

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ СТАВРОПОЛЬСКОГО КРАЯ**  
**Государственное бюджетное профессиональное образовательное учреждение**  
**«Ставропольский строительный техникум»**

М. И. Данилова, Л. А. Антошина

**ИНФОРМАТИКА**

Методические указания по выполнению практических работ

07.02.01 Архитектура

08.02.01 Строительство и эксплуатация зданий и сооружений

08.02.08 Монтаж и эксплуатация оборудования и систем газоснабжения

08.02.12 Строительство и эксплуатация автомобильных дорог, аэродромов и  
городских путей сообщения

08.02.13 Монтаж и эксплуатация внутренних сантехнических устройств,  
кондиционирования воздуха и вентиляции

21.02.19 Землеустройство

1 курс, очная форма обучения

Ставрополь, 2025

УДК 004  
ББК 16.2

*Рекомендовано к применению решением Методического совета ГБПОУ  
«Ставропольский строительный техникум» в качестве методических  
указаний по выполнению практических работ*

Рецензент

**Киричек Ксения Александровна**, заведующий кафедрой математики,  
информатики и цифровых образовательных технологий, кандидат  
педагогических наук, доцент, ГБОУ ВО «Ставропольский государственный  
педагогический институт»

**Данилова М. И., Антошина Л. А.**

Информатика: методические указания по выполнению практических работ в 2 ч.  
Ч. 2. / Данилова М. И., Антошина, Л. А. – издательство ГБПОУ «Ставропольский  
строительный техникум», 2025. – 115 с.: ил.

Методические указания по выполнению практических работ охватывают широкий спектр тем по программированию, работе с данными и электронными таблицами. Рассматриваются основы ввода и вывода данных, реализация линейных, разветвляющихся, циклических и комбинированных алгоритмов, а также обработка строк и списков в Python. Особое внимание уделяется анализу алгоритмов с применением таблиц трассировки, созданию и использованию реляционных баз данных, формированию запросов, сортировке и фильтрации информации. При работе с электронными таблицами изучаются методы адресации, обработка больших массивов данных с использованием математических, статистических, логических, финансовых и текстовых функций, а также построение диаграмм, графиков и моделирование на примерах профессиональных задач.

УДК 004  
ББК 16.2

© ГБПОУ ССТ, 2025 г.  
© Данилова М. И., Антошина Л. А., 2025 г.  
© Оформление. Издательство ГБПОУ ССТ

## **РАССМОТРЕНО**

на заседании цикловой комиссии  
естественно-математических  
дисциплин  
Протокол №10 от «20» мая 2025 г.  
и. о. председателя цикловой комиссии  
Н. В. Корякина

## **РЕКОМЕНДОВАНО**

к применению решением  
Методического совета  
ГБПОУ ССТ протокол №10  
от «27» мая 2025 г.

## **СОГЛАСОВАНО**

Л. В. Белоусова,  
заместитель директора по учебно-  
методической работе и качеству  
«20» мая 2025 г.

## **Рецензенты:**

К. А. Киричек, кандидат педагогических наук, доцент, заведующий  
кафедрой математики, информатики и цифровых образовательных  
технологий, ГБОУ ВО «Ставропольский государственный педагогический  
институт»  
«20» мая 2025 г.

## **Авторы-составители:**

М. И. Данилова, преподаватель цикловой комиссии естественно-  
математических дисциплин, методист ГБПОУ ССТ  
Л. А. Антошина, преподаватель цикловой комиссии естественно-  
математических дисциплин ГБПОУ ССТ

«20» мая 2025 г.

## Содержание

<b>Практическая работа №27.</b> Организация ввода и вывода данных. Программирование линейных, разветвляющихся алгоритмов .....	5
<b>Практическая работа №28.</b> Программирование разветвляющихся, циклических и комбинированных алгоритмов .....	14
<b>Практическая работа №29-30.</b> Работа со строками и циклами в Python.....	26
<b>Практическая работа №31.</b> Работа со списками в Python: создание, модификация и основные методы .....	35
<b>Практическая работа №32.</b> Анализ алгоритмов с использованием таблиц трассировки без использования компьютера .....	42
<b>Практическая работа №33.</b> Создание реляционной базы данных. Сортировка. Фильтрация .....	46
<b>Практическая работа №34.</b> Создание простых и вычислительных запросов в реляционной базе данных.....	54
<b>Практическая работа №35.</b> Вычисление в электронной таблице: относительная, абсолютная и смешанная адресация.....	59
<b>Практическая работа №36.</b> Фильтрация и сортировка большого массива данных в электронных таблицах .....	66
<b>Практическая работа №37.</b> Математические, статистические и логические функции. Обработка большого массива данных .....	74
<b>Практическая работа №38.</b> Финансовые функции.....	81
<b>Практическая работа №39.</b> Текстовые функции .....	84
<b>Практическая работа №40-41.</b> Построение диаграмм для иллюстрации статистических данных. Построение графиков функций .....	90
<b>Практическая работа №42.</b> Моделирование в электронных таблицах (на примерах задач из профессиональной области).....	100
<b>Список информационных источников</b> .....	111
<b>Справочная информация.</b> Сочетания клавиш .....	112

## Практическая работа №27. Организация ввода и вывода данных. Программирование линейных, разветвляющихся алгоритмов

**Цель работы:** получить навыки в организации ввода/вывода данных и порядка работы в среде разработки; овладеть практическими навыками разработки и программирования вычислительного процесса линейной структуры и навыками по отладке и тестированию программ, приобрести навыки в записи выражений на языке программирования *Python*.

### Теоретические сведения

**Переменная** – это элемент данных для хранения информации, имеющий собственное имя.

*Как задать переменную?*

- придумать имя переменной;
- задать значение переменной.

**Пример:**

```
name = 37
    имя      значение
```

Оператор присваивания «= $\Rightarrow$ » задает или меняет значение переменной.

**Примечание:** имя переменной должно начинаться с маленькой латинской буквы, должно быть осмысленным (английские слова или общеупотребимые сокращения) и не должно превышать по длине 10-15 символов.

Пример работы программы с присвоением значений переменной представлен на рис. 1.

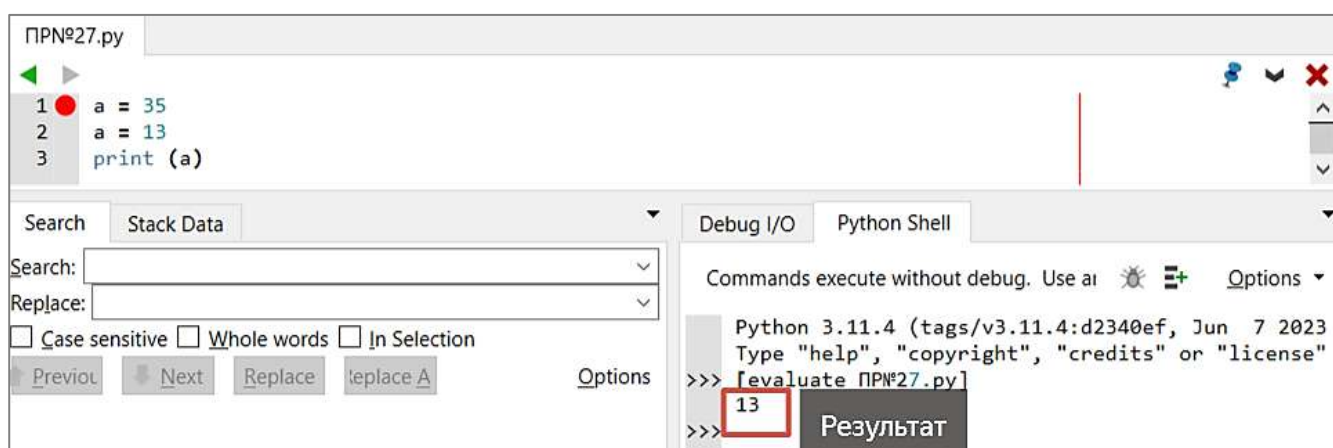


Рис. 1. Запуск программы в среде разработки Wing IDE 101

## Правила для переменных:

### Разрешено

- начинать с латинской буквы (`test`) или символа подчеркивания (`_test`);
- содержать только буквенно-цифровые символы и подчеркивание (`Az`, `0-9` и `_`);
- использовать цифры после букв;
- использовать до 15-ти символов.

### Недопустимо

- начинать с цифры;
- совпадение с зарезервированным ключевыми словами (`print`, `if`, `not`, `while`, `True` и т. д.);
- использовать более 15-ти символов;
- использовать любые знаки (`!`, `@`, `№`, `#`), кроме нижнего подчеркивания;
- внутри названия переменной, которое состоит из нескольких слов ставить пробел (например, `camel Case`).

Таблица 1 – Типы переменных

Тип	Описание	Запись на Python
<code>int</code>	<b>Целые числа</b> ( <code>integer</code> ) – положительные и отрицательные целые числа, а также 0 (например: <code>4</code> , <code>687</code> , <code>-45</code> , <code>0</code> ).	<pre>one_hundred = 100 five_hundred = 500 print (one_hundred + five_hundred) # вывод 600</pre> <p>Дробные числа приводятся к целым функцией <code>int()</code>. Она не округляет числа, а отбрасывает дробную часть:</p> <pre>print (int (3.14)) # вывод целой части, равной трём print (int (2.72)) # вывод целой части, равной двум</pre>
<code>float</code>	Числа с плавающей точкой ( <code>float point</code> ) – <b>дробные числа</b> (например: <code>1.45</code> , <code>3.789654</code> , <code>- 0.00453</code> ). <b>Разделителем</b> целой и дробной части <b>служит точка</b> , а не запятая.	<pre>first = 87.2 second = 50.2 third = 50.242 print (first + second + third) # вывод 187.642</pre>
<code>str</code>	<b>Строки</b> ( <code>string</code> ) – набор символов, заключенных в кавычки (например: <code>"ball"</code> , <code>"What is your name?"</code> , <code>'dkfjUUV'</code> , <code>'6589'</code> , <code>'Привет!'</code> , <code>"Программирование на языке Python"</code> ).	<pre>first = '87.2' second = '50.2' third = '50.242' print (first + second + third) # вывод 87.250.250.242, оператор + объединяет строки</pre> <p>Числа приводятся к строкам функцией <code>str()</code>.</p> <pre>number = 100 # объявляем переменную целого типа rubles = 'рублей' # объявляем переменную</pre>

		<pre># строкового типа # соединить целое число и строку нельзя, # приводим number к строке print (str (number) + ' ' + rubles) # вывод сообщения # «100 рублей» с пробелом</pre>
bool	<p><b>Логический тип</b> данных (<b>boolean</b>), который принимает значение <b>True</b> (1, истина) или <b>False</b> (0, лож). <b>True</b> и <b>False</b> пишутся с большой буквы.</p>	<pre>print (5 + 5 == 20/2 or 1 + 2 == 2) # вывод True</pre>

Функция `type()` используется для получения типа объекта.

### Операции с числовым типом данных

- арифметические операции;
- операции сравнения;
- логические операции.

### Пример:

#### Программный код:

```
x = 56.7
print(type(x))
```

#### Результат вывода:

```
class 'float'
```

**Ввод и вывод данных** осуществляется с помощью встроенных функций

Ввод: `input` (параметры)

Вывод: `print` (параметры)

**Примечание:** функция `input()` – это функция для ввода данных с клавиатуры. Результат работы функции `input()` всегда строка.

Функция `print()` в Python используется для вывода текстовой информации на экран или в консоль.

## Ход выполнения практической работы

Оформите отчёт по практической работе в текстовом процессоре.

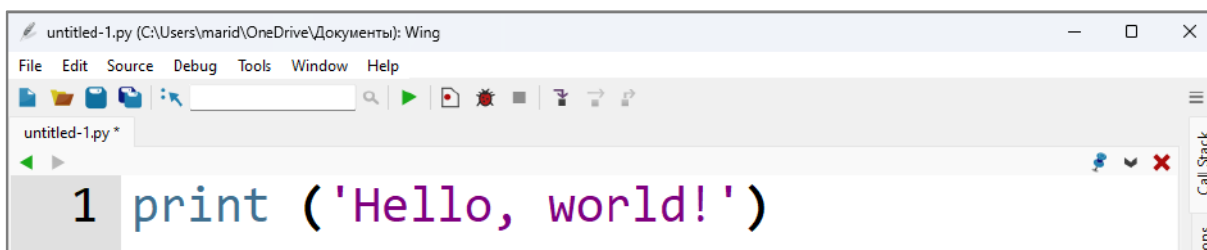
### Для выполнения отчёта выполните следующие действия:

- в папке своего преподавателя найдите шаблон, откройте его и заполните решение задач в виде программного кода;
- сохраните отчёт в свою рабочую папку (в папку Python) с именем **ПР27\_Фамилия\_Группа.odt** (для это выберите вкладку **Файл** → **Сохранить как**, выберите **Рабочий стол** и папку **Моя рабочая папка**); выполняйте текущее сохранение при заполнении отчёта, не реже 1 раза в 5 минут;
- сделайте вывод по практической работе в конце отчёта.



**Задача 1.** Первая программа на Python. Создайте первую программу. Для этого выполните следующие действия:


1. В своей рабочей папке создайте новую папку **Python**.
2. На рабочем столе найдите папку Программы, в ней откройте среду разработки **Wing IDE 101**.
3. В среде Python откройте новый файл для создания программы командой Ctrl + N.
4. В редакторе системы Python наберите программу, представленную на рис. 2.



```
untitled-1.py (C:\Users\marid\OneDrive\Документы): Wing
File Edit Source Debug Tools Window Help
untitled-1.py *
1 print ('Hello, world!')
```

**Рис. 2.** Первая программа на Python

5. Сохраните программу под именем **Код1.py** в папку **Python** на своём компьютере.

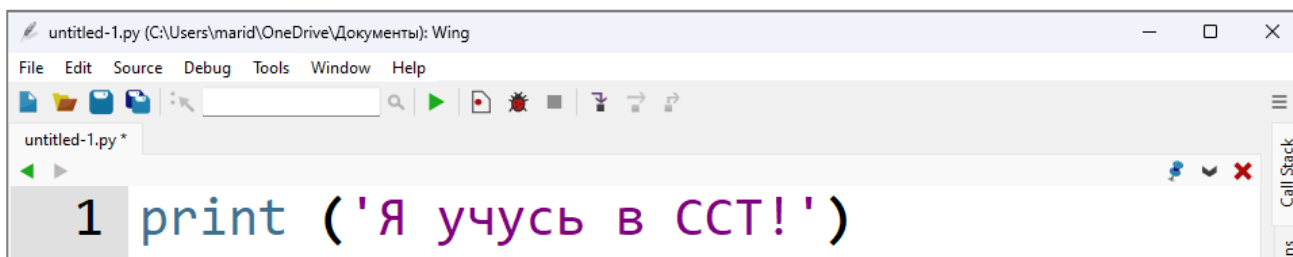
6. Запустите программу (нажатием в интерфейсе среды на Run ). Данная программа выводит на экран текст с помощью функции `print()`. В круглых скобках передаются аргументы функции – информация, которую мы хотим передать функции. Текстовая строка, которую мы хотим напечатать, заключается в двойные или одинарные кавычки.

**Примечание:** функцией `print()` в качестве аргумента можно передавать не только строки, но и числа, и арифметические выражения.

**Пример:**

```
print (123 + 456) # вывод 579
print (2 + 2 * 2 * 6) # вывод 26
```

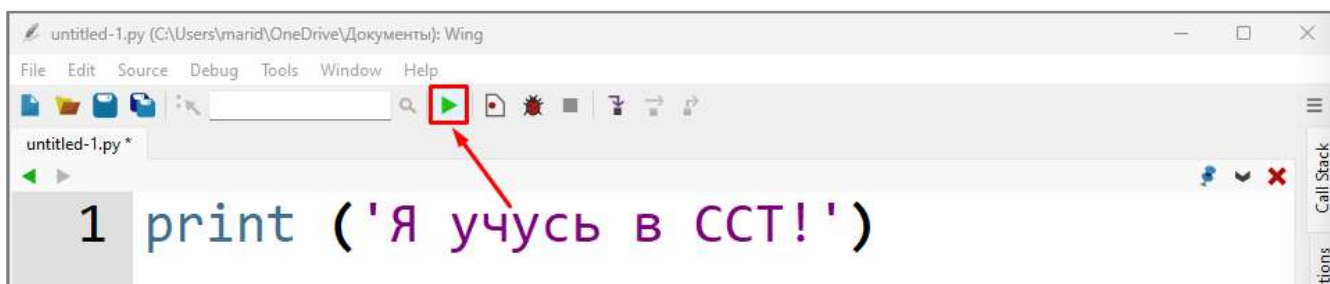
7. Напишите программу, которая выводит на экран текст, представленный на рис. 3.



```
untitled-1.py (C:\Users\marid\OneDrive\Документы): Wing
File Edit Source Debug Tools Window Help
untitled-1.py *
1 print ('Я учусь в ССТ!')
```

**Рис. 3.** Вторая программа на Python

8. Сохраните программу в свой отчёт.
9. Запусти программу (рис. 4).



**Рис. 4.** Запуск программы

10. Обратите внимание на регистр и знаки препинания.

**Таблица 2** – Основные операций для чисел в языке Python

Действия	Математическая запись	Запись на Python
$a + b$ – сумма	$5 + 2 = 7$	<code>print (5+2) # вывод 7</code>
$a - b$ – разность	$5 - 2 = 3$	<code>print (5-2) # вывод 3</code>
$a * b$ – произведение	$5 * 2 = 10$	<code>print (5*2) # вывод 10</code>
$a / b$ – частное от деления	$10 : 2 = 5$	<code>print (10/2) # вывод 5.0</code>
$a ** b$ – возведение в степень	$2 * 2 * 2 = 2^3$	<code>print (2**3) # вывод 8</code>
$a // b$ – нахождение целого от деления	$8 / 3 = 2$ (остаток 2)	<code>print (8//2) # вывод 4</code>
$a \% b$ – остаток от деления	$8 / 5 = 3$ (целая часть 1)	<code>print (8%2) # вывод 0</code>
$abs(a)$ – модуль числа	$ -4  = 4$ или $ 4  = 4$	<code>print (abs(-4)) # вывод 4</code>

Есть также унарные операции – с одним аргументом. Они возвращают число, противоположное данному.

**Пример:** `print (-(2 + 7)) # вывод -9`

**Приоритет операций:**

- 1) возведение в степень;
- 2) смена знака, модуль числа;
- 3) умножение, деление, остаток, целое;
- 4) сложение, вычитание.

**Примечание:** для изменения порядка действий необходимо использовать круглые скобки.

**Пример:** `print ((2 + 3) * 5) # вывод 25`

**Пример:** программа, которая выводит на экран значение выражения:

$$x^{12} + y$$

Переменная  $x$  равна 2, а переменная  $y$  равна 3 (рис. 5).

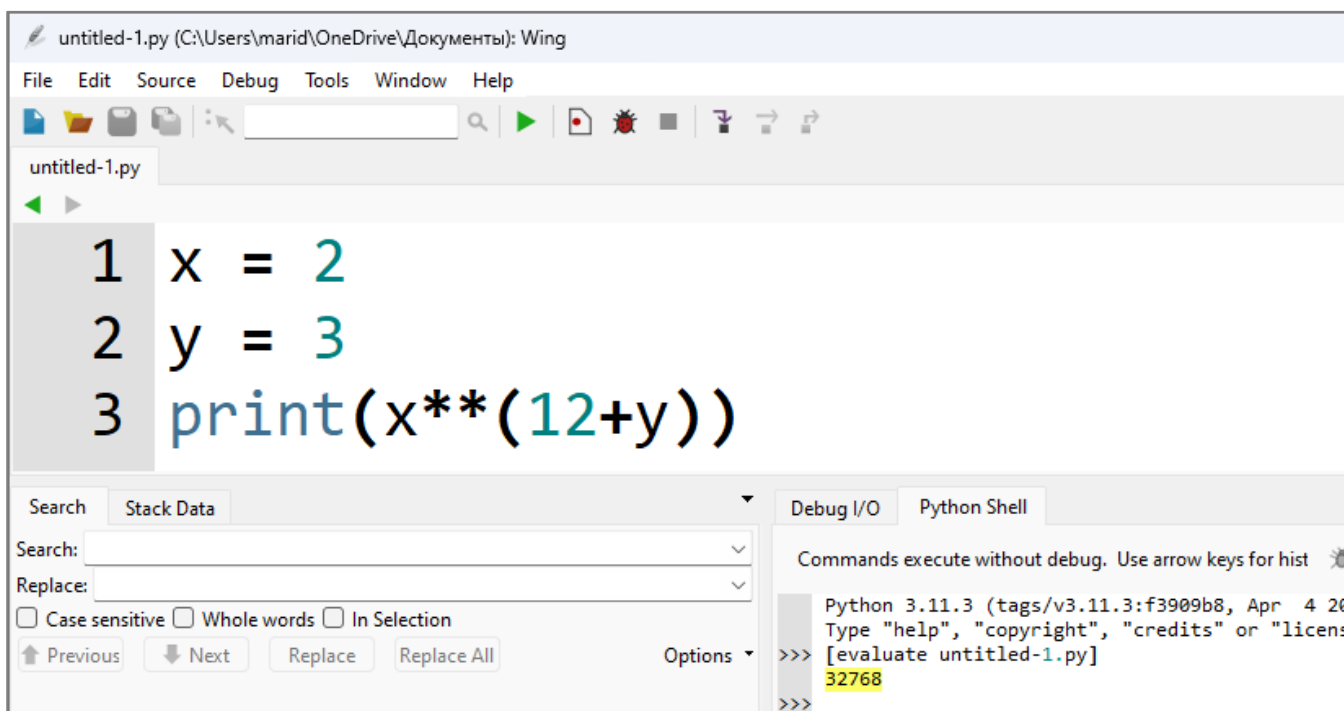


Рис. 5. Вывод значения выражения

**Пример:** Пользователь вводит длину, ширину, высоту прямоугольного параллелепипеда. Программа выводит на экран его объем (рис. 6).

