Министерство науки и высшего образования Российской Федерации
Рубцовский индустриальный институт (филиал)
ФГБОУ ВО «Алтайский государственный технический
университет им. И.И. Ползунова»

Кафедра прикладной математики

Л.А. ПОПОВА

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Методические указания для студентов направления
«Информатика и вычислительная техника»
очной и заочной форм обучения

Рубцовск 2021

ББК 32.973.2

Попова Л.А. Информационные технологии: Методические указания для студентов направления «Информатика и вычислительная техника» очной и заочной форм обучения / Л.А. Попова. – Рубцовск: РИИ, 2021. – 45 с. [ЭР].

Методические указания предназначены для проведения лабораторных работ по курсу «Информационные технологии» у студентов направления 09.03.01 «Информатика и вычислительная техника» очной и заочной форм обучения.

Приведены задания к лабораторным работам, а также вопросы для самостоятельной подготовки к текущему и промежуточному тестированию.

Рассмотрены и одобрены на заседании кафедры прикладной математики Рубцовского индустриального института.
Протокол № 9 от 18.03.2021 г.

Содержание

Введение

Методические указания написаны в соответствии с программой дисциплины «Информационные технологии» для студентов направления «Информатика и вычислительная техника» очной и заочной форм обучения, предназначены для аудиторной и самостоятельной работы по данному курсу.

Указания содержат комплект заданий для лабораторных работ, а также вопросы для самостоятельной работы и подготовки к текущему контролю успеваемости и промежуточной аттестации.

Перечень планируемых результатов обучения по дисциплине, соотнесенных с индикаторами достижения компетенций

Компе-тенция	Содержание компетенции	Индика-тор	Содержание индикатора
ПК-14	Способен осуществлять администрирование программного обеспечения инфокоммуникационной системы организации	ПК-14.1	Инсталлирует ПО для автоматизированных и информационных систем
		ПК-14.2	Анализирует функционирование прикладного программного обеспечения по заданным параметрам

Общий объем дисциплины во 2 семестре: 4 з.е. / 144 час

Форма промежуточной аттестации: Экзамен

Комплект заданий для лабораторных работ

Лабораторная работа № 1

Создание пакета документов

Задание: для организации, занимающейся оптовой продажей продуктов со склада создать пакет документов (в последующих работах предусмотрено их программное заполнение):

1) штатное расписание, форма Т-3;
2) приказ о приеме на работу, форма Т-1;
3) личная карточка работника, форма Т-2;
4) накладная на отпуск товара;
5) карточка учета материалов, форма М-17;
6) акт приема передачи;
7) товарный чек;
8) отчет о наличии товара;
9) отчет об имеющихся товарах.

Порядок выполнения заданий

I. Скачайте с сайта https://lugasoft.ru/blank/t-3-shtatnoe-raspisanie документы 1-3 (формы Т-1, Т-2, Т-3).
II. Найдите в интернете самостоятельно и скачайте «Карточку учета материалов» (форма М-17).
III. Ознакомьтесь с содержанием всех скачанных документов (лучше скачивать не только пустые бланки, но и образцы заполненных документов, чтобы понять, какие именно реквизиты являются постоянными, а какие подлежат заполнению).
IV. Создайте в текстовом процессоре Word (или Writer) документы 6-9 (товарный чек, акт приема передачи и отчеты).
V. Накладную создайте в табличном редакторе Excel (или Calc). ***Примечание***: на рисунке представлен образец заполненной накладной, а нужно оформить пустой бланк, для последующего заполнения.

Наименование организации (ФИО индивидуального предпринимателя):
ИНН:_____
Адрес:_____

Товарный чек №_____ от _____20___ года

№	Артикул	Наименование товара	Ед. изм.	Кол-во	Цена за 1 ед. (руб.)	Сумма (руб.)

Всего наименований_____ на сумму_____руб._____ коп.

Продавец_____ _____
подпись: (инициалы, фамилия)

Приложение №__
к договору
от «__» _____ 20__ г.
№ _____

АКТ ПРИЕМА ПЕРЕДАЧИ ТОВАРА

ООО «_____», в лице _____, действующего на основании _____, именуемое в дальнейшем Продавец, с одной стороны и ИП _____, в лице _____, действующего на основании _____, именуемый в дальнейшем Покупатель, с другой стороны (в дальнейшем вместе именуемые «Стороны» и по отдельности «Сторона»), составили настоящий Акт о нижеследующем:

1. В соответствии с п. ____ Договора между Сторонами № _____ от «__» _____ 20__ года Продавец передает, а Покупатель принимает Товар следующего ассортимента и количества:

№	Наименование	Кол-во	Цена с НДС (руб.)	Сумма с НДС (руб.)
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				
			Итого на сумму	

Стоимость Товара поставленного в соответствии с условиями Договора составляет _____ (_____), с учетом НДС.

2. Принятый Покупателем товар обладает качеством и ассортиментом, соответствующим требованиям Договора. Товар поставлен в установленные в Договоре сроки. Покупатель не имеет никаких претензий к принятому товару.

3. Настоящий Акт составлен в двух экземплярах, имеющих равную юридическую силу, по одному экземпляру для каждой из Сторон и является неотъемлемой частью Договора между Сторонами.

Покупатель:	**Продавец:**
Должность	**Должность**
_____ **И.О.Ф.**	_____ **И.О.Ф.**
М.П.	М.П.

ОТЧЕТ
о наличии товара
на _____ 20___года

Артикул Наименование товара Единица измерения					X(10) A(30) A(10)	
Количество	Цена, руб.	Дата прибытия (отправки)	Номер счета-фактуры	Название организации	ФИО приемщика (отправителя)	Примечания
9(5)	9(5), 9(2)	X(8)	9(6)	X(20)	A(16)	A(1)
Остаток на __9(2).9(2).9(4)_____ Средняя цена за единицу товара (руб.)					9(5) 9(5), 9(2)	

Кладовщик _____ И.О.Ф.
(подпись)

ОТЧЕТ
о товарах, имеющихся на складе
на _____ 20__года

№ п/п	Артикул	Наименование товара	Единица измерения	Количество	Средняя цена, руб.
9(2)	X(10)	A(30)	A(10)	9(5)	9(5), 9(2)

Кладовщик _____ И.О.Ф.
(подпись)

Унифицированная форма № ОП-4
Утверждена постановлением Госкомстата России от 25.12.98 № 132

	Код
Форма по ОКУД	0330504
по ОКПО	
Вид деятельности по ОКДП	
Вид операции	

ООО "Столовая №1"
(организация)
Цех
(структурное подразделение)
(структурное подразделение «получатель»)

Номер документа	Дата составления
12	21.03.2016

НАКЛАДНАЯ
НА ОТПУСК ТОВАРА

Отпущено на основании требования в кладовую №17 Время отпуска 14 ч. 30 мин.
(наименование, номер, дата документа)

Через Полякову Инну Сергеевну
(фамилия, имя, отчество материально ответственного лица)

Номер по порядку	Продукты и товары		Единица измерения	Количество (масса)		По учетным ценам, руб. коп.		Примечание
	наименование, сорт		наименование	затребовано	отпущено всего	цена	сумма	
1	2		4	6	9	10	11	14
	Картофель		кг	12	12	10-00	120-00	
	Морковь		кг	3	3	12-00	36-00	
	Свекла		кг	5	5	7-00	35-00	
		Итого		**20**	**20**	**X**	**191-00**	

Оборотная сторона формы № ОП-4

Номер по порядку	Продукты и товары		Единица измерения	Количество (масса)		По учетным ценам, руб. коп.		Примечание
	наименование, сорт		наименование	затребовано	отпущено всего	цена	сумма	
1	2		4	6	9	10	11	14
		Итого				**X**		**X**
		Всего по накладной		**20**	**20**	**X**	**191-00**	**X**

Всего на сумму Сто девяносто один (прописью) руб. коп.

Отпустил Кладовщик (должность) Сидоркин (подпись) П.М. Сидоркин (расшифровка подписи)

Принял Повар (должность) Уткина (подпись) А.М. Уткина (расшифровка подписи)

Отпуск разрешил:
Руководитель директор (должность) Шутов (подпись) А.Г. Шутов (расшифровка подписи)
Заведующий производством Ившина (подпись) Д.Т. Ившина (расшифровка подписи)

Лабораторная работа № 2

Визуальное и графическое представление данных

Порядок выполнения заданий

I. Запустите среду программирования PyCharm.
II. Создайте новый проект.
III. Добавьте к интерпретатору проекта модуль для построения графиков Matplotlib.
IV. Создайте в проекте три файла с расширением .py для решения задач (код программы для каждой задачи должен быть в отдельном файле).

Задания:

1 Построить график кусочно-заданной функции:

$$y = \begin{cases} (1 + cx^2)^4, & x \le -1, \\ \sin(2kx), & -1 < x \le 3, \\ \sqrt[3]{2x - 1}, & x > 3, \end{cases}$$

где c = –0.02; k=1.5; на отрезке [-5; 5].

a) Все части графика должны быть разным цветом.
b) Расчетные формулы вывести в легенде.
c) Задать другие элементы оформления графика.

2 Построить график изменения температуры воздуха в течение выбранного периода.
a) Добавить в проект тестовый файл **temperature.txt**, содержащий значения температур за каждый час в течение трех месяцев.
b) Построить график изменения среднесуточной температуры в течение всего периода.
c) Вычислить среднемесячные значения температур.

3 Создать в проекте текстовый файл, в который записать динамику изменения курсов валют (доллара и евро) за прошлый месяц.
a) Простроить три графика, расположив их так, как показано на рисунке 1.
b) Задать элементы оформления графиков.

Динамика изменения курса доллара	Динамика изменения курса евро
Динамика изменения курсов валют	

Лабораторная работа № 3

Автоматизированная обработка текстовых документов

Задание: написать программу для заполнения документов: «Штатное расписание» (форма Т-3), «Приказ о приеме на работу» (форма Т-1) и «Личная карточка работника» (форма Т-2), организации, занимающейся оптовой продажей продуктов со склада.

Теоретический материал

Штатный сотрудник – это работник по бессрочному трудовому договору, должность которого зафиксирована в штатном расписании фирмы. Другими словами штат предприятия состоит из определенного числа сотрудников (вакансий), которые являются постоянными. Работники, принимаемые по срочным договорам, сезонные, временные в штат организации не входят.

