверть (1 полугод	Оценка 3 четверть	Оценка 4 четверть (2 полугод	Оценка за год
Алгебра			4		4
Русский язык			4		4
Литература			5		5
Английский язык			4		4

Рисунок 4.3 – Пример формы Ввод оценок

В таблице Ведомость оценок есть поле Название предмета, заполнение которого вручную было бы нецелесообразно. При вводе предмета можно воспользоваться справочной таблицей Список предметов. Для этого необходимо изменить созданную форму Ввод оценок, преобразовав поле Название предмета в поле со списком.

Для изменения формы надо в списке форм выбрать форму Ввод оценок и нажать кнопку «Конструктор». Далее необходимо выполнить следующие действия:

- выделить поле Название предмета и нажать правую кнопку мыши. В контекстном меню выбрать Преобразовать элемент в – Поле со списком;
- выделить поле Название предмета и нажать правую кнопку мыши;
- в контекстном меню выбрать Свойства. В окне Поле со списком: название предмета выбрать вкладку Данные или Все, в строке «Источник строк» щелкнуть левой кнопкой мыши для вывода списка таблиц;
- в списке выбрать таблицу Список предметов и закрыть окно. Сохранить изменения.

3. Создать составную форму для ввода информации о кружках, которые посещает ученик. Для создания формы надо выбрать в окне База данных вкладку Формы, а затем выбрать Создание формы с помощью мастера. Далее осуществить следующие действия:

- В появившемся окне Создание форм в подокне Таблицы и запросы выбрать в качестве источника данных таблицу Ученик. В списке полей таблицы Ученик выбрать из подокна Доступные поля выбрать поля Фамилия, Имя, Отчество, Класс, Буква в качестве полей основной формы, нажимая после выбора каждого поля кнопку «>».

При этом выделенное поле из Доступного поля будет переноситься в подокно Выбранные поля.

- Из подокна Таблицы и запросы выбрать таблицу Кружки, содержащую данные для подчиненной формы.
- В списке полей таблицы Кружки выбрать поля Название кружка и ФИО преподавателя кружка в качестве полей подчиненной формы, а затем нажать кнопку «Далее».
- Выбрать Подчиненные формы и нажать кнопку «Далее».

4. Выбрать вид Табличный и нажать кнопку «Далее». Выбрать Стиль и нажать кнопку «Далее». Ввести заголовок новой формы Посещение кружков и нажать кнопку «Готово». На экране появится окно с выводом данных из таблиц в виде формы, представленной на рисунке

Название кружка	ФИО преподавателя кружка
Компьютерная графика	Смирнова Ирина Павловна
Программирование	Смирнова Ирина Павловна

Рисунок 4.4 - Пример формы Посещение кружков

Закреть форму.

В форме Посещение кружков есть поле Название кружка, заполнение которого вручную было бы нецелесообразно. При вводе названия кружка можно воспользоваться справочной таблицей Список кружков. Для этого необходимо изменить созданную форму Посещение кружков, преобразовав поле Название кружка в поле со списком.

Для изменения формы надо в списке форм выбрать форму Посещение кружков и нажать кнопку «Конструктор». Далее необходимо выполнить следующие действия:

- Выделить поле Название кружка и нажать правую кнопку мыши.
- В контекстнозависимом меню выбрать Преобразовать элемент в – Поле со списком.
- Выделить поле Название кружка и нажать правую кнопку мыши.
- В контекстнозависимом меню выбрать Свойства. В окне Поле со списком: название предмета выбрать вкладку Данные или Все, в

строке «Источник строк» щелкнуть левой кнопкой мыши для вывода списка таблиц.

- В списке выбрать таблицу Список кружков и закрыть окно. Поле ФИО преподавателя кружка тоже целесообразно заполнять с помощью таблицы Учитель.
- Преобразовать поле ФИО преподавателя кружка в поле со списком аналогично предыдущему преобразованию (в качестве источника строк берем таблицу Учитель).
- Сохранить изменения.

5. Создать составную форму для ввода информации о предметах, которые преподает учитель. Для создания формы надо выбрать в окне База данных вкладку Формы, а затем выбрать Создание формы с помощью мастера. Далее осуществить следующие действия:

- В появившемся окне Создание форм в подокне Таблицы и запросы выбрать в качестве источника данных таблицу Учитель.
- В списке полей таблицы Учитель выбрать из подокна Доступные поля выбрать поле Фамилия, Имя, Отчество в качестве поля основной формы.
- Из подокна Таблицы и запросы выбрать таблицу Преподавание, содержащую данные для подчиненной формы. В списке полей таблицы Преподавание выбрать поле Предмет в качестве поля подчиненной формы, а затем нажать кнопку «Далее».
- Выбрать Подчиненные формы и нажать кнопку «Далее». Выбрать вид Табличный и нажать кнопку «Далее».
- Выбрать Стиль и нажать кнопку «Далее». Ввести заголовок новой формы Преподавание и нажать кнопку «Готово». На экране появится окно с выводом данных из таблиц в виде формы, представленной на рисунке 4.5.

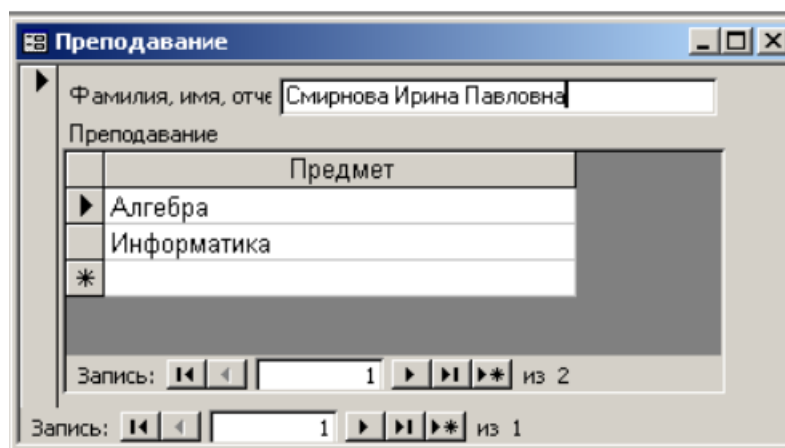


Рисунок 4.5 – Пример формы Преподавание



Поле Предмет тоже целесообразно заполнять с помощью справочной таблицы Список предметов.

- Преобразовать поле Предмет в поле со списком аналогично преобразованию формы Ввод оценок.
- Сохранить изменения.
- Выйти из Access. Выбрать в меню пункты Файл - Выход или нажать кнопку закрытия окна в правом углу окна Access.

## **ЛАБОРАТОРНАЯ РАБОТА 5**

### **ТЕМА: «СОЗДАНИЕ ОТЧЕТОВ В СРЕДЕ СУБД ACCESS»**

Для получения печатного документа в СУБД MS Access предусмотрено несколько возможностей:

- печать таблицы в режиме таблицы (вывод всех записей или записей удовлетворяющих условиям фильтрации в табличном виде);
- печать набора записей запроса в режиме таблицы;
- печать формы в режиме формы или в режиме таблицы;
- печать отчета.

#### **1. Основные сведения об отчетах**

**Отчет** – настраиваемое диалоговое окно, позволяющее организовать форматированное представление данных, предназначенное для вывода на экран, принтер или в файл. В СУБД MS Access отчеты являются объектами базы данных.

Отчеты представляют собой наилучшее средство для представления информации из базы данных в виде печатного документа. По сравнению с другими средствами вывода данных на печать, отчеты обладают 2-мя принципиальными преимуществами:

- предоставляют широкие возможности для группировки, сортировки и вычисления промежуточных и общих итогов для больших наборов записей (вычисления можно производить в разрезе группы, нескольких групп и по отчету в целом);
- позволяют осуществлять вывод данных в виде, приближенном к стандартным бланкам документов.

## Этапы создания отчета.

На первом этапе разрабатывается на бумаге макет будущего отчета. При этом, указывается: из какой таблицы или запроса необходимо отображать данные, определяются уровни группировки и сортировки данных, какие поля должны быть представлены в отчете и какие элементы управления будут использоваться для отображения данных, нужны ли вычисляемые поля, какие графические элементы будут использоваться для придания отчету профессионального вида.

На следующем этапе происходит конструирование отчета средствами СУБД.

## 2. Средства СУБД MS Access для создания отчета

Для создания нового отчета в СУБД MS Access необходимо перейти в окне базы данных на вкладку **Отчеты** и щелкнуть по кнопке **Создать**. В появившемся диалоговом окне **Новый отчет** (рис.5.1) выбрать средство создания отчета.

Все средства, перечисленные на панели инструментов **Отчет**, можно разделить на три основные группы:

- средства автоматического создания отчетов (Автоотчет: в столбец, Автоотчет: ленточный);
- мастера создания отчетов (Мастер отчетов, Мастер диаграмм и Почтовые наклейки);
- конструктор создания отчетов.

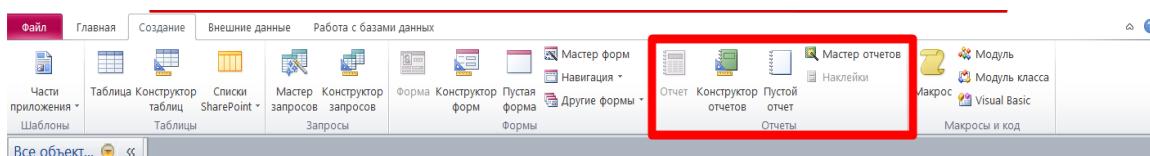





Рисунок 5.1 – Панель инструментов Отчёт

Средства автоматического создания отчетов позволяют на основе выбранного источника данных: таблицы или запроса, создать отчет соответствующего вида.

Мастера создания отчетов позволяют создавать отчеты в режиме диалога с пользователем. С помощью мастеров можно сформировать источник данных для отчета, выбрать вид представления данных, определить поля для группировки, сортировки и подведения итогов, выбрать итоговые функции по группе, определить макет отчета и стиль оформления, а также задать имя отчета.

Конструктор отчетов является основным средством позволяющим создавать отчеты любой сложности вручную. Вид отчета и его содержание зависят от квалификации разработчика отчета.