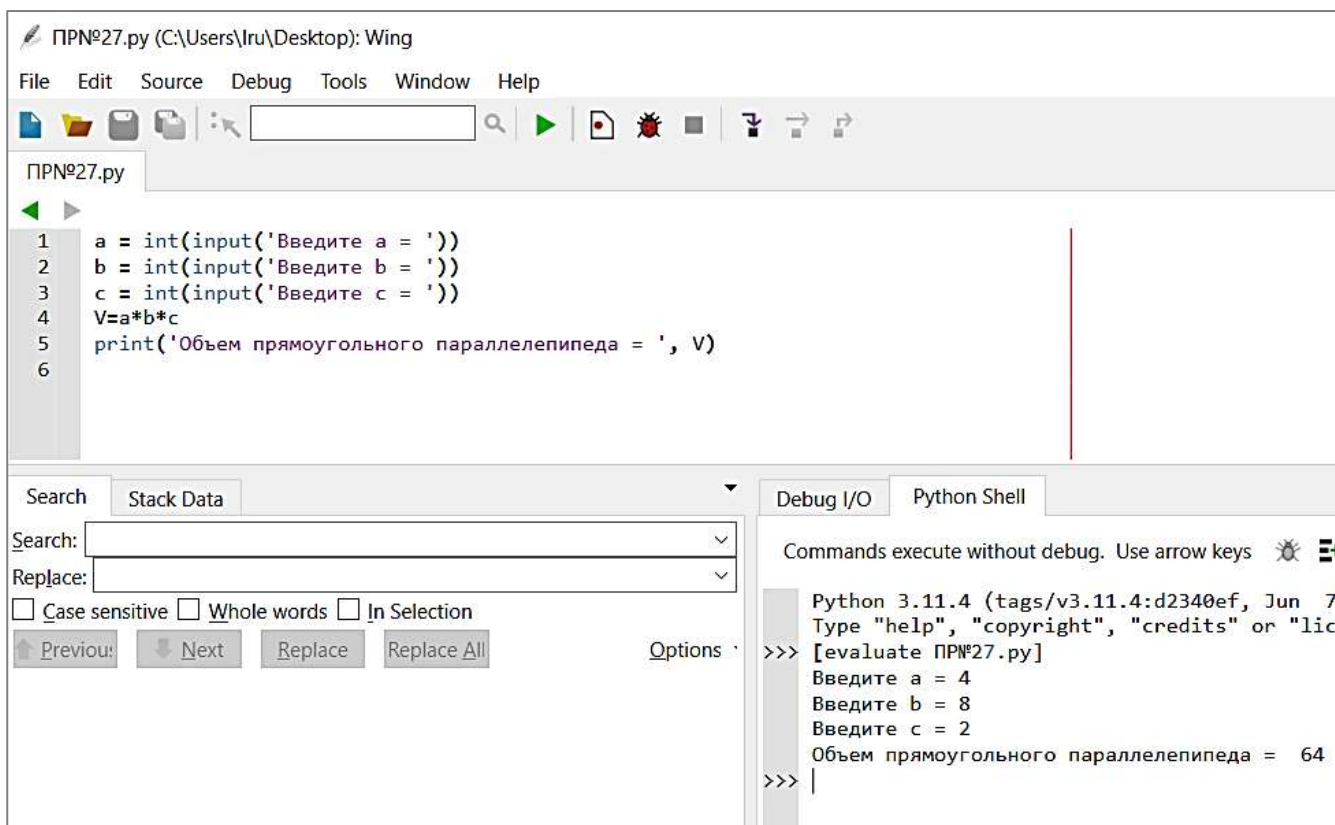



Рис. 6. Решение задания нахождения объёма

 **Задача 2.** Напиши программу, которая вычислит выражение:


$$4 \cdot x^2 + 8 \cdot y^{5+1}$$

Переменная  $x$  равна 3, а переменная  $y$  равна 11. В ответе должен быть только результат вычислений. Используй скобки, чтобы порядок действий был верный. Сохраните программу в свой отчёт.

**Входные данные**

**Выходные данные**

14172524

 **Задача 3.** Напишите программу, которая выводит на экран значение  $20!$  ( $20$  факториал). Сохраните программу в свой отчёт.

Факториал натурального числа  $n$  определяется как произведение всех натуральных чисел от 1 до  $n$  включительно и обозначается  $n!$


$$n! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot n$$

Например,  $3! = 1 \cdot 2 \cdot 3 = 6$ .

**Входные данные**

**Выходные данные**

2432902008176640000

 **Задача 4.** Пользователь вводит любое целое двузначное число. Программа умножает его на 8, а потом находит целую часть от деления на 2 и прибавляет к ней 8. В результате программа выводит полученное значение. Сохраните программу в свой отчёт.

**Пример работы программы:**

Введите целое двузначное число: 27


116

**Входные данные**

**Выходные данные**

27

116

 **Задача 5.** Исправь ошибки в программе и допиши ее. Программа запрашивает сумму счета и прибавляет к нему чаевые 200 рублей, а потом делает посетителю скидку 5%. В результате на экран выводится сумма счета со скидкой. Сохраните программу в свой отчёт.

**Подсказка:** сумма счёта может быть вещественным числом.

```
untitled-1.py *
1  summa = input('Введи, пожалуйста, сумму счёта клиента: ' )
2  summa = summa + 200
```

**Пример работы программы:**

Введи, пожалуйста, сумму счёта клиента: 1985.17  
2075.9115

Входные данные	Выходные данные
1985.17	2075.9115



**Задача 6.** Студент ССТ хочет заказать пиццу на сайте и вводит сумму заказа, а потом количество бонусов для списания «Спасибо» от Сбербанка. Напиши программу, которая вычисляет сумму заказа, после списания бонусов. Сохраните программу в свой отчёт.

**Пример работы программы:**

Введите сумму: 1754  
Введите количество бонусов: 158  
1596

Входные данные	Выходные данные
1754 158	1596



**Задача 7.** Студенты ГБПОУ ССТ на каникулах планируют посетить Исторический парк «Россия – Моя история». С собой ребята хотят взять младших братьев и сестёр, которые учатся в школе. Цена билета для обучающихся равна 250 рублей, а для студентов 528 рублей. Напиши программу, которая запрашивает количество студентов и школьников, а потом считает стоимость экскурсии. Сохраните программу в свой отчёт.

**Пример работы программы:**

Введите количество студентов: 30  
Введите количество школьников: 15  
Итого: 19590

Входные данные	Выходные данные
30 15	Итого: 19590



**Задача 8.** Напиши программу, которая рассчитывает стоимость товара по акции в строительном магазине «ДомСтрой». Программа должна запрашивать у пользователя первоначальную цену товара на строительные материалы, размер

скидки (размер скидки указывается в %) и выводить цену товара с учетом скидки (вводятся и выводятся целые числа). Сохраните программу в свой отчет.

**Пример работы программы:**

Введите цену: 25000  
Процент скидки: 5  
23750

Входные данные	Выходные данные
25000 5	Процент скидки: 5 23750



**Задача 9.** Студент ССТ хочет проверить свои знания в программировании на Python решив задачу для своего младшего брата по нахождению объема и площади комнаты. Условия для задачи: пользователь вводит длину, ширину и высоту комнаты. Программа вычисляет площадь и объем комнаты и выводит на экран. Сохраните программу в свой отчет.

**Пример работы программы:**

Длина: 5  
Ширина: 12  
Высота: 4.9  
Площадь комнаты: 60.0  
Объем комнаты: 294.0

Входные данные:	Выходные данные:
5 12 4.9	Площадь комнаты: 60.0 Объем комнаты: 294.0



**Задача 10.** Напишите программу на Python. Дано натуральное число  $X$ . Вычислить  $y = 1 - 2x + 3x^2 - 4x^3$ . Разрешается использовать не более 8 арифметических операций. Допустимы: операции сложение, вычитание, умножение. **Протестировать** программу для  $x = 0$ ,  $x = 1$ ,  $x = -2$ . Сохраните программу в свой отчет.

**Пример работы программы:**

Введите  $x$ : 0  
1

Входные данные:	Выходные данные:
0 1 -2	1 -2 49

## Практическая работа №28. Программирование разветвляющихся, циклических и комбинированных алгоритмов

**Цель работы:** научиться правильно использовать условный оператор **if**, научиться составлять программы решения задач на разветвляющиеся алгоритмы; познакомиться с циклом **while** и научиться его правильно использовать.

### Теоретические сведения

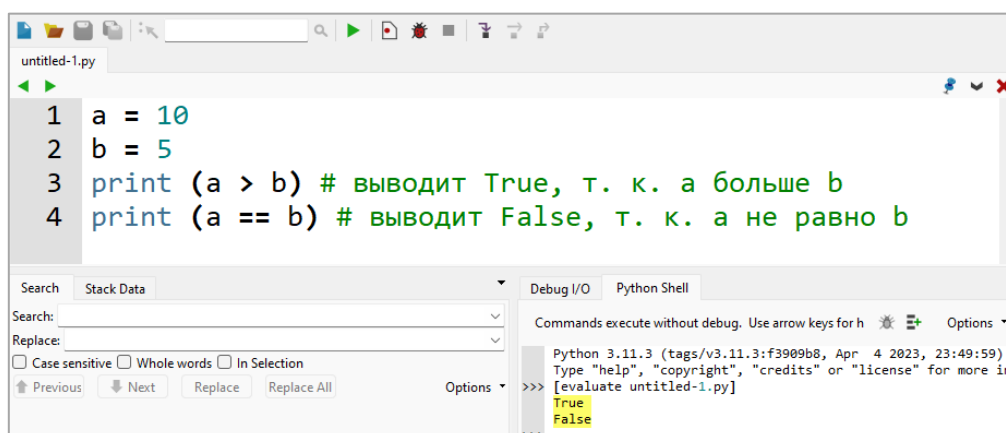
**Логические выражения** могут принимать значения **True** (истина) и **False** (ложь).

В Python, операции сравнения используются для сравнения значений возвращая логическое значение **True** или **False**. Например, оператор сравнения **==** (двойной знак равенства) проверяет, равны ли два значения, и возвращает **True**, если они равны, и **False** в противном случае.

Таблица 1 – Операторы сравнения

Обозначение	Наименование
==	равно
<	меньше
>	больше
>=	больше или равно
<=	меньше или равно
!=	не равно

На рис. 7 представлен код, который сравнивает два числа и выводит результат.



```
untitled-1.py
1 a = 10
2 b = 5
3 print (a > b) # выводит True, т. к. a больше b
4 print (a == b) # выводит False, т. к. a не равно b

Search Stack Data
Search:
Replace:
 Case sensitive  Whole words  In Selection
Previous Next Replace Replace All Options

Debug I/O Python Shell
Commands execute without debug. Use arrow keys for h
Options
Python 3.11.3 (tags/v3.11.3:f3909b8, Apr 4 2023, 23:49:59)
Type "help", "copyright", "credits" or "license" for more in
[evaluate untitled-1.py]
>>> True
>>> False
>>>
```

Рис. 7. Сравнение чисел

**Условный алгоритм** – алгоритм, который выполняет или не выполняет действие в зависимости от значения логического выражения. **Условный оператор** `if` (в англ. – «если») `else` (в англ. – «иначе»).

### Синтаксис условного оператора:

```

if <условие>:                # если выполнено условие, то:
    действие 1                # действие 1
else:                          # иначе:
    действие 2                # действие 2

```

Первое действие будет выполнен, если условие истинно. Если условие ложно, будет выполнено второе действие.

Для выделения блока инструкций, относящихся к инструкции `if` или `else`, в языке Python используются отступы. Все инструкции, которые относятся к одному блоку, должны иметь равную величину отступа, то есть одинаковое число пробелов в начале строки. Рекомендуется использовать отступ в 4 пробела и **не рекомендуется** использовать в качестве отступа символ табуляции.

Это одно из существенных отличий синтаксиса Python от синтаксиса большинства языков, в которых блоки выделяются специальными словами или фигурными скобками.

**Таблица 2 – Формы ветвления**

Полная	Неполная
<b>Синтаксис</b>	
<pre> if выражение истинно:     действие 1     ... else:     действие 2     ... </pre>	<pre> if выражение истинно:     действие 1     действие 2     ... </pre>
<b>Запись на Python</b>	
<pre> x = int (input ('Введите x: ')) if x &gt; 0:     print (x) else:     print (-x) </pre>	<pre> x = int (input ('Введите x ')) if x &lt; 0:     x = -x print (x) </pre>
<b>Описание</b>	
<pre> x = int(input('Введите x: ')) # Запрос у пользователя значения переменной x и преобразование в целое число if x &gt; 0: # Проверка условия: если x больше нуля. После условия ставится двоеточие </pre>	<pre> x = int(input('Введите x: ')) # Запрос у пользователя значения переменной x и преобразование в целое число if x &lt; 0: # Проверка условия: если x меньше нуля. После условия ставится двоеточие </pre>

<pre>print(x) # Вывод значения x, если условие истинно. Перед print выставляется 4 пробела  else: # Если условие не выполнено (x меньше или равно нулю). После else ставится двоеточие  print(-x) # Вывод значения если условие ложно (не выполняется). Перед print выставляется 4 пробела</pre>	<pre>x = -x # Если условие выполнено, меняем знак x на противоположный. Перед x выставляется 4 пробела  print (x) # Вывод значения x на экран</pre>
--	---

**Пример 1:** Даны два числа, необходимо найти наибольшее из них и напечатать его на экран. На рисунке ниже представлено решение.

```

1 a = int (input ('Введите число a: '))
2 b = int (input ('Введите число b: '))
3 if a > b:
4     m = b
5 else:
6     m = b
7 print ('Наибольшее число:',m)

```

Python 3.11.3 (tags/v3.11.3:f3909b8, Apr 4 2023, 23:49:59)  
Type "help", "copyright", "credits" or "license" for more info  
[evaluate untitled-1.py]  
>>> Введите число a: 5  
Введите число b: 6  
Наибольшее число: 6  
>>>

**Рис. 8.** Нахождение максимального числа (полное ветвление)

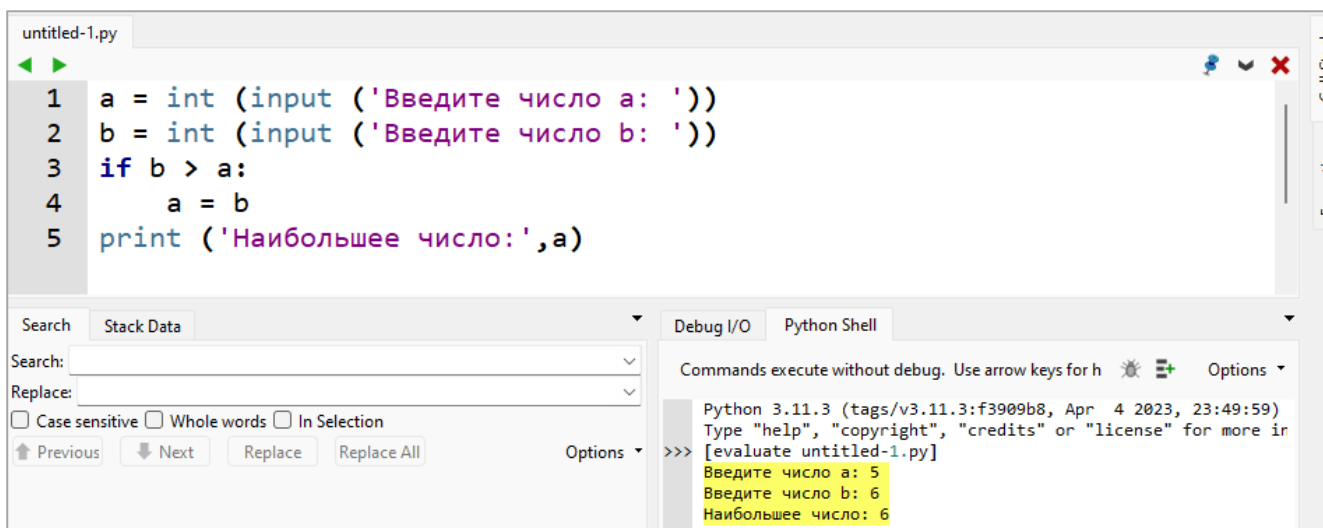
**Код для копирования:**

```

a = int (input ('Введите число a: '))
b = int (input ('Введите число b: '))
if a > b:
    m = b
else:
    m = b
print ('Наибольшее число:',m)

```

Также эту задачу можно решить с помощью неполного ветвления. В этом случае нам не нужно использовать дополнительную переменную. На рис. 9 представлено решение.



**Рис. 9.** Нахождение максимального числа (неполное ветвление)

**Код для копирования:**

```

a = int (input ('Введите число a: '))
b = int (input ('Введите число b: '))
if b > a:
    a = b
print ('Наибольшее число:', a)

```

**Таблица 3** – Логические операторы

Обозначение	Запись на Python
<b>or</b> (или) – логическое сложение	if beaufort == 7 <b>or</b> beaufort == 8: print('крепкий ветер')
<b>and</b> (и) – логическое умножение	if m >= 1 <b>and</b> m <= 12: print('корректно')
<b>not</b> (не) – отрицание	x = int(input('Введите x ')) if <b>not</b> x > 15: print (x) else: print (x+1)

## Пример 2: Условный оператор `if` для записи ветвления «если – то»:

# в переменной `beaufort` хранится скорость ветра по шкале Бофорта

Неполная форма ветвления	Конструкция <code>if-else</code> для записи ветвления «если – то – иначе»:	Множественное ветвление:
<pre>if beaufort == 0:     print ('штиль')</pre>	<pre>if beaufort == 0:     print ('штиль') else:     print ('есть ветер')</pre>	<pre>if beaufort == 0:     print ('штиль') elif beaufort == 1:     print ('тихий ветер') elif beaufort == 2:     print ('лёгкий ветер') elif beaufort == 3:     print ('слабый ветер') elif beaufort == 4:     print ('умеренный ветер') elif beaufort == 5:     print ('свежий ветер') elif beaufort == 6:     print ('сильный ветер')</pre>

**Примечание:** как только выполняется одно из условий – все нижеследующие `elif` и `else` пропускаются.

**Цикл** – это алгоритм, который выполняет указанные действие пока истинно условие.

Цикл `while` (в переводе с англ. «пока»):

- может использоваться, когда неизвестно количество повторений;
- выполняет действия, пока истинно условие;
- может быть, не выполнен ни разу, если условие ложно.

**Синтаксис** цикла `while`:

```
действие 1
действие 2
действие 3
```

**Примечание:** пока условие будет истинным, будут выполняться первые два действия.

**Счётчик** – переменная, которая считает количество повторений цикла,  
`i = 0`

`while` выражение истинно:

```
действие 1
действие 2
i = i + 1
```

действие 3

**Пример 3:** программа напечатает на экран все целые числа, не превосходящие  $n$ , и их сумму:

```
s = 0
i = 1
while i <= n:
    print (i)
    s = s + i
    i = i + 1
```

**Пример 4:** напиши программу, которая найдет максимальную степень двойки, не превосходящую 100.

```
p = 1
while p * 2 <= 100:
    p = p * 2
print (p)
```

После окончания цикла значение переменной  $p$  будет максимальной степенью двойки, не превосходящей 100, потому что на следующем шаге условие уже не будет выполняться.

**Пример 5:** напиши программу, используя цикл `while`, для определения количества цифр натурального числа  $n$  и их суммы:

```
s = 0
n = int(input())
count = 0
while n > 0:
    count = count + 1
    s = s + n % 10
    n = n // 10
print (count)
print (s)
```

В этом цикле мы отбрасываем по одной цифре числа, начиная с конца, что эквивалентно целочисленному делению на 10 (`n // 10`), при этом считаем в переменной `count`, сколько раз это было сделано, а в переменной `s` – сумму отброшенных цифр.

Оформите отчёт по практической работе в текстовом процессоре.

Для выполнения отчёта выполните следующие действия:

- в папке своего преподавателя найдите шаблон, откройте его и заполните решение задач в виде программного кода;
- сохраните отчёт в свою рабочую папку (в папку Python) с именем **ПР28\_Фамилия\_Группа.odt** (для это выберите вкладку **Файл** → **Сохранить как**, выберите **Рабочий стол** и папку **Моя рабочая папка**); выполняйте текущее сохранение при заполнении отчёта, не реже 1 раза в 5 минут;
- сделайте вывод по практической работе в конце отчёта.



