



Некоммерческое  
акционерное  
общество

**АЛМАТИНСКИЙ  
УНИВЕРСИТЕТ  
ЭНЕРГЕТИКИ И  
СВЯЗИ**

Кафедра компьютерных  
технологий

## **УЧЕБНАЯ ПРАКТИКА**

Методические указания для студентов специальности  
5В070400 –Вычислительная техника и программное обеспечение

Алматы 2016

Составители: З.А.Акижанова. Учебная практика. Методические указания для студентов специальности 5В070400 – Вычислительная техника и программное обеспечение. – Алматы: АУЭС, 2016. – 20 с.

Методическое указание содержит рекомендации по подготовке, проведению учебной практики, а также рекомендации к темам индивидуальных заданий и требования к оформлению отчета.

Программа предназначена для студентов специальности 5В070400 – Вычислительная техника программное обеспечение, проходящих практику в кафедральных компьютерных классах кафедры «Компьютерные технологии».

Рецензент: доц. Г.Д. Мусапирова.

Печатается по плану издания некоммерческого акционерного общества «Алматинский университет энергетики и связи» на 2015 г.

© НАО «Алматинский университет энергетики и связи», 2016 г.

Сводный план издания 2015 г., поз. 200

Акижанова Зауре Абеевна

## УЧЕБНАЯ ПРАКТИКА

Методические указания для студентов специальности  
5В070400 – «Вычислительная техника и программное обеспечение»

Редактор Л.Т. Сластихина  
Специалист по стандартизации Н.К. Молдабекова

Подписано в печать «\_\_» \_\_\_\_ 2016г.  
Тираж 30 экз  
Объем 1,25уч.-изд.л.

Формат 60X84 1/16  
Бумага типографская № 1  
Заказ \_\_\_\_ Цена 625т.

Копировально-множительное бюро  
некоммерческого акционерного общества  
«Алматинский университет энергетики и связи»  
050013 Алматы, Байтурсынова, 126

Некоммерческое акционерное общество  
АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ  
Кафедра компьютерных технологий

УТВЕРЖДАЮ  
Проректор по УМР

С.В. Коньшин

« \_\_\_\_\_ » \_\_\_\_\_ 2016 г.

### УЧЕБНАЯ ПРАКТИКА

Методические указания для студентов специальности  
5В070400 – «Вычислительная техника и программное обеспечение»

СОГЛАСОВАНО

Начальник УМО

\_\_\_\_\_ М.А. Мустафин

“ \_\_\_\_\_ ” \_\_\_\_\_ 2016 г.

Рассмотрено и одобрено на  
заседании кафедры КТ,  
протокол №3 от 27.10.2016 г.

Согласовано:

Зав. кафедрой КТ

\_\_\_\_\_ З.К.Куралбаев

Председатель УМС

\_\_\_\_\_ Б.К.Курпенов

“ \_\_\_\_\_ ” \_\_\_\_\_ 2016 г.

Специалист по стандартизации

\_\_\_\_\_

“ \_\_\_\_\_ ” \_\_\_\_\_ 2016 г.

Редактор

\_\_\_\_\_

“ \_\_\_\_\_ ” \_\_\_\_\_ 2016 г.

Составители:

\_\_\_\_\_ Г.Т.Мусатаева

## **Введение**

Учебная практика является обязательным компонентом образовательной программы и проводится на первом курсе.

Учебная практика студентов, обучающихся по специальности 5В070400 –Вычислительная техника и программное обеспечение, является важным элементом учебного процесса по подготовке высококвалифицированных специалистов.

Методические указания по учебной практике составлены на основании Типового плана специальности 5В070400–Вычислительная техника и программное обеспечение.

Методические указания по учебной практике используют ссылки на документы:

1) Правила организации и проведения профессиональной практики и правила определения организаций в качестве баз практик. Утверждены приказом Министра образования и науки Республики Казахстан от 29 января 2016 года №107.

2) Положение об организации и проведения профессиональной практики в некоммерческом акционерном обществе «Алматинский университет энергетики и связи», 2016 г.

В соответствии с рабочим учебным планом устанавливается срок прохождения учебной практики на кафедре «Компьютерные технологии».

### **1 Цели и задачи практики**

1.1 Целью учебной (ознакомительной) практики обучающихся АУЭС является приобретение первичных профессиональных компетенций, т.е. закрепление теоретических знаний, полученных в процессе обучения, получение первых навыков исследовательской деятельности, приобретение практических умений и навыков работы по специальности.

1.2 Задачами учебной (ознакомительной) практики являются:

- ознакомление с основными направлениями деятельности высшего учебного заведения;

- общее знакомство с организационно- правовой формой, структурой, системой управления организаций, являющихся объектами будущей профессии;

-изучение видов, функций и задач будущей профессиональной деятельности;

- приобретение навыков работы в трудовом коллективе;

- знакомство с процессом производства с помощью экскурсий на профильные предприятия по специальности.

## **2 Организация проведения практики**

2.1 График проведения практики соответствует графику учебного процесса. Практика осуществляется преподавателями-руководителями практики, закрепленными за студентами кафедрой “Компьютерные технологии”.

2.2 Направление на практику оформляется приказом ректора об учебной практике. В приказе указываются график практики (сроки, база, аудитория, дата и время защиты отчета), список студентов, преподаватели– руководители практики.