Обычно для создания прототипа отчета используются автоматические средства создания отчетов или мастера, а уже в конструкторе отчетов вид отчета доводится до профессионального вида.

Работа с отчетами может происходить в трех режимах: в режиме **Предварительный просмотр** , в режиме **Образец**  и в режиме **Конструктор** .

Режим **Предварительный просмотр** предназначен для формирования отчета и вывода всех данных из источника записей на экран, режим **Образец** предназначен для быстрого формирования отчета на основе механической подстановки данных из нескольких записей источника данных (критерии выборки и сортировки игнорируются), режим **Конструктор** предназначен для создания и изменения структуры отчета. Для изменения режима работы с отчетом можно воспользоваться кнопкой **Вид** на панели инструментов **Конструктор отчетов** или аналогичной командой меню **Вид**.

### 3. Создание отчета в режиме Конструктор

Для создания отчета в режиме **Конструктор** необходимо выполнить следующие действия:

1. Настроить параметры страницы.
2. Выбрать или создать источник записей для отчета.
3. Определить уровни группировки и сортировки.
4. Разместить поля источника записей в отчете.
5. Добавить вычисляемые и свободные элементы управления.
6. Настроить свойства элементов управления и разделов отчета.
7. Отформатировать отчет с целью придания профессионального вида.
8. Сохранить отчет, задав имя.

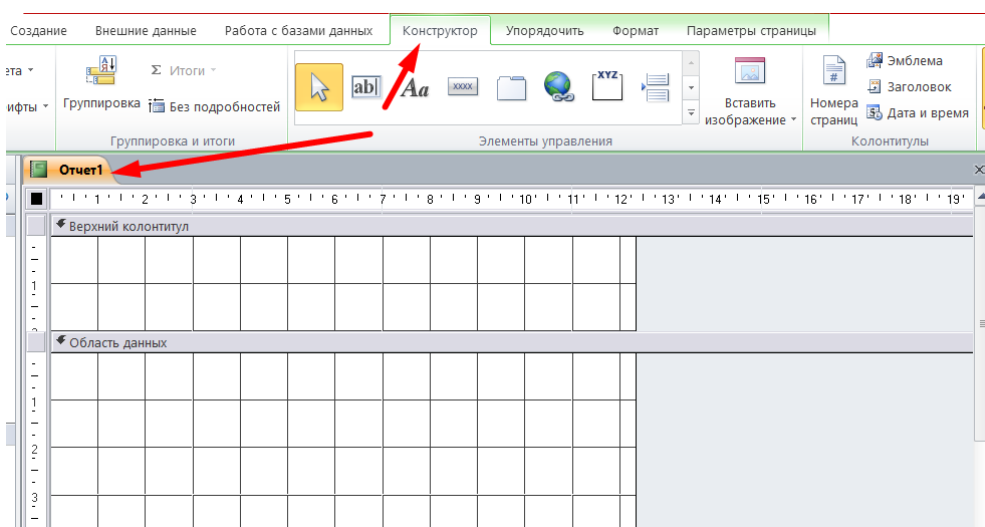


Рисунок 5.2 – Создание отчёта в режиме конструктора

### 3.1. Настройка параметров страницы отчета

Основное назначение отчета – вывод на печать информации из базы данных в форматированном виде (приблизенном к бланку бумажного документа). Поэтому прежде чем конструировать отчет (размещать в отчете элементы управления) необходимо настроить параметры страницы отчета (выбрать принтер, определить размер листа и его ориентацию, установить размеры полей и т. п.).

При физическом отсутствии принтера необходимо установить принтер программно с помощью команды операционной системы **Пуск | Настройка | Принтеры | Установка принтера**.

Для настройки параметров страницы отчета необходимо выполнить команду **Файл | Параметры страницы ...**. Диалоговое окно **Параметры страницы** представлено на рис.5.3.

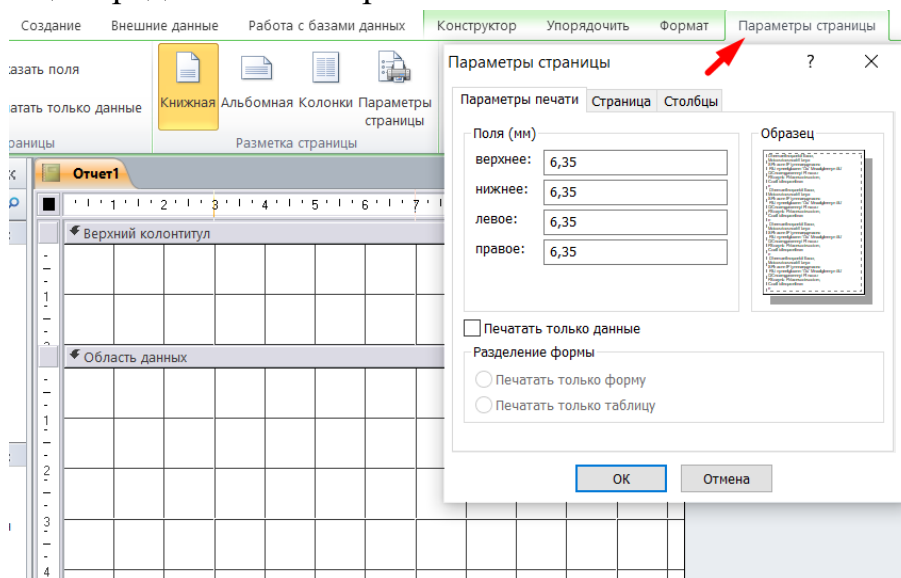


Рисунок 5.3 – Настройка параметров страницы

Параметры страницы настраиваются для каждого отчета.

Если существует возможность вывода отчетов на различные принтеры, то в диалоговом окне **Параметры страницы** на вкладке **Страница** можно выбрать необходимый тип принтера. Диалоговое окно **Выбор принтера** представлено на рис.5.4.

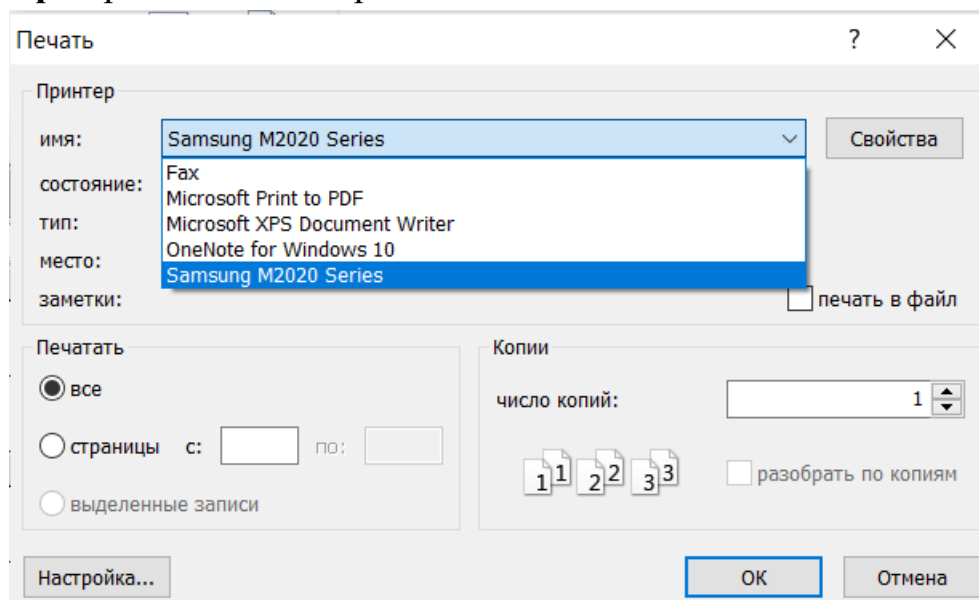


Рисунок 5.4 – Диалоговое окно Выбор принтера

### 3.2. Структура отчета

Структура отчета в режиме **Конструктор** представлена на рис. 5.5. Любой отчет имеет следующие разделы:

– **заголовок отчета** (выводится только на 1-м листе отчета (титальном листе) и может содержать наименование отчета, логотип фирмы, юридический адрес предприятия и т. п.);

– **верхний колонтитул** (выводится в верхней части каждой страницы и может содержать заголовки, дату печати, номер страницы и т. п.);

– **область данных** (предназначена для вывода полей из запроса или таблицы – источника записей для отчета);

– **нижний колонтитул** (выводится в нижней части каждой страницы и может содержать заголовки, дату печати, номер страницы и т. п.);

– **примечание отчета** (выводится только на последнем листе отчета и может содержать итоговые расчеты, выводы по отчету и т. п.).

Из всех перечисленных разделов обязательным является раздел – область данных.

Разделы заголовок отчета и примечание отчета отображаются/скрываются в окне **Конструктора** одновременно. Такой же

принцип отображения/скрытия и у разделов верхний и нижний колонтитул (см. рис.10.4).