Заполнение разделов документа «Штатное расписание»:

В 1 разделе («Наименование структурного подразделения») указывают подразделения, представительства, филиалы. Сначала прописывают руководящий состав, затем – по отделам всех остальных. Например, администрация, бухгалтерия, отдел кадров, данные о производстве (цехи, участки и т. д.).

Для заполнения раздела 2 («Код структурного подразделения») пользуются отраслевыми классификаторами. Каждому подразделению присваивают определенный код для облегчения ведения документооборота.

Раздел 3 («Должность (специальность, профессия), разряд, класс (категория) квалификации работника») заполняют на основе ***Классификатора профессий*** рабочих, должностей служащих и тарифных разрядов.

Раздел 4 («Количество штатных единиц») содержит сведения о числе сотрудников штата в соответствии с потребностями, целесообразностью и особенностями конкретного предприятия.

В 5 разделе («Тарифная ставка (оклад) и пр.») указывается установленный оклад или тариф подчиненного в рублях в соответствии с условиями ***трудового договора***.

В разделе «Надбавки и доплаты» содержатся данные о стимулирующих выплатах и компенсациях.

Порядок выполнения заданий

I. Создайте файл, содержащий данные об организации, занимающейся хранением и реализацией продуктов со склада. Для этого:
   a. разработайте логотип,
   b. придумайте название и реквизиты организации (адрес, ОКПО, ОГРН, ИНН, КПП).

II. Разработайте шаблон штатного расписания (на основе формы Т-3) для данной организации (не менее 20 штатных единиц, включая различные категории персонала, например, директор, главный бухгалтер, водитель, уборщица).
   a. Создайте два текстовых документа, в которые введите данные о структурных подразделениях организации и о штате сотрудников.
   b. Структура данных в документе «Структурные подразделения»: код (01 – 05), наименование.
   c. Структура данных в документе «Штат сотрудников»: код подразделения, должность, количество штатных единиц (может быть дробной величиной), тарифная ставка.

III. Откройте документ «Штатное расписание» в текстовом процессоре (LibreOffice Writer). Разделите таблицу на три части (отделите основную таблицу от остальных данных пустыми абзацами).
   a. Удалите пустые строки из основной таблицы (так как количество должностей может меняться и заранее может быть не известно).
   b. Введите наименование организации и ОКПО.
   c. Введите категории надбавок (премия и районный коэффициент).
   d. Введите переменные (например, {{Date}}, {{Period}}) в ячейки, подлежащие заполнению, первой таблицы. Образец шаблона документа представлен на рисунке 1.

Унифицированная форма № Т-3
Утверждена Постановлением Госкомстата России
от 05.01.2004 № 1

	Код
Форма по ОКУД	0301017
по ОКПО	???

??? _____
наименование организации

Номер документа	Дата составления
	{{Date}}

ШТАТНОЕ РАСПИСАНИЕ

УТВЕРЖДЕНО

Приказом организации от "____" _____ 20____ г. № _____

на период {{Period}} с "{{Day}}" {{Month}} 20{{Year}} г.

Штат в количестве {{Total}} единиц

Структурное подразделение		Должность (специальность, профессия), разряд, класс (категория) квалификации	Количество штатных единиц	Тарифная ставка (оклад) и пр., руб.	Надбавки, руб.			Всего, руб. ((гр.5+гр.6 +гр.7+гр.8) × гр.4)	Примечание
наименование	код				премия	районный коэффициент			
1	2	3	4	5	6	7	8	9	10

Руководитель кадровой службы _____ _____ _____
должность личная подпись расшифровка подписи

Главный бухгалтер _____ _____
личная подпись расшифровка подписи

Рисунок 1 – Образец шаблона штатного расписания

IV. Сохраните документ в формате .docx (документ Word) в папке **Шаблоны** (в которой будут расположены шаблоны всех документов).

V. Подготовьте программную среду и создать проект для работы с текстовыми документами.
   a. Запустите PyCharm.
   b. Задайте путь доступа и имя проекта (папки, в которой будут храниться программные файлы), подключите интерпретатор.
   c. Установите пакеты docxtpl и python-docx. Проверьте их подключение к вашему проекту (File – Settings – Project Interpreter).
   d. Переместите папку Шаблоны и текстовые файлы (с данными для заполнения документов) в папку проекта.
   e. Создайте папку Документы (для выходных документов) в папке проекта.

VI. Составьте программу для заполнения формы Т-3 («Штатное расписание»).
   a. В папке проекта создайте основной файл под именем Main.py, в котором будет содержаться меню программы и вызов необходимых функций. Сами функции следует описать в других файлах проекта.
   b. В папке проекта создайте файл Staffing_table.py с описанием методов для формирования штатного расписания.
   c. Введите программный код, дополнив его инструкциями, необходимыми для заполнения документа «Штатное расписание», где величина премии равна 30% и районный коэффициент равен 15% от тарифной ставки.

Содержимое файла Main.py:

```
import tkinter as tk
```

```
from Staffing_table import *

root = tk.Tk() # Создание главного окна
root.geometry("400x300+200+200")
staff = tk.Button(root, text="Заполнить штатное расписание")
staff.place(x=100, y=100)
# Вызов события при нажатии на левую кнопку мыши
staff.bind("<Button - 1>", staff_doc)

# Зацикливание обработчика событий
root.mainloop()
```

Содержимое файла Staffing_table.py:

```
import docx
import datetime
from docxtpl import DocxTemplate

def creat_dict():
    """
    Заполнение словаря для структурных подразделений
    :return: dict
    """
    f1 = open("Структурные подразделения.txt", encoding="utf-8")
    dict_sp = {}
    for line in f1:
        arr = line.strip().split(" : ")
        if len(arr) > 1:
            dict_sp[arr[0]] = arr[1]
    f1.close()
    return dict_sp

def staff_doc(event):
    """
    Заполнение штатного расписания
    :param event: событие
    :return: --
    """
    dict_sp = creat_dict()
    f1 = open("Штат сотрудников.txt", encoding="utf-8")
    # Открытие шаблона документа для редактирования
    # с помощью библиотеки docx
    document = docx.Document("Шаблоны\\shtatnoe_raspisanie.docx")
    # Доступ к первой таблице документа
    table = document.tables[1]
    # Создание массива для вычисления данных в строке Итого
    in_total = [0] * 6
    # Считывание построчно данных из файла
    for line in f1:
        arr = line.strip().split(" : ")
        # Добавление строки в таблицу
        row = table.add_row()
        # Формирование данных для каждой должности
```

```
# Здесь должен быть ваш код!!!

        # Вывод данных в строку таблицы

# Здесь должен быть ваш код!!!

        # Вычисление Итого

# Здесь должен быть ваш код!!!

    # Вывод строки Итого в таблицу

# Здесь должен быть ваш код!!!

    # Сохранение документа под новым именем
    # (промежуточная версия документа)
    document.save("test1.docx")

    # Открытие шаблона документа для редактирования
    # с помощью библиотеки docxtpl
    doc = DocxTemplate("test1.docx")
    today = datetime.datetime.today()
    # Формирование словаря для замены переменных
    context = {"Data": today.strftime("%m.%d.%Y"),
               "Day": ("%02d" % (today.day + 1)),
               "Month": ("%02d" % today.month),
               "Year": ("%02d" % today.year)[2:],
               }
    # Period (количество месяцев до конца года)
    # и Total задать самостоятельно
    doc.render(context)
    # Сохранение готового документа
    doc.save("Документы\\Штатное расписание (" +
             today.strftime("%m.%d.%Y") + ").docx")
```

Лабораторная работа № 4

Создание шаблонов документов и их автоматизированное заполнение

Задание: Предполагая, что отдел кадров принял на работу 5 сотрудников, составил трудовые договоры, и на их основе подготовил документ **Список сотрудников.xlsx** (структура в таблице 1), подготовить автоматическое заполнение приказов о приеме на работу каждого сотрудника и частичного заполнения личной карточки сотрудника, опираясь на данные из документов **Штат сотрудников.txt** и **Список сотрудников.xlsx**.

Порядок выполнения заданий

I. Составьте табличный документ **Список сотрудников_текущ_дата.xlsx**, содержащий данные не менее чем о 5 работниках, имеющий следующую структуру:

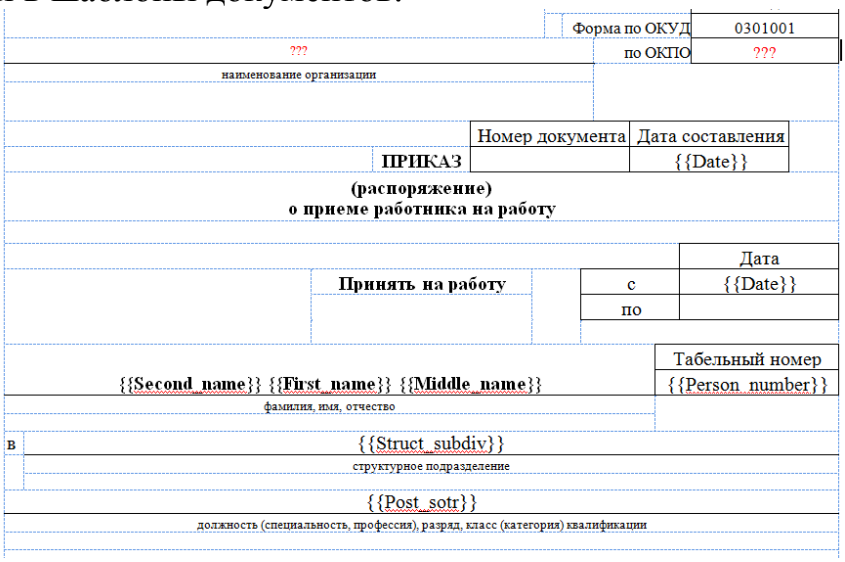
Таблица 1– Структура документа «Список сотрудников»