2.3 При направлении на прохождение профессиональной практики студентам выдаются рабочий план-график профессиональной практики, направление, дневник-отчет о прохождении практики.

2.4 По итогам практики студенты представляют отчеты и дневники на кафедру «Компьютерные технологии». Отчет проверяется руководителем практики. Для защиты отчета распоряжением заведующего кафедрой создается комиссия.

## **3 Содержание учебной практики**

3.1 Учебная практика проводится в университете на кафедре «Компьютерные технологии» с ознакомительными экскурсиями в организации, являющиеся объектами будущей профессиональной деятельности: АО «Казактелеком», ТОО «MS-ProfessionalSystems», ТОО «ЭкоЭнергоГаз», филиал АО «Казактелеком» Дирекция «Академия инфокоммуникационных технологии», АО «Национальный центр информатизации», ТОО «Институт Казсельэнергопроект».

3.2 Практика проводится в соответствии с графиком практики согласно приказу ректора. Во время практики студенты должны посетить предприятие для знакомства с производством, выполнить учебные программы, создать базу данных на СУБД MS Access.

3.3 В дневник практики заносятся индивидуальные задания. Дневник практики заполняется ежедневно, указываются содержание и характер выполненной работы, руководитель ставит подпись. По итогам практики по индивидуальным заданиям составляется отчет.

3.4 Выполнение индивидуальных практических работ проходит под руководством преподавателя. Руководитель регулярно проверяет выполнение заданий и отчета по ним.

Защита отчета по практике производится в соответствии с графиком практики.

Студенты должны пройти инструктаж по технике безопасности.

3.5 По окончании работы над индивидуальным заданием по практике студент должен защитить свой отчет перед комиссией, сдать дневник, отчет и диск с программами.

Отчет по учебной практике должен содержать следующие разделы:

1) Титульный лист.

2) Введение:

- цель и задачи практики, место проведения, продолжительность практики;

- перечень работ, выполненных в процессе практики;

- посещение предприятия для знакомства с производством, описание общей структуры предприятия и функции подразделений, связанных с будущей профессиональной деятельностью.

3) Основная часть:

- описание задач, решаемых студентом во время практики, проиллюстрированное таблицами, рисунками, графиками и др.;

- для программ: задание, постановка задачи, текст программы, результат выполнения программы.

4) Заключение:

- знания и практические навыки, приобретенные в процессе практики;

- предложения по совершенствованию решений практических вопросов и задач.

5) Список литературы.

#### **4 Задание и этапы выполнения**

В ходе учебной практики на кафедре студенты могут углубить свои знания, полученные ими при преподавании дисциплин «Информатика» и «Алгоритмизация и программирование», получить новые практические навыки.

Задания были выбраны по следующим причинам: студенты по дисциплине информатика научились работать в СУБД MS Access; задание проектирования реляционной базы данных подготавливает студентов к новому уровню по работе с базами, к созданию модели базы данных. Для определения правильности модели базы данных создается база данных на СУБД MS Access. Для использования данных из других баз данных выполняется задание 3. Задания по программированию закрепляют знания по программированию.

Предусмотрено выполнение заданий:

- Задание 1. Проектирование реляционной базы данных.

- Задание 2. Создание базы данных.

- Задание 3. Работа в СУБД MS Access с внешними данными.

- Задание 4. Составление программ на языке программирования C++ с использованием функций.

## 4.1 Задание 1. Проектирование базы данных

Цель работы: научиться проектировать реляционную базу данных информационной системы с использованием ER-модели.

Задание:

- 1) Определить цель создания базы данных.
- 2) Описать предметную область, указать границы предметной области (ограничения). Источниками данных являются документы, функционирующие в организации, или словесное описание функционирования базы данных. Составить перечень функций, которые должна выполнять база данных.
- 3) Сделать анализ предметной области. Это дает возможность определить набор информационных объектов базы данных и их поля (реквизиты), определить первичные ключи объектов.
- 4) Определить связи между объектами. Изменить структуру объектов (включить внешние ключи), включить при необходимости новые объекты. Информационно - логическая модель отображает данные предметной области в виде информационных объектов и связей между ними. Составить информационно-логическую модель в MS Visio или др.
- 5) Сформулировать запросы к базе данных и определить информационные объекты, из которых можно получить искомую информацию, определить условия получения запроса (ограничения, например, «только те товары, которые поставлены в 1 квартале текущего года»).

Индивидуальное задание представляется преподавателем или выбирается самостоятельно. В ходе проектирования производится обсуждение темы с преподавателем, корректировка структуры базы. В базе должно быть достаточное количество таблиц, позволяющих полно выполнить задание.

Отчет по заданию 1 включает:

- цель создания базы данных;
- индивидуальное задание;
- описание предметной области;
- документы, имеющиеся в предметной области или словесное описание функционирования базы данных;
- информационные объекты, первичные ключи объектов;
- связь между двумя информационными объектами, описание связи (тип связи и класс принадлежности), изменение структуры информационных объектов (за счет ввода внешних ключей), включение новых информационных объектов;
- окончательный вариант информационно – логической модели;
- вид модели в MS Visio;
- запросы (тексты запросов), которые можно предъявить к базе данных, указание информационных объектов, из которых можно получить запрос, условия получения запроса.