**Задача 1.** Скопируй код и исправь все ошибки в синтаксисе кода.

Код для копирования:

```
n = int (input ('Число:'))
if n >= 0
    print ('Положительное число')
Else:
    print ('Отрицательное число')
```

**Входные данные**

9

**Выходные данные**

Положительное число



**Задача 2.** В 2024 году ГБПОУ ССТ проводит товарищеские матчи по баскетболу. Напиши программу, которая запрашивает у пользователя количество очков, набранных командой (натуральное число), если количество очков больше или равно 10, то программа выводит на экран «Ты победил!» иначе «Ты проиграл!».

**Пример работы программы:**

Введите количество очков: 10

Ты победил!

**Входные данные**

10

**Выходные данные**

Ты победил!



**Задача 3.** Шахматная ладья ходит по горизонтали или вертикали. Даны две различные клетки шахматной доски, определите, может ли ладья попасть с первой клетки на вторую одним ходом.

у	A	B	C	D	E	F	G	H
8								
7								
6								
5								
4								
3								
2								
1								
х	1	2	3	4	5	6	7	8

**Пример работы программы:**

Координата  $x_1$  первой ладьи: 7  
Координата  $y_1$  первой ладьи: 2  
Координата  $x_2$  второй ладьи: 3  
Координата  $y_2$  второй ладьи: 2  
YES

<b>Входные данные</b>	<b>Выходные данные</b>
Программа получает на вход четыре числа от 1 до 8 каждое, задающие номер столбца и номер строки сначала для первой клетки, потом для второй клетки	Программа должна вывести YES, если из первой клетки ходом ладьи можно попасть во вторую, или NO в противном случае
7 2 3 2	YES



**Задача 4.** В строительном магазине «Держава» действует скидка 5% при покупке на сумму от 25 000 рублей. Напиши программу, которая будет запрашивать сумму покупок и вычислять итоговую сумму с учетом скидки.

**Пример работы программы:**

Введите сумму покупки: 25000  
23750.0

Входные данные	Выходные данные
25000	23750.0
24000	24000



**Задача 5.** Напиши программу, которая запрашивает у пользователя его год рождения и текущий год. Затем программа должна проверить, является ли пользователь совершеннолетним. Если да, то программа выводит сообщение «Доступ разрешён», иначе «Доступ ограничен».

**Пример работы программы:**

Введите год своего рождения: 2004  
Введите текущий год: 2024  
Доступ разрешён

Входные данные	Выходные данные
2004 2024	Доступ разрешён



**Задача 6.** Пользователь вводит двузначное число. Напиши программу, которая вычисляет сумму цифр этого числа. Если сумма его цифр четная и больше 10, то программа умножает число на 2, в противном случае уменьшает число на 2. В результате выводится полученное число.

**Пример работы программы:**

Введите двузначное число: 12  
10

Входные данные	Выходные данные
12	10
48	96



**Задача 7.** Напиши программу, которая запрашивает возраст пользователя и определяет стоимость билета в кино в зависимости от возраста. Билет в кино стоит 450 рублей. Если пользователю меньше 18 лет, то билет идет со скидкой 5%, иначе выводится полная стоимость билета без скидок.

**Пример работы программы:**

Введите возраст: 14

Ваша скидка 5%

427,5

Входные данные	Выходные данные
14	Ваша скидка 5% 427,5
18	450

**Цикл while**

**Задача 8.** Напишите программу, которая при вводе целого числа будет выводить количество цифр.

**Пример работы программы:**

Введите целое число: 123

Всего: 3

Входные данные	Выходные данные
123	3
1968	4
5847126	7



**Задача 9.** Напишите программу, которая при вводе целого числа будет выводить количество чётных цифр и сумму всех цифр.


**Пример работы программы:**

Введите целое число: 123

Всего чётных цифр: 1

Сумма цифр: 6

Входные данные	Выходные данные
123	1 6
258	2 15
1458756	3 36

 **Задача 10.** Напишите программу, которая при вводе целого числа будет выводить произведение всех цифр.

**Пример работы программы:**

Введите целое число: 123

Всего чётных цифр: 1

Сумма цифр: 6

Входные данные	Выходные данные
123	1 6
258	2 15
1458756	3 36

 **Задача 11.** Напишите программу, которая выводит числа от 1 до 10 с помощью while.

**Пример работы программы:**

1

2

3

4

5

6


7

8

9

10

Входные данные	Выходные данные
	1
	2
	3
	4
	5
	6
	7
	8
	9
	10

 **Задача 12.** Напишите программу, которая запрашивает числа у пользователя, пока он не введёт 0. После этого программа выводит сумму всех

введённых чисел (без 0). Используйте **while**, чтобы запрашивать числа, и складывайте их в переменную **total**.

**Пример работы программы:**

Введите число: 4

Введите число: 7

Введите число: -2

Введите число: 0

Сумма чисел: 9

Входные данные	Выходные данные
4	9
7	
-2	
0	

## Практическая работа №29-30. Работа со строками и циклами в Python

**Цель работы:** научиться работать с методами строк `lower()`, `replace()`, `count()`, циклом `for`, функцией `range`, а также научиться извлекать данные из строк и производить вычисления.

### Ход выполнения практической работы

Оформите отчёт по практической работе в текстовом процессоре.

**Для выполнения отчёта выполните следующие действия:**

– в папке своего преподавателя найдите шаблон, откройте его и заполните решение задач в виде программного кода;

– сохраните отчёт в свою рабочую папку (в папку Python) с именем **ПР29-30\_Фамилия\_Группа.odt** (для это выберите вкладку **Файл** → **Сохранить как**, выберите **Рабочий стол** и папку **Моя рабочая папка**); выполняйте текущее сохранение при заполнении отчёта, не реже 1 раза в 5 минут;

– сделайте вывод по практической работе в конце отчёта.

**Строка** – объект, состоящий из нескольких символов.

Нумерация символов в строке начинается с 0:

П	р	и	в	е	т	,		С	С	Т	!
0	1	2	3	4	5	6	7	8	9	10	11

**Подстрока** – некоторая часть строки.

Метод `replace` находит и заменяет подстроку. Синтаксис:

```
stroka = "..."  
a = stroka.replace (old, new)
```

`old` – старая подстрока, которая будет заменена, `new` – новая подстрока, которая заменит старую.


Пример использования метода `replace`:

	Программа	Результат работы
1.	<pre>stroka = "Это интересно! Это классно! Это необычно!"</pre>	<pre>&lt;&lt; интересно!  классно! Это необычно!</pre>
2.	<pre>print (stroka.replace("Это","", 2))</pre>	

Необязательный параметр указывает на количество замен подстроки.

Функция `len()` – возвращает длину строки. Пример:

Программа	Результат работы
<pre>1. a = 'Привет' 2. b = len(a) 3. print(b)</pre>	6

 **Задача 1.** Напиши программу, которая будет заменять слово «это» в строке на «–». Используй метод `replace`.

Входные данные	Выходные данные
Ставропольский строительный техникум (ССТ) это одно из самых престижных средних образовательных учреждений Юга России	Ставропольский строительный техникум (ССТ) – одно из самых престижных средних образовательных учреждений Юга России

Метод `count` – считает сколько раз встречается подстрока. Синтаксис:

```
stroka = "..."  
str_cnt = stroka.count ("подстрока", start_index, end_index)
```

Необязательные параметры: `start_index` и `end_index`


Пример использования метода `count`:

Программа	Результат работы
<pre>1. comment = 'Ничего не понравилось' 2. stroka = input ('Введи подстроку ' ) 3. print (comment.count(stroka, 5))</pre>	<< Введи подстроку о 3


Методы изменения регистра:

`lower()` – преобразование в нижний регистр (Привет – привет).

`upper()` – преобразование в верхний регистр (Привет – ПРИВЕТ).

 **Задача 2.** Напиши программу, которая запрашивает строку и заменяет 2 первые подстроки «то» на «иногда». А также вычисляет сколько раз встречается подстрока «то» в строке до замены. Используй методы `lower`, `replace`, `count`.

Входные данные	Выходные данные
То опаздываю на пару, то просыпаю, то не успеваю, то не учу.	иногда опаздываю на пару, иногда просыпаю, то не успеваю, то не учу. 4


 **Задача 3.** Напиши программу, которая выводит символы с 7 индекса до конца строки без шага.

**Входные данные**

Я люблю ГБПОУ ССТ

**Выходные данные**


ГБПОУ ССТ

 **Задача 4.** Дана строка, в которой есть слово «или». Определить, сколько раз оно встречается. `stroka = 'Или опаздывает на пару, или спит на паре, или отвлекает одногруппников'`. Используй методы `lower`, `count`.

**Входные данные**

**Выходные данные**

3


 **Задача 5.** Строительная компания использует каталог материалов. В одном из названий ошибка: вместо «цемент» написано «семент». Напишите программу, которая исправляет ошибку в названии.

**Входные данные**

Портландсемент высокого качества

**Выходные данные**

Портландцемент высокого качества


 **Задача 6.** Строитель читает техническое задание, в котором говорится о количестве кирпичей. Нужно посчитать, сколько раз встречается слово «кирпич» в описании.

**Входные данные**

Для строительства используется красный кирпич. Кирпич должен быть высокого качества. Кирпичи укладываются в шахматном порядке.

**Выходные данные**

Слово 'кирпич' встречается: 3


 **Задача 7.** Инженер записал текст разными регистрами. Нужно привести весь текст к заглавным буквам для оформления отчёта.

**Входные данные**

выполнена кладка стен из кирпича.  
Залит фундамент. Работы завершены.

**Выходные данные**

Отформатированный текст: ВЫПОЛНЕНА  
КЛАДКА СТЕН ИЗ КИРПИЧА. ЗАЛИТ  
ФУНДАМЕНТ. РАБОТЫ ЗАВЕРШЕНЫ.

 **Задача 8.** При проектировании здания используется разный бетон. Инженеру нужно узнать, сколько раз упоминается слово «бетон», не зависимо от регистра.

### Входные данные

Бетон марки М300 используется для фундамента. Также возможен бетон М200 для внутренних перегородок.

### Выходные данные

Слово «бетон» встречается 2 раза.

**Срез строки** – механизм извлечения одного символа строки или подстроки.

Пример строки с датой рождения:

a =	'	0	1	.	0	2	.	2	0	2	3	'
		0	1	2	3	4	5	6	7	8	9	

Пользователи вводят дату рождения, нам нужно извлечь 6-й и 9-й символы:

### Программа

1. `a = '01.02.2023'`
2. `year1 = a[6]`
3. `year2 = a[9]`

### Результат работы

2  
3

Если нам нужно извлечь только год рождения, то воспользуемся диапазоном индексов, которые отделяются двоеточием:

### Программа

1. `a = '01.02.2023'`
2. `year1 = a[6:10]`
3. `print(year1)`

### Результат работы

2023

Срез с двумя параметрами – `s[a:b]` возвращает подстроку, начиная с символа с индексом `a` до символа с индексом `b`, **не включая его**. Если опустить второй параметр (но поставить двоеточие), то срез берётся до конца строки, например: `year1 = a[6:]` выведет подстроку 2023.



**Задача 9.** В электронный журнал студенту выставили оценки «Журнал 1 С колледж: 4, 5, 3». Напиши программу, которая извлечет из строки только оценки и определит средний балл среди них. Используй `int()`.

### Входные данные

Журнал 1 С колледж: 4, 5, 3

### Выходные данные

4 5 3  
4.0



**Задача 10.** Код здания состоит из типа (2 буквы) + года постройки. Например, "ПЗ1987" (Панельный, жилой, 1987). Нужно извлечь тип и год.

### Входные данные

ПЗ2025

### Выходные данные

Тип: ПЗ

Год постройки: 2025



**Задача 11.** В техническом паспорте указано "Этаж 5, подъезд 3". Нужно извлечь номер этажа.

### Входные данные

Этаж 5, подъезд 3

### Выходные данные

5



**Задача 12.** В номере проекта "А-203/15-450" содержится информация: "А" – тип проекта, "203" – номер проекта, "15" – район строительства, "450" – площадь здания в м<sup>2</sup>. Нужно извлечь все параметры и вывести их отдельно.

### Входные данные

А-203/15-450

### Выходные данные

Тип проекта: А

Номер проекта: 203

Район строительства: 15

Площадь: 450 м<sup>2</sup>

**Цикл for** – перебирает элементы в последовательности. Синтаксис цикла:

```
for элемент in последовательность:  
    действие 1  
    действие 2  
    ...
```

**Элемент** – переменная, которая изменяется в цикле. **Последовательность** – значения, для которых выполняется повторение.

**Функция range** – создает последовательность чисел в диапазоне от 0 до n – 1. Синтаксис:

```
for элемент in range (n):  
    действие 1  
    действие 2  
    ...
```

### Цикл for

- Количество повторений известно заранее
- Используется для перебора элементов

```
for i in range (4,10):  
    print(i)
```

**Вывод на экран**

4

### Цикл while

- Количество повторений неизвестно заранее
- Выполняется пока верно условие

```
i = 4  
while i != 10:  
    print(i)  
    i = i + 1
```

5  
6  
7  
8  
9

### Вывод на экран

4  
5  
6  
7  
8  
9

Способы вызова `range()`:

#### `range(stop)`

Берёт один аргумент.  
Возвращает все  
целые числа от 0 до  
`stop`.

#### `range(start, stop)`

Берёт два аргумента.  
Возвращает все целые числа в  
промежутке от `start`  
(включая) до `stop` (не  
включая).

#### `range(start, stop, шаг)`

Берёт три аргумента. Позволяет выбрать  
не только то, где ряд чисел начнётся и  
остановится, но и то, на сколько велика  
будет разница между одним числом и  
следующим. Если не задавать этот «шаг»,  
то `range()` автоматически будет вести  
себя так, как если бы шаг был бы равен 1.

Пример цикла, который будет выполнен 4 раза:

```
for i in range (4):  
    a = input('Ваше желание? ')  
    print('Желание записано')
```

**Пример использования функции `range`:** в многоэтажном здании от 1 до n пронумерованы этажи. Нужно вывести номера всех этажей.

```
n = int(input("Введите количество этажей: "))  
for floor in range(1, n+1):  
    print("Этаж", floor)
```

#### Входные данные

5

#### Выходные данные

Этаж 1  
Этаж 2  
Этаж 3  
Этаж 4  
Этаж 5

**Рассмотрим пример:** напишите программу, которая запрашивает у пользователя ввод строки и проверяет, содержит ли она запрещённые символы (`/`, `*`, `@`). Если хотя бы один из них встречается в строке, программа сообщает об этом.

```
s = input('Введите строку: ') # Запрашиваем строку у пользователя  
wrong = '/*@' # Запрещённые символы  
for symbol in s: # Перебираем каждый символ в строке  
    if symbol in wrong: # Если символ есть в списке запрещённых  
        print('Содержит', symbol) # Выводим сообщение
```



**Задача 13.** Программа проверяет содержит ли цифры строка, которую

вводит пользователь. Используй цикл `for`, оператор `if`.

**Пример работы программы:**

Введите пароль: ГБПОУ ССТ 2024

Содержит: 2

Содержит: 0

Содержит: 2

Содержит: 4

Входные данные	Выходные данные
ГБПОУ ССТ 2024	Содержит: 2 Содержит: 0 Содержит: 2 Содержит: 4



**Задача 14.** Скопируй код ниже и исправь все ошибки в синтаксисе

кода:

**Код для копирования:**

```
letter = 'ГБПОУ ССТ'  
For in letter  
    print(i)
```

Входные данные

Выходные данные

Г  
Б  
П  
О  
У  
  
С  
С  
Т



**Задача 15.** Напишите программу, которая печатает все нечетные двузначные числа, у которых последняя цифра равна 3 или 7. Используй цикл `for i in range`, оператор `if` и математическое выражение:

```
i % 10 == 3 or i % 10 == 7:
```

Как извлечь последнюю цифру из числа 746? Эту цифру нужно записать в отдельную переменную.

Используем остаток от деления на 10 ( $746 \% 10 == 6$ ):

```
a = int(input())  
a3 = a % 10  
    print(a3)
```

**Входные данные**

**Выходные данные**

13  
17  
23  
27  
33  
...  
87  
93  
97



**Задача 16.** Программа запрашивает количество оценок, а потом запрашивает оценки и считает их средний балл. Используйте цикл `for i in range`. Используйте счётчик `total` для подсчёта суммы всех введённых отметок. Используйте переменную `number` для ввода количества оценок и используйте эту переменную для функции `range`.

**Пример работы программы:**

Количество оценок: 4  
Оценка: 2  
Оценка: 5  
Оценка: 3  
Оценка: 4  
Среднее: 3.5

**Входные данные**

4  
2  
5  
3  
4

**Выходные данные**

Среднее: 3.5



**Задача 17.** На фасаде здания есть `n` окон, и каждое окно подписано. Нужно вывести номер каждого окна.

**Входные данные**

4

**Выходные данные**

Окно 1  
Окно 2  
Окно 3  
Окно 4



**Задача 18.** Давление в газовой трубе постепенно снижается с 100 до 60 с шагом 10. Выведите все значения. Используйте функцию `range` с указанием отрицательного шага.

**Входные данные**

**Выходные данные**

Давление: 100

Давление: 90

Давление: 80

Давление: 70

Давление: 60



**Задача 19.** Выведите номера земельных участков от 101 до 105.

**Входные данные**

**Выходные данные**

Участок 101

Участок 102

Участок 103

Участок 104

Участок 105

## Практическая работа №31. Работа со списками в Python: создание, модификация и основные методы

**Цель работы:** научиться создавать, изменять и обрабатывать списки, используя базовые операции (индексация, срезы, добавление и удаление элементов) и встроенные методы (`len()`, `append()`, `sort()`, `min()`, `max()`, `count()`).

### Теоретические сведения

**Массивы Python.** В Python нет структуры данных, полностью соответствующей массиву. Однако, есть **списки**, которые являются их надмножеством, то есть это те же массивы, но с расширенным функционалом.

**Массив и список** – это упорядоченные коллекции, но разница между ними заключается в том, что классический массив должен содержать элементы только **одного типа**, а **список Python может содержать любые элементы**. Для хранения группы однотипных (но и не только однотипных) объектов используют списки – объекты типа `list`. Отличие списков от массивов заключается в том, что список – это динамическая структура, размер которого можно изменять во время выполнения программы (удалять, добавлять элементы) не задумываясь над операциями по управлению памятью (это делает транслятор).

**Список** – это набор или последовательность элементов, которые имеют свой индекс. Могут хранить **данные любого типа**. Пример списка:

```
numbers = [ 2 , 4 , 5 ]
            0   1   2
            имя списка      индексы элементов
                           списка
```

Содержимое списка пишется в квадратных скобках, элементы списка разделяются запятой:

```
new_list = [<элемент>, <элемент>, <элемент>, <элемент>, <элемент>]
```

У каждого элемента есть свой порядковый номер – индекс. С помощью индекса можно получить значение элемента списка. Индексы начинаются с **0**.

Списки могут хранить данные любого типа.

Пример списка с однотипными элементами:

```
people = ["Лера", "Дима", "Ваня"]
```

Пример списка с элементами разных типов:

```
objects = [132, 5.67, "Привет", True]
```

**Как задаются списки?**

`s = []` – пустой список

`s1 = list()` – создание пустого списка

**Пример 1.** Вам дан список русского алфавита. Напечатайте букву, которая находится на 1-м индексе списка. Напечатайте букву, которая находится на 2-м индексе списка. Подсказка: Индексация в списках начинается с 0, поэтому буква с индексом 1 – это вторая буква алфавита.

#### Код программы

```
russian_alphabet = ['а', 'б', 'в', 'г',  
'д', 'е', 'ё', 'ж', 'з', 'и', 'й', 'к',  
'л', 'м', 'н', 'о', 'п', 'р', 'с', 'т',  
'у', 'ф', 'х', 'ц', 'ч', 'ш', 'щ', 'ъ',  
'ы', 'ь', 'э', 'ю', 'я']  
print(russian_alphabet[1])  
print(russian_alphabet[2])
```

б  
в

#### Результат

**Пример 2.** В сказке о Бременских музыкантах участвуют несколько персонажей. Данный список содержит их имена: `bremen_musicians = ['Трубадур', 'Кот', 'Пёс', 'Осёл', 'Петух']`. Подсчитайте, сколько персонажей в этом списке. Выведите полученное количество на экран. Используйте стандартную функцию `len()`, которая позволяет узнать количество элементов в списке.

#### Код программы

```
bremen_musicians = ['Трубадур', 'Кот',  
'Пёс', 'Осёл', 'Петух']  
count = len(bremen_musicians)  
print(count)
```

5

#### Результат

**Пример 3.** У вас есть список чисел: `num = ['21', '123', '15']`. Проверьте, содержится ли число '21' в списке. Если число есть в списке, выведите: "Список содержит значение 21". Если числа нет в списке, выведите: "Нет такого элемента". Используйте оператор `in`, чтобы проверить, присутствует ли нужное значение в списке.

#### Код программы

```
num = ['21', '123', '15']  
if '21' in num:  
    print('Список содержит значение 12')  
else:  
    print('Нет такого элемента')
```

Результат  
Список содержит значение 21

#### Действия со списками:

- срез списков;
- изменение элементов;
- объединение (конкатенация);
- дублирование;
- разложение;
- сравнение;

## Функции для типа данных «списки» в Python:

`min()` принимает на вход список и возвращает наименьший элемент из этого списка. Элементы списка должны быть сравнимыми между собой.