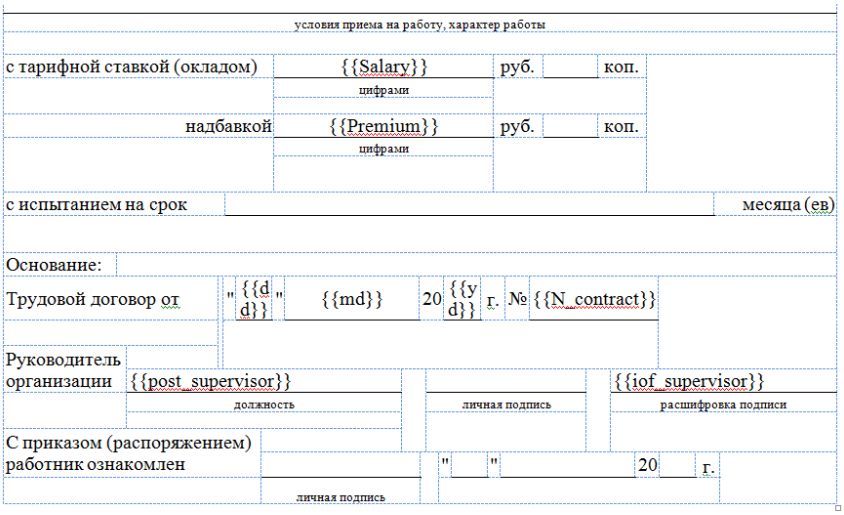
Табельный номер	Фамилия	Имя	Отчество	Код структурного подразделения	Должность	Дата трудового договора	№ трудового договора
9(3)	A(20)	A(15)	A(15)	9(2)	X(20)	9(2).9(2).9(4)	9(4)

Примечания:

- Пока не используем возможности базы данных, табельный номер сотрудника и номер трудового договора назначить самостоятельно, контролировать, чтобы они не повторялись.
- С табличным документом можно работать в программе LibreOffice.Calc, но сохранять документ нужно в формате .xlsx!
- При вводе данных код подразделения и должность следует согласовывать со штатным расписанием.

II. Создайте шаблоны для документов **Приказ о приеме на работу** и **Личная карточка,** сохраните их в папке **Шаблоны** в формате .docx. Определите поля для программного заполнения и вставьте в них переменные (рисунки 1-2). Реквизиты организации должны быть введены в шаблоны документов.





условия приема на работу, характер работы

с тарифной ставкой (окладом)	{{Salary}}	руб.		коп.
	цифрами			
надбавкой	{{Premium}}	руб.		коп.
	цифрами			

с испытанием на срок _____ месяца (ев)

Основание: _____

Трудовой договор от " {{dd}} " {{md}} 20 {{yd}} г. № {{N_contract}}

Руководитель организации {{post_supervisor}} (должность) _____ (личная подпись) {{iof_supervisor}} (расшифровка подписи)

С приказом (распоряжением) работник ознакомлен _____ (личная подпись) " " 20 г.

Рисунок 1 – Шаблон приказа о приеме на работу

	Форма по ОКУД	0301002
???	по ОКПО	???
(наименование организации)		

Дата составления	Табельный номер	Идентификационный номер налогоплательщика	Номер страхового свидетельства государственного пенсионного страхования	Алфавит	Характер работы	Вид работы (основная, по совместительству)	Пол (мужской, женский)
{{Date}}	{{Person_number}}						

ЛИЧНАЯ КАРТОЧКА
работника

I. ОБЩИЕ СВЕДЕНИЯ

Трудовой договор	номер	{{N_contract}}	
	дата	{{Date}}	

1. Фамилия {{Second_name}} Имя {{First_name}} Отчество {{Middle_name}}

II. СВЕДЕНИЯ О ВОИНСКОМ УЧЕТЕ

1. Категория запаса	6. Наименование военного комиссариата по месту жительства
2. Воинское звание	
3. Состав (профиль)	7. Состоит на воинском учете:
4. Полное кодовое обозначение ВУС	а) общем (номер команды, партии)
5. Категория годности к военной службе	б) специальном
	8. _____ (отметка о снятии с воинского учета)

Работник кадровой службы {{post_OK}} (должность) _____ (личная подпись) {{iof_OK}} (расшифровка подписи)

Работник _____ (личная подпись)

" " 20 г.

Рисунок 2 – Шаблон личной карточки

III. Программно заполните Приказы о приеме на работу для всех сотрудников из списка (должно получиться 5 файлов!). Имя файла формируйте по шаблону «Приказ о приеме на работу_ № табеля.docx».

IV. Одновременно с приказом сформируйте Личную карточку работника, куда должны быть перенесены поля: табельный номер, фамилия, имя, отчество, дата трудового договора, номер трудового договора. Также программно заполните поля: дата составления (текущая), должность ОК, ИОФ_ОК.

a. Добавьте в проект (созданный в лабораторной работе 2) новый файл **Recruitment.py** с содержимым:

```
import openpyxl
import datetime
from docxtpl import DocxTemplate

class Recruitment:
    """
```

```
        Класс для заполнения документов
        Приказ о приеме на работу и Личная карточка
        на основе текстовых документов и
        табличного документа Список сотрудников.
    """

    def save_data(self, event):
        # Метод для сохранения данных на основе шаблонов
        # и файлов с данными.
        wb = ... # Открыть "рабочую книгу"
                 # "Список сотрудников.xlsx"

        # Получить список имен всех листов.
        sheets_name = wb.sheetnames
        # Получить ссылку на первый лист.
        sheet1 = wb[sheets_name[0]]
        max_row = ... # Номер максимальной строки.

        today = ... # Получить текущую дату

        date = today.strftime("%d.%m.%Y")
        # Открыть файл "Структурные подразделения.txt"
        # и сформировать словарь dict sp
        # см. предыдущую работу

        for i in range(2, max_row + 1):
        # Сформировать словарь для замены данных в файлах .docx.
            self.__context = {}
        # Определить данные для "переменных" из файлов-шаблонов.
        # Значение табельно номера получить
        # из первого столбца документа "Список сотрудников.xlsx".
            self.__context["Person_number"] =
str(sheet1.cell(row=i, column=1).value)
        # Затем аналогично задать значения остальных переменных.
            self.__context["Second_name"] = ...

        # Для переменных "Salary" (оклад) и "Premium" (надбавка)
        # данные нужно взять из файла Штат сотрдников.txt

        # Для остальных данных, например, "iof  supervisor"
        # значения задать константами.

        # Вызов методов для заполнения и сохранения документов.
              self.__order()

    def __order(self):
        # Открытие шаблона документа prikaz o prieme.docx для
        # редактирования с помощью библиотеки docxtpl.
        # см. предыдущую работу
        doc = ...
        doc.render(self.__context)
        # Сохранить готовый документ
        # в папке Документы (имя документа описано в п. III)
```

```
    def __personal_card(self):
        # Открытие шаблона «Личной карточки»
        # Содержимое аналогично предыдущему методу.
```

b. Добавьте собственный код в тех местах, где это необходимо (задания в подчеркнутых комментариях).

c. В метод Main.py добавьте следующий код (перед командой root.mainloop())

```
recr = tk.Button(root, text="Заполнить приказ и личную карточку")
recr.place(x=85, y=150)
recruitment = Recruitment()
recr.bind("<Button - 1>", recruitment.save_data)
```

Примечание: два знака подчеркивания перед именем означает, что метод (или поле) будет скрытым (недоступным вне класса). Таким образом поддерживается инкапсуляция объекта.

Лабораторная работа № 5

Заполнение товарного чека

Задание: Создать оконное приложение для ведения диалога с пользователем. Организовать программное заполнение товарных чеков данными, вводимыми с клавиатуры.

Порядок выполнения заданий

I. Откройте документ, содержащий образец *товарного чека* (см. лабораторную работу 1). Создайте на его основе шаблон (рисунок 1).

Наименование организации (ФИО индивидуального предпринимателя):

ИНН: ???

Адрес: ???

Товарный чек № {{Number}} от {{Date}} года

№	Артикул	Наименование товара	Ед. изм.	Кол-во	Цена за 1 ед. (руб.)	Сумма (руб.)

Всего наименований {{Count}} на сумму {{Summa}}

Продавец {{FIO}}

подпись: (инициалы, фамилия)

Рисунок 1 – Образец шаблона товарного чека

II. Введите реквизиты «вашей» организации (выделены красным цветом).

III. Сохраните документ под именем **Sales_receipt** в формате **.docx** в папке Шаблоны.

IV. Внутри папки с предыдущим проектом (см. лабораторные работы 2 и 3) создайте проект для ввода данных с клавиатуры и заполнения документа «Товарный чек».

Примечание: расположение проекта может быть другим, но в этом случае нужно правильно определить пути доступа к файлам, нужным для работы.

V. Добавьте в проект три файла с описанием классов (**Product.py**, **Packing_list.py**, **Interface.py**) и файл с основным исполняемым кодом (**Main.py**):

Код файла **Product.py**:

```
class Product:
    """
    Класс, описывающий данные о товаре.
    """

    def __init__(self, code, name, unit, count, price):
        # Это конструктор.
        # В нем характеристики товара:
        # артикул, наименование, ед.измерения, количество и цена,
        # определяют поля для экземпляров класса.
        self.code = code
        self.name = name
        self.unit = unit
        self.count = count
        self.price = price

    @property
    def sum(self):
        # Это свойство.
        # Вычисление суммы товара.
        return self.count * self.price

    def __str__(self):
        # Формат вывода данных
        # при обращении к экземпляру класса.
        return
f"{self.code}\t{self.name}\t{self.unit}\t{self.count}\t{self.price
}\t{self.sum}"
```