Пример 4.1.1. Построить базу данных «Доставка товара клиентам в магазине бытовой техники». База данных содержит сведения о доставке купленных товаров клиенту. Заявление дает клиент с предъявлением чека, в котором указаны дата, данные о товарах, оплата. Доставка производится в течение нескольких дней. Данная база данных должна обеспечить подготовку, хранение, просмотр данных по заявлениям клиентов, по фактическим доставкам товаров, а также по анализу выполнения обязательств магазина на поставку по срокам и объемам, т.е. заказанных товаров.

Информационное обеспечение приложения включает:

- справочную информацию о доставляемых товарах;
- данные о планируемых доставках;
- данные о фактических доставках.

В магазине имеются:

1) информация о доставляемых товарах (это справочник товаров):

Код товара	Наименование	Ед. измерен.	Цена	Доп.сведения

2) данные о планируемых доставках: список заявлений. Заявление (договор) клиента на доставку товара содержит данные о клиенте, о товаре, количестве товара, номер чека на оплату товара:

Заявление № \_\_\_\_\_  
« \_\_\_\_ » \_\_\_\_\_ 201\_\_ г.

Клиент  
Ф.И.О. \_\_\_\_\_  
Адрес \_\_\_\_\_  
Телефон \_\_\_\_\_  
Чек № \_\_\_\_\_ от \_\_\_\_\_

Код товара	Наименование	Ед. измерения	Цена	Количество	Сумма	Срок поставки

Сумма всего \_\_\_\_\_

3) Данные о фактических доставках товара со склада: список накладных. Накладная на поставку содержит сведения о поставляемых товарах:

Накладная № \_\_\_\_\_  
« \_\_\_\_ » \_\_\_\_\_ 201\_\_ г.

Магазин \_\_\_\_\_ Ф.И.О. \_\_\_\_\_

Заявление №\_\_

Адрес \_\_\_\_\_

от «\_\_» \_\_\_\_\_ 201\_\_ г.

Телефон \_\_\_\_\_

Код товара	Наименование	Ед. измерения	Цена	Количество	Сумма

Сумма всего \_\_\_\_\_

Анализ документов, функционирующих в организации, позволяет выделить реквизиты системы, определяет информационные объекты, ключи этих объектов и связи между объектами.

При проектировании были выделены объекты Товар, Покупатель и их первичные ключи. Объекты имеют реквизиты: Товар (*код товара, наименование товара, ед. измерения товара, цена товара*), Покупатель (*код клиента, Ф.И.О., адрес клиента, телефон клиента*).

Каждый объект имеет первичный ключ, однозначно определяющий экземпляр объекта. Объекты обозначаются прямоугольниками, имена объектов – существительные, имена связей – глаголы. Первичные ключи выделены курсивом.

Связи двух объектов (бинарные) могут быть:

1) 1:1 («один к одному»), где одному экземпляру первого объекта соответствует один экземпляр второго объекта и, наоборот, одному экземпляру второго объекта соответствует один экземпляр первого объекта.

2) 1:M («один ко многим»), где одному экземпляру первого объекта соответствует 0,1,... экземпляров второго объекта, а одному экземпляру второго объекта соответствует один экземпляр первого объекта.

3) M:M («многие ко многим»), где одному экземпляру первого объекта соответствует 0,1,... экземпляров второго объекта, а одному экземпляру второго объекта соответствует несколько экземпляров первого объекта.

Объекты в связи могут иметь слабую связь (класс принадлежности этого объекта необязательный) - тогда не все экземпляры объекта участвуют в этой связи, а также иметь сильную связь (класс принадлежности этого объекта обязательный) - тогда все экземпляры объекта участвуют в этой связи.

В объектах связи «один к одному»: если оба объекта имеют сильную связь, то обычно объекты объединяются в один объект.

В объектах связи «один к одному»: если один объект имеет слабую связь, то сохраняется два объекта, а в объект с сильной связью добавляется дополнительное поле (внешний ключ), соответствующее первичному ключу объекта со слабой связью.

В объектах связи «один ко многим»: если оба объекта имеют сильную связь, то во второй объект (М) вводится для связи дополнительное поле для связи с первым объектом (внешний ключ), соответствующее первичному ключу в первом объекте.

В объектах связи «один ко многим»: если второй объект (М) имеет слабую связь, то вводится дополнительный связующий объект, содержащий внешние ключи для связи с первым (1) и со вторым (М) объектом.

В объектах связи «многие ко многим»: всегда вводится дополнительный связующий объект, содержащий внешние ключи для связи с первым и со вторым объектом.

Рассмотрим покупку товара. Покупателю выписывают Чек (*код чека, дата*). Связь «Выписывают» между объектами Покупатель и Чек имеет тип 1:1; связь сильная для обоих объектов, данные покупателя не сохраняются, а каждый чек принадлежит одному покупателю. Тогда можно заменить оба объекта на один объект Чек, включающий данные чека и данные покупателя (*код чека, дата чека, Ф.И.О., адрес клиента, телефон клиента*). Ключ объекта Покупатель – *код клиента* - заменяется ключом *код чека*.