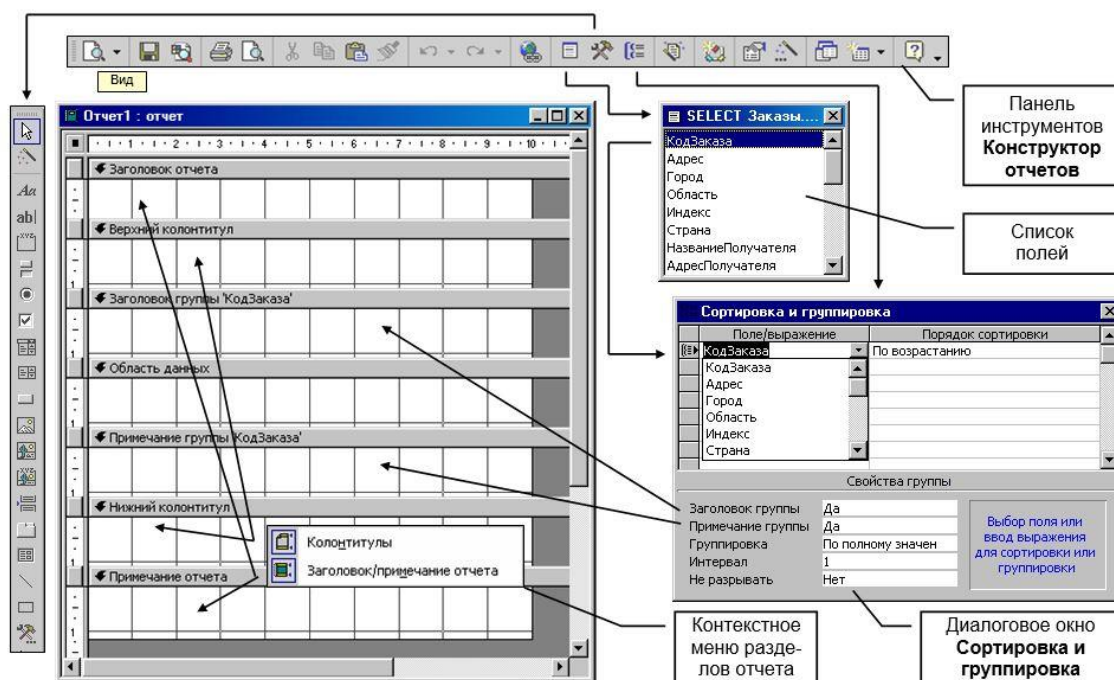


Рисунок 5.5 – Разделы отчёта

При группировке записей в окно **Конструктора** отчетов (см. рис.5.5) могут быть добавлены для каждой группы, а в отчетах можно определить до 10 уровней группировки и сортировки, независимо следующие разделы:

- **заголовок группы** (выводится в начале каждой новой группы и может содержать поле, по которому проводится группировка);
- **примечание группы** (выводится в конце каждой группы и может содержать итоговые значения по группе).

Высоту области любого раздела отчета можно изменять с помощью мыши, перетаскивая нижнюю часть раздела вверх или вниз.

Ширину отчета настраивают исходя из выбранного размера бумаги, ориентации листа и размеров печатных полей.

Ширина отчета не должна превышать расчетного размера, т. к. информация, выводимая в отчете, будет разорвана на нескольких страницах.

Изменить ширину отчета можно с помощью мыши, перетаскивая правую границу отчета влево или вправо.

Также как и в форме, в любой раздел отчета можно вставлять различные элементы управления: присоединенные, вычисляемые и свободные. Выражения в вычисляемых полях отчета формируются по тем же правилам, что и для форм:

$$= [\text{Цена}] * [\text{Количество}]$$

= SUM([Стоимость])  
 = Фамилия & " " & Left(Имя; 1) & "."

С помощью элемента управления **Подчиненная форма/отчет** в главный отчет можно внедрять подчиненные отчеты.

Для каждого элемента управления, раздела и отчета в целом могут быть уточнены свойства.

Сформированный отчет **Бланк заказа** в режиме **Конструктор** и режиме **Предварительный просмотр** представлен на рис.5.6.

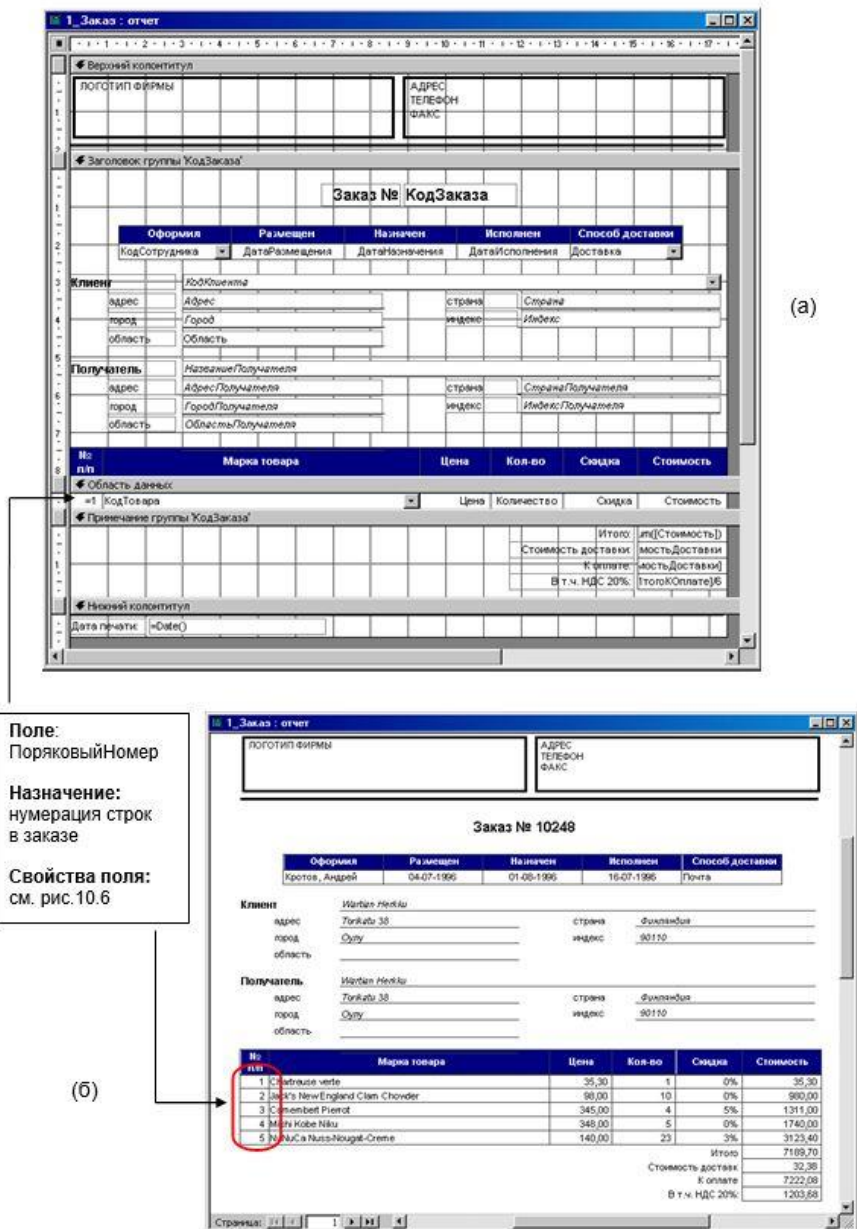


Рисунок 5.6 – Отчет Бланк заказа в режиме Конструктор (а) и в режиме Предварительный просмотр (б)

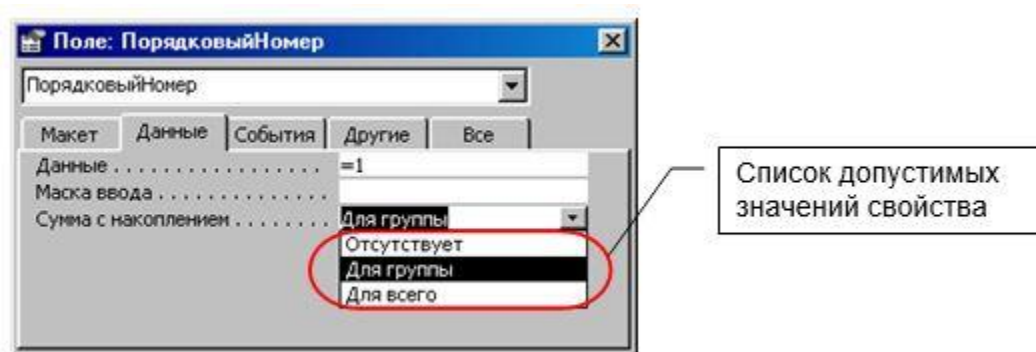


Рисунок 5.7 – Окно свойств поля Порядковый Номер

Для вывода каждой группы (бланка заказа) на отдельной странице необходимо установить в окне свойств раздела **Примечание группы "КодЗаказа"** на вкладке **Макет** значение свойства **Конец страницы: После раздела** (рис.5.8).

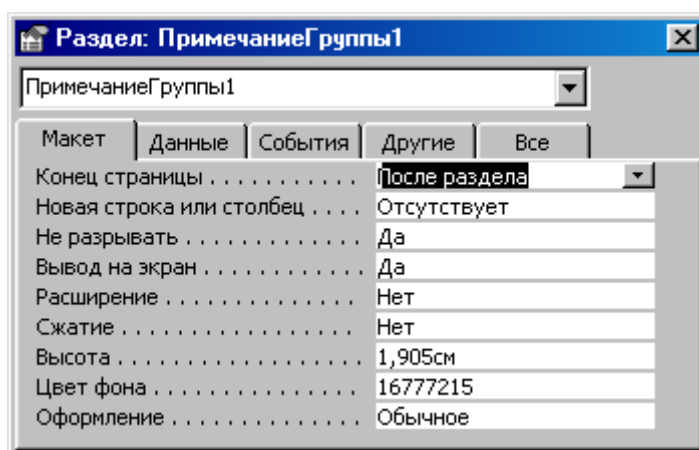


Рисунок 5.8 – Окно свойств раздела ПримечаниеГруппы1

### 3.3. Форматирование отчета

Одним из элементов окна **Конструктора** отчетов является панель инструментов **Формат форма/отчет**, содержащая кнопки и раскрывающиеся списки, облегчающие выбор параметров форматирования (цвета, шрифта, толщины линий, способа выравнивания и т. п.).

Под форматированием понимается и размещение элементов управления в разделах отчета, выравнивание элементов управления, изменение размеров элементов управления, установка интервалов между элементами управления. Для выполнения эти операций можно использовать мышшь или воспользоваться командами меню **Формат (Выровнять ..., Размер ..., Интервал по вертикали ... и Интервал по горизонтали ...)**.

Для более быстрого выполнения операций по форматированию отчета можно настроить панель инструментов **Формат форма/отчет**, скопировав на нее кнопки из меню **Формат**.



## Задание

1. Открыть базу данных, созданную в лабораторной работе 2 и соответствующую вашему варианту задания (варианты заданий приведены в лабораторной работе №2).
2. Спроектировать формы для всех таблиц вашей базы данных. В каждой форме должен присутствовать заголовок. Формы должны быть спроектированы как в ленточном виде, так и в один столбец.
3. Запретить доступ к одному из элементов управления. Для этого надо открыть форму в конструкторе; выделить элемент; установить свойства: “Доступ” – “Нет” и “Блокировка” – “Да”.
4. Создать формы для основных запросов. Использовать подчиненные формы для таблиц, связанных отношением один ко многим.
5. Спроектировать отчеты для всех таблиц вашей базы данных.
6. Создать отчет с группировкой данных.
7. Подготовить отчет по работе.