Код программы	Результат
<pre>my_list = [3, 1, 4, 1, 5, 9, 2, 6, 5, 3, 5] print(min(my_list))</pre>	1
<pre>my_list = ["apple", "banana", "cherry", "durian", "elderberry"] print(min(my_list))</pre>	apple

`max()` используется для поиска максимального элемента в списке. Она принимает один аргумент – список, и возвращает максимальный элемент.

Код программы	Результат
<pre>my_list = [2, 4, 1, 7, 5] max_num = max(my_list) print(max_num)</pre>	7
<pre>list1 = [3, 5, 9] list2 = [2, 4, 8] list3 = [1, 7, 6] max_num = max(list1, list2, list3) print(max_num)</pre>	9

`count()` – метод, который используется для подсчёта количества элементов списка, которые равны заданному значению. Синтаксис метода: `list.count(element)`, где `list` – список, в котором нужно подсчитать количество элементов, `element` – элемент, который нужно найти в списке. Метод возвращает количество элементов списка, которые равны `element`.

Код программы	Результат
<pre>my_list = [1, 2, 3, 1, 4, 1] print(my_list.count(1))</pre>	3
<pre>my_list = [1, 2, 3, 4, 5] print(my_list.count(6))</pre>	0

`append()` – это метод списков, который позволяет добавлять элементы в конец списка. Синтаксис метода: `list_name.append(item)`, где `list_name` – имя списка, а `item` – элемент, который нужно добавить в конец списка. Метод `append()` может быть использован для добавления любых элементов в конец списка, включая другие списки.

Код программы	Результат
<pre>fruits = ["apple", "banana", "cherry"] fruits.append("orange") print(fruits)</pre>	<pre>['apple', 'banana', 'cherry', 'orange']</pre>
<pre>fruits = ["apple", "banana", "cherry"] more_fruits = ["mango", "papaya", "kiwi"] fruits.append(more_fruits) print(fruits)</pre>	<pre>['apple', 'banana', 'cherry', ['mango', 'papaya', 'kiwi']]</pre>

`sort()` сортирует элементы данного списка в определенном порядке возрастания или убывания. Синтаксис метода: `list.sort(key=..., reverse=...)`, где `key` – функция, которая служит ключом для сравнения сортировок, `reverse` – если `True`, отсортированный список переворачивается (или сортируется в порядке убывания). **Сортировать можно только однотипные списки.**


Код программы	Результат
<pre>vowels = ['e', 'a', 'u', 'o', 'i'] vowels.sort() print('Sorted list:', vowels)</pre>	<pre>Sorted list: ['a', 'e', 'i', 'o', 'u']</pre>

## Ход выполнения практической работы

Оформите отчёт по практической работе в текстовом процессоре.

**Для выполнения отчёта выполните следующие действия:**


- в папке своего преподавателя найдите шаблон, откройте его и заполните решение задач в виде программного кода;
- сохраните отчёт в свою рабочую папку (в папку Python) с именем **ПР31\_Фамилия\_Группа.odt** (для это выберите вкладку **Файл** → **Сохранить как**, выберите **Рабочий стол** и папку **Моя рабочая папка**); выполняйте текущее сохранение при заполнении отчёта, не реже 1 раза в 5 минут;
- сделайте вывод по практической работе в конце отчёта.

 **Задача 1.** С помощью функции `list()` создайте одномерный массив `a` из букв своего имени и фамилии.

Входные данные

Выходные данные

```
['А', 'л', 'е', 'к', 'с', 'а', 'н', 'д', 'р', ' ', 'П', 'у', 'ш',
'к', 'и', 'н']
```

 **Задача 2.** К задаче 1:

- Создайте ещё один массив `b`, состоящий из элементов `['студент', 'первого', 'курса']` и соедините два массива с помощью функции конкатенации `+` в переменную `c`. Выведите результат переменной `c`.

- С помощью функции `len(a)` определите и выведите длину массива `a`.
- С помощью метода `count()`, определите количество букв `a`, в массиве `a`.
- Упорядочите элементы массива `a` по возрастанию с помощью метода `sort()` без смены регистра.

#### Входные данные

#### Выходные данные

```
['A', 'л', 'е', 'к', 'с', 'а', 'н', 'д', 'р', ' ', 'П', 'у', 'ш', 'к', 'и', 'н', 'студент', 'первого', 'курса']
```

```
16
```

```
1
```

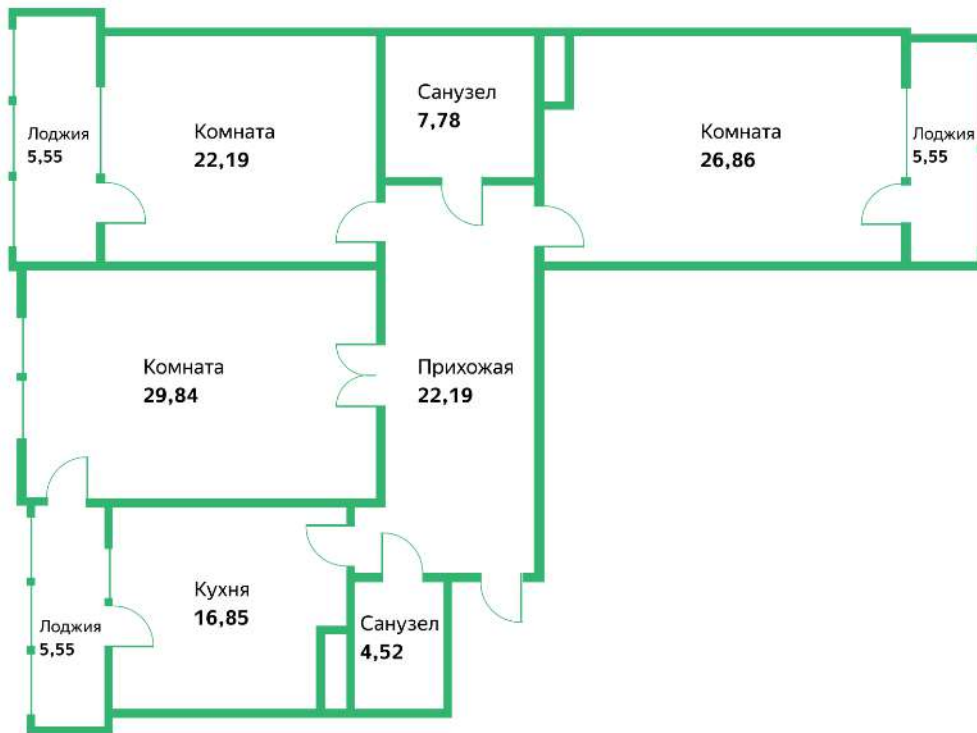
```
[' ', 'A', 'П', 'а', 'д', 'е', 'и', 'к', 'к', 'л', 'н', 'н', 'р', 'с', 'у', 'ш']
```



**Задача 3.** Составить программу, которая найдёт количество помещений, площадь которых равна 22.19 м<sup>2</sup>.

#### Алгоритм решения задачи:

1. Создаём список площадей помещений (см. ниже) – `flat`.
2. Создаём переменную, в которой будем хранить постоянное значение комнаты площадью 22.19 м<sup>2</sup>.
3. Создаём переменную-счётчик: в ней будем хранить количество обнаруженных помещений нужной площади. Пока что значение этой переменной – ноль.
4. Применим цикл `for room in созданном списке`; `room` – это имя переменной, которая поочередно принимает значения из списка (отдельная комната в квартире).
5. Сравниваем каждое значение с 22.19 (в виде переменной).
6. Если совпадает – увеличиваем счётчик.
7. После завершения цикла выводим результат: `'Комнат площадью'`, `[переменная]`, `'кв. м: '`, `[значение счётчика]`.



**Рис. 10.** План квартиры на 2 этаже

**Задача 4.** Дан список чисел: `numbers = [10, 20, 30, 40, 50, 60]`. Пользователь вводит число. Проверьте, содержится ли оно в списке. Если содержится, выведите: «Число найдено в списке!». Иначе выведите: «Число отсутствует в списке!».

**Входные данные**

Введите число: 30  
Введите число: 39

**Выходные данные**

Число найдено в списке!  
Число отсутствует в списке!

**Задача 5.** Дан список товаров: `products = ["бетон", "кирпич", "песок", "цемент", "бетон", "арматура", "бетон"]`, определите, сколько раз в списке встречается слово "бетон" (не учитывая регистр).

**Входные данные**

**Выходные данные**

Количество упоминаний слова 'бетон': 3

**Задача 6.** У вас есть список строительных материалов: `materials = ["кирпич", "цемент", "арматура", "песок"]`. Добавьте в этот список новый материал – "бетон", затем отсортируйте список и выведите результат.

**Входные данные**

**Выходные данные**

Отсортированный список материалов:  
['арматура', 'бетон', 'кирпич', 'песок', 'цемент']

**Задача 7.** Дан список температурных значений за неделю:


`temperatures = [15, 20, 18, 25, 30, 28, 22]`. Найдите максимальную и минимальную температуру, используя функции `max()` и `min()`.

**Входные данные**

**Выходные данные**

Максимальная температура: 30

Минимальная температура: 15

 **Задача 8.** Дан список букв алфавита: `russian_alphabet = ['a', 'б', 'в', 'г', 'д', 'е', 'ё', 'ж', 'з', 'и', 'й', 'к', 'л', 'м', 'н', 'о', 'п', 'р', 'с', 'т', 'у', 'ф', 'х', 'ц', 'ч', 'ш', 'щ', 'ъ', 'ы', 'ь', 'э', 'ю', 'я']`. Выведите: первую букву списка, последнюю букву списка (используйте отрицательный индекс), пятую букву списка.

## Практическая работа 32. Анализ алгоритмов с использованием таблиц трассировки без использования компьютера

**Цель работы:** научиться анализировать работу алгоритмов, используя таблицы трассировки, без непосредственного выполнения кода на компьютере.

### Теоретические сведения

**Таблица трассировки** (таблица отладки) – это инструмент, используемый для анализа алгоритмов. Она помогает понять, как изменяются переменные на каждом шаге выполнения программы. Таблицы трассировки (матрицы отслеживания) получили свое название от английского слова *trace*, что означает «проследивать» или «отслеживать».

#### Как строится таблица трассировки?

1. Записываем входные данные.
2. Определяем переменные, используемые в алгоритме.
3. Прописываем последовательные шаги работы программы.
4. В каждой строке таблицы фиксируем изменения значений переменных.
5. Делаем вывод о работе алгоритма.

**Пример:** анализ работы алгоритма сложения чисел.

Дан алгоритм:

```
sum = 0
for i in range(1, 6):
    sum = sum + i
```

**Построим таблицу трассировки:**

<b>i</b>	<b>sum</b> (до операции)	Промежуточные вычисления	<b>sum</b> (после операции $sum = sum + i$ )
1	0	$0 + 1$	1
2	1	$1 + 2$	3
3	3	$3 + 3$	6
4	6	$6 + 4$	10
5	10	$10 + 5$	15

**Вывод:** после выполнения цикла переменная **sum** содержит сумму чисел от 1 до 5, то есть 15.

## Ход выполнения практической работы

Оформите отчёт по практической работе в тетради.

**Задание 1.** Анализ поиска максимального элемента. Дан алгоритм:

```
numbers = [4, 8, 2, 9, 5]
max_num = numbers[0]
for num in numbers:
    if num > max_num:
        max_num = num
```

**Заполним таблицу трассировки:**

num	max_num (до проверки)	Промежуточные вычисления num > max_num	max_num (после обновления)

**Вывод:** после завершения работы алгоритма max\_num = \_\_\_\_, что является максимальным элементом списка.

**Задание 2.** Проверка, есть ли в списке число 7. Дан алгоритм:

```
numbers = [3, 5, 7, 9]
found = False
for num in numbers:
    if num == 7:
        found = True
```

**Заполним таблицу трассировки:**

num	found (до проверки)	num == 7?	found (после обновления)
			True, как только мы нашли 7 и выполнили found = True, переменная навсегда остается True

**Вывод:** алгоритм успешно нашёл число \_\_\_\_, и переменная found стала \_\_\_\_.

**Задание 3.** Изменение элементов списка. Дан алгоритм:

```
n = [10, 20, 30, 40, 50]
n[2] = 99
n[4] = n[0] + n[1]
n[3] = 115
```

**Заполним таблицу трассировки:**

Шаг	Операция	n[0]	n[1]	n[2]	n[3]	n[4]

**Ответ:** [\_\_, \_\_, \_\_, \_\_, \_\_].

**Задание 4.** Подсчет количества входящих элементов. Дан алгоритм:

```
letters = ['a', 'b', 'c', 'a', 'd', 'a']
count_a = letters.count('a')
```

**Заполним таблицу трассировки:**

Шаг	Операция	letters	Текущий символ	count_a

**Ответ:** count\_a = \_\_.

**Задание 5.** Добавление элементов в список. Дан алгоритм:

```
fruits = ["apple", "banana"]
fruits.append("cherry")
fruits.append("mango")
```

**Заполним таблицу трассировки:**

<b>Шаг</b>	<b>Операция</b>	<b>fruits</b>

**Ответ:** ["\_\_\_\_\_", "\_\_\_\_\_", "\_\_\_\_\_", "\_\_\_\_\_"]

## Практическая работа №33. Создание реляционной базы данных. Сортировка. Фильтрация

**Цель работы:** научиться создавать таблицу базы данных в режиме **Конструктор**, сохранять и загружать базу данных; изучить основные приёмы заполнения и редактирования таблиц базы данных; ознакомиться с простой сортировкой значений таблицы, с поиском записей по образцу.

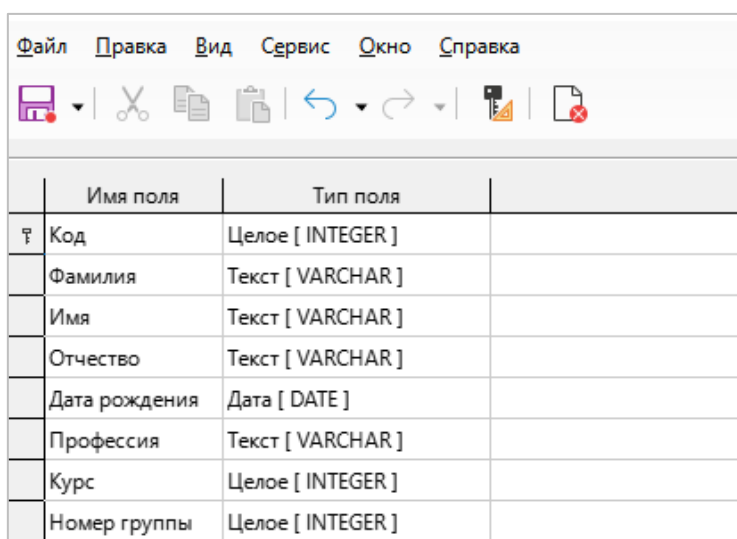
### Ход выполнения практической работы

#### **Задание 1. Создать базу данных.**

1. На рабочем столе откройте папку **Папки преподавателей** и в ней найдите и откройте файл с заготовкой **ПР33\_Заготовка**. Сохраните его в свою рабочую папку (**Рабочий стол** → **Моя рабочая папка**) и введите имя файла **ПР33\_Фамилия\_Группа**

2. Нажмите **Создать таблицу в режиме дизайна** и определите поля будущей таблицы. Для этого введите в строку столбца **Имя поля**, имя первого поля – **Код**, а в строке **Тип данных** щёлкните по ячейке напротив и из списка выберите тип данных **Целое [INTEGER]**.


3. Остальные поля заполните согласно рис. 11.



	Имя поля	Тип поля	
☐	Код	Целое [ INTEGER ]	
	Фамилия	Текст [ VARCHAR ]	
	Имя	Текст [ VARCHAR ]	
	Отчество	Текст [ VARCHAR ]	
	Дата рождения	Дата [ DATE ]	
	Профессия	Текст [ VARCHAR ]	
	Курс	Целое [ INTEGER ]	
	Номер группы	Целое [ INTEGER ]	

**Рис. 11.** Поля таблицы и типы данных


4. Создайте первичный ключ слева от поля **Код**, для этого, слева (пустой квадрат), нажмите ПКМ и выберите пункт **Первичный ключ**. Слева должен появиться ключ.

5. Сохраните созданную таблицу, нажав на пиктограмму , в появившемся окне введите имя таблицы **Список** и щёлкните **ОК**.

6. Заполните базу данных согласно таблице (ниже).


**Таблица 1** – Таблица данных


Код	Фамилия	Имя	Отчество	Дата рождения	Профессия	Курс	Номер группы
1	Елагин	Денис	Олегович	06.04.1993	Автомеханик	1	27
2	Жибаков	Дмитрий	Сергеевич	25.10.1992	Автомеханик	1	27
3	Усов	Константин	Иванович	08.05.1992	Автомеханик	2	28
4	Христофоров	Михаил	Валерьевич	21.03.1991	Автомеханик	2	28
5	Бузина	Ирина	Алексеевна	18.11.1993	Портной	1	35
6	Андросова	Июина	Ивановна	14.02.1992	Портной	2	16
7	Володина	Екатерина	Петровна	23.06.1991	Портной	2	14
8	Ягупова	Ольга	Алексеевна	17.03.1991	Портной	2	14
9	Тимашевич	Евгений	Сергеевич	09.04.1993	Сварщик	1	8
10	Машкарин	Павел	Николаевич	11.07.1992	Сварщик	2	2

7. Сохраните данные, нажав на пиктограмму  и, закройте таблицу.  
 8. Добавьте в конец таблицы записи согласно рис. 12.

11	Третьяков	Алексей	Петрович	07.07.1991	Сварщик	3	9
12	Николаев	Владимир	Михайлович	10.12.1992	Портной	3	5

**Рис. 12.** Добавление записей в таблицу

9. Отсортируйте данные в поле **Фамилия** по алфавиту. Для этого щёлкните в любой записи поля **Фамилия** и нажмите на пиктограмму  сверху на панели управления.


10. Отсортируйте данные в поле **Номер группы** по убыванию. Для этого щёлкните в любой записи поля **Номер группы** и нажмите на пиктограмму  сверху на панели управления.

11. Отсортируйте данные в поле **Курс** по убыванию.

12. Отсортируйте данные в поле **Код** по возрастанию.


13. Сохраните текущую таблицу.


14. **Поиск записей по образцу.** Для поиска в таблице обучающегося Третьякова:


- установите курсор в первую строку поля **Фамилия**;
- щёлкните по пиктограмме , которая позволяет найти запись по введённому значению;
- в появившемся окне **Поиск записи** напротив **Текст:** введите **Третьяков** и щёлкните **Найти** – после чего курсор перейдёт на 11-ю запись и выделит фамилию **Третьяков**.

– Закройте окно **Поиск записи**.

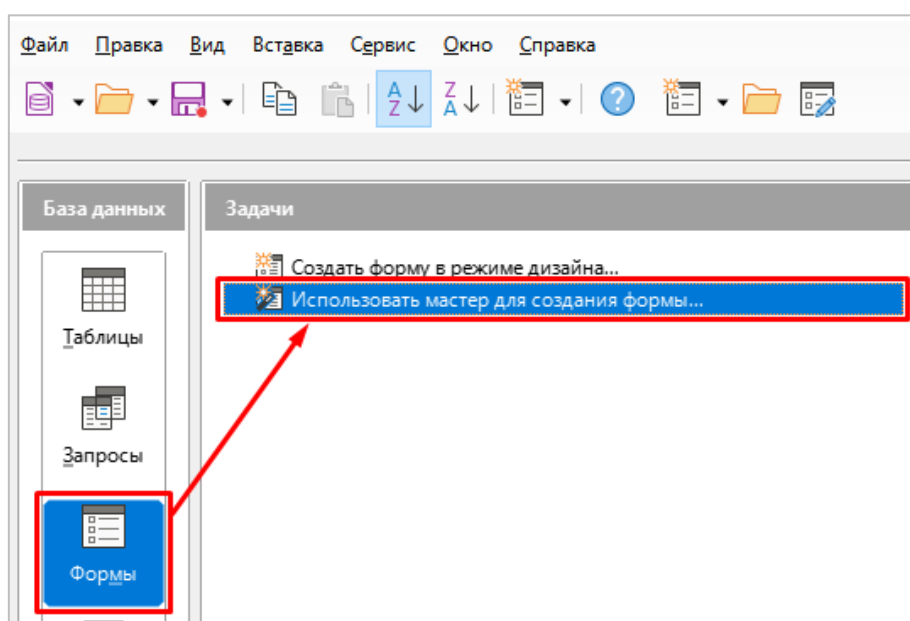
15. **Отбор данных с помощью фильтра.** Для фильтрации по полям **Профессия** и **Курс** выполните следующие действия:

– щёлкните по записи **Автомеханик** поля **Профессия** и щёлкните по пиктограмме Автофильтр ;

– щёлкните по записи **1** поля **Курс** и щёлкните по пиктограмме Автофильтр ;

– для отмены фильтрации щёлкните по пиктограмме  и в таблице появятся все записи, сохраните таблицу и закройте её.