Связь объектов Товар и Чек называется «Содержатся в Чеке» и имеет тип М:М, поэтому возникает связующий объект Содержимое\_Чека с составным ключом (*код чека, код товара*). Связь объектов Чек и Содержимое\_Чека 1:М, класс принадлежности экземпляров М-связного объекта обязательный (сильная связь), поэтому в объект Содержимое\_Чека включается поле для связи с Чеком. Связь объектов Товар и Содержимое\_Чека 1:М, класс принадлежности экземпляров М-связного объекта обязательный (сильная связь), поэтому в объект Содержимое\_Чека включается поле для связи с Товаром. Дополнительно в объект Содержимое\_Чека включаются данные о количестве товаров и стоимости каждого товара. Итоговая стоимость по чеку включаются в объект Чек.

Таким образом, имеем объекты Чек (*код чека, дата чека, Ф.И.О., адрес клиента, телефон клиента, итоговая стоимость*), Товар (*код товара, наименование товара, ед. измерения товара, цена товара*), Содержимое\_Чека (*код чека, код товара, количество товара*). Стоимость каждого товара является вычисляемым полем (*стоимость = количество товара \* цена товара*).

Рассмотрим оформление покупателем заявлений на доставку товара. Объект Заявление имеет реквизиты: номер заявления, дата заявления. Связь объектов Заявление и Чек называется «Заявление\_по\_Чеку», тип связи 1:1. Но связь со стороны Чека слабая, т.к. не все покупатели обращаются с заявлением. Поэтому в объект Заявление вводится код чека как внешний ключ (*код заявления, дата заявления, код чека*).

В заявлении указываются доставляемые товары из списка товаров, оплаченных по чеку, объекта Содержимое\_Чека. Связь объектов Заявление и Содержимое\_Чека называется «Содержатся\_в\_Заявлении», тип связи 1:М. Связь объекта Содержимое\_Чека слабая, т.к. не все товары вносятся в заявление, поэтому возникает новый объект Содержимое\_Заявления. В объект Содержимое\_Заявления вводятся внешние ключи для связи с объектом Заявление и с объектом Содержимое\_Чека. В объекте Содержимое\_Заявления количество товара отсутствует, т.к. значение берется из объекта Содержимое\_Чека.

Таким образом, имеем объекты Заявление (*код заявления, дата заявления, код чека, итоговая стоимость*), Содержимое\_Заявления (*код заявления, код чека, код товара*), стоимость каждого товара является вычисляемым полем.

Рассмотрим доставку товара. По одной накладной отпускается товар по одному заявлению. Связь объектов Накладная и Заявление называется «Выписывают по заявлению», тип связи 1:1. Следует учесть, что сначала формируется Заявление, и до формирования объекта Накладная объект Заявление не участвует в связи и связь со стороны объекта Заявление слабая. Поэтому объекты не объединяются и в Накладной указывается код Заявления.

В документе накладная записываются доставляемые товары, указанные в заявлении. Вводится объект Содержимое\_Накладной, в который входят все товары из Содержимое\_Заявления. Связь объектов Содержимое\_Накладной и Содержимое\_Заявления 1:1, но со слабой связью со стороны Содержимое\_Заявления, поэтому в объект Содержимое\_Накладной вводится внешний ключ для связи с объектом Содержимое\_Заявления.

Таким образом, имеем объекты Накладная (*код накладной, дата накладной, код заявления*), СодержимоеНакладной (*код накладной, код заявления, код чека, код товара*).

Логическая структура реляционной базы данных предметной области имеет вид (рисунок 4.1).