## ЛАБОРАТОРНАЯ РАБОТА 6

### ТЕМА: «ПОСТРОЕНИЕ ДИАГРАММ В СРЕДЕ СУБД ACCESS»

Файл базы данных содержит информацию о результатах олимпиады:

Страна	золото	серебро	бронза
Беларусь	3	3	11
Германия	14	17	15
Италия	13	8	13
Казахстан	3	4	0
Канада	3	3	8
Китай	28	16	15
Россия	32	28	28
США	39	25	33
Украина	3	10	10
Франция	13	14	11
Чехия	2	3	3
Шри-Ланка	0	0	1

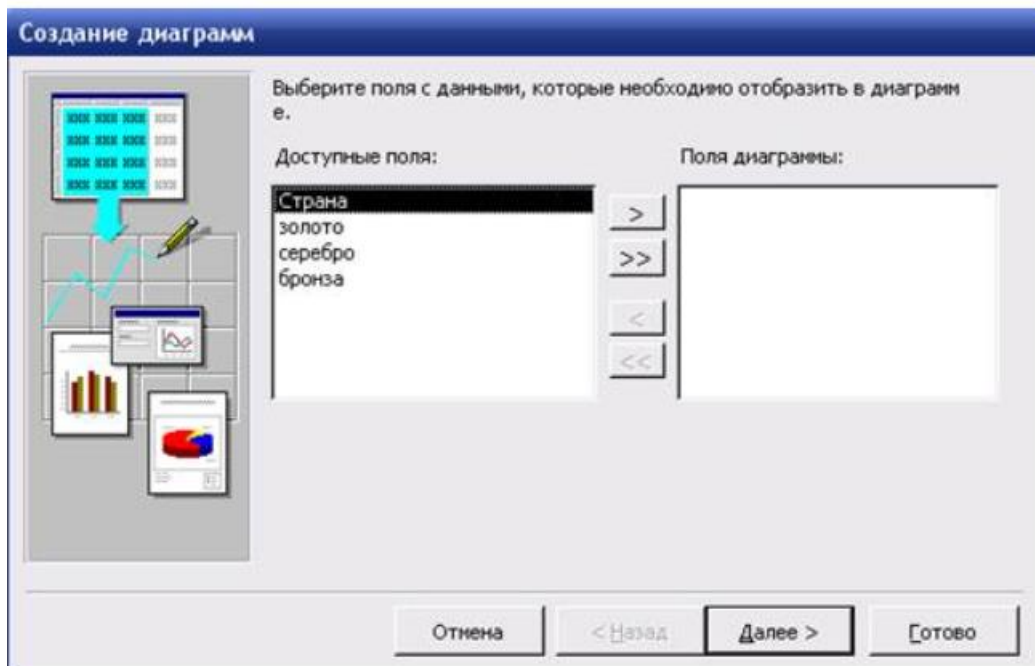
Рисунок 6.1 – Таблица базы данных

С помощью мастера диаграмм построить диаграмму (объёмную гистограмму), отображающую количество медалей, полученных спортсменами разных стран.

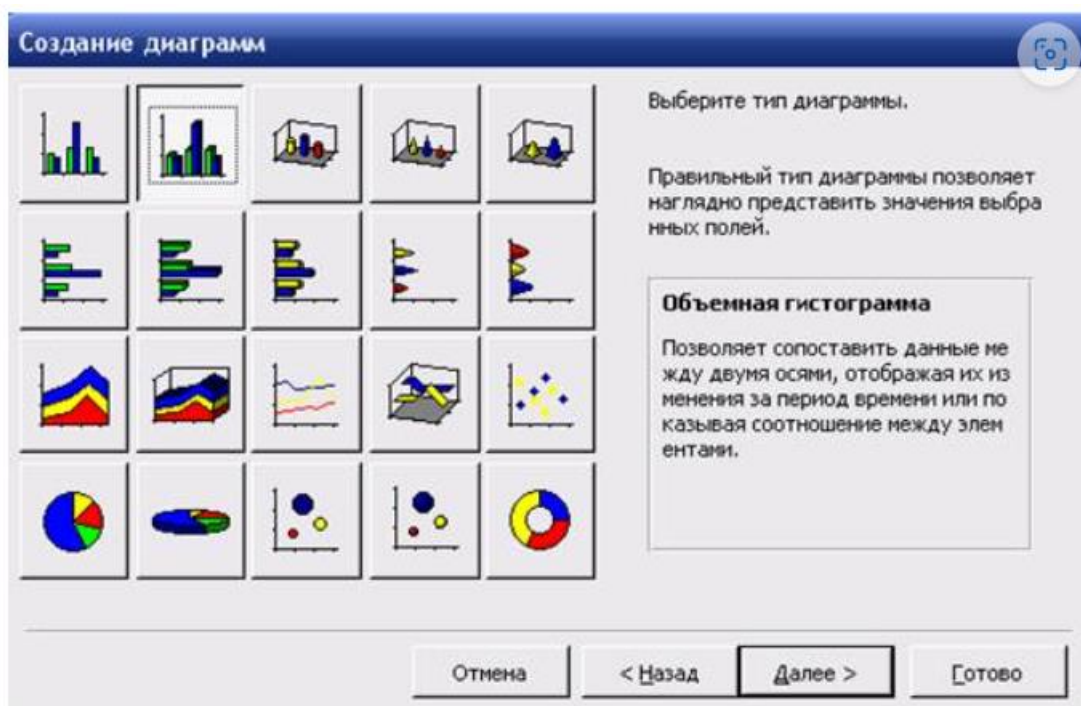
1. Создайте таблицу базы данных, представленную на рисунке 6.1.
2. В списке Объекты выбрать Отчёты, нажать кнопку Создать, в диалоговом окне Новый отчёт выбрать вариант Мастер диаграмм и выбрать в качестве источника данных таблицу Олимпиада и нажать

кнопку ОК.

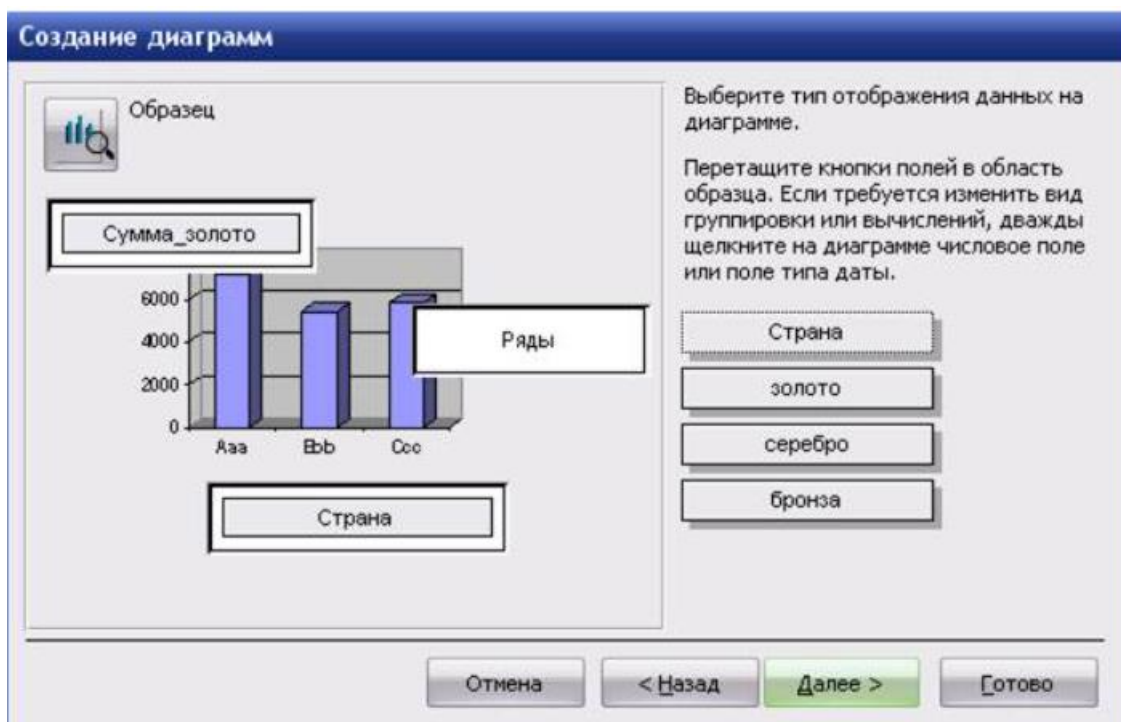
3. Выбрать поля для отображения в диаграмме с помощью кнопки>>.



4. Выбрать тип диаграммы Объёмная гистограмма.

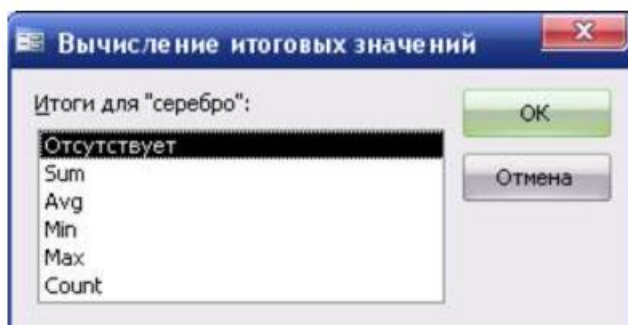


5. Выбрать способ отображения данных на диаграмме. Для этого в открывшемся окне путём перетаскивания поместить поле золото, бронза в область Данные, а поле Страна – в область Оси.