Код файла **Packing_list.py**:

```
import docx
import datetime
from docxtpl import DocxTemplate
# Внешний модуль необходимо добавить к интерпретатору
# (в проекте на PyCharm или в командной строке).
from number_to_string import get_string_by_number
from Product import Product
```

```
class Pack:
    """
    Класс, описывающий создание списка данных
    для заполнения товарного чека
    и создания текстового файла на основе шаблона.
    """

    def __init__(self):
        # Очистка списка данных о товаре.
        self.clear()

    def clear(self):
        # Создание полей, доступных только внутри класса.
        # Список товаров.
        self.__prod = []
        # Словарь для замены данных в файле.
        self.__context = {}

    def add(self, code, name, unit, count, price):
        # Добавление данных о товаре в список.
        self.__prod.append(Product(code, name, unit, count,
price))

    def __getitem__(self, item):
        # Индексатор (итератор) для доступа
        # к элементам списка вне класса.
        return self.__prod[item]

    def __setitem__(self, key, value):
        # Индексатор для редактирования
        # элементов списка вне класса.
        self.__prod[key] = value

    def __len__(self):
        # Количество элементов в списке.
        return len(self.__prod)

    def save(self, number, fio):
        # Сохранение данных в файле.
        self.__context["Number"] = number
        self.__context["FIO"] = fio
        # Вызов инкапсулированных методов класса.
        self.__save_table()
        self.__save_data()

    def __save_table(self):
        # Открытие шаблона документа для редактирования
        # с помощью библиотеки docx
        document =
docx.Document("..\\Шаблоны\\Sales_receipt.docx")
        # Доступ к таблице документа.
        table = document.tables[0]
        # Номер строки с данными о товаре в таблице.
```

```
        i = 0
        # Общая сумма.
        summa = 0
        for prod in self.__prod:
            # prod - данные о выводимом в таблицу товаре.
            # Добавление строки в таблицу.
            row = table.add_row()
            i += 1
            # Заполнение строки данными.
            row.cells[0].text = str(i)
            row.cells[1].text = prod.code
            row.cells[2].text = prod.name
            row.cells[3].text = prod.unit
            row.cells[4].text = str(prod.count)
            row.cells[5].text = str(prod.price)
            row.cells[6].text = str(prod.sum)
            summa += prod.sum


        # Сохранение документа под новым именем
        # (промежуточная версия документа).
        document.save("test1.docx")
        # Добавление данных словарь для замены переменных.
        self.__context["Count"] = str(i)
        self.__context["Summa"] = get_string_by_number(summa)

    def __save_data(self):
        # Открытие шаблона документа для редактирования
        # с помощью библиотеки docxtpl.
        doc = DocxTemplate("test1.docx")
        today = datetime.datetime.today()
        # Добавление данных словарь для замены переменных.
        self.__context["Date"] = today.strftime("%d.%m.%Y")
        doc.render(self.__context)
        # Сохранение готового документа
        doc.save("..\\Документы\\Товарный чек (" +
self.__context["Number"] + ").docx")
```

Код файла **Interface.py**:

```
import tkinter as tk
from Packing_list import Pack

class Interface:
    """
    Класс создания интерфейса окна.
    """
    def __init__(self, form):
        self.form = form
        # Создание экземпляра класса для формирования товарного
чека.
        self.prod = Pack()
        # Порядковый номер добавленного товара.
        # Начало отсчета.
```

```
        self.count = 0
        self.widgets()

    def widgets(self):
        # Добавление виджетов на форму.
        # grid - упаковщик (размещает данные в виде таблицы).
        tk.Label(self.form, text="Введите данные о товаре",
font="20").grid(row=0, column=0)
        # Создание списка текстовых полей.
        self.entry = [0] * 7
        # Создание списка подписей (меток).
        label = ["Артикул", "Наименование товара", "Ед. изм.",
"Кол-во", "Цена (руб.)"]
        for i in range(5):
            tk.Label(self.form, text=label[i],
font="20").grid(row=1, column=i, padx=5)
            self.entry[i] = tk.Entry(self.form, font="Arial 12")
            self.entry[i].grid(row=2, column=i, padx=5)
        # Метка нужна для разделения данных на форме.
        tk.Label(self.form).grid(row=3, column=1)
        # Добавление кнопок на форму.
        self.button_add = self.do_button(4, 1, "Добавить",
self.add)
        self.button_save = self.do_button(4, 2, "Сохранить",
self.save)
        self.button_clear = self.do_button(4, 3, "Очистить",
self.clear)
        # Метка для разделения данных на форме.
        tk.Label(self.form).grid(row=6, column=1)
        tk.Label(self.form, text="Товарный чек",
font="20").grid(row=7, column=1)
        # Номер товарного чека.
        self.entry[5]=tk.Entry(self.form, font="Arial 12")
        self.entry[5].grid(row=7, column=2)
        # Многострочный вывод текста.
        self.text = tk.Text(self.form, height=20, width=80,
font="Arial 12", wrap=tk.WORD)
        self.text.grid(row=8, column=0, columnspan=4)
        # Данные о продавце для заполнения товарного чека.
        tk.Label(self.form, text="Продавец",
font="20").grid(row=9, column=1)
        self.entry[6]=tk.Entry(self.form, font="Arial 12")
        self.entry[6].grid(row=9, column=2)

    def do_button(self, row, column, text, func):
        # Метод для формирования кнопок.
        button = tk.Button(self.form, text=text, font="Arial 12",
command=func)
        button.grid(row=row, column=column, padx=15)
        return button

    def add(self):
        # Метод для добавления данных в товарный чек и поле text.
```

```
            self.prod.add(self.entry[0].get(), self.entry[1].get(),
                          self.entry[2].get(),
int(self.entry[3].get()),
                          int(self.entry[4].get()))
            self.count += 1
            self.text.insert(0.0, f"{self.count})  {self.prod[-1]}\n")

    def clear(self):
        # Метод для очитска товарного чека и поля text.
        self.prod.clear()
        self.text.delete(0.0, tk.END)
        self.count = 0

    def save(self):
        # Метод для сохранения данных.
        self.prod.save(self.entry[5].get(), self.entry[6].get())
```

Код файла **Main.py**:

```
import tkinter as tk
from Interface import Interface

root = tk.Tk()
root.geometry("1000x600")
pack_list = Interface(root)
root.mainloop()
```

Примечание:

Для того чтобы не усложнять программный код, в нем отсутствует проверка корректности вводимых данных, в том числе номеров товарных чеков.

Дополнительные задания

Добавьте в класс **Packing_list.py** методы для удаления товара (последнего в списке) и редактирования данных о товаре (по номеру элемента).

Добавьте на форму кнопки для выполнения этих операций и свяжите их с методами класса.

Лабораторная работа № 6

Автоматизированная обработка табличных документов

Задание: Создать оконное приложение для:

- заполнения табличного документа со списком всех прибывших и отправленных товаров,
- составления отчетов о наличии товара и об имеющихся товарах на складе.

Порядок выполнения заданий

I. Создайте новый проект **Отчеты**.
II. В папке проекта создайте две папки: **Прибытие**, **Отправка**.

III. В каждой папке создайте несколько текстовых файлов. Образец представлен на рисунке 1.

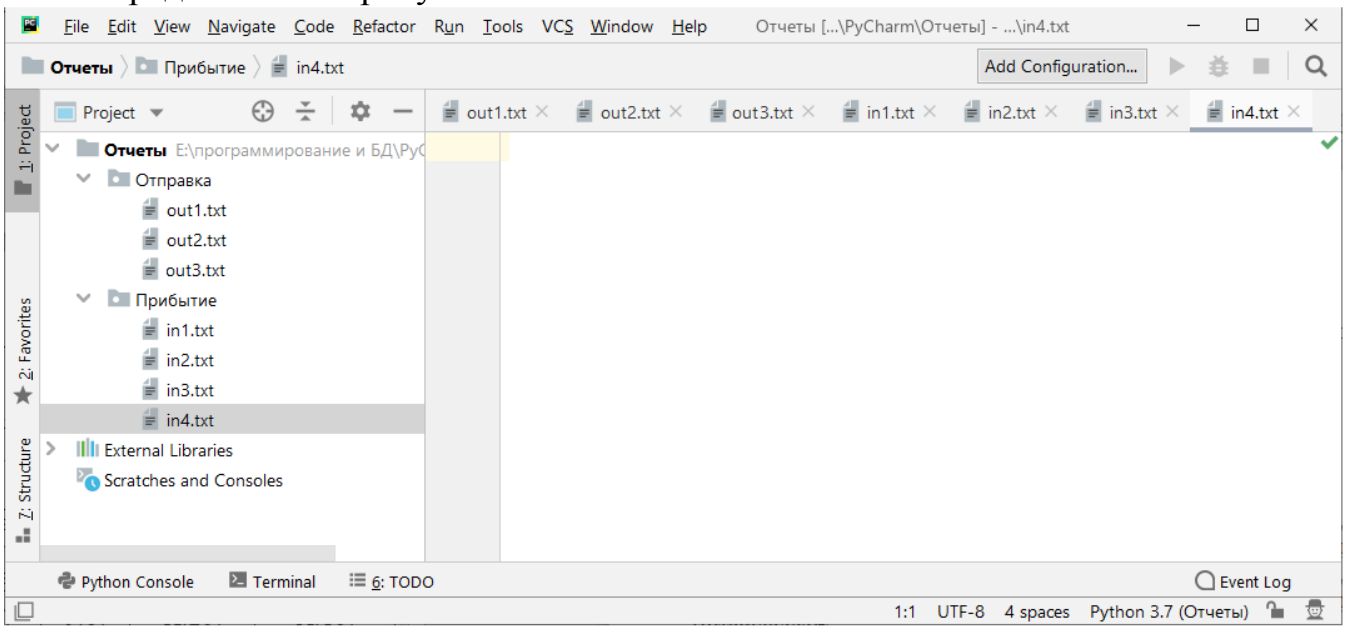


Рисунок 1 – Окно приложения на начальном этапе

IV. Введите в файлы данные следующей структуры:

Дата прибытия (отправки)
Номер счета-фактуры
Название организации
ФИО приемщика (отправителя)
[список товаров в формате]
Артикул [разделитель] Наименование товара [разделитель] Единица измерения [разделитель] Количество [разделитель] Цена, руб. [разделитель]
Артикул [разделитель] Наименование товара [разделитель] Единица измерения [разделитель] Количество [разделитель] Цена, руб. [разделитель]
[и так далее]

Примечания:

1) чтобы не усложнять программный код, нужно следить за тем, чтобы было совпадение между данными: ***Артикул***, ***Наименование товара*** и ***Единица измерения***, в различных файлах;
2) количество отправленного товара должно быть не больше, чем прибывшего.
3) Пример ввода данных:

```
10.04.2020
101
ООО "Продукты"
Иванов И.В.
С35322 : Сахар : кг. : 100 : 30
М3654112 : Мука 1 сорта : кг. : 200 : 23.50
```

V. Создайте в папке проекта каталоги: **Архив** (для хранения обработанных файлов), **Шаблоны** и **Документы**.

VI. Создайте в папке проекта табличный документ **Список товаров** (в формате .xlsx), имеющий структуру, представленную на рисунке 2.