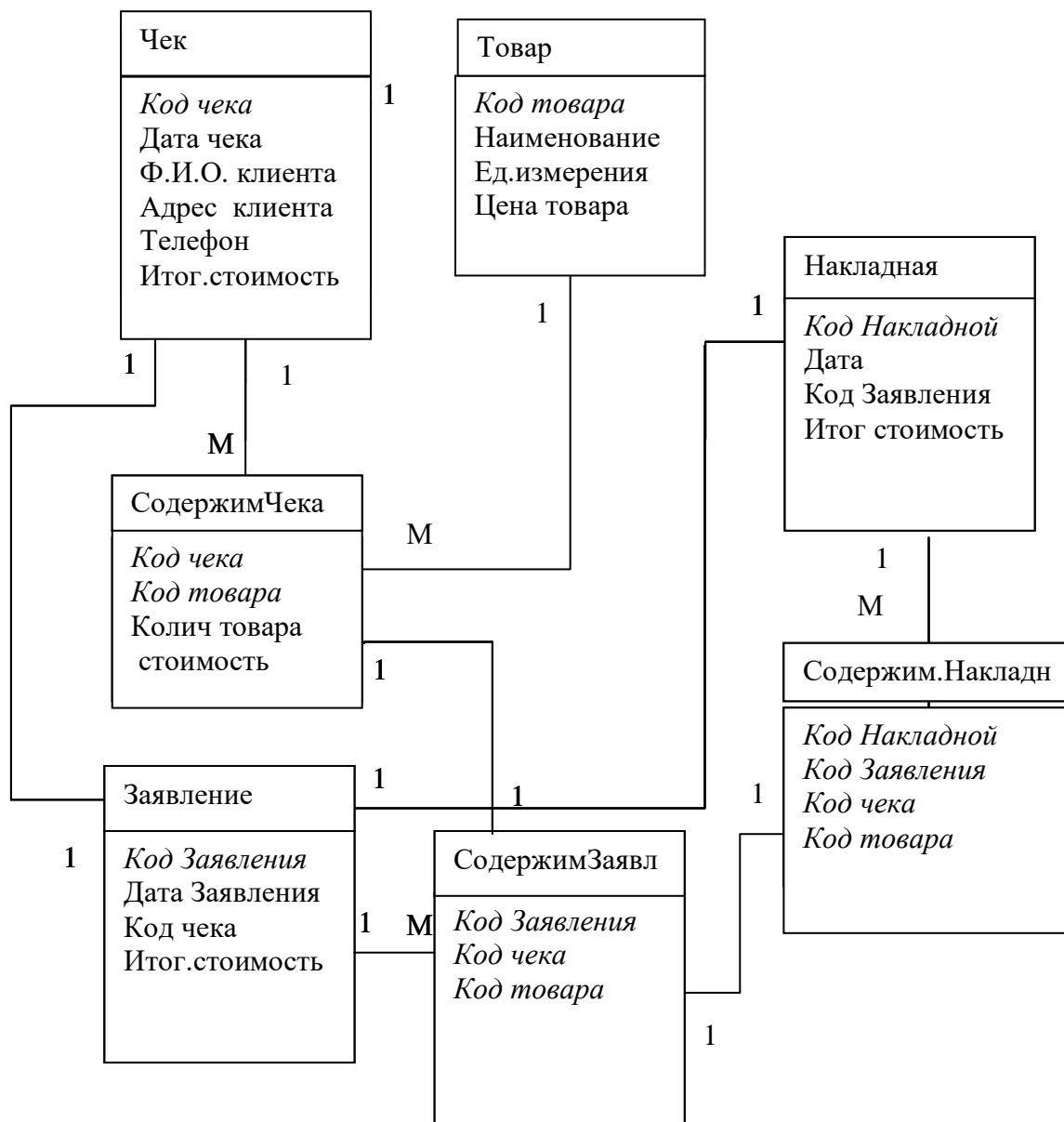


Рисунок 4.1 - Логическая структура реляционной базы данных предметной области

К базе данных можно создать запросы:

а) вычислить стоимость каждого товара в чеке. Для этого надо вычислить по формуле  $\text{стоимость} = \text{Колич товара} * \text{Цена товара}$ .

Данные берутся из информационных объектов Товар и Содержимое\_Чека;

б) вычислить итоговую стоимость по всем товарам в чеке. Данные берутся из информационных объектов Чек, Товар, Содержимое\_Чека, вычисляется стоимость и суммируется по всем записям, относящимся к данному чеку;

в) определить, выполнена ли заявка №\_\_. Данные берутся из таблиц Заявление, Содержимое\_Заявления, Накладная, Содержимое\_Накладной.

Для проектирования базы данных можно использовать MS Visio, ERWin или др. В MS Visio модель имеет вид (рисунок 4.2).