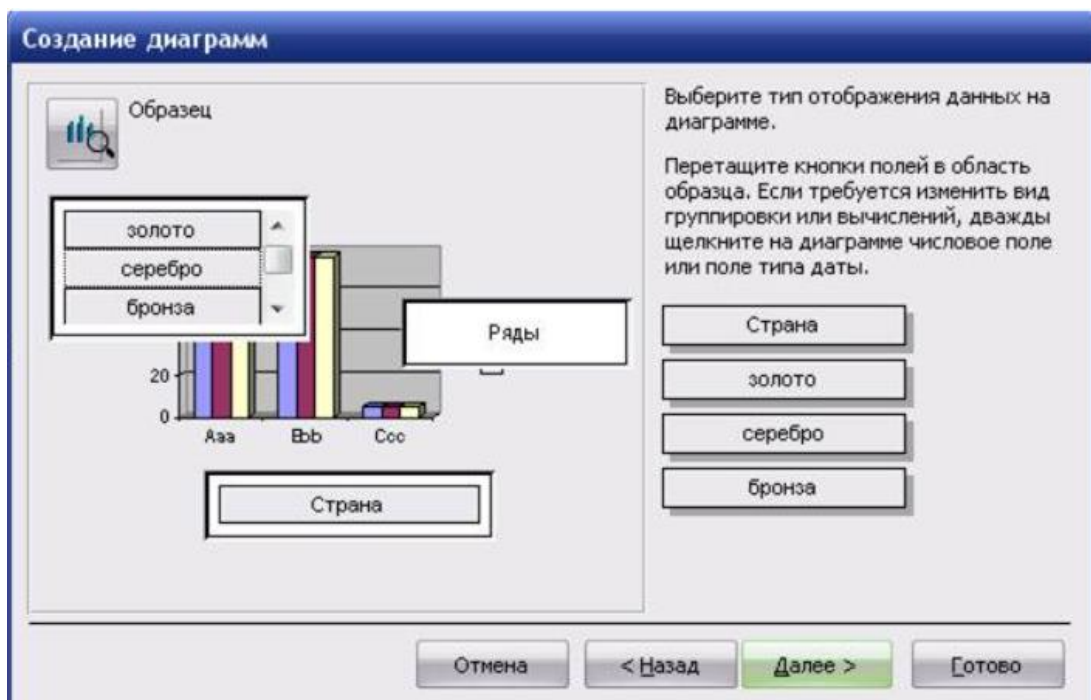


В области Данные двойным щелчком по полю серебро открыть диалоговое окно Вычисление итоговых значений и в списке Итоги для «серебро» выбрать вариант Отсутствует.

Аналогичные действия выполнить для полей золото и бронза.

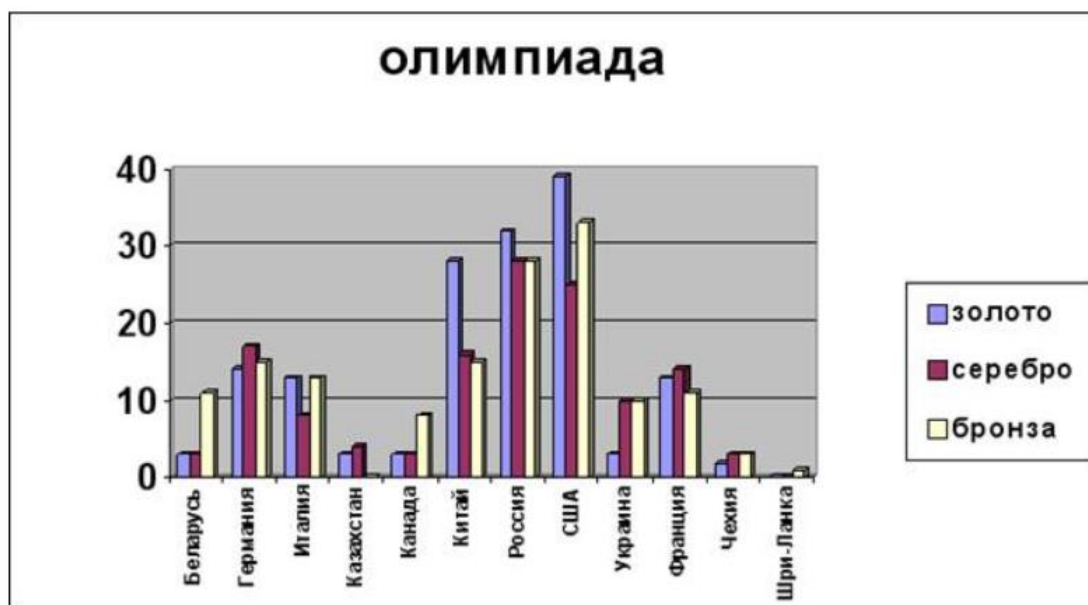


6. В результате диалоговое окно примет вид:



Дать диаграмме заголовок Олимпиада.

7. В результате диаграмма примет вид:



При необходимости диаграмму можно отредактировать в режиме конструктора.

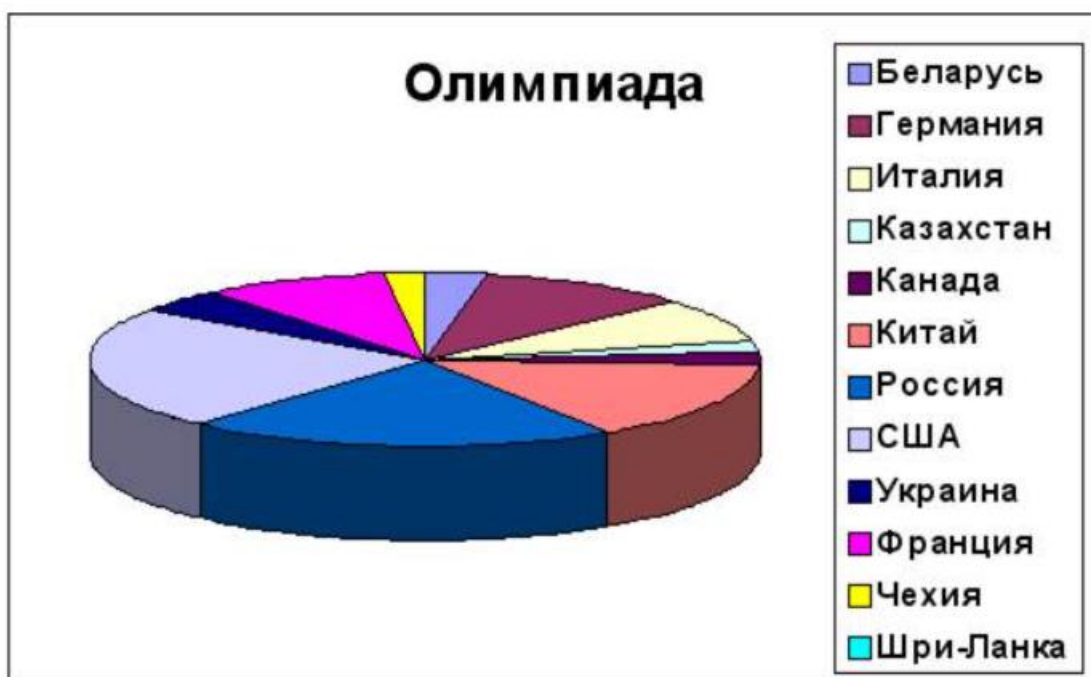
8. Дополнить отчёт данными таблицы.

### Задание.

На базе таблицы Олимпиада файла базы данных создать запрос, который подсчитывает количество очков, набранных каждой страной-

участницей (за золотую медаль насчитывается 3 очка, за серебряную – 2, за бронзовую – 1). По запросу построить круговую диаграмму, которая показывает, сколько очков получила каждая страна.

1. Открыть файл базы данных.
  2. Создать запрос, вычисляющий количество очков, набранных каждой страной, по формуле  
Очки:  $3 * [\text{золото}] + 2 * [\text{серебро}] + [\text{бронза}]$ .
  3. Построить по запросу круговую диаграмму, показывающую количество очков, набранное каждой страной.
- В результате фрагмент отчёта с диаграммой примет вид:



4. Добавить отчёт в таблицу данных.

## ЛАБОРАТОРНАЯ РАБОТА 7

### ТЕМА: «МАКРОСЫ И ПРОЦЕДУРЫ В СРЕДЕ СУБД ACCESS»

В предыдущих лабораторных работах рассматривалось создание базы данных с точки зрения пользователя среды Microsoft Access. Однако Access содержит дополнительные средства, доступные для использования программисту: макросы (небольшие программы на языке макрокоманд системы Access) и модули (процедуры на языке Visual Basic for Application, VBA).

*Использование макросов в базе данных Microsoft Access.*

С помощью макросов можно выполнить практически все действия над объектами Access из тех, которые были описаны в предыдущих работах.

Макрос в Access представляет собой структуру, состоящую из одной или нескольких макрокоманд, которые выполняются либо последовательно, либо в порядке, заданном определенными условиями. Набор макрокоманд в Access очень широк, с помощью макросов можно реализовать многое из того, что позволяют сделать процедуры на VBA. Каждая макрокоманда имеет определенное имя и, возможно, один или несколько аргументов, которые задаются пользователем. Например, при использовании макрокоманды ОткрытьФорму (OpenForm) в качестве аргументов необходимо задать, по крайней мере, имя открываемой формы и режим вывода ее на экран.

В табл. 7.1 содержится список макрокоманд Microsoft Access, сгруппированных по категориям: работа с данными в формах и отчетах; выполнение команд, макросов, процедур и запросов; работа с объектами; импорт/экспорт данных и объектов и прочие. В категорию "Прочие" собраны все макрокоманды, которые позволяют влиять на интерфейс приложения.