	A	B	C	D	E	F	G	H	I	J
1	**Список товаров**									
2	Дата прибытия (отправки)	Номер счета-фактуры	Название организации	ФИО приемщика (отправителя)	Артикул	Наименование товара	Единица измерения	Количество	Цена, руб.	Примечание (П – прибытие, О – отправка)
3										

Рисунок 2 – Структура таблицы документа Список товаров.xlsx

VII. Дополните табличный документ данными из текстовых документов. Для этого:

1) добавьте в проект файл Make_Reports.py, содержащий класс с методами для работы с документами.
2) введите программный код (вместо троеточия вставить правильные данные):

```
# Подключить модуль для работы в файловой системой.
import os
# Подключить модуль для работы с датой и временем.
...
# Подключить модуль для работы с табличными документами.
...
# Подключить два модуля для работы с текстовыми документами.
...
...

class Reports:
    def Product_List(self):
        """
        Запись данных из текстовых файлов
        в табличный документ.
        """
        # Открыть "рабочую книгу" Список товаров.xlsx
        wb = ...
        # Получить ссылку на активный лист.
        sheet = ...
        # Определить номер последней строки с данными.
        row = ...
        # Создать список для заполнения строки таблицы.
        data = [0] * 10
        # Создать список из имен каталогов с файлами данных.
        dirs = ["Прибытие", "Отправка"]
        for dir in dirs:
            # Получить список файлов из каталога.
            files = os.listdir(dir)
            for file in files:
                # Получить имя файла относительно текущего каталога.
                name = dir + "\\" + file
                # Открыть файл на чтение.
                f = open(name, encoding="UTF-8")
                # Считать данные из первых 4 строк.
                for i in range(4):
                    data[i] = f.readline().strip()
                # Считать данные из оставшихся строк.
                for line in f:
                    # Если строка в файле не пустая
```

```
                # (чтобы не было ошибки в конце файла).
                if ...:
                    # Определить номер строки
                    # (увеличить ее на 1).
                    row ...
                    # Разбить строку на части.
                    mas = line.split(" : ")
                    # Изменить данные в списке
                    # (с индексами 4-6).
                    for i in range(3):
                        data[...] = ...
                    # Записать в список "количество" и "цену".
                    data[7] = float(...)
                    data[8] = ...
                    # Записать в список "примечание".
                    data[9] = dir[0]
                    # Переписать данные на строку row из списка data.
                    for j in range(...):
                        sheet.cell(...).value = ...
            # Закрыть текстовый файл.
            ...
            # Переместить обработанный файл в папку Архив
            # (чтобы данные в табличном документе не повторялись).
            os.rename(name, "Архив" + "\\" + file)
    # Сохранить изменения в табличном документе.
    ...
```

VIII. Сформируйте шаблон из документа **Отчет о наличии товара** (лаб. работа 1). Образец шаблона представлен на рисунке 3.

ОТЧЕТ¶
о·наличии·товара¶
на·{{Date}}·года¶

Артикул·¶ {{Vendor_code}}¶
Наименование·товара¶ {{Name}}¶
Единица·измерения¤ {{Unit}}¤

Количество¤	Цена,· руб.¤	Дата· прибытия· (отправки)¤	Номер· счета-фактуры¤	Название· организации¤	ФИО· приемщика· (отправителя)¤	Примечания¤

¶

Остаток·на·····{{Date}} → ¶ {{Count}}¶
Средняя·цена·за·единицу·товара·(руб.)¤ {{Average_price}}¤

Кладовщик → _____ → И.О.Ф.¶
(подпись)¶

Рисунок 3 – Образец шаблона отчета о наличии товара

IX. Сохраните документ в папке **Шаблоны** под именем **Product_Report.docx**.

X. Добавьте в проект файл **Interface.py**, описывающий программный интерфейс:

```
import tkinter as tk
# Подключить модуль для работы с диалоговыми окнами.
from tkinter import messagebox as mb

from Make_Reports import *

class Interface:
    def __init__(self, form):
        self.form = form
        # Задать заголовок и размеры окна.
        self.form.title("Формирование отчетов о товарах")
        self.form.geometry("500x400+300+200")

        # Создать экземпляр класса Reports().
        self.report = ...
        self.set_menu()

    def set_menu(self):
        """
        Формирование главного меню окна.
        """
        # Создать объект Меню на форме.
        form_menu = tk.Menu(self.form)
        # Конфигурировать окно с указанием меню для него.
        self.form.config(menu=form_menu)
        # Создать пункт меню с размещением на основном меню.
        pm1 = tk.Menu(form_menu)
        # Создать подменю для пункта "Табл. документ".
        form_menu.add_cascade(label="Табл. документ", menu=pm1)
        # Сформировать список команд пункта меню:
        # добавить пункт подменю с указанием команды,
        pm1.add_command(label="Добавить данные",
                        command=self.report.Product_List)
        # команда – это метод класса Reports().

        # Добавить второй пункт меню.
        pm2 = tk.Menu(form_menu)
        form_menu.add_cascade(label="Отчеты", menu=pm2)
        # Добавить пункты подменю.
        pm2.add_command(label="О наличии товара...",
command=self.new_win)
        pm2.add_command(label="Об остатках товаров",
command=self.report.Balance_Report)

        pm3 = tk.Menu(form_menu)  # Третий пункт меню.
        form_menu.add_cascade(label="Помощь", menu=pm3)
        pm3.add_command(label="О программе", command=self.about)

        # Четвертый пункт меню не будет содержать вложенных подпунктов.
        tk.Menu(form_menu)
        form_menu.add_cascade(label="Выход", command=self.close_win)

    def new_win(self):
        """
        Создание нового окна с элементами управления.
        Вызывается из главного окна.
        """
        form = tk.Toplevel(self.form)
```

```
        # Задать заголовок и размеры окна.
        ...
        ...
        # Создать виджеты окна.
        tk.Label(form, text="Введите артикул товара", height=3).pack()
        entry = tk.Entry(form)
        entry.pack(pady=10)
        # Получить данные из поля.
        ch = entry.get()
        # Задать команду, которая будет выполняться
        # при нажатии на кнопку.
        # Если данные не введены,
        # вызвать диалоговое окно ошибки.
        create = lambda x=ch: mb.showerror("Ошибка", "Данные
отсутствуют!") \
            if ch == "" else self.report.Product_Report(x)
        tk.Button(form, text='Создать', command=create).pack(pady=10)
        tk.Button(form, text='Отменить',
command=form.destroy).pack(pady=10)

    def close_win(self):
        """
        Действия при завершении работы с программой.
        Вызов диалогового окна.
        """
        answer = mb.askyesno(title="Выход", message="Завершить работу?")
        if answer == True:
            self.form.destroy()

    def about(self):
        """
        Создание нового окна,
        содержащего сведения о программе.
        """
        form = tk.Toplevel(self.form)
        # Задать заголовок и размеры окна.
        ...
        ...
        tk.Label(form,font="16", text="Программа предназначена для \n "
                              "формирования отчетов ... ").pack()
        # Дописать назначение программы.
```

XI. Добавьте в проект файл **Main.py** – основной запускаемый файл проекта:

```
import tkinter as tk

root = tk.Tk()
# Создать экземпляр класса Interface.
...
root.mainloop()
```

XII. Добавьте в класс **Reports** метод для создания документа по переданному в него (в качестве параметра) Артикулу товара (vendor_code):

```
def Product_Report(self, vendor_code):
    """
    Отчет о наличии товара.
    :param vendor_code: Артикул
    """
```

```
# Создать словарь для замены данных в шаблоне.
context = {}
# Открыть шаблон документа.
document = ...
# Получить доступ к первой таблице документа.
table = ...
# Задать начальные значения
# для общей суммы и количества.
summa = ...
count = ...
# Открыть табличный документ ("рабочую книгу").
wb = ...
# Создать "флаг" для проверки наличия товара.
flag = False
# Получить доступ к активному листу.
sheet = ...
# Получить номер последней строки с данными.
max_row = ...
# В цикле от строки 3 до последней:
for i in range(...):
    # Если данные в ячейке с row=i, column=5
    # равны переданному в метод параметру:
    if ...:
        # Получить данные для переменных шаблона
        # из соответствующих им ячеек строки.
        if not flag:
            context["Vendor_code"] = ...
            context["Name"] = ...
            context["Unit"] = ...
            context["Date"] = ... # Текущая дата.
            # Изменить значение переменной flag.
            flag = True
        # Определить оставшиеся данные для записи
        # в строке таблицы шаблона.
        quantity = ...  # Количество
        price = ...  # Цена
        data = ...  # Дата прибытия (отправки)
        number = ...  # Номер счета-фактуры
        organization = ... # Название организации
        fio = ...  # ФИО приемщика (отправителя)
        note = ...  # Примечание
        # Если товар "прибыл",
        # то значения count и summa увеличиваем,
        # иначе уменьшаем.
        if note == "П":
            count += ...
            summa += ...
        else:
            ...
            ...
        # Добавить строки в таблицу шаблона.
        row = ...
        # Заполнить ячейки строки данными
        # в соответствии с их назначением.
        row.cells[0].text = ...
        row.cells[1].text = ...
        row.cells[2].text = ...
        row.cells[3].text = ...
```

```
            row.cells[4].text = ...
            row.cells[5].text = ...
            row.cells[6].text = ...
    # Если данные найдены в табличном документе:
    if flag:
        # Дополнить словарь для замены.
        context["Count"] = ...
        if count != 0:
            context["Average_price"] = ...
        # Сохранить документ под новым именем
        # (промежуточная версия документа).
        document.save("test1.docx")

        # Открыть документ test1.docx для редактирования
        # с помощью библиотеки docxtpl.
        doc = ...
        # Выполнить замену в документе.
        ...
        # Сохранить готовый документ:
        # в папке "Документы" под именем
        # Отчет о товаре артикул (дата).docx
        ...
        # Удалить промежуточный документ.
        os.remove("test1.docx")
```

XIII. Добавьте в класс **Reports** метод (для того, чтобы на этом этапе работы программа запускалась на выполнение):

```
def Balance_Report(self):
    """
    Отчет об имеющихся товарах.
    """
    pass
```

Дополнительные задания

1) Сформируйте шаблон из документа **Отчет об имеющихся товарах** (лаб. работа 1). Сохраните его в папке **Шаблоны** под именем **Balance_Report.docx**.
2) Составьте код метода **Balance_Report** из класса **Reports:**
   - сначала сформируйте любую структуру, содержащую данные о товарах (словарь, список из словарей или экземпляров класса);
   - если количество товара больше 0, то выведите данные о нем в файл шаблона;
   - сохраните полученный документ в папке **Документы.**

Лабораторные работы 7 (Разработка структуры БД и ее реализация в СУБД) и 8 (Реализация функций БД в программном обеспечении) следует выполнять по заданиям, содержащимся в электронном курсе https://stepik.org/course/Информационные-системы-в-экономике-Работа-с-СУБД-MS-Access-58692/ Информационные системы в экономике. Работа с СУБД MS Access. Задачей курса является обучение практическим навыкам работы с системой управления базами данных MS Access для решения профессиональных задач.