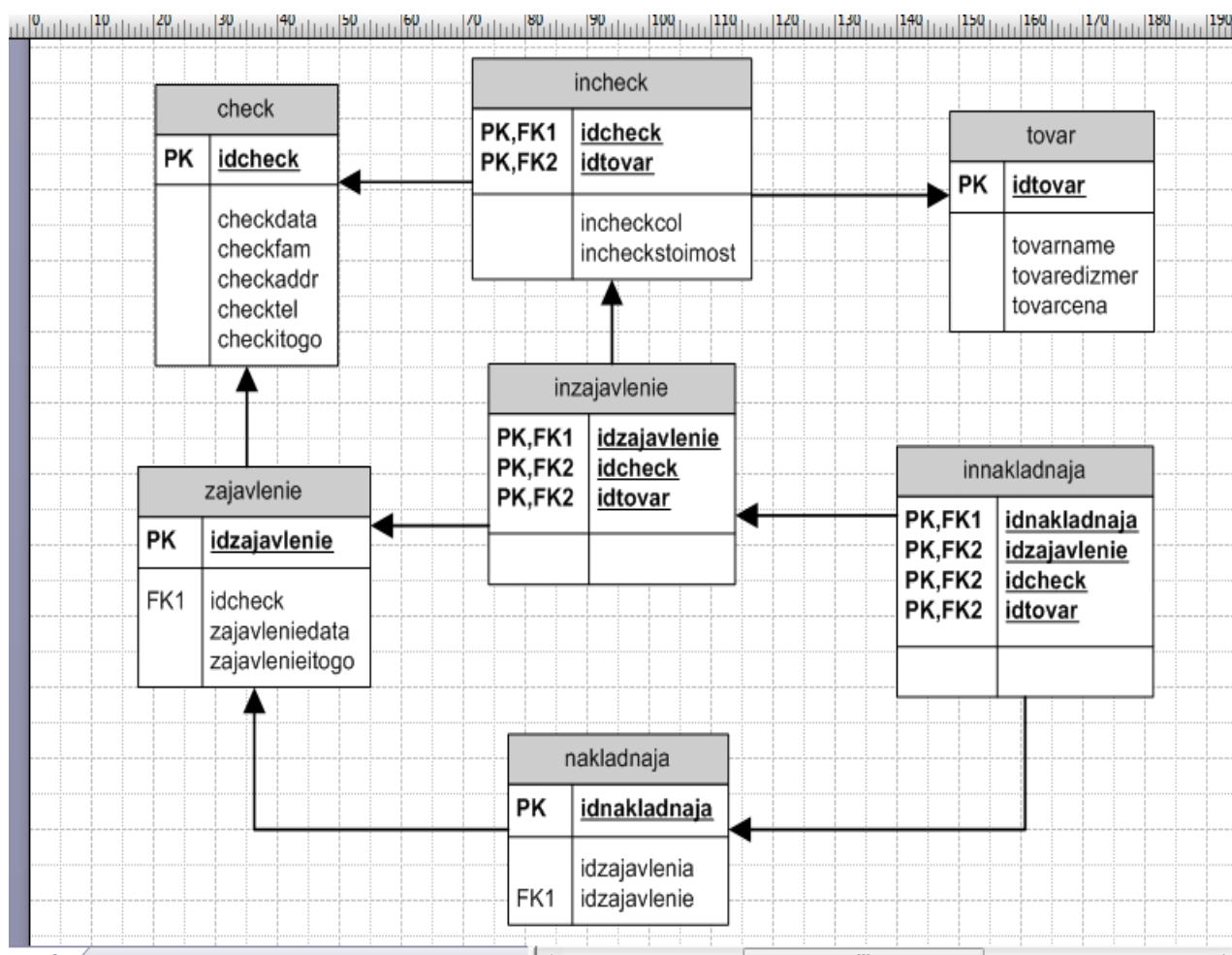


Рисунок 4.2 - Структура реляционной базы данных предметной области в MS Visio

Пример 4.1.2. Построить базу данных «Фитнес - центр. Запись клиентов на получение услуг». У нас не имеется никаких документов, по которым можно было бы получить сведения о данных и структуре данных в базе. Мы опираемся на описание работы, полученное от сотрудников фитнес-центра. Клиент покупает месячный абонемент, затем по прейскуранту услуг выбирает те услуги, которые являются дополнительными к базовым услугам. Цена абонемента изменяется и состоит из базовой цены абонемента и стоимости всех получаемых услуг. Эти услуги оказывают сотрудники центра. Клиент может выбрать любого сотрудника. При согласовании даты и времени сотрудник смотрит свою таблицу занятости и затем по согласованию с клиентом назначает дату и время. Таким образом, мы можем определить реквизиты базы данных, сгруппировать их по информационным объектам, затем связать эти объекты в информационно - логическую модель. База данных содержит стоимость абонемента, перечень услуг с указанием цен, список сотрудников, сведения о клиенте и его абонементе, сведения о

сотруднике, виде услуги, клиенте, которому оказывается услуга, дату и время оказания услуги. Исходя из этих сведений, можно получить запрос на занятость данного сотрудника в данном месяце.

## **4.2 Задание 2. Создание базы данных**

Цель работы: создать реляционную базу данных информационной системы, модель которой создана в задании 1.

Задание:

- 1) Создать базу данных на СУБД Access.
- 2) Создать схему данных.
- 3) Создать запросы.

Отчет по заданию 2 включает:

а) структуру таблиц с описанием: назначения таблицы, назначения полей таблицы, типа данных полей, указания первичного ключа, указания внешних ключей и таблиц, с которыми связывается исходная таблица с помощью этих ключей;

б) данные таблиц: можно показать только часть данных, которые затем будут участвовать в запросах;

в) структуру базы данных; объяснить связь между таблицами;

г) запросы: для каждого запроса представляется текст запроса, конструктор запроса, результат выполнения запроса.

Затем описывается функционирование базы данных, работа с базой данных, выполнение запросов.

В качестве программного обеспечения выбрана СУБД Microsoft Access. Создание базы данных производится в соответствии с информационно – логической моделью данных. Создание базы данных начинается с формирования структуры таблиц, затем создается схема данных для поддержания целостности связи данных. Заполнение таблиц можно производить непосредственно в таблицы или через формы.

Контрольные вопросы к заданиям 1 и 2:

- 1) Что собой представляет база данных?
- 2) Что такое «реляционная база данных»?
- 3) Назовите основные объекты базы данных Microsoft Access.
- 4) Первичный ключ, его назначение.
- 5) Вторичный ключ, его назначение.
- 6) Что такое схема данных?
- 7) Какую роль играют связи между таблицами?

## **4.3 Задание 3. Работа Access с внешними данными**

Цель работы: научиться работать из Access с внешними данными.

Задание:

- использовать данные из другой базы данных Access;
- экспорт данных в другую базу данных Access.

В задании студенты должны самостоятельно изучить способы импорта и экспорта данных из своей базы данных в другую базу данных Access:

- 1) Создать новую базу данных.
- 2) Импортировать часть данных из имеющейся базы данных.
- 3) Выполнить запросы с использованием импортированных данных.
- 4) Экспортировать измененные и дополненные данные в старую базу данных.
- 5) Сравнить результаты в новой и старой базе данных.