Таблица 7.1 Макрокоманды Microsoft Access

<b>Категория</b>	<b>Назначение</b>	<b>Макрокоманда</b>
<b>Работа с данными в формах и отчетах</b>	Отбор данных	Применить Фильтр (ApplyFilter)
	Перемещение по данным	СледующаяЗапись (FindNext), НайтиЗапись (FindRecord), КЭлементу-Управления (GoToControl), НаСтраницу (GoToPage), НаЗапись (GoToRecord)
	Обновление данных или экрана	Обновление (Requery), ПоказатьВсеЗаписи (ShowAllRecords)
<b>Выполнение</b>	Выполнение команды	Выполнить Команду (RunCommand)
	Выполнение макроса, процедуры или запроса	ЗапускМакроса (RunMacro), Запуск Программы (RunCode), ОткрытьЗапрос (OpenQuery), ЗапускЗапроса SQL (RunSQL)
	Выполнение другого приложения	Запуск Приложения (RunApp)
	Прерывание выполнения	ОтменитьСобытие (CancelEvent), ОстановитьВсеМакросы (StopAllMacros), ОстановитьМакрос (StopMacro)

<b>Им-порт/экспорт</b>	Передача объектов Microsoft Access в другие приложения	Вывести В Формате (OutputTo), ОтправитьОбъект (SendObject)
	Преобразование формата данных	Преобразовать БазуДанных (TransferDatabase), ПереносБазыДанныхSQL (TransferSQLDatabase), Преобразовать ЭлектроннуюТаблицу (TransferSpreadsheet), ПреобразоватьТекст (TransferText)
<b>Работа с объектами</b>	Копирование, переименование и сохранение объекта	КопироватьОбъект (CopyObject), Копировать ФайлБазыДанных (CopyDatabaseFile) Переименовать (Rename) Сохранить (Save)
<b>Прочие</b>	Вывод на экран или скрытие встроенной или специальной панели инструментов Подача звукового сигнала	ПанельИнструментов (ShowToolbar) Сигнал (Beep)

Как видно из этой таблицы, действия, которые могут быть выполнены с помощью макросов, очень разнообразны. Все эти макрокоманды можно использовать в процедурах VBA.

Отметим особенности некоторых макрокоманд:

1. **ПереносБазыДанныхSQL (TransferSQLDatabase)** – переносит базу данных Microsoft SQL Server 7.0 и выше на другой сервер.

2. **КопироватьФайлБазыДанных (CopyDatabaseFile)** – выполняет копирование текущей базы данных Microsoft SQL Server 7.0 и выше, присоединенной к проекту Access.

3. **ОткрытьФункцию (OpenFunction)** – выполняет функцию, определенную пользователем, в проекте Access. Эта функция представляет собой запрос, который, используя входные параметры, возвращает результаты так же, как и хранимая процедура. Отличительной чертой макросов, в отличие от VBA, является легкое их создание, т. е. не требуется знание синтаксиса языка программирования. Для создания макроса требуется знать лишь основные приемы работы в Microsoft Access и Windows, такие как перетаскивание объектов из окна База данных (Database) в специальное окно – Конструктор макросов (Macro Design), выбор действия из списка и ввод выражений в качестве аргументов макрокоманды. Основное назначение макросов – это создание удобного интерфейса приложения: чтобы формы и отчеты открывались при нажатии кнопок в форме или на панели инструментов или же привычным выбором команды меню; чтобы при открытии приложения пользователь видел на экране не окно База данных (Database), наполненное множеством таблиц, запросов, форм и отчетов, а

некую понятную форму, с помощью которой можно было бы сразу производить желаемые действия и т. д.

Однако использование макросов имеет и некоторые недостатки:

1. Возможности макрокоманд ограничены по сравнению с возможностями языка VBA, поэтому в ряде случаев без программирования на VBA не обойтись. Язык VBA предоставляет более широкие возможности для работы с данными, позволяет использовать механизм программирования объектов для связи с другими приложениями, вызывать функции из библиотек динамической загрузки (DLL) Windows и создавать собственные специализированные функции.

2. Макросы можно использовать практически везде, где применяются процедуры VBA, однако процедуры VBA, как правило, выполняются быстрее.

3. Макросы являются объектами, существующими отдельно от форм и отчетов, в которых они используются, поэтому, когда этих объектов становится очень много, их поддержка достаточно трудоемка. Процедуры обработки событий VBA являются неотъемлемой частью форм и отчетов. Например, при переносе форм и отчетов из одной базы данных в другую с ними автоматически переносятся связанные процедуры.

**Создание макросов.** В Microsoft Access существует специальное средство для создания макросов – Макрос (Macro Design) (рис. 7.1). Фактически это Конструктор, в котором выбирается последовательность действий. Вызывается он через раздел Создание и выбора элемента Макрос.

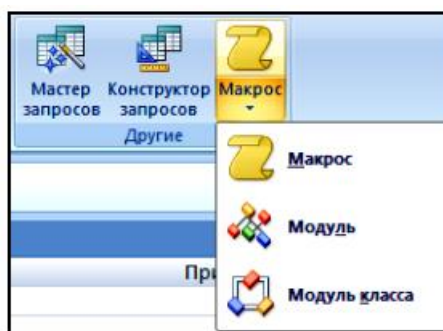


Рисунок 7.1 – Макрос

Окно Конструктора макросов построено аналогично окну Конструктора таблиц, то есть разделено по горизонтали на две части: панель описаний и панель аргументов (рис. 7.2).





выполнения перейдите в раздел Конструктор (работа с макросами), выберите Выполнить (рис. 7.4).

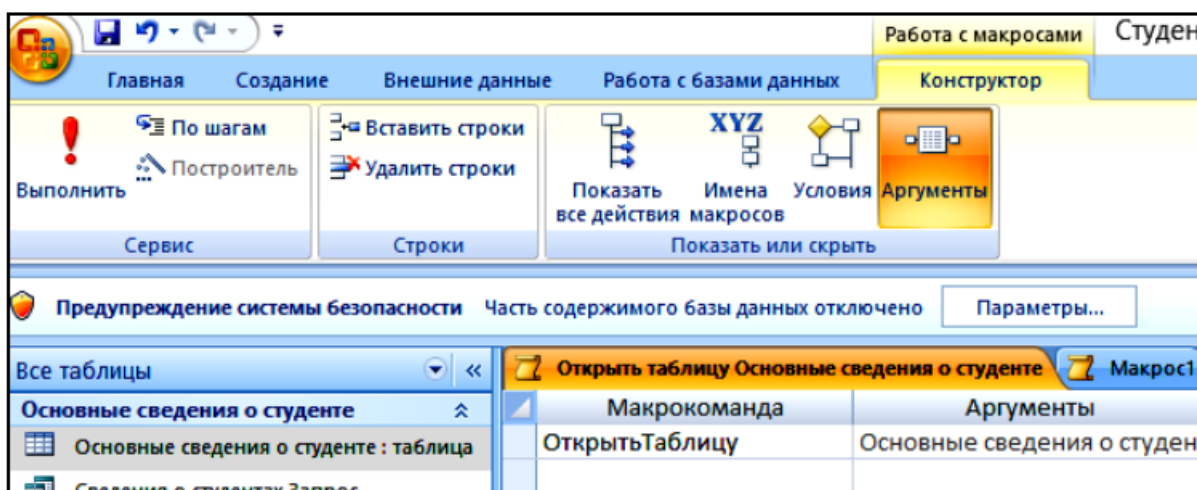


Рисунок 7.4 – Конструктор макросов

После сохранения макроса и его последующего выполнения автоматически откроется форма, в которой можно будет работать с данными.

Для ввода аргументов макрокоманды чаще всего требуется выбирать значения из списков или вводить выражения. Для ввода выражений можно воспользоваться Построителем выражений. Вызов возможен через Условие по нажатию правой кнопки мыши в поле и выборе нужного элемента (рис. 7.5).

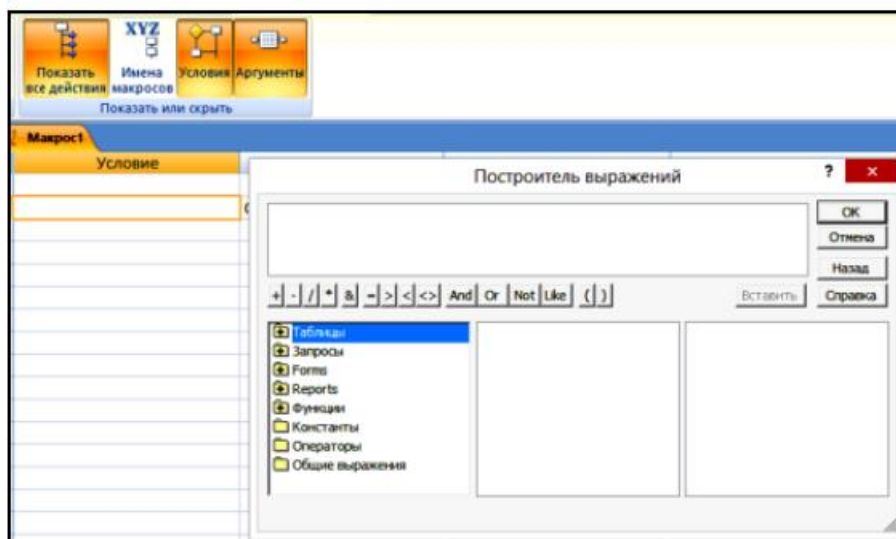


Рисунок 7.5 – Построитель выражений

Если в качестве аргумента макрокоманды необходимо задать имя объекта базы данных, то его можно либо ввести с помощью клавиатуры, либо выбрать из раскрывающегося списка, либо указать название объекта, перетащив его из окна базы данных.

Как уже было сказано ранее, макрос может содержать последовательность из нескольких макрокоманд. Поэтому в этом же макросе в следующей строке мы выберем макрокоманду Сообщение (MsgBox), которая после открытия формы будет отображать диалоговое окно с сообщением о том, какая именно таблица была открыта. Для этого введем в поле Сообщение (Message) на панели аргументов текст сообщения, например «Открыта таблица Основные сведения о студенте», остальные поля можем оставить без изменения. Аргумент Сигнал (Веер) определяет, будет ли вывод сообщения сопровождаться звуковым сигналом. Аргумент Тип (Type) определяет вид диалогового окна сообщения. Аргумент Заголовок (Title) позволяет задать текст, который будет выведен в заголовке окна сообщения (рис. 7.6).