Комплект заданий для расчетного задания (для студентов очной формы обучения) или контрольной работы (для студентов заочной формы обучения)

Вариант 1

Описание БД: В базе данных содержится список учреждения, оказывающих услуги населению, и их характеристики. В таблице *Предприниматели* содержатся сведения о предпринимателях, имеющих лицензию на оказание услуг населению.

План работы:

1. Создание БД, таблиц и связей между ними.
2. Ввод данных в таблицы.
3. Создание форм, запросов и отчетов.

База данных должна содержать таблицы (знаком ✓ отмечены ключевые поля).

Учреждения услуг		
Поле	Тип данных	Свойства поля
✓Код учреждения	Счетчик	
Название	Текстовый	
Статус	Текстовый	
Категория	Текстовый	
Продукция	Текстовый	
Адрес	Текстовый	
Предприниматель	Текстовый	

Предприниматели		
Поле	Тип данных	Свойства поля
✓Предприниматель	Текстовый	
Лицензия №	Текстовый	
Дата выдачи лицензии	Дата/время	
Юридический адрес	Текстовый	
Телефон	Текстовый	
Фотография	Поле объекта OLE	

Рабочие места		
Поле	Тип данных	Свойства поля
Код учреждения	Числовой	
Должность	Текстовый	
Кол-во рабочих мест	Числовой	
Заработная плата	Числовой	

Таблицу ***Предприниматели*** создать с помощью ***Мастера***.
Таблицу ***Учреждения услуг*** создать путем *ввода данных*. ***Рабочие места*** – в режиме *Конструктора*.

4. Создайте *связи* между таблицами.
5. Введите произвольные данные в таблицы.
6. Создайте форму (с помощью Мастера форм) в *столбец* для таблицы ***Учреждения услуг.***

   В режиме *конструктора* внесите изменения: поля ***Название*** и ***Предприниматель*** переместите в *центр*, поля ***Статус*** и ***Категория*** преобразуйте в поле со списком (вызовите *контекстное меню* для поля, выберите нужные команды, в том числе воспользуйтесь пунктом *свойства*), измените подпись поля ***Код учреждения.*** Форму назовите так же, как названа таблица.
7. Создайте *ленточную* форму (с помощью Мастера форм) для таблицы ***Предприниматели***. Форму назовите так же, как названа таблица.
8. Создайте форму с помощью *Мастера*. Из таблицы ***Учреждения услуг*** выберите поля ***Название, Статус, Адрес***, из таблицы ***Рабочие места*** – *все* поля. Форму назовите ***Рабочие места***. Введите данные в таблицы, используя данную форму. Обратите внимание, в какие поля данные вводить не нужно!
9. Отсортируйте по *возрастанию* данные в форме ***Учреждения услуг*** по полю ***Название.***
10. Выполните для формы ***Рабочие места*** фильтрацию по *выделенному* наименованию в поле ***Должность.***
11. Выполните *поиск* и *замену* в форме ***Предприниматели*** (замените название ***Юридического адреса***).
12. Создайте следующие ***запросы***:
    - *с параметром*. В запросе должны быть поля ***Название, Статус, Предпринимател***ь (из таблицы ***Учреждения услуг***); ***Лицензия №, Телефон*** (из таблицы ***Предприниматели***). В качестве параметра используйте поле ***Категория***.
    - *на выборку с вычислениями* на основе таблицы ***Рабочие места.*** Вычислите суммарную зарплату для каждой должности ([***Кол-во рабочих мест***]*[***Заработная плата***]).
13. Создайте ***отчеты*** по своим запросам.

Вариант 2

Описание БД: В базе данных содержатся сведения о выпуске товаров на некотором предприятии и о списке сотрудников этого предприятия.

План работы:

1. Создание БД, таблиц и связей между ними.
2. Ввод данных в таблицы.
3. Создание форм, запросов и отчетов.

База данных должна содержать таблицы (знаком ✓ отмечены ключевые поля).

Сотрудники		
Поле	Тип данных	Свойства поля
✓Табельный номер	Счетчик	
Фамилия	Текстовый	
Имя	Текстовый	
Отчество	Текстовый	
Код отдела	Числовой	
Должность	Текстовый	
Дата найма	Дата/время	
Фотография	Поле объекта OLE	

Вид выпускаемой продукции		
Поле	Тип данных	Свойства поля
✓Наименование товара	Текстовый	
Цена изготовителя	Денежный	
План	Числовой	

Товары		
Поле	Тип данных	Свойства поля
✓Код отдела	Счетчик	
Отдел поставки	Текстовый	
Наименование товара	Текстовый	
Кол-во выпущенной продукции	Числовой	

Таблицу ***Сотрудники*** создать с помощью ***Мастера***

Таблицу ***Товары*** создать путем *ввода данных*. ***Вид выпускаемой продукции*** – в режиме *Конструктора*.

4. Создайте *связи* между таблицами.
5. Введите произвольные данные в таблицы.
6. Создайте форму (с помощью Мастера форм) в *столбец* для таблицы ***Сотрудники***.

   В режиме *конструктора* внесите изменения: поля ***Фамилия*** и ***Должность*** оставьте *слева*, остальные поместите *справа*, поле ***Должность*** преобразуйте в поле со списком (вызовите *контекстное меню* для поля, выберите нужные команды, в том числе воспользуйтесь пунктом *свойства*). Форму назовите так же, как названа таблица.
7. Создайте *ленточную* форму (с помощью Мастера форм) для таблицы ***Вид выпускаемой продукции***. Измените подпись поля ***Цена изготовителя*** (в режиме *конструктора*). Форму назовите так же, как названа таблица.
8. Создайте форму с помощью *Мастера*. Из таблицы ***Товары*** выберите *все* поля, из таблицы ***Вид выпускаемой продукции*** выберите поля ***Цена изготовителя***, ***План***. Форму назовите ***Товары***. Введите данные в таблицы, используя данную форму. Обратите внимание, в какие поля данные вводить не нужно!

9. Отсортируйте по *возрастанию* данные в форме ***Сотрудники*** по полю ***Фамилия***.
10. Выполните для формы ***Товары*** фильтрацию по *выделенному* наименованию в поле ***Наименование товара***.
11. Выполните *поиск* и *замену* в форме ***Сотрудники*** (замените ***Фамилию*** сотрудника).
12. Создайте следующие ***запросы***:
    — *с параметром*. В запросе должны быть поля ***Фамилия***, ***Имя***, ***Отчество***, ***Код отдела*** (из таблицы ***Сотрудники***); ***Отдел поставки*** (из таблицы ***Товары***). В качестве параметра используйте поле ***Должность***.
    — *на выборку с вычислениями* на основе таблиц ***Вид выпускаемой продукции*** и ***Товары***. Вычислите стоимость каждого товара ([***Цена изготовителя***]*[***Кол-во выпущенной продукции***]).
13. Создайте ***отчеты*** по своим запросам.

Вариант 3

План работы:
1. Создание БД, таблиц и связей между ними.
2. Ввод данных в таблицы.
3. Создание форм, запросов и отчетов.

База данных должна содержать таблицы (знаком ✓ отмечены ключевые поля).

Товары		
Поле	Тип данных	Свойства поля
✓Код товара	Счетчик	
Наименование	Текстовый	
Опт цена за ед.	Денежный	
Единица измерения	Текстовый	
Кол-во в партии	Числовой	

Заказы		
Поле	Тип данных	Свойства поля
Код товара	Числовой	
Кол-во партий	Числовой	
Заказчик	Текстовый	
Дата заказа	Дата/время	

Заказчики		
Поле	Тип данных	Свойства поля
✓Код заказчика	Счетчик	
Наименование	Текстовый	
Город	Текстовый	
Адрес	Текстовый	
Контактный телефон	Текстовый	
Примечания	Поле МЕМО	

Таблицу ***Заказчики*** создать с помощью ***Мастера***.
Таблицу ***Товары*** создать путем *ввода данных*. ***Заказы*** – в режиме *Конструктора*.

4. Создайте *связи* между таблицами.
5. Введите произвольные данные в таблицы.
6. Создайте форму (с помощью Мастера форм) в *столбец* для таблицы ***Товары***.

   В режиме *конструктора* внесите изменения: поле ***Наименование*** переместите в *центр*, поле ***Единица измерения*** преобразуйте в поле со списком (вызовите *контекстное меню* для поля, выберите нужные команды, в том числе воспользуйтесь пунктом *свойства*), измените подпись поля ***Опт цена за ед.*** Форму назовите так же, как названа таблица.
7. Создайте *ленточную* форму (с помощью Мастера форм) для таблицы ***Заказчик***. Форму назовите так же, как названа таблица.
8. Создайте форму с помощью *Мастера*. Из таблицы ***Заказчик*** выберите поля ***Наименование, Контактный телефон, Адрес***, из таблицы ***Заказы*** – *все* поля. Форму назовите ***Заказы***. Для поля Дата заказа сделайте *маску ввода* 00.00.00 (в режиме *конструктора*). Введите данные в таблицы, используя данную форму. Обратите внимание, в какие поля данные вводить не нужно!
9. Отсортируйте по *возрастанию* данные в форме ***Товары*** по полю ***Название***.
10. Выполните для формы ***Заказчик*** фильтрацию по *выделенному* наименованию в поле ***Город***.
11. Выполните *поиск* и *замену* в форме ***Заказы*** (замените ***Количество партий*** товара).
12. Создайте следующие ***запросы***:
    — *с параметром*. В запросе должны быть все поля из таблицы ***Заказы***; ***Наименование*** (из таблицы ***Товары***); ***Адрес***, ***Контактный телефон*** (из таблицы ***Заказчики***). В качестве параметра используйте поле ***Код товара***.
    — *на выборку с вычислениями* на основе таблиц ***Товары*** и ***Заказы***. Вычислите суммарную стоимость заказы по каждому наименованию ([***Опт цена за ед***]*[***Кол-во в партии***]*[***Кол-во партий***]).
13. Создайте ***отчеты*** по своим запросам.