#### 4.4 Задание 4. Составление программ с использованием функций

Цель работы: закрепление умений по программированию с использованием функций.

Задание:

- 1) Создать функции, которые ничего не возвращают и которые возвращают одно значение.
- 2) При передаче параметров обязательно должны использоваться передача параметров по значению и по ссылке.
- 3) Использовать передачу массивов.
- 4) В функции main произвести вызовы пользовательских функций и проверить передаваемые значения до и после выполнения функции.

Примеры программ приведены ниже.

Пример 4.4.1.

Создать структуру struct, представляющую студента, с именем, номером зачетной книжки. Создать массив студентов группы. Ввести массив a из n элементов. Создать функции ввода, вывода элементов массива. Используя арифметику указателей, вывести элементы массива с индексами 2, 3 и 6.

```
#include "stdafx.h"  
#include <iostream>  
#include <conio.h>  
#include <stdlib.h>  
using namespace std;
```

```
structure struct{  
char name[20];  
long zach;  
}
```

```
void vvod(structure struct a[], unsigned int N);
```



```

void shygaru(structure struct a[], unsigned int N);
void esetue(structure struct a[], unsigned int N);
void read(structure struct s);
void show(structure struct s);

int main(){
    system("color 0f");
    setlocale (LC_ALL, "Russian");
    unsigned int N=0;
    int T;
    cout<<"\nВведите размер N = ";
    cin>>N;
    structure struct * ptr;
    structure struct * a=new structure struct[N];
    cout<<"\nВведите элементы массива : \n"<<endl;
    vvod(a,N);
    cout<<"\nИсходный массив : ";
    shygaru(a,N);
    esetue(a, N);

getch();
    return 0;
}

void vvod(structure struct a[], unsigned int N){
for (int T=0; T<N; T++){
    cout<<"a["<<T<<"]="";
    a[T].read();
    }
}

void shygaru(structure struct a[], unsigned int N){
for (int T=0; T<N; T++){
    a[T].show()<<" ";
    }
}

void esetue(structure struct a[], unsigned int N){
structure struct * ptr;
ptr=a;
cout<<"\n*a = "<<*a<<"\t *ptr = "<<*ptr<<endl;
ptr+=2; a+=2;
cout<<"\n*a = "<<*a<<"\t *ptr = "<<*ptr<<endl;
ptr++; a++;
}

```

```

cout<<"\n*a = "<<*a<<"\t *ptr = "<<*ptr<<endl;
if (N>6){
    cout<<"\n*a = "<<*(a+3)<<"\t *ptr =
"<<*(ptr+3)<<endl;
}
}
void read(struct s){
cout<<"введите имя студента";
gets(s.name);
cout<<"введите номер зачетной книжки студента";
cin>>s.zach;
}
void show(struct s){
cout<<"\n имя студента - "<<s.name<<endl;
cout<<"\номер зачетной книжки студента -
"<<s.zach<<endl;
}
}

```

Пример 4.4.2. Создать структуру - Температура в городе (месяц (три первых символа), день, температура) и массив Данные о Температуре в городе за сезон, отсортировать по температуре, вывести данные о самой низкой температуре за первый месяц сезона.

```

#include<iostream>

using namespace std;
struct Temp
{
    string mon;
    int day;
    float temp;
};
struct Temp_S
{
    Temp t[1000];
    int size;
};
void show(Temp &t)
{
    cout<<"Mes`yac \'"<<t.mon<<"\",
den`:"<<t.day<<"\tTemperatura: "<<t.temp<<"*C"<<endl;
}

```

```

void show(Temp_S T)
{
    for(int i = 0; i < T.size; i++)
    {
        show(T.t[i]);
    }
}
void set(string m, int d, float t, Temp &T)
{
    T.mon = m; T.day = d; T.temp = t;
}
void add(Temp_S &T, Temp t)
{
    T.t[T.size] = t;
    T.size++;
}
void zap(Temp_S &T, Temp &t)
{
    set("JUN", 15, 28, t); add(T, t);
    set("JUL", 23, 35, t); add(T, t);
    set("AUG", 15, 30, t); add(T, t);
    set("JUN", 12, 20, t); add(T, t);
    set("JUL", 24, 31, t); add(T, t);
    set("AUG", 6, 28, t); add(T, t);
    set("AUG", 7, 19, t); add(T, t);
    set("JUL", 19, 24, t); add(T, t);
    set("AUG", 23, 26, t); add(T, t);
    set("JUN", 30, 28, t); add(T, t);
    set("JUL", 12, 30, t); add(T, t);
    set("AUG", 20, 31, t); add(T, t);
}
void swap(Temp &t1, Temp &t2)
{
    Temp temp;
    temp = t1;
    t1 = t2;
    t2 = temp;
}
void sort(Temp_S &T)
{
    for( int i = T.size-1; i >= 0; i--)
        for(int j = 0; j < i; j++)
            {

```

```

        if(T.t[j].temp < T.t[j+1].temp)
            swap(T.t[j].temp, T.t[j+1].temp);
    }
}
main()
{
    Temp_S x;
    Temp y;
    x.size = 0;
    zap(x, y);
    show(x);
    sort(x);
    cout<<"\n"<<endl;
    show(x);
    float flag1 = 49;
    int flag2 = 0;
    for(int i = 0; i < x.size; i++)
    {
        if(x.t[i].temp < flag1 && x.t[i].mon == "JUN")
        {
            flag1 = x.t[i].temp;
            flag2 = i;
        }
    }
    cout<<"\n"<<endl;
    show(x.t[flag2]);
}

```

Примеры индивидуальных заданий.