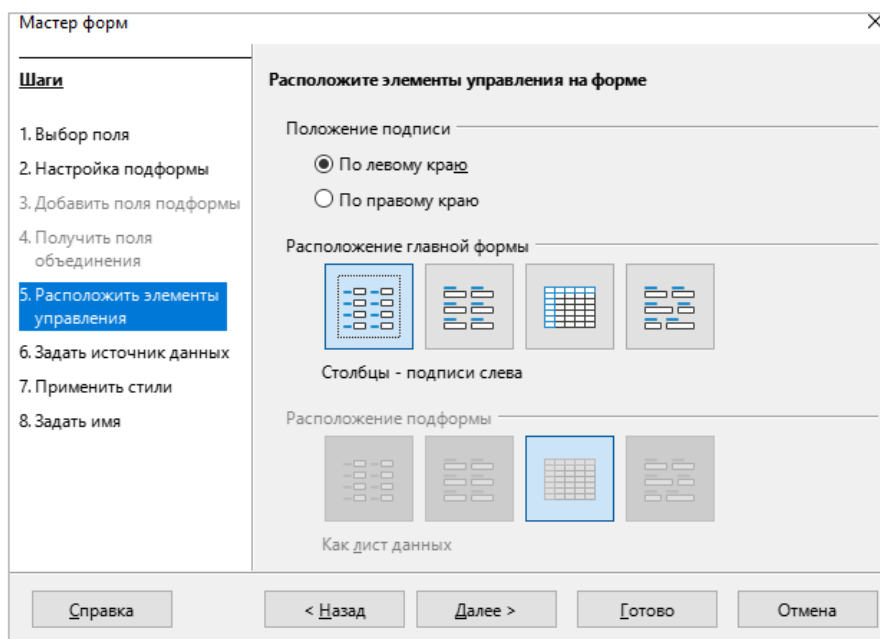
16. **Ввод и просмотр данных посредством формы.** Для создания формы по таблице **Список** перейдите на вкладку **Формы** и выберите **Использовать мастер для создания форм...** (рис. 13).



**Рис. 13.** Создание формы при помощи мастера

17. В появившемся окне выберите поля, которые будут присутствовать в форме. В данном случае будут присутствовать все поля, поэтому щёлкните по кнопке **>>**.

18. Перейдите к **Шагу 5. Расположить элементы управления** и настройте, как показано на рис. 14 и нажмите **Готово**.




**Рис. 14.** Выбор расположения главной формы

19. Закройте создавшую форму. Она появится в разделе **Формы**.

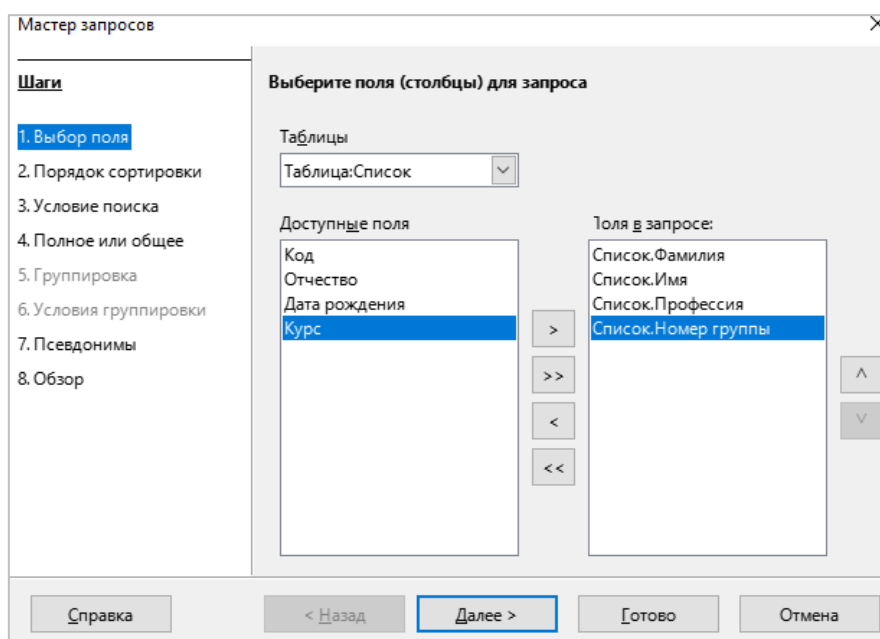
20. **Формирование запросов на выборку. Создание простого запроса.**

Для создания простого запроса:

– в окне базы данных откройте вкладку **Запросы** и щёлкните **Использовать мастер для создания запроса...**;

– в окне *Доступные поля* щёлкните по параметру **Фамилия** и нажмите  после чего поле **Фамилия** перейдёт в окно **Поля в запросе**;

– аналогично переведите поля **Имя**, **Профессия** и **Номер группы** в окно **Выбранные поля** (порядок важен, рис. 15);



**Рис. 15.** Создание простого запроса

– перейдите на шаг №8. *Обзор* и введите имя запроса – **Профессии**

**обучающихся;**

– щёлкните по кнопке Готово.

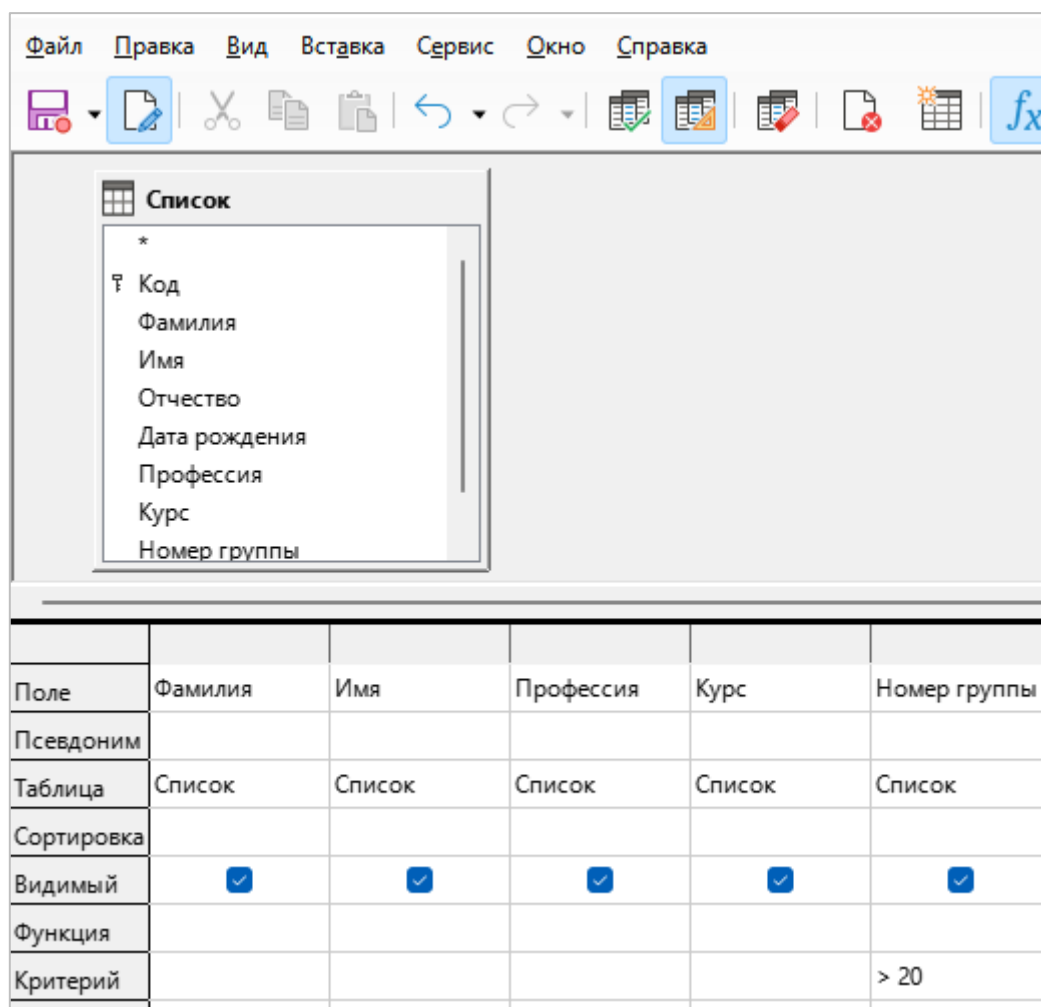
21. Отсортируйте данные по полю **Профессия** по возрастанию и сохраните запрос.

22. Создание запроса с параметром. Для создания простого запроса:

– в окне базы данных откройте вкладку **Запросы** и щёлкните **Создать запрос в режиме дизайна...**;

– добавьте таблицу **Список**;

– выберите оттуда поля **Фамилия, Имя, Профессия, Курс, Номер группы**;



**Рис. 16.** Создание запроса с параметром

– в строке **Критерий** введите напротив поля **Номер группы** условие **>20** (рис. 16);

– сохраните запрос с именем **Запрос с параметром**, в получившемся запросе будут те студенты, номер группы у которых больше числа 20.

– закройте окно запроса.



## Задание 2. Работа с однотабличной базой данных в СУБД.

1. Установите режим работы с таблицей (вкладка **Таблицы** – левая боковая панель программы). Откройте таблицу **Фильмы** и изучите содержимое. В столбце «Просмотрено», отметьте те фильмы, которые вы просмотрели.

2. Закройте таблицу и откройте её в режиме правки, для этого нажмите ПКМ по таблице **Фильмы** и выберите **Правка** (рис. 17).

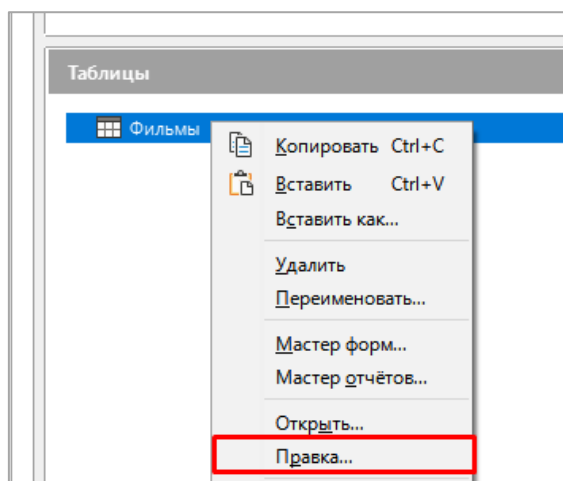


Рис. 17. Открытие таблицы в режиме правки

3. Познакомьтесь с именем полей, их типами и описанием. Добавьте новое поле **Сборы в мире**:

– **Имя поля:** **Сборы в мире**

– **Тип поля:** **Число**

– **Описание:** Сумма, которая заработана фильмом от продажи билетов в кинотеатрах

4. Выполните текущее сохранение **Ctrl + S**. Закройте **Конструктор таблиц**. Откройте таблицу двойным щелчком ЛКМ. В базе данных появился новый столбец. Изменим его формат, для этого нажмите ПКМ по заголовку «**Сборы в мире**» и выберите **Формат столбца**. Во вкладке **Формат** настройте категорию, формат в соответствии с рис. 18. Там же проверьте наличие разделителя разрядов.

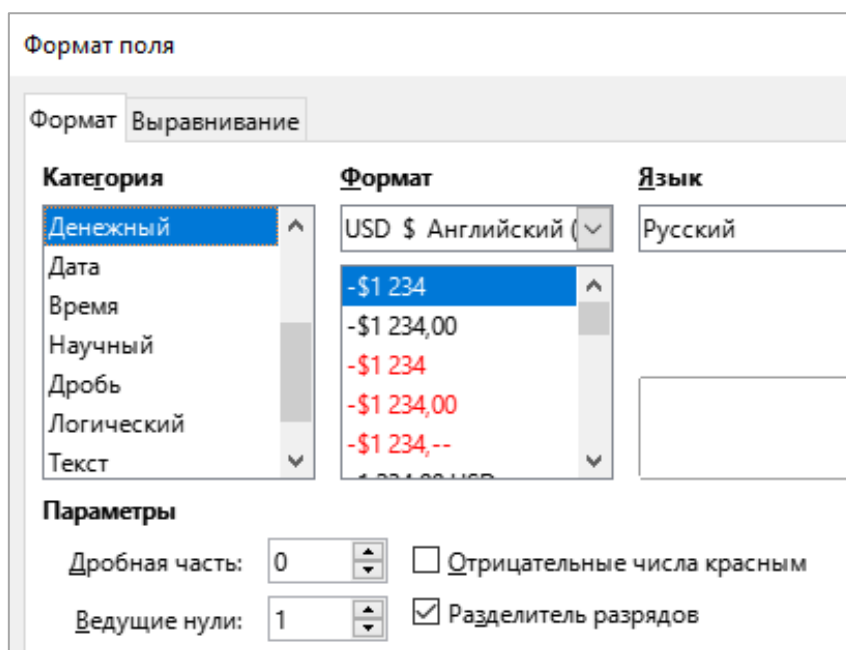



Рис. 18. Настройка поля «Сборы в мире»

5. Заполните информацию по каждому фильму (см. таблицу 2).

Таблица 2 – Мировые сборы фильмов

№	Фильм	Сборы в мире	№	Фильм	Сборы в мире
1	Форрест Гамп	\$677 387 716	11	Головоломка	\$857 611 174
2	1 + 1	\$426 588 510	12	Один дома 2: Затерянный в Нью-Йорке	\$358 994 850
3	Интерстеллар	\$677 896 797	13	Матрица: Воскрешение	\$156 575 035
4	Начало	\$828 322 032	14	Человек-паук: Нет пути домой	\$1 535 856 116
5	Унесённые призраками	\$355 467 076	15	Бегущий по лезвию 2049	\$259 239 658
6	Остров проклятых	\$294 804 195	16	Вечное сияние чистого разума	\$72 042 875
7	Игры разума	\$313 542 341	17	Побег из Шоушенка	\$28 418 687
8	Титаник	\$1 843 478 449	18	Поймай меня если сможешь	\$352 114 312
9	Гарри Поттер и Тайная комната	\$878 979 634	19	Время	\$173 930 596
10	Хатико: Самый верный друг	\$46 184 345	20	Бойцовский клуб	\$100 853 753

6. Использование фильтра. На панели инструментов сверху выберите **Стандартный фильтр** . Выполните (по отдельности) сортировку по следующим критериям:


- все фильмы, которые были сняты только в США;

- все фильмы с рейтингом выше 8,1;
- все фильмы с рейтингом выше 8,2 и старше 2001 года;
- все фильмы, снятые в США с бюджетов больше \$500 000.

7. Отмените фильтр. Для этого на панели инструментов сверху выберите

Удалить фильтр/сортировку .

8. Закройте таблицу. Выполните текущее сохранение Ctrl + S.

 **Задание 3.** Создайте базу данных как показано в таблице 3.

**Таблица 3** – Данные для зачётного задания 3

Код	Номер группы	Фамилия	Имя	Пол	ЕГЭ	Количество баллов
1	ТВ-2	Фролов	Игорь	м	<input checked="" type="checkbox"/>	189
2	ТВ-2	Полякова	Виктория	ж	<input type="checkbox"/>	201
3	ТВ-1	Виноградов	Алекс	м	<input checked="" type="checkbox"/>	221
4	ТВ-2	Марков	Пётр	м	<input checked="" type="checkbox"/>	196
5	ТВ-3	Данилов	Денис	м	<input type="checkbox"/>	179
6	ТВ-3	Чернова	Арина	ж	<input type="checkbox"/>	201
7	ТВ-1	Абрамов	Виктор	м	<input checked="" type="checkbox"/>	200
8	ТВ-2	Афанасьева	Вера	ж	<input checked="" type="checkbox"/>	234
9	ТВ-2	Исаев	Давид	м	<input type="checkbox"/>	250
10	ТВ-1	Маслова	Мирослава	ж	<input type="checkbox"/>	199
11	ТВ-3	Климова	Валентина	ж	<input checked="" type="checkbox"/>	198
12	ТВ-3	Филатов	Олег	м	<input checked="" type="checkbox"/>	189
13	ТВ-3	Грицаев	Дмитрий	м	<input checked="" type="checkbox"/>	197
14	ТВ-2	Соболев	Виктор	м	<input type="checkbox"/>	189
15	ТВ-1	Миронов	Артём	м	<input checked="" type="checkbox"/>	177

1. Создайте простой запрос по таблице *Список группы* с полями **Фамилия**, **Имя**, **ЕГЭ**.

2. Создайте запрос с параметром на **Количество баллов** больше 190.

3. Создайте форму по таблице *Список группы* по **всем полям** таблицы.

4. Отсортируйте в таблице *Список группы* поле **Количество баллов** по возрастанию.

5. Выполните отбор данных по фильтру, выбрав в поле **Номер группы** группу **ТВ-2**.

## Практическая работа №34. Создание простых и вычислительных запросов в реляционной базе данных

**Цель работы:** освоить навыки создания простых и вычислительных запросов в реляционной базе данных LibreOffice Base.

### Теоретические сведения

В LibreOffice Base вы можете создавать **вычисляемые запросы**, которые позволяют выполнять математические операции или вычисления на основе данных из таблиц.

**Запрос** – это специальное представление таблицы. Запрос может отображать выбранные записи или выбранные поля в записях; он также может сортировать эти записи. Запрос может применяться к одной или к нескольким таблицам, если они связаны общими полями данных.

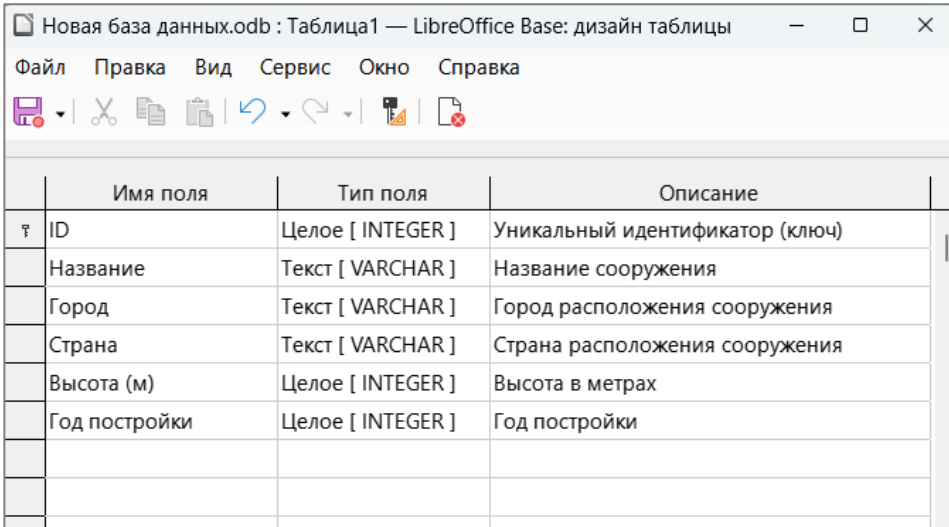
Используйте запросы для поиска записей из таблиц данных на основе определённых критериев. Все запросы, созданные для базы данных, перечислены под записью **Запросы**.

### Ход выполнения практической работы

1. На рабочем столе в папке *Программы* найдите и откройте LibreOffice Base. Сохраните файл в свою рабочую папку (*Рабочий стол* → *Моя рабочая папка*) с именем ПР34\_Фамилия\_Группа. Тип файла: ODF – База данных (\*.odb).

2. Нажмите **Создать таблицу в режиме дизайна** и определите поля будущей таблицы. Для этого введите в строку столбца **Имя поля**, имя первого поля – **ID**, а в строке **Тип данных** щёлкните по ячейке напротив и из списка выберите тип данных **Целое [INTEGER]**.


3. Остальные поля заполните согласно рисунку (см. ниже).



	Имя поля	Тип поля	Описание
☑	ID	Целое [ INTEGER ]	Уникальный идентификатор (ключ)
	Название	Текст [ VARCHAR ]	Название сооружения
	Город	Текст [ VARCHAR ]	Город расположения сооружения
	Страна	Текст [ VARCHAR ]	Страна расположения сооружения
	Высота (м)	Целое [ INTEGER ]	Высота в метрах
	Год постройки	Целое [ INTEGER ]	Год постройки

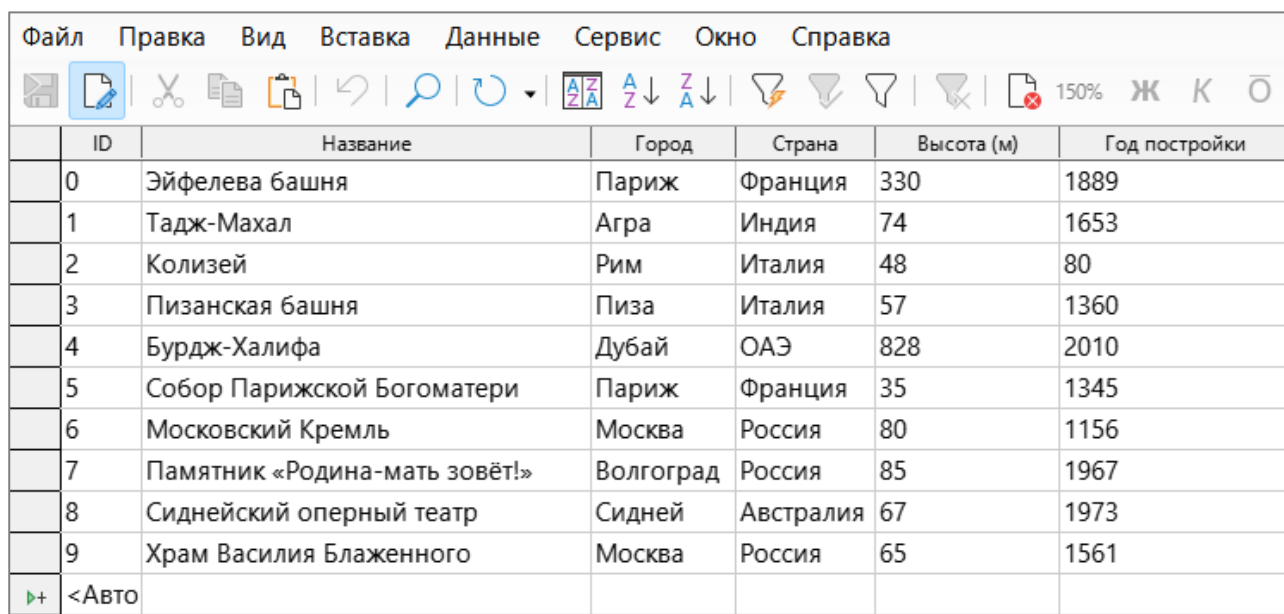
Рис. 19. Поля таблицы и типы данных

4. Создайте первичный ключ слева от поля **ID**, для этого, слева (пустой квадрат), нажмите ПКМ и выберите пункт *Первичный ключ*. Слева должен появиться ключ.