Вариант 4

Описание БД: В базе данных содержатся сведения о наличии книг в школьной библиотеке, а также информация о читателях и о прочитанных ими книгах.
План работы:

1. Создание БД, таблиц и связей между ними.
2. Ввод данных в таблицы.
3. Создание форм, запросов и отчетов.

База данных должна содержать таблицы (знаком ✓ отмечены ключевые поля).

Книги		
Поле	Тип данных	Свойства поля
✓Шифр книги	Счетчик	
Автор	Текстовый	
Название	Текстовый	
Год издания	Числовой	
Издательство	Текстовый	

Список читателей		
Поле	Тип данных	Свойства поля
✓Код читателя	Счетчик	
Фамилия	Текстовый	
Имя	Текстовый	
Дата рождения	Дата/время	
Класс	Текстовый	

Классы		
Поле	Тип данных	Свойства поля
✓№ класса	Текстовый	
Профиль	Текстовый	
№ кабинета	Числовой	
Классный руководитель	Текстовый	

Выдача книг		
Поле	Тип данных	Свойства поля
Код читателя	Числовой	
Шифр книги	Числовой	
Дата выдачи	Дата/время	
Возврат	Логический	

Таблицу ***Книги*** создать с помощью ***Мастера***.

Таблицу ***Выдача книг*** создать путем *ввода данных*. ***Классы, Список читателей*** – в режиме *Конструктора*.

4. Создайте *связи* между таблицами.
5. Введите произвольные данные в таблицы.
6. Создайте форму (с помощью Мастера форм) в *столбец* для таблицы ***Книги***.

   В режиме *конструктора* внесите изменения: поля ***Автор*** и ***Название*** оставьте *слева*, остальные переместите в *центр*, поле ***Издательство*** преобразуйте в поле со списком (вызовите *контекстное меню* для поля, выберите нужные команды, в том числе воспользуйтесь пунктом *свойства*). Форму назовите так же, как названа таблица.
7. Создайте *ленточную* форму (с помощью Мастера форм) для таблицы ***Классы***. Измените подпись поля ***Классный руководитель*** (в режиме *конструктора*). Форму назовите так же, как названа таблица.

8. Создайте форму с помощью *Мастера*. Из таблицы ***Выдача книг*** выберите *все* поля, из таблицы ***Список читателей*** выберите поля ***Фамилия***, ***Имя, Класс***. Форму назовите ***Выдача книг***. Введите данные в таблицы, используя данную форму. Обратите внимание, в какие поля данные вводить не нужно!
9. Отсортируйте по *возрастанию* данные в форме ***Книги*** по полю ***Автор***.
10. Выполните для формы ***Классы*** фильтрацию по *выделенному* наименованию в поле ***Профиль***.
11. Выполните *поиск* и *замену* в форме ***Выдача книг*** (замените ***Шифр книги***).
12. Создайте следующие ***запросы***:
    — *на выборку*. В запросе должны быть поля ***Код читателя***, ***Шифр книги***, ***Дата выдачи*** (из таблицы ***Выдача книг***); ***Автор***, ***Название*** (из таблицы ***Книги***); ***Фамилия***, ***Имя***, ***Класс*** (из таблицы ***Список читателей***). Выведите список не возвращенных книг.
    — *с параметром*. Выведите список книг по введенному ***Автору***.
13. Создайте ***отчеты*** по своим запросам.

Вариант 5

Описание БД: В базе данных содержится список телефонов некоторого учреждения (таблица ***Справочник***), а также список сотрудников (в таблицах ***Клиенты*** и ***Переговоры***), которым разрешено пользование рабочими телефонами. Таблица *Оплата* содержит тарифы за оказание услуг телефонной связи.

План работы:
1. Создание БД, таблиц и связей между ними.
2. Ввод данных в таблицы.
3. Создание форм, запросов и отчетов.

База данных должна содержать таблицы (знаком ✓ отмечены ключевые поля).

Справочник		
Поле	Тип данных	Свойства поля
✓№ телефона	Текстовый	
Ответственный абонент	Текстовый	
Год установки	Дата/время	
Вид установки	Текстовый	

Клиенты		
Поле	Тип данных	Свойства поля
✓Код клиента	Счетчик	
Фамилия	Текстовый	
Отдел	Текстовый	
Номер телефона	Текстовый	
Заметки	Поле MEMO	
Разрешение на переговоры	Логический	

Оплата		
Поле	Тип данных	Свойства поля
✓Вид установки	Текстовый	
Абонентская плата	Денежный	
Переговоры за мин	Денежный	

Переговоры		
Поле	Тип данных	Свойства поля
Код клиента	Числовой	
Дата	Дата/время	
Время переговоров, мин	Числовой	

Таблицу ***Клиенты*** создать с помощью ***Мастера***.
Таблицу ***Переговоры*** создать путем *ввода данных*. ***Оплата*** и ***Справочник*** – в режиме *Конструктора*.

4. Создайте *связи* между таблицами.
5. Введите произвольные данные в таблицы.
6. Создайте форму (с помощью Мастера форм) в *столбец* для таблицы ***Справочник.***

   В режиме *конструктора* внесите изменения: поле ***№ телефона*** переместите в *центр*, поле ***Вид установки*** преобразуйте в поле со списком (вызовите *контекстное меню* для поля, выберите нужные команды, в том числе воспользуйтесь пунктом *свойства*), измените подпись поля ***Ответственный абонент.*** Форму назовите так же, как названа таблица.
7. Создайте *ленточную* форму (с помощью Мастера форм) для таблицы ***Клиенты***. Форму назовите так же, как названа таблица.
8. Создайте форму с помощью *Мастера*. Из таблицы ***Переговоры*** выберите *все* поля, из таблицы ***Клиенты*** выберите поля ***Фамилия***, ***Номер телефона***, ***Отдел***. Форму назовите ***Переговоры***. Введите данные в таблицы, используя данную форму. Обратите внимание, в какие поля данные вводить не нужно!
9. Отсортируйте по *возрастанию* данные в форме ***Клиенты*** по полю ***Фамилия***.
10. Выполните для формы ***Справочник*** фильтрацию по *выделенному* наименованию в поле ***№ телефона***.
11. Выполните *поиск* и *замену* в форме ***Переговоры*** (замените ***Дату***).
12. Создайте следующие ***запросы***:

    — *с параметром*. В запросе должны быть поля ***Отдел***, ***Номер телефона***, ***Разрешение на переговоры*** (из таблицы ***Клиенты***); ***Ответственный абонент***, ***Вид установки*** (из таблицы ***Справочник***). В качестве параметра используйте поле ***Фамилия*** (клиента).

    — *на выборку с вычислениями* на основе таблиц ***Переговоры*** и ***Оплата***. Вычислите оплату за переговоры по каждому клиенту за

каждую дату (имеющуюся в списке) ([***Время переговоров***]*[***Переговоры за мин***]).

13. Создайте ***отчеты*** по своим запросам.

Вариант 6

Описание БД: В таблице *Кабинет* содержатся данные обо всех школьных кабинетах и об учителях, отвечающих за содержание кабинетов. В таблице *Имущество* содержится список всего школьного имущества с указанием кабинетов, в которых оно содержится. В таблице *Сроки эксплуатации* приведена износостойкость всех имеющихся видов имущества.

План работы:

1. Создание БД, таблиц и связей между ними.
2. Ввод данных в таблицы.
3. Создание форм, запросов и отчетов.

База данных должна содержать таблицы (знаком ✓ отмечены ключевые поля).

Имущество		
Поле	Тип данных	Свойства поля
✓Код имущества	Счетчик	
Название	Текстовый	
Модель	Текстовый	
Изготовитель	Текстовый	
Дата покупки	Дата/время	
Цена покупки	Числовой	
Количество	Числовой	
Заметки	Поле MEMO	
№ кабинета	Числовой	

Кабинет		
Поле	Тип данных	Свойства поля
✓№ кабинета	Счетчик	
Этаж	Числовой	
Преподаваемый предмет	Текстовый	
Учитель	Текстовый	

Сроки эксплуатации		
Поле	Тип данных	Свойства поля
✓Название	Текстовый	
Единица измерения срока	Текстовый	
Гарантийный срок	Числовой	
Срок износа	Числовой	

Таблицу ***Имущество*** создать с помощью ***Мастера***.

Таблицу ***кабинет*** создать путем *ввода данных*. ***Сроки эксплуатации*** – в режиме *Конструктора*.

4. Создайте *связи* между таблицами.
5. Введите произвольные данные в таблицы.
6. Создайте форму (с помощью Мастера форм) в *столбец* для таблицы ***Сроки эксплуатации***.

   В режиме *конструктора* внесите изменения: поле ***Название*** оставьте *слева*, остальные переместите в *центр*, поле ***Единица измерения срока*** преобразуйте в поле со списком (вызовите *контекстное меню* для поля, выберите нужные команды, в том числе воспользуйтесь пунктом *свойства*), сделайте для этого поля подпись. Форму назовите так же, как названа таблица.
7. Создайте *ленточную* форму (с помощью Мастера форм) для таблицы ***Кабинет***. Измените подпись поля ***Преподаваемый предмет***. Форму назовите так же, как названа таблица.
8. Создайте форму с помощью *Мастера*. Из таблицы ***Имущество*** выберите поля ***Код имущества, Название, Изготовитель, Дата покупки, Количество***, из таблицы ***Сроки эксплуатации –*** поля ***Гарантийный срок, Единица измерения срока***. Форму назовите ***Имущество***. Для поля ***Дата покупки*** сделайте *маску ввода* 00.00.00 (в режиме *конструктора*). Введите данные в таблицы, используя данную форму. Обратите внимание, в какие поля данные вводить не нужно!
9. Отсортируйте по *возрастанию* данные в форме ***Кабинет*** по полю ***Учитель***.
10. Выполните для формы ***Имущество*** фильтрацию по *выделенному* наименованию в поле ***Название***.
11. Выполните *поиск* и *замену* в форме ***Сроки эксплуатации*** (замените ***Срок износа***).
12. Создайте следующие ***запросы***:
    — *с параметром*. В запросе должны быть поля ***Название, Модель, Количество, № кабинета*** (из таблицы ***Имущество***); ***Преподаваемый предмет*** и ***Учитель*** (из таблицы ***Кабинеты***). В качестве параметра используйте поле ***Дата покупки***.
    — *на выборку с вычислениями* на основе таблицы ***Имущество***. Вычислите стоимость имущества для каждого названия ([***Цена покупки***]*[***Количество***]).
13. Создайте ***отчеты*** по своим запросам.