1. Дана последовательность символов  $a_1a_2a_3\dots a_n$ , и  $n$ . Группы символов, разделенные одним пробелом, будем называть словами. Пусть задано предложение из нескольких слов. Оно начинается с символа и заканчивается точкой. Предложение является записью арифметических действий (+ и -) над целыми числами (например, «25 + 4230 =. »). Вычислить результат, добавить результат в предложение и вывести.

2. Ввести натуральное число  $n$  (где  $n \leq 10$ ), значения координат центра  $x_i, y_i$  и радиуса  $r_i$  для  $n$  кругов. Если пересекаются или соприкасаются три или более круга, то вывести номера этих кругов, их количество, координаты их центров и радиусы.

3. Создать структуру Блюдо в меню (название, цена) и массив Меню кафе, отсортировать блюда по цене, вывести блюда, стоимость которых отличается от средней цены всех блюд, но не более, чем на 20 тенге.

## **5 Защита отчета**

Студент защищает отчет перед комиссией и преподавателем по практике. На защите студент должен хорошо ориентироваться в содержании представленного отчета, отвечать на вопросы теоретического и практического характера.

Защита отчета оценивается по установленной балльно-рейтинговой буквенной системе оценок, а результат заносится в зачетную книжку. Отчеты, получившие неудовлетворительную оценку, подлежат доработке и представляются на повторное рецензирование после прохождения дополнительной сессии.

## Список литературы

- 1 Информатика. Базовый курс. Учебник для вузов. Под.ред. С.В.Симоновича. – СПб.: Питер, 1999. – 640 с.
- 2 Бекаревич Ю.Б., Пушкина Н.В. Самоучитель Access 2007.— СПб.: БХВ-Петербург, 2007.
- 3 Шилдт Г. С++: базовый курс, 3-е издание.: Пер. с англ.-М.: Издательский дом «Вильямс», 2010.- 624 с.:ил.-Парал.тит. англ.
- 4 Лафоре Р. Объектно-ориентированное программирование в С++. Классика Computer Science. 4-е изд. - СПб.: Питер, 2011.- 928 с.: ил.
- 5 Роб П., Коронел К. Системы баз данных: проектирование, реализация и управление.- 5-е изд., перераб. и доп.; Пер. с англ.- СПб.: БХВ- Петербург, 2004. – 1040 с.:ил.

## Содержание

Введение .....	3
1 Цели и задачи практики .....	3
2 Организация проведения практики .....	4
3 Содержание учебной практики .....	4
4 Задание и этапы выполнения .....	5
5 Защита отчета .....	19
Список литературы .....	20

Акижанова Зауре Абеевна

## УЧЕБНАЯ ПРАКТИКА

Методические указания для студентов специальности  
5В070400 –Вычислительная техника и программное обеспечение

Редактор Л.Т. Сластихина  
Специалист по стандартизации Н.К.Молдабекова

Подписано в печать «\_\_» \_\_\_\_ 2016г.  
Тираж 30 экз  
Объем 1,25уч.-изд.л.

Формат 60X84 1/16  
Бумага типографская № 1  
Заказ \_\_\_\_ Цена 625т.

Копировально-множительное бюро  
Некоммерческого акционерного общества  
«Алматинский университет энергетики и связи»  
050013, Алматы, Байтурсынова, 126

Некоммерческое акционерное общество

АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ

Кафедра компьютерных технологий

УТВЕРЖДАЮ  
Проректор по УМР

С.В. Коньшин

« \_\_\_\_\_ » \_\_\_\_\_ 201\_ г.

### УЧЕБНАЯ ПРАКТИКА

Методические указания для студентов специальности  
5В070400 – «Вычислительная техника и программное обеспечение»

СОГЛАСОВАНО

Начальник УМО

\_\_\_\_\_ М.А. Мустафин  
« \_\_\_\_\_ » \_\_\_\_\_ 201\_ г.

Рассмотрено и одобрено на  
заседании кафедры КТ,  
протокол №3 от \_\_.\_\_.20\_\_ г.

Согласовано:

Зав. кафедрой КТ

\_\_\_\_\_ З.К.Куралбаев

Председатель УМС

\_\_\_\_\_ Б.К.Курпенов  
« \_\_\_\_\_ » \_\_\_\_\_ 201\_ г.

Специалист по стандартизации

\_\_\_\_\_ 201\_ г.

Редактор

\_\_\_\_\_ 201\_ г.

Составители:

\_\_\_\_\_ З.А.Акижанова





**Некоммерческое  
акционерное  
общество**

**АЛМАТИНСКИЙ  
УНИВЕРСИТЕТ  
ЭНЕРГЕТИКИ И  
СВЯЗИ**

Кафедра компьютерных  
технологий

## **УЧЕБНАЯ ПРАКТИКА**

Методические указания для студентов специальности  
5В070400 –Вычислительная техника и программное обеспечение

Алматы 2016