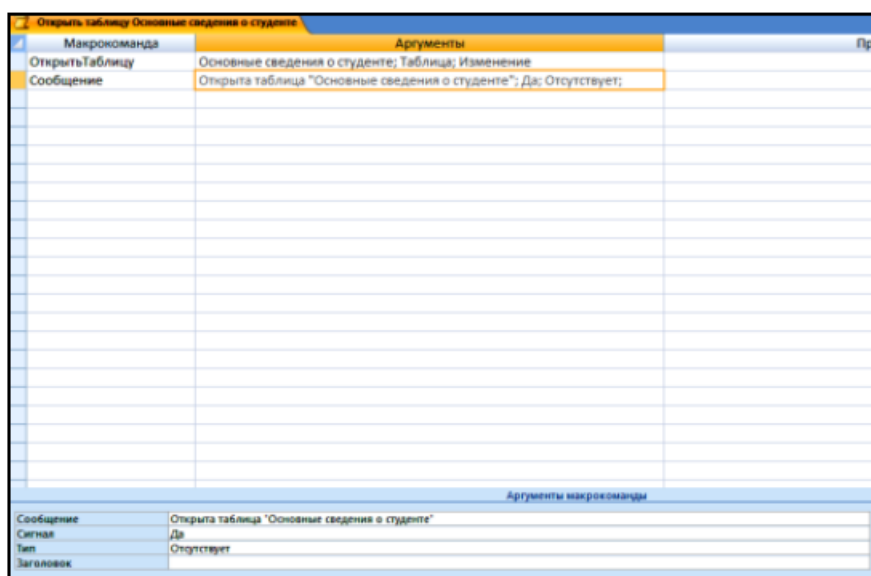


Рисунок 7.6 – Создание последовательности макрокоманд

Результат выполнения представлен на рис. 7.7.

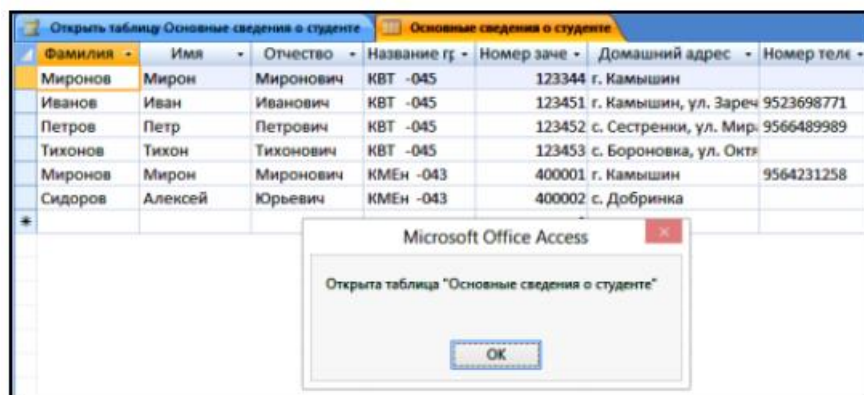


Рисунок 7.7 – Выполнение макроса

Если база данных большая, отбор записей в форме может занять

некоторое время, поэтому перед выполнением открытия формы можно вывести на экран курсор в форме песочных часов, который будет показывать, что идет обработка данных. Для того, чтобы сделать это, воспользуемся макрокомандой ПесочныеЧасы (Hourglass). Макрокоманды в макросе выполняются в той последовательности, в которой они записаны в столбце Макрокоманда (Action), т. е. сверху вниз. Макрокоманду ПесочныеЧасы (Hourglass) следует поставить первой, т. к. курсор должен измениться перед выводом формы. Таким образом, нам нужно вставить новую строку на панели описаний. Для этого установим курсор в первую строку панели и нажмем кнопку Добавить строки (Insert Rows) на панели инструментов или просто клавишу <Insert>. Появится пустая строка. В этой строке в поле Макрокоманда (Action) выберем макрокоманду ПесочныеЧасы (Hourglass). В поле Включить (Hourglass On) панели аргументов появится значение Да (Yes), установленное по умолчанию, что нам и нужно. Чтобы вернуть первоначальный вид указателю мыши, в конце созданного макроса нужно ввести такую же макрокоманду ПесочныеЧасы (Hourglass), но со значением Нет (No) аргумента Включить (Hourglass On). Поставим эту макрокоманду после открытия формы. В общем виде полученная последовательность макрокоманд будет выглядеть следующим образом (рис. 7.8).

Макрокоманда	Аргументы
ПесочныеЧасы	Да
ОткрытьТаблицу	Основные сведения о студенте; Таблица; Изменение
ПесочныеЧасы	Нет
Сообщение	Открыта таблица "Основные сведения о студенте"; Да; Отсутствует;

Рисунок 7.8 – Последовательность макрокоманд

Сохраните макрос и выполните его.

#### *Применение условий в макросах.*

Как и в любой программе, порядок выполнения макрокоманд в макросе может быть изменен. Для этого вводятся условия выполнения или пропуска макрокоманд. Эти условия задаются в виде выражений в специальном столбце, который появляется в окне Конструктора макросов, если нажата кнопка Условия (Conditions) на панели инструментов (см. рис. 7.5).

Если требуется при истинности условия выполнить сразу несколько макрокоманд, то для всех макрокоманд, кроме первой, в столбце Условие (Condition) ставится многоточие (...) (см. рис. 7.5). Условие, которое должно проверяться, пишется в строке первой макрокоманды. Тогда, если оно истинно, выполняется весь набор макрокоманд от этого условия и до следующей макрокоманды с заданным условием, до следующего макроса

или до конца макроса. Если условие ложно, пропускаются все команды, помеченные многоточием, включая макрокоманду с заданным условием. Далее выполняется макрокоманда, следующая за пропущенными, в которой содержится новое условие или поле Условие (Condition) не заполнено. Таким образом, в отличие от большинства языков программирования, в макросах нет альтернативного ветвления. Для того чтобы создать макрос с двумя альтернативными ветвями, нужно сначала ввести условие и определить макрокоманды, исполняемые при выполнении этого условия. Сразу после них необходимо указать обратное условие и определить макрокоманды, исполняемые в альтернативной ветви. Затем можно ввести макрокоманды, которые исполняются безусловно.

Допускается использование макрокоманды ОстановитьМакрос (StopMacro). Она позволяет прекратить выполнение макроса, при этом все макрокоманды, следующие по порядку за макрокомандой ОстановитьМакрос (StopMacro), не будут выполняться.

*Запуск макроса при открытии базы данных.*

При открытии базы данных Microsoft Access могут выполняться некоторые действия. Чаще всего это открытие специальной формы, так называемой Главной кнопочной формы, вывод специальных меню или панелей инструментов, скрывание стандартных меню и т. д. Для определения этих действий используется диалоговое окно Параметры запуска (Startup). Однако иногда требуется при запуске приложения выполнить более сложный набор действий, чем тот, что позволяет задать это диалоговое окно. Например, может потребоваться заранее открыть несколько форм (не показывая их на экране), чтобы потом их отображение не занимало много времени, проверить некоторые условия или запросить ввод некоторых данных. Все это можно сделать с помощью специального макроса, который называется «AutoExec» (рис. 7.9). При открытии базы данных Access проверяет наличие этого макроса и, если он существует, выполняет его.

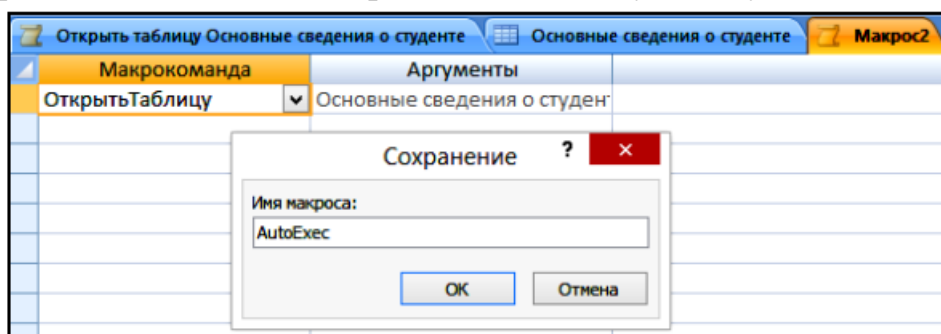


Рисунок 7.9 – Макрос AutoExec

Перечислены только некоторые из способов запуска макросов.

Дополнительно можно вызывать макросы через комбинацию «горячих» клавиш.

#### *Назначение макроса событию.*

Наиболее часто макросы используются в приложении Access для обработки событий. Событие – это любое действие, распознаваемое объектом, и можно определить реакцию объекта на событие. События происходят в результате действий пользователя, выполнения инструкций VBA или генерируются системой. Примером событий является вывод на экран формы, отчета, ввод данных в текстовое поле, нажатие кнопки мыши или клавиши. Каждому из этих событий можно назначить макрос или процедуру VBA, которые будут автоматически выполняться в ответ на произошедшее событие. Практически все программирование в Access сводится к написанию макросов или процедур, обрабатывающих события, т. е. программируется реакция объектов на события. Существует большое количество различного рода событий, на которые реагируют объекты, причем часто возникает не одно, а целая последовательность событий. Поэтому необходимо обладать определенным умением, чтобы решить, какому событию следует назначить созданный вами макрос или процедуру VBA.

#### **Задание.**

Создайте новую форму – «Основная форма». Сделайте ее название «База данных Институт». Добавим еще одну кнопку, при нажатии которой должна открываться форма «Основные сведения о студенте форма».

Для назначения макроса событию Нажатие кнопки (OnClick) сделайте на созданной форме следующие операции:

1. Откройте форму «Основная форма» в режиме Конструктора.
2. Создайте кнопку в области заголовка формы.
3. Откройте окно свойств только что созданной кнопки, если оно еще не открыто, и раскройте вкладку События (Event).

4. Обратите внимание, сколько разных событий связано только с командной кнопкой. Помимо обычного нажатия, в набор событий кнопки входят получение и потеря фокуса, двойной щелчок кнопкой мыши, простое перемещение указателя мыши над кнопкой и пр. Такое многообразие событий дает разработчику большие возможности по созданию удобного интерфейса пользователя. Теперь найдите в списке событие Нажатие клавиши (OnClick) и установите курсор в соответствующую ячейку. Оно может быть пустым или уже связано с выполнением макроса, если при создании кнопки выбрали действие (рис. 7.10).