Вариант 7

Описание БД: В базе данных содержится информация о студентах (например, третьего курса) и преподавателях, а также все экзаменационные оценки.

План работы:

1. Создание БД, таблиц и связей между ними.
2. Ввод данных в таблицы.
3. Создание форм, запросов и отчетов.

База данных должна содержать таблицы (знаком ✓ отмечены ключевые поля).

Личное дело		
Поле	Тип данных	Свойства поля
✓Код студента	Счетчик	
Фамилия	Текстовый	
Имя	Текстовый	
Отчество	Текстовый	
Адрес	Текстовый	
№ телефона	Текстовый	
Специализация	Текстовый	
Фотография	Поле объекта OLE	
№ группы	Текстовый	

Преподаватели		
Поле	Тип данных	Свойства поля
✓Код преподавателя	Счетчик	
ФИО преподавателя	Текстовый	
Кафедра	Текстовый	
Должность	Текстовый	
Звание	Текстовый	
Нагрузка, ч	Числовой	

Экзамены		
Поле	Тип данных	Свойства поля
Код студента	Числовой	
Дисциплина	Текстовый	
Код преподавателя	Числовой	
Оценка	Числовой	

Таблицу ***Студенты*** создать с помощью ***Мастера***.

Таблицу ***Темы*** создать путем *ввода данных*. ***План просмотра*** – в режиме *Конструктора*.

4. Создайте связи между таблицами.
5. Введите произвольные данные в таблицы.
6. Создайте форму (с помощью Мастера форм) в *столбец* для таблицы ***Преподаватели.***

   В режиме *конструктора* внесите изменения: поля ***ФИО преподавателя*** и ***Должность*** переместите в *центр*, поля ***Должность*** и ***Звание*** преобразуйте в поле со списком (вызовите *контекстное меню* для поля, выберите нужные команды, в том числе воспользуйтесь пунктом *свойства*), измените подпись поля ***Код преподавателя.*** Форму назовите так же, как названа таблица.
7. Создайте *ленточную* форму (с помощью Мастера форм) для таблицы ***Экзамены***. Форму назовите так же, как названа таблица.
8. Создайте форму с помощью *Мастера*. Из таблицы ***Экзамены*** выберите *все* поля, из таблицы ***Личное дело*** выберите поля ***Фамилия***, ***Имя***,

Отчество, из таблицы ***Преподаватели*** – ***ФИО преподавателя***. Форму назовите ***Студенты***. Введите данные в таблицы, используя данную форму. Обратите внимание, в какие поля данные вводить не нужно!

9. Отсортируйте по *возрастанию* данные в форме ***Студенты*** по полю ***Фамилия***.
10. Выполните для формы ***Экзамены*** фильтрацию по *выделенному* наименованию в поле ***Дисциплина***.
11. Выполните *поиск* и *замену* в форме ***Преподаватели*** (замените ***Нагрузку***).
12. Создайте следующие ***запросы***:
    — *на выборку с вычислениями*. Вычислите для каждого преподавателя количество студентов, у которых он принимал экзамены (на основе таблиц ***Экзамены*** и ***Преподаватели***).
    — *с параметром*. Выведите список студентов по введенной ***Специализации*** (на основе таблицы ***Личное дело***).
13. Создайте ***отчеты*** по своим запросам.

Вариант 8

Описание БД: В базе данных содержится информация о работе проектной организации. В том числе о разработанных проектах, об имеющихся в структуре организации отделах и об исполнителях проектов.

План работы:

1. Создание БД, таблиц и связей между ними.
2. Ввод данных в таблицы.
3. Создание форм, запросов и отчетов.

База данных должна содержать таблицы (знаком ✓ отмечены ключевые поля).

Проекты		
Поле	Тип данных	Свойства поля
✓Код проекта	Счетчик	
Название проекта	Текстовый	
Описание проекта	Текстовый	
Номер заказа	Числовой	
Оценочная стоимость	Денежный	
Дата начала проекта	Дата/время	
Дата завершения проекта	Дата/время	
Признак исполнения	Логический	

Отделы		
Поле	Тип данных	Свойства поля
✓Код отдела	Счетчик	
Название	Текстовый	
Подразделение	Текстовый	
Руководитель	Текстовый	
Телефон	Текстовый	

Исполнение		
Поле	Тип данных	Свойства поля
Код отдела	Числовой	
Код проекта	Числовой	
Ответственный за исполнение	Текстовый	
Кол-во исполнителей	Числовой	

Таблицу ***Проекты*** создать с помощью ***Мастера***.

Таблицу ***Отделы*** создать путем *ввода данных*. ***Исполнение*** – в режиме *Конструктора*.

4. Создайте *связи* между таблицами.
5. Введите произвольные данные в таблицы.
6. Создайте форму (с помощью Мастера форм) в *столбец* для таблицы ***Проекты***.

   В режиме *конструктора* внесите изменения: поля ***Название проекта*** и ***Номер заказа*** поместите в *центре*, поле ***Описание проекта*** преобразуйте в поле со списком (вызовите *контекстное меню* для поля, выберите нужные команды, в том числе воспользуйтесь пунктом *свойства*), для полей ***Дата начала проекта*** и ***Дата завершения проекта*** сделайте *маску ввода* 00.00.00 и *подпись*. Форму назовите так же, как названа таблица.
7. Создайте *ленточную* форму (с помощью Мастера форм) для таблицы ***Отделы***. Форму назовите так же, как названа таблица.
8. Создайте форму с помощью *Мастера*. Из таблицы ***Исполнение*** выберите *все* поля, из таблицы ***Проекты*** выберите поля ***Название проекта***, ***Признак исполнения***. Форму назовите ***Исполнение***. Введите данные в таблицы, используя данную форму. Обратите внимание, в какие поля данные вводить не нужно!
9. Отсортируйте по *возрастанию* данные в форме ***Проекты*** по полю ***Название проекта***.
10. Выполните для формы ***Исполнение*** фильтрацию по *выделенному* наименованию в поле ***Ответственный за исполнение***.
11. Выполните *поиск* и *замену* в форме ***Отделы*** (замените ***Руководителя***).
12. Создайте следующие ***запросы***:

    — *на выборку*. В запросе должны быть все поля из таблицы ***Исполнение***; ***Название проекта***, ***Номер заказа***, ***Оценочная стоимость*** (из таблицы ***Проекты***); ***Название, Подразделение***, ***Руководитель*** (из таблицы ***Отделы***). Выведите список исполненных проектов.

    — *с параметром*. Выведите данные о проекте по введенному ***Коду проекта*** (на основе таблицы ***Проекты***).
13. Создайте ***отчеты*** по своим запросам.

ПРАВИЛА ОФОРМЛЕНИЯ РАБОТЫ

Отчет должен содержать следующие страницы: титульный лист с указанием варианта, формулировку заданий по вашему варианту, решение задач с пояснениями по ходу выполнения.

Отчет выполняется на листах формата А4 средствами текстового процессора **OpenOffice.org Writer** или любого другого офисного приложения подобного класса**.** Для оформления текста, кроме случаев со специальным форматированием, необходимо использовать следующие параметры форматирования:

- *Поля документа*: *левое* – 3 см, правое – 1,5 см, верхнее и нижнее по 2 см.
- *Текст*: шрифт **Times New Roman**, размер **14**, выравнивание **по ширине,** отступ **первой строки** на **1,25 см**, межстрочный интервал **полуторный**.
- *Заголовки*: шрифт **Times New Roman,** размер **14,** начертание **полужирное**, буквы **все прописные**, выравнивание **по центру**, интервал после – 0.5 см.

Отчет предоставляется в распечатанном и скрепленном виде за один месяц до начала сессии.

Список использованной литературы

1. Информационные технологии : учебник / Ю.Ю. Громов, И.В. Дидрих, О.Г. Иванова, и др. ; Тамбовский государственный технический университет. – Тамбов : Тамбовский государственный технический университет (ТГТУ), 2015. – 260 с. : ил., табл., схем. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=444641 (дата обращения: 17.12.2020). – Библиогр. в кн. – ISBN 978-5-8265-1428-3. – Текст : электронный.

2. Иванов, В.И. Информатика. Информационные технологии : учебное пособие / В.И. Иванов, Н.В. Баскакова ; Кемеровский государственный университет. – Кемерово : Кемеровский государственный университет, 2015. – 228 с. : 2015 – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=437474 (дата обращения: 09.01.2021). – ISBN 978-5-8353-1811-7. – Текст : электронный.

3. Хныкина, А.Г. Информационные технологии : учебное пособие / А.Г. Хныкина, Т.В. Минкина ; Северо-Кавказский федеральный университет. – Ставрополь : Северо-Кавказский Федеральный университет (СКФУ), 2017. – 126 с. : схем., ил. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=494703 (дата обращения: 19.11.2020). – Библиогр. в кн. – Текст : электронный.

4. Информационные технологии: лабораторный практикум : [16+] / авт.-сост. А.Г. Хныкина, Т.В. Минкина ; Северо-Кавказский федеральный университет. – Ставрополь : Северо-Кавказский Федеральный университет (СКФУ), 2018. – 122 с. : ил. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=562883 (дата обращения: 19.11.2020). – Библиогр. в кн. – Текст : электронный.

5. Лисяк, В.В. Разработка информационных систем : учебное пособие : [16+] / В.В. Лисяк ; Южный федеральный университет. – Ростов-на-Дону ; Таганрог : Южный федеральный университет, 2019. – 97 с. : ил. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=577875 (дата обращения: 17.12.2020). – Библиогр.: с. 91 - 93. – ISBN 978-5-9275-3168-4. – Текст :

6. Электронно-библиотечная система «Университетская библиотека Online» [Электронный ресурс]. – М.: Издательство «Директ-Медиа». – Режим доступа: http://www.biblioclub.ru

7. Онлайн-курсы [режим доступа] https://stepik.org/catalog