5. Сохраните созданную таблицу, нажав на пиктограмму , в появившемся окне введите имя таблицы **Архитектура** и щёлкните *ОК*.

6. Закройте Конструктор таблиц.

7. Нажмите двойным щелчком на таблицу **Архитектура**. Заполните таблицу данными согласно рис. 20.



ID	Название	Город	Страна	Высота (м)	Год постройки
0	Эйфелева башня	Париж	Франция	330	1889
1	Тадж-Махал	Агра	Индия	74	1653
2	Колизей	Рим	Италия	48	80
3	Пизанская башня	Пиза	Италия	57	1360
4	Бурдж-Халифа	Дубай	ОАЭ	828	2010
5	Собор Парижской Богоматери	Париж	Франция	35	1345
6	Московский Кремль	Москва	Россия	80	1156
7	Памятник «Родина-мать зовёт!»	Волгоград	Россия	85	1967
8	Сиднейский оперный театр	Сидней	Австралия	67	1973
9	Храм Василия Блаженного	Москва	Россия	65	1561

**Рис. 20.** Таблица данных

8. Добавьте столбец **Архитектор**, самостоятельно найдите информацию и внесите в свою таблицу.

### Простые запросы:

9. Вывести список всех архитектурных сооружений с их годом постройки и высотой. Для этого в окне базы данных откройте вкладку **Запросы** и щёлкните **Создать запрос в режиме дизайнера...**

10. Добавьте таблицу **Архитектура**. Выберите оттуда поля **Название**, **Год постройки**, **Высота (м)**. Нажмите **Выполнить запрос** (рис. 21) или кнопку **F5**. Сохранить с именем **Годы постройки**.



**Рис. 21.** Выполнение запроса

11. Найти все архитектурные объекты, построенные после 1950 года. Для этого в поле **Критерий** столбца **Год постройки** укажите условие. Нажмите

Выполнить запрос (рис. 21) или кнопку F5. Сохранить с именем После 1950 (Файл – Сохранить как).

12. Вывести названия сооружений и страны, в которых они расположены. Нажмите Выполнить запрос или кнопку F5. Сохранить с именем Название и страна (Файл – Сохранить как).

13. Найти все сооружения, высота которых превышает 100 метров. Нажмите Выполнить запрос или кнопку F5. Сохранить с именем Выше 100 метров (Файл – Сохранить как).

#### Запросы с сортировкой и фильтрацией:

14. Вывести список всех сооружений, отсортированный по году постройки (по возрастанию). Нажмите Выполнить запрос или кнопку F5. Сохранить с именем По возрастанию (Файл – Сохранить как).

15. Отобразить 5 самых высоких архитектурных объектов. Для этого из таблицы Архитектура выберите поле Название и Высота (м) Установите сортировку По убыванию высоты (рис. 22). Запустите запрос F5.

Поле	Название	Высота (м)
Псевдоним		
Таблица	Архитектура	Архитектура
Сортировка		по убыванию
Видимый	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Функция		

Рис. 22. Сортировка в запросах

16. Для создания ограничения на 5 записей дополним SQL-запрос. Нажмите Режим дизайна вкл./выкл. (рис. 23).

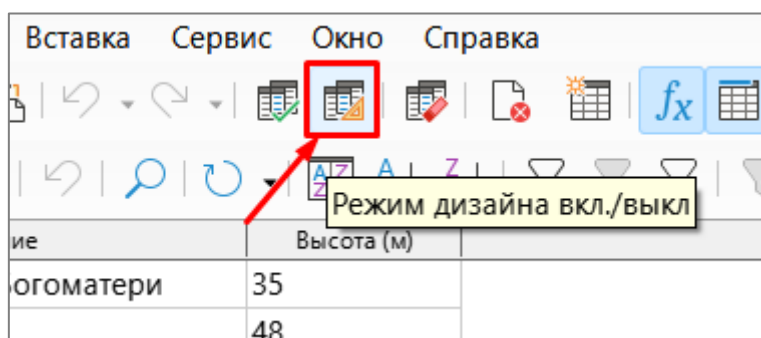


Рис. 23. Включение режима дизайна

SQL-запрос выглядит следующим образом:

```
SELECT "Название", "Высота (м)" FROM "Архитектура" ORDER BY  
"Высота (м)" DESC
```

**SELECT, FROM** – обязательные элементы запроса, которые определяют выбранные столбцы, их порядок и источник данных.

**ORDER BY** с ключевым словом **DESC** в SQL позволяет отсортировать данные в порядке убывания.

17. Воспользуемся элементом **LIMIT** для ограничения количества возвращаемых записей на основании предельного значения.

18. Запрос будет выглядеть так:

```
SELECT "Название", "Высота (м)" FROM "Архитектура" ORDER BY  
"Высота (м)" DESC LIMIT 5
```

19. Нажмите **Выполнить запрос** или кнопку **F5**. Сохраните запрос с именем **5 самых высоких объектов**.

20. Найти все сооружения, построенные в XX веке (1901-2000 годы) Для этого в критерии столбца Год постройки используйте выражение **BETWEEN 1901 AND 2000**. Сохраните запрос с именем **XX век**.

21. Показать архитектурные сооружения, находящиеся в России. Сохраните запрос с именем **Архитектура России**.

#### Вычислительные запросы:

22. **Вычисление возраста архитектурного сооружения.** Рассчитаем, сколько лет прошло с момента постройки каждого архитектурного сооружения. Для этого нужно вычесть год постройки из текущего года.

23. Закройте конструктор запросов и заново в окне базы данных откройте вкладку **Запросы** и щёлкните **Создать запрос в режиме дизайна...**

24. Добавьте таблицу Архитектура. Выберите оттуда поля Название, Год постройки. Добавьте вычисляемое поле как показано на рисунке ниже.

Поле	Название	Год постройки	2025 - "Год постройки"
Псевдоним			Возраст
Таблица	Архитектура	Архитектура	
Сортировка			
Видимый	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Функция			
Критерий			

**Рис. 24.** Создание вычисляемого запроса

25. Это поле будет вычислять возраст каждого сооружения на основе 2025 года. Нажмите **Выполнить запрос** или кнопку **F5**. Сохраните запрос с именем **Возраст**.

26. **Суммирование высоты всех сооружений.** Рассчитаем общую высоту всех архитектурных сооружений, представленных в базе данных.

– В *Конструкторе запроса* добавьте поле: Высота (м).

– В строке *Функция* выберите *Сумма* для поля *Высота (м)*:

– Нажмите **Выполнить запрос** или кнопку **F5**.

27. Пример результата: 1669. Сохраните запрос с именем **Высота** (*Файл – Сохранить как*). Это поле суммирует высоту всех архитектурных сооружений в базе данных.

28. Рассчитать среднюю высоту всех архитектурных сооружений. Выберите в конструкторе поле **Высота (м)**, добавьте *Псевдоним – Средняя высота* и выберите функцию **AVG (или Среднее)**. Сохраните запрос с именем **Средняя высота по всем странам**.

29. Определить количество архитектурных объектов в базе данных. В поле введите знак \*, добавьте *Псевдоним – Количество сооружений*, выберите функцию **Количество**. Сохраните запрос с именем **Количество сооружений**.

30. Найти минимальную высоту архитектурных сооружений. Выберите в конструкторе поле **Высота (м)**, добавьте *Псевдоним – Минимальная высота*, выберите функцию **Минимум**. Аналогично сделайте второй столбец для нахождения максимальной высоты. Сохраните запрос с именем **Мин и макс высота**.

31. Для подсчёта сооружений в каждой стране создайте следующие условия (рис. 25). Сохраните запрос с именем **Количество сооружений по странам**.

Поле	Страна	Архитектура.*
Псевдоним		Количество объектов
Таблица	Архитектура	Архитектура
Сортировка		
Видимый	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Функция	группировка	Количество

**Рис. 25.** Запрос на группировку и количество

32. Найдите среднюю высоту сооружений по странам. Для этого добавьте два поля **Страна** и **Высота (м)**. Столбцу **Высота (м)** добавьте *Псевдоним Средняя высота*. Используйте функции для столбца **Страна** – **группировка**, для столбца **Высота (м)** – **AVG (или Среднее)**. Сохраните запрос с именем **Средняя высота по отдельным странам**.

## Практическая работа №35. Вычисление в электронной таблице: относительная, абсолютная и смешанная адресация

**Цель работы:** получить навыки практической работы по созданию и форматированию электронных таблиц; научиться вводить и копировать формулы, подсчитывать итоги с использованием функции Автосумма.

### Ход выполнения практической работы

1. На **Рабочем столе** в папке своего преподавателя и найдите файл ПР35\_Заготовка.

2. Откройте файл при помощи программы **LibreOffice Calc** ПКМ по файлу → **Открыть с помощью** → **Выбрать другое приложение** (рис. 26).

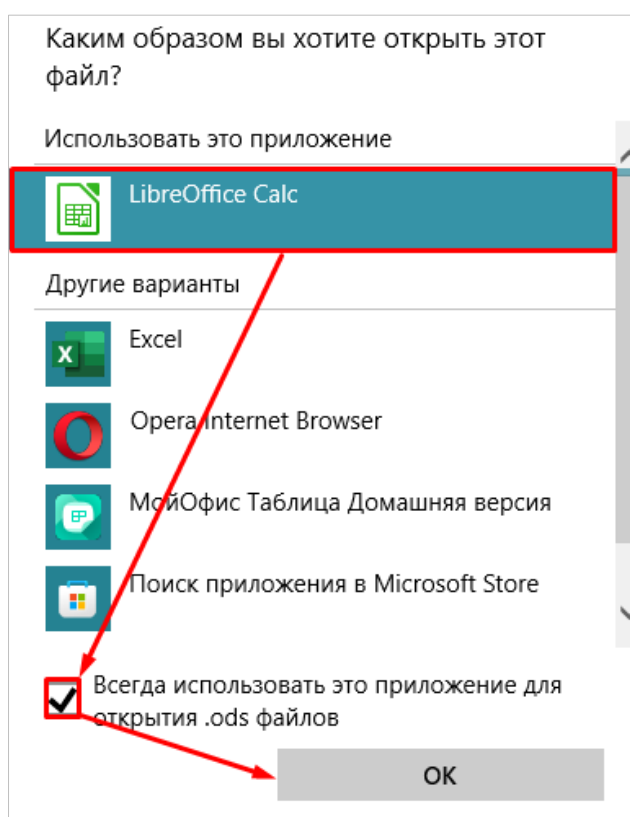



Рис. 26. Выбор приложения для открытия файла

3. Сохраните файл в свою рабочую папку с именем **ПР35\_Фамилия\_Группа** (вкладка **Файл** → **Сохранить как...** → **Рабочий стол** → **Моя рабочая папка**).

 **Задание 1. Ввод данных в таблицу.** Составьте смету расходов на строительно-отделочные работы (Приложение №1 К Договору №57-К от 20.03.2025).

### Алгоритм построения таблицы

1. Настроим столбец **F**. Для этого нажмите на столбец **F** (после чего должен выделиться весь столбец), *ЛКМ* нажмите на вкладку **Формат** → **Столбцы** → **Ширина** → введите 3,70 см.

2. Выделите диапазон ячеек **F1:F3**, на панели форматирования (сверху) нажмите на Курсив (**Ctrl + I**). Наберите текст в каждую ячейку по отдельности как показано в таблице 1.

3. Настройте денежный формат некоторых ячеек. Для этого выделите диапазон **E8:F21**, нажмите **Ctrl + 1** и выполните команды в соответствии с рис. 27.

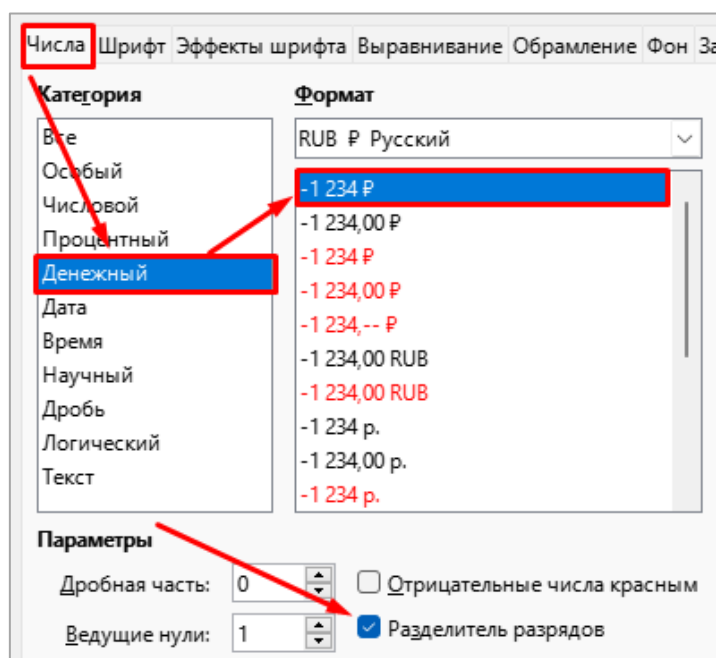



Рис. 27. Настройка формата ячеек

4. Выделите ячейки **A5:F5** и объедините их, нажав на кнопку **Объединить и центрировать**  на панели форматирования (сверху) нажмите на **Жирный (Полужирный)** (**Ctrl + B**).

5. Настройте ширину столбцов: **A** (0,90), **B** (5,60), **C** (1,38), **D** (2,53), **E** (2,45).

6. Выделите строку **7** (целиком). Для этого нажмите на строку **7** (после чего должна выделиться вся строка), на панели форматирования (сверху) нажмите на **Жирный (Полужирный)**, выравнивание **по центру** (**Ctrl + E**), переносить текст, центрировать по вертикали.

7. Наберите заголовочные данные таблицы по образцу рисунка 3 (заполните диапазон **A7:F7**).

8. В ячейку **A8** введите число **1**. Нажмите на ячейку **A8** и при помощи маркера автозаполнения (чёрный квадратик в правом нижнем углу ячейки) протяните вниз до 21-й строки.

9. Выделите столбец **A** (целиком) настройте параметры шрифта: жирный, курсив, по центру, центрировать по вертикали.

10. Выделите диапазон ячеек **B8:B21** настройте параметры шрифта: обычный, выравнивание по левому краю, перенос текста, центрировать по вертикали.

11. Выделите диапазон ячеек **C8:F21** настройте параметры шрифта: обычный, выравнивание по центру, центрировать по вертикали.

12. Заполните диапазон **D8:D21** данными как показано в таблице 1.

13. Выделите диапазон **E8:F21** и нажмите **Ctrl + 1**. Настройте формат ячеек как показано в таблице 1.

**Таблица 1** – Исходные данные для задания 1

	<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>E</b>	<b>F</b>
<b>1</b>						Приложение №1
<b>2</b>						к Договору № 57-К
<b>3</b>						от 20.03.2025
<b>4</b>						
<b>5</b>	<b>Смета на строительно-отделочные работы</b>					
<b>6</b>						
<b>7</b>	<b>№ п/п</b>	<b>Наименование работ</b>	<b>Ед. изм.</b>	<b>Количество</b>	<b>Цена</b>	<b>Стоимость</b>
<b>8</b>	<b>1</b>	Укладка ламината	кв. м.	200	240	
<b>9</b>	<b>2</b>	Укладка кафеля	кв. м.	15	600	
<b>10</b>	<b>3</b>	Оклейка обоями	кв. м.	300	200	
<b>11</b>	<b>4</b>	Установка натяжного потолка	кв. м.	200	300	
<b>12</b>	<b>5</b>	Установка раковины	шт.	1	1 000	
<b>13</b>	<b>6</b>	Установка бойлера	шт.	1	2 000	
<b>14</b>	<b>7</b>	Монтаж межкомнатной двери (полный комплект)	шт.	1	17 500	
<b>15</b>	<b>8</b>	Монтаж розеток, выключателей	шт.	7	150	
<b>16</b>	<b>9</b>	Монтаж светильников (точечных)	шт.	6	200	
<b>17</b>	<b>10</b>	Замена счётчика	шт.	1	2 000	
<b>18</b>	<b>11</b>	Монтаж галтель	м.	60	80	
<b>19</b>	<b>12</b>	Монтаж тепло-шумоизоляции на потолок	кв. м.	200	200	
<b>20</b>	<b>13</b>	Установка водонагревателя проточного типа	шт.	1	3 100	
<b>21</b>	<b>14</b>	Погрузочно-разгрузочные работы	час	1	250	
<b>22</b>	<b>ИТОГО:</b>					

14. Заполнять столбец **Стоимость** (диапазон **F8:F21**) вручную **не нужно**. Нажмите на ячейку **F8**. Введите формулу **=D8\*E8**. После ввода формулы

нажмите кнопку **Enter**. Используя маркер автозаполнения протяните его до **21-й строки** (электронной таблицы) включительно.

15. Объедините диапазон ячеек **A22:E22**, настройте параметры шрифта: жирный, выравнивание по правому краю, наберите текст большими буквами как показано в таблице выше.

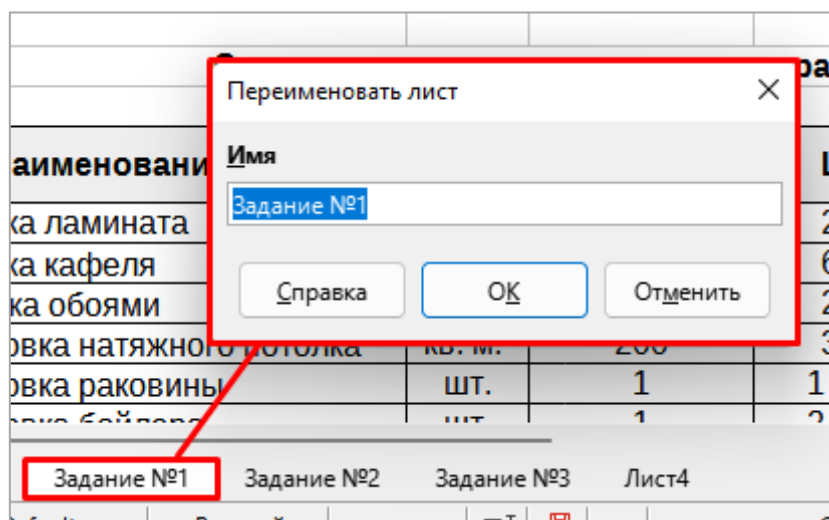
**Примечание.** Если вы набрали текст маленькими (строчными буквами), а его нужно было набрать большими (прописными) это можно исправить при помощи программы. Для этого нужно выделить слово или сочетание слов и нажать последовательность действий: вкладка **Формат** → **Текст** → **ПРОПИСНЫЕ**.

16. В ячейке **F22** посчитаем итог всех стоимостей. Для этого нажмите на ячейку **F22** и щёлкните по пиктограмме Автосумма  $\Sigma$ , выберите СУММ. Проверьте, диапазон внутри функции должен быть – **F8:F21**. Таким образом общий вид формулы: **=СУММ(F8:F21)**. Общий расход составит **249 900 Р**

17. В ячейке **F22** настройте денежный формат.

18. Оформите оставшуюся часть таблицы в соответствии с образцом.

19. Переименуйте ярлычок (Лист1), нажав по нему двойным щелчком ЛКМ (рис. 28).



**Рис. 28.** Переименование листа

20. Выполните текущее сохранение (**Ctrl + S**).

21. Нажмите на **Лист2**, внизу, где расположены все листы (все задания должны быть выполнены в одном документе).

**💡 Задание 2. Использование абсолютной ссылки в формуле и сортировка данных.** Постройте таблицу, содержащую сведения о стоимости туристических путёвок по отдельным услугам в разные страны мира.

1. Перейдите на второй лист. Занесите исходные данные в таблицу и оформите её в соответствии с таблицей 2.

2. Актуальный курс доллара и евро найдите самостоятельно на сайте Банка России: <https://cbr.ru/>. Внесите данные в ячейки **B1** и **B2**. Введите только числа.