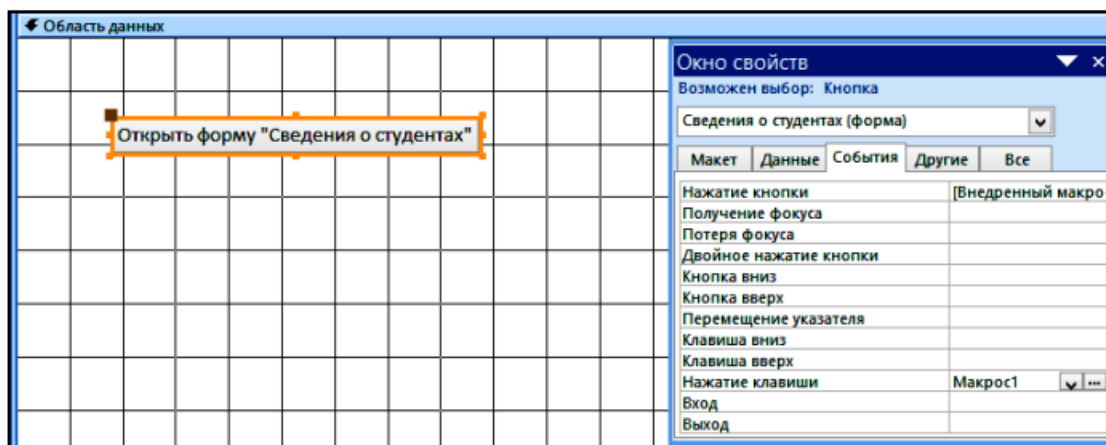


Рисунок 7.10 – Кнопка и её свойства (События)

5. В случае, если нужного нам макроса в списке нет, то можно сразу создать его. Для этого нажмите кнопку Построителя, находящуюся справа от поля. Появится окно Построитель (Choose Builder), в котором предлагается выбрать один из трех Построителей: Выражения (Expression Builder), Макросы (Macro Builder) и Программы (Code Builder). Далее выбирается создание макроса и уже переходим к непосредственному его созданию.

6. В нашем случае уже имеется макрос, который обеспечивает открытие формы.

После этого у нас выполнятся все действия, описанные в макросе.

Примечание: При переходе от одной формы к другой возможно закрытие «старой» формы, а при «ненужности» новой, предусмотреть ее закрытие и возврат к предыдущей.

### Задание.

Попробуйте реализовать этот переход самостоятельно (возможно добавление в форму кнопки возврата на предыдущую форму).

### *Отладка макросов и поиск ошибок.*

Обычно для отладки макроса его выполняют в пошаговом режиме. Для включения этого режима необходимо нажать кнопку По шагам (Single Step) на панели инструментов в окне Конструктора макросов выбрать команду меню Запуск, По шагам (Run, Single Step). Тогда перед выполнением каждой макрокоманды будет появляться диалоговое окно Пошаговое исполнение макроса (Macro Single Step) (рис. 7.11).

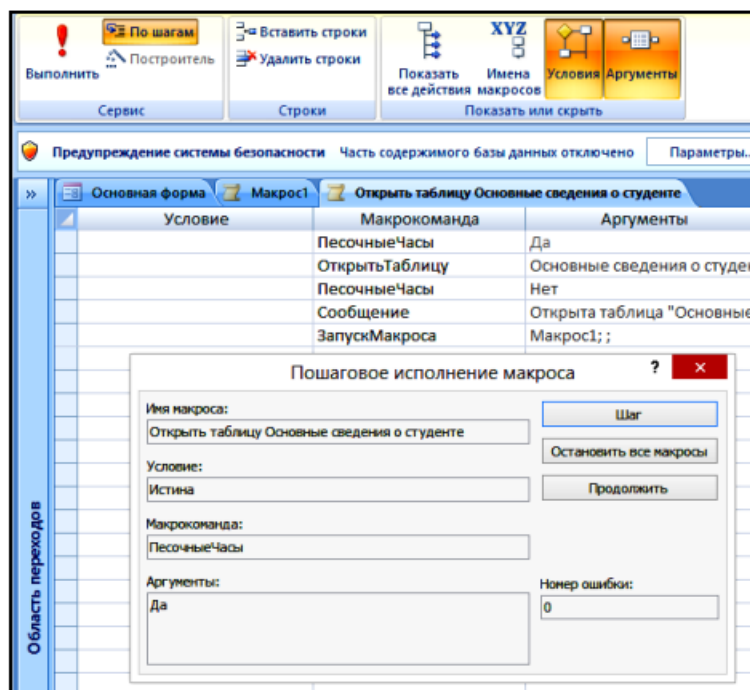


Рисунок 7.11 – Пошаговое выполнение макроса

В этом окне отображаются имя макроса, имя исполняемой макрокоманды, условие ее исполнения и аргументы. При этом в полях Условие (Condition) и Аргументы (Arguments) видны значения условного выражения и аргументов макрокоманды. Если в строке макрокоманды условие не указано, в поле Условие (Condition) всегда будет показываться значение Истина (True). Кнопки справа от полей в этом окне позволяют выполнить очередную макрокоманду, прервать выполнение макроса (например, если видно, что аргументы вычислены неправильно) и продолжить выполнение в обычном режиме, т. е. не по шагам.

Если в процессе выполнения макроса возникла ошибка, появляется диалоговое окно Ошибка выполнения макроса (Action Failed), аналогичное окну Пошаговое исполнение макроса (Macro Single Step), но без возможности продолжить процесс (кнопки Шаг (Step) и Продолжить (Continue) недоступны). Можно только прервать выполнение макроса, нажав кнопку Прервать (Halt). В окне отображается макрокоманда, содержащая ошибку.

Поиск ошибок в макросах, как правило, не вызывает больших затруднений. Это связано с тем, что:

- большинство синтаксических ошибок проверяется при вводе и система просто не даст ввести неправильное название макрокоманды или условное выражение (правда, она не защитит от ввода неправильного имени объекта, например несуществующей формы или элемента управления в форме);



- макросы, как правило, достаточно просты и не содержат длинных последовательностей макрокоманд, в противном случае лучше написать процедуру на VBA, т. к. язык VBA предоставляет больше возможностей и имеет существенно лучшие средства отладки.

Основные ошибки, которые возникают в макросах, связаны либо с неправильным вводом имен объектов, и тогда Access не может найти соответствующий объект, либо с тем, что объекты, на которые ссылается макрокоманда, недоступны, например форма не открыта. Поэтому при возникновении ошибки в первую очередь нужно проверить, правильно ли введены имена объектов и открыты ли формы, использованные в условиях или аргументах макрокоманд.

Стандартным приемом отладки как процедур, так и макросов, является вывод на экран специальных сообщений, говорящих о прохождении определенных этапов процесса. В макросах это осуществляется путем вставки дополнительных макрокоманд Вывод Сообщения (MsgBox) для отображения таких сообщений.

Можно легко отключить вывод на экран отладочных сообщений, для этого достаточно ввести в столбец Условия (Condition) напротив соответствующих макрокоманд значение Ложь.

Еще один стандартный прием – установка точек останова. Его применяют для того, чтобы остановить выполнение макроса в определенной точке до завершения процесса. Для этого используется макрокоманда ОстановитьМакрос (StopMacro). Макрос будет выполняться до макрокоманды ОстановитьМакрос (StopMacro), после этого можно проверить результаты. Таким образом, можно локализовать ошибку с точностью до макрокоманды.

### **Задание на проведение лабораторной работы**

#### **Задание на 3 балла:**

1. Выполните все задания, рассмотренные в теоретическом материале.
2. Сделать так, чтобы при открытии базы данных автоматически отображалась «Основная форма».

#### **Задание на 4-5 баллов:**

1. Выполните все задания, рассмотренные в теоретическом материале.
2. Сделать так, чтобы при открытии базы данных автоматически отображалась «Основная форма».
3. На «Основная форма» создайте элементы управления,

обеспечивающие вызов макросов, отвечающих за открытие форм для таблицы «Группы» и «Преподаватели». Учтеть, что при переходе от одной формы к другой, «старая» должна закрываться и открываться заново, когда текущая закрывается.

4. Создать форму, на которой выбираются:

- отчеты по заданию из лабораторной работы № 6;
- отображаются результаты запросов из лабораторной работы № 5.

### **Список использованной и рекомендуемой литературы**

1. Сеннов, А. С. Access 2007. Учебный курс (+CD) / А. С. Сеннов. СПб.: Питер, 2007. – 267 с.
2. taurion.ru – Иллюстрированный самоучитель по Microsoft Access. [Электронный ресурс]. – Режим доступа: [http://www.taurion.ru /access](http://www.taurion.ru/access), свободный – Загл. с экрана.
3. support.office.microsoft.com – Руководство по интерфейсу пользователя Access 2007. [Электронный ресурс]. – Режим доступа: <http://support.office.microsoft.com/client/HA010039892?lcid=1049&query=&NS=&Version=12&ShowNav=false>, свободный – Загл. с экрана.

Учебное издание

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ЛАБОРАТОРНЫМ РАБОТАМ**  
по дисциплине  
**«УПРАВЛЕНИЕ ИНФОРМАЦИЕЙ И ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ»**

для студентов направления подготовки  
Профессиональное обучение (по отраслям),  
профиль «Информационные технологии и системы».  
(в 2-х частях). Часть 2

**С о с т а в и т е л ь:**  
Елена Сергеевна Чёрная

Печатается в авторской редакции.  
Компьютерная верстка и оригинал-макет автора.

Подписано в печать \_\_\_\_\_  
Формат 60x84<sup>1</sup>/<sub>16</sub>. Бумага типограф. Гарнитура Times  
Печать офсетная. Усл. печ. л. \_\_\_\_\_. Уч.-изд. л. \_\_\_\_\_  
Тираж 100 экз. Изд. № \_\_\_\_\_. Заказ № \_\_\_\_\_. Цена договорная.

Издательство Луганского государственного  
университета имени Владимира Даля

*Свидетельство о государственной регистрации издательства  
МИ-СРГ ИД 000003 от 20 ноября 2015г.*

**Адрес издательства:** 91034, г. Луганск, кв. Молодежный, 20а  
**Телефон:** 8 (0642) 41-34-12, **факс:** 8 (0642) 41-31-60  
**E-mail:** izdat.lguv.dal@gmail.com **http:** //izdat.dahluniver.ru/