**Таблица 2** – Исходные данные для задания 2

	<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>E</b>	<b>F</b>	<b>G</b>	<b>H</b>
<b>1</b>	Курс доллара	#ввести значение						
<b>2</b>	Курс евро	#ввести значение						
<b>3</b>								
<b>4</b>	<b>Страна</b>	<b>Дорога</b>	<b>Проживание в отеле</b>	<b>Питание</b>	<b>Страховка</b>	<b>Итого, руб.</b>	<b>Итого, долл.</b>	<b>Итого, евро</b>
<b>5</b>	Турция	40 139 Р	22 834 Р	21 000 Р	2 000 Р			
<b>6</b>	Испания	40 651 Р	29 652 Р	23000 Р	2 000 Р			
<b>7</b>	Италия	41 500 Р	31 568 Р	25602 Р	2 000 Р			
<b>8</b>	Германия	49 715 Р	29 663 Р	32500 Р	2 000 Р			
<b>9</b>	Греция	57 256 Р	24 567 Р	24522 Р	2 000 Р			
<b>10</b>	Чехия	38 962 Р	23 600 Р	23699 Р	2 000 Р			
<b>11</b>	Грузия	55 053 Р	19 370 Р	30200 Р	2 000 Р			
<b>12</b>	Индонезия	96 635 Р	41 200 Р	34200 Р	2 000 Р			
<b>13</b>	ОАЭ	76 598 Р	51 000 Р	36000 Р	2 000 Р			
<b>14</b>	Кипр	56 793 Р	45 200 Р	34700 Р	2 000 Р			

3. В столбце **F** посчитайте сумму в рублях по каждой стране, используя функцию **СУММ**.

4. Нажмите на ячейку **G5** и введите формулу для расчёта стоимости путёвки, используя абсолютный адрес ячейки **B1**. Формула для расчёта: **=F5/\$B\$1**. Для того чтобы быстро поставить абсолютную ссылку (\$), достаточно поставить курсор перед адресом ячейки и нажать клавишу **F4**.

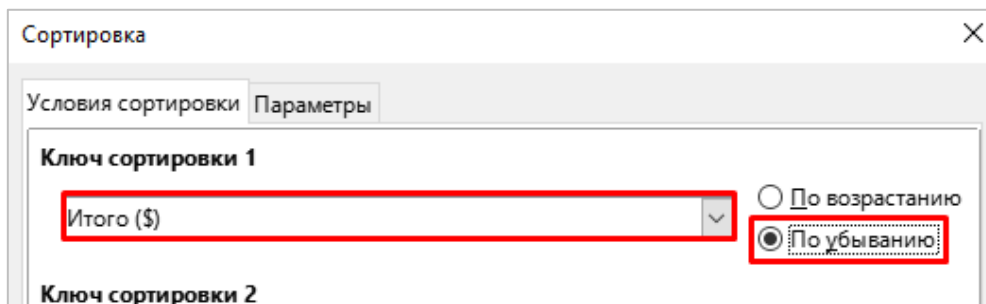
5. При помощи **маркера автозаполнения** (маленький чёрный квадрат в правом нижнем углу ячейки) скопируйте (зажмите маркер и протяните его вниз) до ячейки **G14**. Выделите диапазон **G5:G14** и на панели форматирования сверху нажмите на кнопку **Формат – денежный** и из раскрывающегося списка выберите **USD \$ Английский (США)**.

6. Нажмите на ячейку **H5** и введите формулу для расчёта стоимости путёвки, используя абсолютный адрес ячейки **B2**. Формула для расчёта: **=F5/\$B\$2**.

7. При помощи **маркера автозаполнения** (маленький чёрный квадрат в правом нижнем углу ячейки) скопируйте (зажмите маркер и протяните его вниз) до ячейки **H14**. Выделите диапазон **H5:H14** и на панели форматирования сверху нажмите на кнопку **Формат – денежный** и из раскрывающегося списка выберите **EUR € Итальянский (Италия)**.


8. Выделите диапазон **B5:E14** и на панели форматирования сверху нажмите на кнопку **Формат – денежный** и из раскрывающегося списка выберите **Р Стандарт**.

9. Произведите сортировку данных. Для этого выделите диапазон **A4:H14**, выполните команды **Данные** → **Сортировка**. В диалоговом окне Сортировка установите следующее (рис. 29), после чего нажмите **ОК**.



**Рис. 29.** Настройка параметров сортировки

10. Внизу переименуйте ярлычок (**Лист2**), нажав на него двойным щелчком мыши и, введите новое название **Задание 2**.

 **Задание 3.** Требуется составить таблицу сложения чисел первого десятка, т. е. заполнить таблицу, представленную ниже.

**Таблица 3** – Исходные данные для задания 2

	A	B	C	D	E	F	G	H	I	J
1		1	2	3	4	5	6	7	8	9
2	1									
3	2									
4	3									
5	4									
6	5									
7	6									
8	7									
9	8									
10	9									

1. При заполнении любой ячейки этой таблицы складываются соответствующие ей значения ячеек столбца A и строки 1. Иначе говоря, у первого слагаемого неизменным остаётся имя столбца (на него следует дать абсолютную ссылку), но изменяется номер строки (на неё следует дать относительную ссылку); у второго слагаемого изменяется номер столбца (относительная ссылка), но остаётся неизменным номер строки (абсолютная ссылка).

2. Внесём в ячейку B2 формулу  $=\$A2+B\$1$  и скопируем её на весь диапазон **B2:J10**. Получится таблица сложения, знакомая каждому первокласснику.
3. Уберём неиспользуемые ячейки в таблице. Для этого нажмите на вкладку Вид → Разметка страниц.
4. Переименуйте ярлычок (**Лист3**), нажав на него двойным щелчком мыши и, введите новое название **Задание 3**.

## Практическая работа №36. Фильтрация и сортировка большого массива данных в электронных таблицах

**Цель работы:** научиться использовать стандартные функции при решении задач; познакомиться с методами фильтрации, сортировки данных.

### Теоретические сведения

Важной частью анализа данных является их сортировка. С помощью сортировки данные можно расположить по возрастанию или по убыванию содержимого ячеек; также можно организовать сортировку по цвету ячеек, цвету шрифта и некоторым другим параметрам.

**Сортировка** – это расстановка элементов списка в заданном порядке. Для чисел обычно используют сортировку по возрастанию или убыванию, для текста – алфавитный (от А до Я) или обратный алфавитный порядок (от Я до А).

Программа Calc по умолчанию предполагает, что первая строка выделенного диапазона – это заголовок, и он в сортировке не участвует. Изменить этот режим можно с помощью меню **Данные** → **Сортировать**.

В таблице, состоящей из нескольких столбцов, можно осуществлять сортировку в двух вариантах:


1) **сортировка конкретного столбца:** тогда данные в каждой строке не будут связаны между собой и менять местоположение будут ячейки только конкретного выбранного столбца (диапазона); сортировка по одному столбцу диапазона может привести к нежелательным результатам.

2) **сортировка всей таблицы:** тогда данные в каждой строке будут связаны между собой и в результате изменения порядка ячеек выбранного столбца (диапазона) будет происходить одновременно с другими столбцами построчно.

В табличном процессоре данные кнопки  и  сортируются в следующем алфавитном порядке.

 – для чисел – сортировка **по возрастанию**;

 – для текста – сортировка **по алфавиту**;

 – для чисел – сортировка **по убыванию**;

 – для текста – сортировка **по обратному алфавитному порядку**;

## Ход выполнения практической работы



**Задание 1.** Фильтрация данных с использованием автофильтра, сортировки данных.

Фильтрация данных позволяет отобразить те строки, содержимое ячеек которых отвечает заданному условию (или нескольким условиям). Эта операция может выполняться с помощью автофильтра или расширенного фильтра.


1. На Рабочем столе в папке своего преподавателя и найдите файл **ПР36\_Заготовка**.

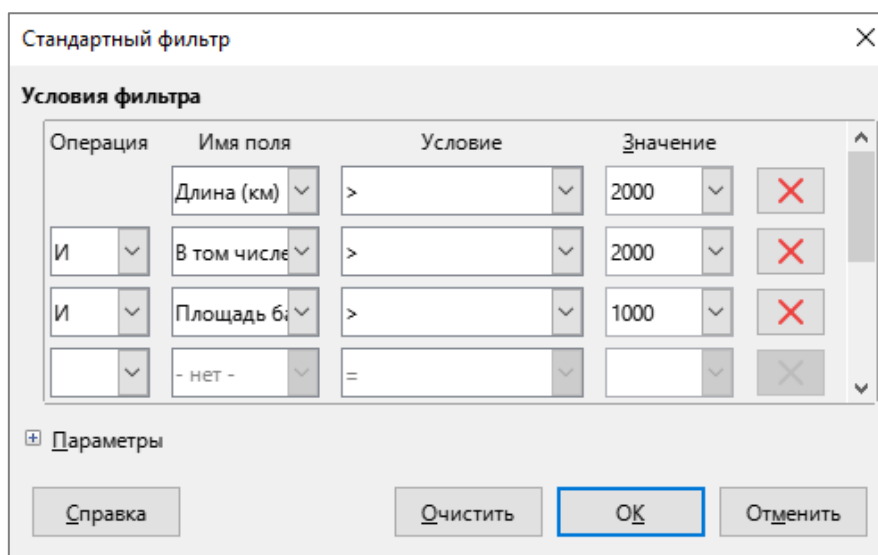
2. Оформите таблицу в соответствии с таблицей 1.

**Таблица 1** – Исходные данные для задания 1


	A	B	C	D	E	F	G
1	№	Название	Длина (км)	В том числе в РФ (км)	Площадь бассейна (тыс. кв. км)	Расход воды (куб. м./с)	Впадает в
2	1	Лена	4400	4400	2490	16350	море Лаптевых
3	2	Иртыш	4248	1900	1643	3000	Обь
4	3	Енисей	3487	3488	2580	19800	Карское море
5	4	Обь	3650	3650	2990	12492	Карское море
6	5	Амур	2824	2824	1855	12800	Амурский лиман
7	6	Волга	3531	3571	1360	8060	Каспийское море
8	7	Нижняя Тунгуска	2989	2989	473	3680	река Енисей
9	8	Вилюй	2650	2650	454	1468	река Лена
10	9	Колыма	2129	2129	643	3800	Восточно-Сибирское море
11	10	Ишим	2450	800	177	56,3	река Иртыш
12	11	Урал	2422	1550	237	400	Каспийское море
13	12	Оленёк	2292	2292	219	1210	Оленёкский залив, море Лаптевых
14	13	Алдан	2273	2273	729	5060	река Лена
15	14	Днепр	2201	485	504	1670	Чёрное море
16	15	Витим	1837	1837	225	1520	река Лена


3. Выделите диапазон **A1:G16** скопируйте его, нажмите на ячейку **A18** и выполните вставку. Аналогичным образом вставьте скопированную таблицу в ячейку **A35**. Таким образом, на **листе 2** у вас должно быть 3 одинаковые таблицы.

4. Выделите диапазон **B2:G2** и выполните команды **Данные** → **Автофильтр (Быстрый фильтр)**. В столбце **B** щёлкните левой кнопкой мыши по пиктограмме  в появившемся списке выберите **Стандартный фильтр**. В диалоговом окне **Стандартный фильтр** установите параметры согласно рисунку ниже и щёлкните **ОК**.




**Рис. 30.** Установка параметров для фильтрации данных

5. Выделите диапазон **B18:G18** и выполните команды **Данные** → **Автофильтр (Быстрый фильтр)**. В столбце **B** щёлкните левой кнопкой мыши по пиктограмме  в появившемся списке выберите **Стандартный фильтр**. В диалоговом окне **Стандартный фильтр** установите параметры: **площадь бассейна (тыс. кв. км) больше 1200** и **Расход воды (куб. м./с) меньше 12 000**.

6. Выделите диапазон **B35:G35** и выполните команды **Данные** → **Автофильтр (Быстрый фильтр)**. В столбце **B** щёлкните левой кнопкой мыши по пиктограмме  в появившемся списке выберите **Стандартный фильтр**. В диалоговом окне **Стандартный фильтр** установите 3 параметра, соединённых операцией **И**:

- Расход воды (куб. м./с) больше 1000;
- Впадает в реку Лена
- Площадь бассейна (тыс. кв. км) меньше 1000.

7. Внизу переименуйте ярлычок (Лист1), нажав на него двойным щелчком мыши и введите новое название **Задание 1**.

 **Задание 2.** Дана таблица с некоторыми характеристиками продуктов. Выполните фильтрацию данных в таблице.

1. Перейдите на новый лист, нажав внизу **Лист2**.
2. На втором листе продублирована одна и та же таблица несколько раз (см. табл. 2). Работайте отдельно с каждой таблицей для создания фильтра, предварительно выделяя заголовки таблицы и выбирая команды **Данные** → **Автофильтр (Быстрый фильтр)**.

3. Настройте следующие фильтры:

**Фильтр 1.** Нужно составить сложный фильтр и отобразить продукты по цене не меньше 70 рублей за штуку с весом не менее 150 грамм.

**Фильтр 2.** Вес – не менее 100 г, но не более 300 г.

**Фильтр 3.** Цена – в диапазоне от 50 до 90 включительно.

- Фильтр 4.** Дата производства – с 1 мая 2023 года и позже.
- Фильтр 5.** Срок годности – не менее 35 дней.
- Фильтр 6.** Жирность – больше 2%, но не превышает 5%.
- Фильтр 7.** Содержание углеводов – не менее 10 г на 100 г продукта.
- Фильтр 8.** Энергетическая ценность – не менее 80 ккал на 100 г.
- Фильтр 9.** Вес – не менее 120 г, но не более 500 г.
- Фильтр 10.** Цена – больше 35, но не превышает 85.
- Фильтр 11.** Дата производства – с 2 мая 2023 года и позже.
- Фильтр 12.** Срок годности – от 30 до 50 дней включительно.
- Фильтр 13.** Жирность – либо 2%, либо 7%.
- Фильтр 14.** Содержание углеводов – больше 5 г, но не более 15 г на 100 г.
- Фильтр 15.** Энергетическая ценность – от 60 до 130 ккал на 100 г.

**Таблица 2** – Исходные данные для задания 2

	A	B	C	D	E	F	G
<b>1</b>	<b>Вес (г)</b>	<b>Цена</b>	<b>Дата производства</b>	<b>Срок годности, день</b>	<b>Жирность, %</b>	<b>Углеводы (на 100 грамм)</b>	<b>Энергетическая ценность, ккал (на 100 грамм)</b>
<b>2</b>	140	51	30.04.2023	30	2%	7,0	63,00
<b>3</b>	130	53	01.05.2023	40	5%	16,0	130,00
<b>4</b>	105	55	02.05.2023	40	7%	12,3	123,00
<b>5</b>	250	80	02.05.2023	35	2%	4,2	66,80
<b>6</b>	260	67	02.05.2023	38	3%	11,9	85,00
<b>7</b>	90	36	02.05.2023	35	2%	11,1	70,00
<b>8</b>	120	36	03.05.2023	65	2%	14,5	87,60
<b>9</b>	290	84	06.05.2023	35	3%	3,0	99,00
<b>10</b>	950	130	06.05.2023	21	2%	11,5	70,70
<b>11</b>	138	81	07.05.2023	40	7%	16,2	150,00



**Задание 3.** Выполните фильтрацию и сортировку данных в таблице.

1. Перейдите на новый лист, нажав внизу Лист3.
2. На втором листе продублирована одна и та же таблица несколько раз (см. табл. 3). Работайте отдельно с каждой таблицей для создания фильтра, предварительно выделяя заголовки таблицы и выбирая команды **Данные** → **Автофильтр (Быстрый фильтр)**.

3. Настройте следующие фильтры и сортировку:

**Фильтр и сортировка 1.** При помощи стандартного фильтра найди гостиницы, в которых в номере два места и завтрак включён в стоимость. Отсортируйте данные по названию гостиниц в алфавитном порядке.

**Фильтр и сортировка 2.** При помощи стандартного фильтра найди гостиницы, в которых в стоимость номер не входит завтрак и ценой меньше 7000. Отсортируй данные по цене в порядке возрастания.

**Фильтр и сортировка 3.** При помощи стандартного фильтра найди гостиницы, которые начинаются на букву «А» или на букву «И». Отсортируй данные по цене в порядке убывания.

**Таблица 3** – Исходные данные для задания 3

	А	В	С	Д	Е	Ф
1	№ гостиницы	Название гостиницы	Количество мест в номере	Завтрак (входит / не входит в стоимость)	Число звёзд	Цена
2	1	Гостевой дом	3	не входит	2	3 100 руб.
3	2	Астория	2	входит	4	9 390 руб.
4	3	Ибис	3	не входит	3	6 790 руб.
5	4	Космос	2	входит	3	3 700 руб.
6	5	Рэдиссон	1	входит	5	5 700 руб.
7	6	Азимут	2	не входит	4	2 990 руб.
8	7	Мариотт	2	входит	5	8 190 руб.
9	8	Интурист	3	не входит	3	2 790 руб.
10	9	Хилтон	2	входит	5	11 990 руб.



**Задание 4.** Необходимо определить показатели для данных, которые соответствуют какому-то условию (например, для определения дней, месяцев, городов, предметов или групп). Такими показателями могут быть:

- количество значений;
- среднее арифметические значений;
- сумма значений;
- максимальное или минимальное.

В столбце А записан код округа, в котором учится студент; в столбце В – код фамилии студента; в столбце С – выбранный студентом предмет; в столбце Д – тестовый балл. Всего в электронную таблицу были занесены данные по 1000 студентам.


Сколько студентов, которые проходили тестирование по информатике, набрали более 600 баллов? Ответ запишите в ячейку Н2 таблицы.

Каков средний тестовый балл студентов, которые проходили тестирование по информатике? Ответ запишите в ячейку Н3 таблицы с точностью не менее двух знаков после запятой.

**Решение:** задания такого типа удобно решать, используя «фильтры». Фильтр – это предусмотренная в программе – электронной таблице возможность отбирать значения в столбцах, удовлетворяющие некоторому условию. В

качестве условия можно выбирать значения даты, населенных пунктов, классов, групп. Использование фильтров показано в задании 1.

1. Откройте Лист4.

2. Выделите заголовок таблицы A1:D1 и включите Автофильтр . Вид заголовка изменится (рис. 31). Небольшие треугольники в первой строке – это инструмент, позволяющий использовать фильтр в том или ином столбце.

	A	B	C	D
1	округ	фамилия	предмет	балл
2	C	Студент 1	физика	37
3	B	Студент 2	физкультура	37

Рис. 31. Автофильтр

3. Щёлкните мышью на треугольник столбца C (рис. 32). Видно, что в нижней части перечислены все значения из **столбца C** в алфавитном порядке без повторений. Если убрать «галочку» у каких-либо значений, то строки с такими значениями на листе отображаться не будут.

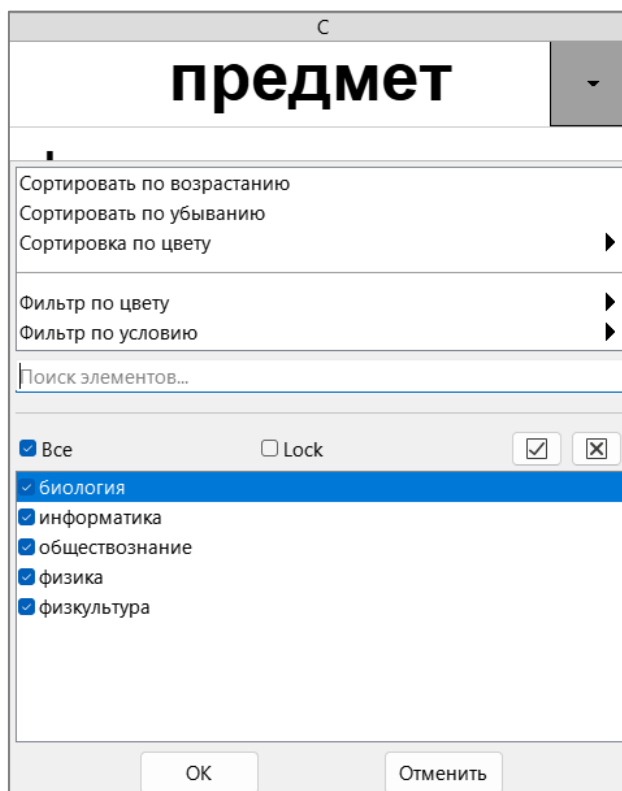


Рис. 32. Стандартный фильтр столбца C

4. Настроим фильтром так, чтобы остались только строки с предметом «Информатика» и определим количество этих строк. В раскрывшемся автофильтре столбца C, уберите «галочку» на варианте «**Все**» и поставьте галочку на строке «**информатика**», после чего нажать кнопку **ОК**.

В результате, останутся только строки с нужным предметом. Их

количество будет указано в нижней левой части окна программы: 72 of 1000 records found / 72 из 1000 найденных.

5. В решаемой нами задаче следует найти студентов, которые набрали по информатике более 600 баллов.

6. Используем стандартный фильтр в столбце D (**Фильтр по условию – Стандартный фильтр**). Создайте дополнительное условие (рис. 33).

Операция	Имя поля	Условие	Значение	
	предмет	=	информати	X
И	балл	>	600	X
	- нет -	=		X
	- нет -	=		X

Параметры

Справка Очистить OK Отменить

**Рис. 33.** Создание сложного условия

7. После применения второго фильтра искомое значение также появится в нижней левой части окна программы: *selected: 36 rows, 4 columns* / выделено: 36 строк, 4 столбца, но **нужно учитывать, что одна строка – заголовок таблицы**. В итоге всего по данному условию будет отобрано 35 студентов.

8. Ответ запишите в ячейку H2 таблицы.

9. Отмените стандартный фильтр (**Данные – Еще фильтры – Удалить фильтр**).

10. Определим средний тестовый балл студентов, которые проходили тестирование по информатике.

11. Выберите команду **Данные – Промежуточные итоги** (рис. 34).

1-я группа 2-я группа 3-я группа Параметры

Группировать по: предмет  Выбрать все столбцы

Вычислить промежуточные итоги для:

- округ
- фамилия
- предмет
- балл

Использовать функцию:

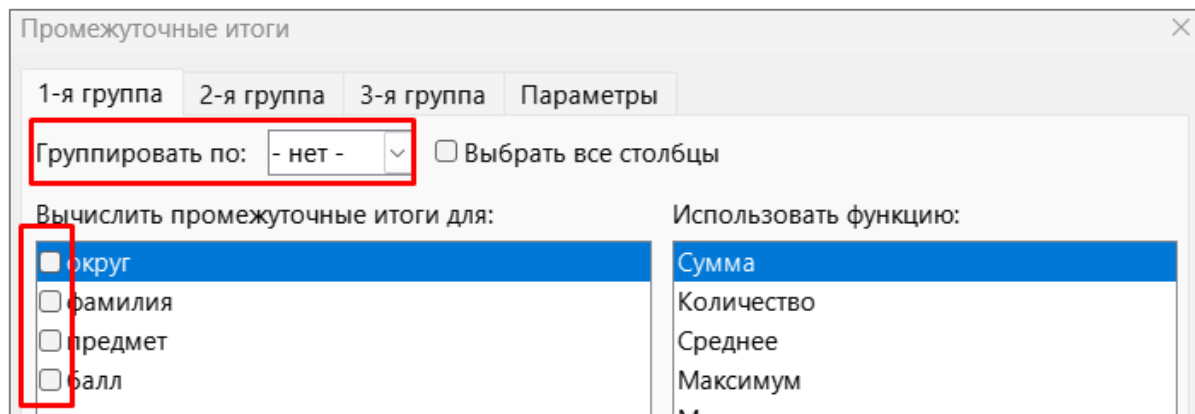
- Сумма
- Количество
- Среднее**
- Максимум
- Минимум
- Произведение
- Количество (только числа)

**Рис. 34.** Промежуточные итоги

12. Будут сформированы итоги по каждому предмету. Найдите итоги по предмету информатика и запишите их в ячейку **H3** таблицы с точностью **не менее**

двух знаков после запятой.

13. Отмените группировку промежуточных итогов. Для этого выберите команду **Данные – Промежуточные итоги** и отмените всё (см. рис. 35).



**Рис. 35.** Отмена промежуточных итогов

## Практическая работа №37. Математические, статистические и логические функции. Обработка большого массива данных

**Цель:** освоение методов работы с математическими, статистическими и логическими функциями в табличном процессоре для обработки и анализа больших массивов данных.

### Теоретические сведения

**Функция** (от лат. *functio* – исполнение, осуществление), одно из основных понятий математики, означающее зависимость одних переменных величин от других. Слово «величина» в этом определении понимается в самом широком смысле: это может быть именованное число, отвлечённое число (действительное или комплексное), несколько чисел (т. е. точка пространства) и вообще элемент любого множества.

Математические, статистические и логические функции позволяют эффективно анализировать, моделировать и контролировать большие массивы данных в разных профессиональных сферах. Они помогают автоматизировать сложные расчёты, находить закономерности и повышать точность принимаемых решений.

Таблица 1 – Основные математические функции

Функция	Описание
=СУММ(число1; число2; ...)	Добавляет значения. Вы можете складывать отдельные значения, диапазоны ячеек, ссылки на ячейки или данные всех этих трех видов.
=СУММЕСЛИ (диапазон; условие; диапазон суммирования)	Используется, если необходимо просуммировать значения диапазон, соответствующие указанному критерию. Предположим, например, что в столбце с числами необходимо просуммировать только значения, превышающие 5. Для этого можно использовать следующую формулу: =СУММЕСЛИ(B2:B25;"> 5").
=СУММЕСЛИМН (диапазон суммирования; диапазон условия1; условие1; диапазон условия2; условие2; ...)	Суммирует все аргументы, удовлетворяющие нескольким условиям.
=ПРОИЗВЕД(число1; число2;...)	Перемножает все числа, переданные как аргументы, и возвращает произведение. Например, если в ячейках A1 и A2 содержатся числа, формула =ПРОИЗВЕД(A1;A2) возвращает произведение этих чисел. Эту операцию также можно выполнить с помощью математического оператора умножения (*). Пример: =A1 * A2.
=ОКРУГЛ(число; число разрядов)	округляет число до указанного количества дробных разрядов. Например, если в ячейке A1 находится число 23,7825, и его необходимо округлить до двух дробных разрядов, можно использовать следующую формулу: =ОКРУГЛ(A1;2). В результате получается число 23,78.

= <b>ЦЕЛОЕ</b> (число)	Округляет число до ближайшего меньшего целого. Например, если использовать формулу =ЦЕЛОЕ(8,9), то в результате получается число 8.
= <b>ОСТАТ</b> (число; делитель)	Возвращает остаток от деления аргумента "число" на значение аргумента "делитель". Результат имеет тот же знак, что и делитель. Например, если использовать формулу =ОСТАТ(3; 2), то в результате получается число 1.
= <b>КОРЕНЬ</b> (число)	Возвращает положительное значение квадратного корня. =КОРЕНЬ(16) даёт результат 4.
= <b>ФАКТР</b> (число)	Возвращает факториал числа. Факториал числа – это значение, равное $1*2*3*...*$ число. =ФАКТР(5) даёт результат 120.
= <b>СЛУЧМЕЖДУ</b> (нижняя граница; верхняя граница)	Возвращает случайное число в интервале между двумя заданными числами.

**Таблица 2 – Основные статистические функции**

<b>Функция</b>	<b>Описание</b>
= <b>СРЗНАЧ</b> (число1; число2; ...)	Возвращает среднее (среднее арифметическое) аргументов. Например, если диапазон A1:A20 содержит числа, формула =СРЗНАЧ (A1:A20) возвращает среднее значение этих чисел.
= <b>МАКС</b> (число1; число2; ...)	Возвращает наибольшее значение из набора значений.
= <b>МИН</b> (число1; число2; ...)	Возвращает наименьшее значение в списке аргументов.
= <b>СЧЁТ</b> (значение1; значение2; ...)	Функция СЧЁТ используется для определения количества числовых ячеек в диапазонах и массивах чисел. Например, для вычисления количества чисел в диапазоне A1:A20 можно ввести следующую формулу: =СЧЁТ(A1:A20). Если в данном примере пять ячеек из диапазона содержат числа, то результатом будет значение 5.
= <b>СЧЁТЕСЛИ</b> (диапазон; критерий)	Посчитывает количество ячеек, отвечающих определенному условию (например, число клиентов в списке из определенного города). Самая простая функция СЧЁТЕСЛИ означает следующее: =СЧЁТЕСЛИ(где нужно искать; что нужно найти).
= <b>СЧЁТЕСЛИМН</b> (диапазон условия1; условие1; диапазон условия2; условие2; ...)	Применяет критерии к ячейкам в нескольких диапазонах и вычисляет количество соответствий всем критериям.
= <b>СЧЁТЗ</b> (значение1; значение2;...)	Подсчитывает количество заполненных ячеек в диапазон.

**Таблица 3 – Основные логические функции**

<b>Функция</b>	<b>Описание</b>
= <b>ЕСЛИ</b> (лог выражение; значение если истина; значение если ложь)	Позволяет выполнять логические сравнения значений и ожидаемых результатов. У функции возможны два результата. Первый результат возвращается в случае, если сравнение истинно, второй – если сравнение ложно.

	Например, функция =ЕСЛИ(С2="Да";1;2) означает следующее: ЕСЛИ(С2="Да", то вернуть 1, в противном случае вернуть 2).
=ЕСЛИМН (условие1; значение1; условие2; значение2; условие3; значение3)	Проверяет соответствие одному или нескольким условиям.
= ИЛИ(логическое значение1; логическое значение2; ...)	Функция ИЛИ возвращает значение ИСТИНА, если в результате вычисления хотя бы одного из ее аргументов получается значение ИСТИНА, и значение ЛОЖЬ, если в результате вычисления всех ее аргументов получается значение ЛОЖЬ.
= И(логическое значение1; логическое значение2; ...)	Функция И возвращает значение ИСТИНА, если в результате вычисления всех аргументов получается значение ИСТИНА, и значение ЛОЖЬ, если вычисление хотя бы одного из аргументов дает значение ЛОЖЬ.
=НЕ(логическое значение)	Функция НЕ меняет значение своего аргумента на обратное.

## Ход выполнения практической работы



**Задание 1.** В таблицу внесены данные об электронной очереди в больницу. В столбце А записан кабинет приёма врача; в столбце В – фамилия и имя пациента; в столбце С – инициалы принимающего врача; в столбце D – номер пациента в очереди. **Сколько пациентов было принято в 5 кабинет?** Ответ запишите в ячейку J1. **Каков средний номер пациента, принятого врачом ВЛЕ?** Ответ запишите в ячейку K1. Используйте функции СЧЁТЕСЛИ и СУММЕСЛИ.

1. На Рабочем столе в папке своего преподавателя найдите файл **ПР37\_Заготовка**. Откройте файл при помощи программы **LibreOffice Calc**.

2. Сохраните файл в свою рабочую папку с именем **ПР37\_Фамилия\_Группа** (вкладка **Файл** → **Сохранить как...** → **Рабочий стол** → **Моя рабочая папка**).

3. Откройте Лист1. Переименуйте его в **Задание 1**.

4. Для того, чтобы посчитать сколько пациентов было принято в 5 кабинет, в ячейке **J1** воспользуемся функцией СЧЁТЕСЛИ:

=СЧЁТЕСЛИ(диапазон; критерий)

Используем диапазон столбца – **Кабинет**, критерий – **номер кабинета**:

=СЧЁТЕСЛИ(A2:A501; 5)

5. Для подсчёта среднего номер пациента, принятого врачом ВЛЕ, нужно знать сколько раз врач ВЛЕ принимал. Найдите это значение используя функцию СЧЁТЕСЛИ. Текстовые данные в формулах нужно указывать в двойных кавычках «"»». Используйте для этого ячейку **J2**.

6. В ячейке **J3** найдём сумму всех номеров пациентов, которых принял врач ВЛЕ. Для этого воспользуемся функцией СУММЕСЛИ:

=СУММЕСЛИ (диапазон; условие; [диапазон суммирования])

Подсказка: диапазон – **столбец С** (Инициалы врача), **условие "ВЛЕ"**, диапазон суммирования – **столбец D** (Номер в очереди).

7. Для того, чтобы найти средний номер пациента, принятого врачом ВЛЕ, нужно сумму всех номеров пациентов, которых принял врач ВЛЕ разделить на количество принятых пациентов врачом ВЛЕ. Внесите эту формулу в ячейку K1.

8. Выполните текущее сохранение документа.

*Используя функции из таблиц 1, 2 и 3, выполни вычисления по данным таблицы листа 1.*

– В ячейке **N1** определите средний номер в очереди среди всех пациентов.  
– В ячейке **N2** определите максимальный номер в очереди среди всех пациентов.

– В ячейке **N3** определите минимальный номер в очереди среди всех пациентов.

– В ячейке **N4** найдите общее количество пациентов, используя функцию.

– В ячейке **N5** определите, сколько пациентов приняли в кабинете №1.

– В ячейке **N6** определите, сколько пациентов обслуживает врач с инициалами «ДЛЯ».

– В ячейке **N7** подчитайте пациентов, которых приняли в кабинетах №1 и №5.

– В ячейке **N8** подчитайте, сколько заполненных записей есть в столбце с пациентами.

– В ячейке **N9** подчитайте сумму номеров в очереди для кабинета №2.

– В ячейке **N10** подчитайте сумму номеров в очереди для кабинетов №1 и №4.

– В ячейке **N11**, **используя функцию**, умножьте между собой номера в очереди первых трёх пациентов.

– В ячейке **N12** найдите целую часть из среднего номера в очереди.

– В ячейке **N13** определите остаток от деления общего количества номеров в очереди на 5.

– В ячейке **N14** найдите квадратный корень от минимального номера в очереди.

– В ячейке **N15** сгенерируйте случайное число от минимального до максимального номера в очереди.

– В ячейке **N16** вычислите факториал минимального номера в очереди.

9. Выполните текущее сохранение документа.



**Задание 2.** В таблицу внесены данные о тестировании студентов. В столбце А записан номер студента; в столбце В – район, в котором проживает студент; в столбце С и D – количество баллов, набранных студентом за тест по математике и русскому языку соответственно. **Найдите наибольшую сумму баллов среди студентов, проживающих в Космическом районе. Ответ запишите в ячейку Н1.**

1. Откройте Лист2. Переименуйте его в **Задание 2**.

2. Для нахождения наибольшей суммы баллов среди студентов, проживающих в Космическом районе, воспользуемся функцией ЕСЛИ:

=ЕСЛИ(лог выражение; значение если истина; значение если ложь)

**Логическое выражение** – проживающие в Космическом районе. Вспомним, что текстовые данные в формулах нужно указывать в двойных кавычках «"»». Формула будет анализировать и сравнивать каждую отдельную ячейку, поэтому логическое выражение будет выглядеть так: **В2="Космический"**.

**Значение если истина** – сумма баллов тестирования по математике и русскому языку: **С2+D2**.

**Значение если ложь** запишем 0.

Собираем формулу в ячейку E2:

=ЕСЛИ(В2="Космический";С2+D2;0)

3. Такая формула будет считать сумму баллов только в случае, если студент из Космического района. Мы посчитали значение только для первого студента. Щёлкните по ячейке **E2**, в которой набрана формула. Двойным щелчком нажмите на маркер автозаполнения, и он автоматически просчитает значения по каждому студенту.

4. Самостоятельно посчитайте наибольшую сумму баллов. Ответ запишите в ячейку **Н1**.

5. Подсчитаем количество учеников из района "Яснево" с баллом по математике выше 70. Для этого воспользуемся функцией СЧЁТЕСЛИМН:

=СЧЁТЕСЛИМН (диапазон условия1; условие1; диапазон условия2; условие2)

**Первое условие** – район **Яснево**, **второе условие** – балл по математике **выше 70**.

Собираем формулу в ячейку **Н2**:

=СЧЁТЕСЛИМН(В2:В1020; "Яснево"; С2:С1024; ">70")

6. Такая формула будет считать только тех учеников, у которых выполняется два условия. Не нужно протягивать маркер автозаполнения, так как мы считаем всех студентов сразу.

7. Определим, есть ли у ученика хотя бы один выше 90. Для этого воспользуемся функцией ЕСЛИ и ИЛИ:

– нажмите в пустую ячейку в **F2**, введи знак "=";

- используем функцию ЕСЛИ: пишем **ЕСЛИ**( это означает: "Если выполняется условие, то выводим один результат, иначе – другой");
- добавляем проверку с ИЛИ: пишем **ИЛИ(C2>90; D2>90)**. Это значит: "Проверяем, есть ли в C2 или D2 число больше 90".
- указываем, что писать, если условие выполняется. Если хотя бы одна оценка больше 90, напишем **"Есть высокий балл"**;
- указываем, что писать, если условие НЕ выполняется: если нет высоких баллов, напишем **"Нет"**.
- закрываем скобку и нажимаем Enter. Полная формула:  
=ЕСЛИ(ИЛИ(C2>90; B2>90); "Есть высокий балл"; "Нет")

8. Выполните текущее сохранение документа.

*Используя функции из таблиц 1, 2 и 3, выполни вычисления по данным таблицы листа 2.*

- В ячейке **I2** найдите средний балл по математике для всех учеников.
- В ячейке **J2** определите максимальный балл по русскому языку.
- В ячейке **K2** найдите минимальный балл по математике.
- В ячейке **L2** подсчитайте количество учеников, имеющих оценки по математике.
- В ячейке **M2** подсчитайте количество учеников из района "Космический".
- В ячейке **N2** напишите условие – если ученик сдал тест по математике, вывести "Сдал", иначе – "Не сдал". Выполните автозаполнение по всем ученикам (проходной балл 60).
- В ячейке **O2** напишите условие – если ученик сдал тест по русскому языку, вывести "Сдал", иначе – "Не сдал". Выполните автозаполнение по всем ученикам (проходной балл 70).
- В ячейке **P2** напишите условие – если ученик набрал по математике >80 баллов, то он "Отличник", иначе – "Не отличник". Выполните автозаполнение по всем ученикам.
- В ячейке **Q2** напишите условие – если ученик набрал по математике <50 баллов, то он должен пересдать тест ("Пересдача"), иначе – "Нет". Выполните автозаполнение по всем ученикам.
- В ячейке **R2** напишите условие – если ученик из района "Космический", вывести "Городской", иначе – "Пригородный". Выполните автозаполнение по всем ученикам.
- В ячейке **S2** напишите условие, проверяющее является ли ученик отличником (по обоим предметам > 85). Выполните автозаполнение по всем ученикам.

9. Выполните текущее сохранение документа.



### Задание 3. Изучите работу функции ПРЕОБР и выполните расчёты.

Функция ПРЕОБР преобразует число из одной системы мер в другую. Например, с помощью функции ПРЕОБР можно перевести таблицу расстояний в милях в таблицу расстояний в километрах.

Синтаксис:

**ПРЕОБР**(число; исходная единица измерения; конечная единица измерения)

где **число** – значение в исходных единицах измерения, которое нужно преобразовать, может быть задано в виде числа или ссылки на ячейку; **исходная единица измерения** – единицы измерения аргумента "число"; **конечная единица измерения** – единицы измерения результата.

Исходную единицу измерения и конечную единицу измерения необходимо вводить в кавычках.

1. Перейдите на Лист3. Переименуйте его в **Задание 3**. Создайте таблицу в соответствии с таблицей ниже.

**Таблица 4** – Исходные данные для задания 3

	<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>E</b>
<b>1</b>	<b>№ п/п</b>	<b>Фамилия</b>	<b>Рост, см</b>	<b>Рост, дюйм</b>	<b>Рост, мм</b>
<b>2</b>	1	Студент 1	152	#формула	#формула
<b>3</b>	2	Студент 2	182	#формула	#формула
<b>4</b>	3	Студент 3	191	#формула	#формула
<b>5</b>	4	Студент 4	184	#формула	#формула
<b>6</b>	5	Студент 5	158	#формула	#формула
<b>7</b>	6	Студент 6	194	#формула	#формула
<b>8</b>	7	Студент 7	187	#формула	#формула
<b>9</b>	8	Студент 8	179	#формула	#формула

2. Используя функцию преобразования переведите рост студентов из сантиметров в дюймы и миллиметры. Сантиметры в формуле обозначьте как **"cm"**, миллиметры обозначьте как **"mm"**. Используйте обозначения единиц измерения [в статье](#).

3. Выполните текущее сохранение документа.

## Практическая работа №38. Финансовые функции

**Цель работы:** научиться применять финансовые функции в электронных таблицах для расчётов, связанных с кредитами, депозитами и платежами. Освоить базовые функции, такие как ПЛТ, ПС, ОСПЛТ, и научиться использовать их для решения простых финансовых задач.

### Ход выполнения работы

Наиболее часто используемая из финансовых функций – функция ПЛТ, с помощью которой можно определить выплаты по месяцам при взятии кредита в банке.

**Функция ПЛТ (Платёж)** в электронной таблице используется для расчёта ежемесячного платежа по кредиту или займу.

*Синтаксис:*

$$=ПЛТ(\text{ставка; кол-во периодов; сумма кредита})$$

где:

- **ставка** – процентная ставка за 1 период (например, за месяц);
- **кол-во периодов** – общее число платежей (например, месяцев);
- **сумма кредита** – сумма займа (пишется со знаком минус, так как это расход).

**Пример 1.** Кредит на 100 000 руб. на 3 года под 12% годовых.

Условия: сумма кредита – 100 000 руб.; срок – 3 года (36 месяцев); процентная ставка – 12% годовых (то есть  $12\% / 12 = 1\%$  в месяц).

*Формула:*

$$=ПЛТ(12\%/12; 36; -100000)$$

*Что нужно знать?*

- Если ставка годовая, делим её на 12.
- Если ставка ежемесячная, оставляем как есть.
- Сумма кредита с минусом: в функции ПЛТ кредит указывается отрицательным числом (чтобы показать расход).
- Результат – это фиксированный платеж. Функция ПЛТ даёт аннуитетный платёж (он одинаковый каждый месяц).

**Пример 2.** Ипотека на 3 000 000 руб. на 20 лет под 8% годовых.

*Формула:*

$$=ПЛТ(8\%/12; 20*12; -3000000)$$

*Результат:* 25 093,41 руб. в месяц.

Функция ПЛТ помогает быстро узнать сумму ежемесячного платежа по кредиту. Она полезна при расчёте ипотеки, автокредита или других займов.

10. На рабочем столе в папке Программы найдите и откройте LibreOffice Calc и сохраните файл в свою рабочую папку с именем **ПР38\_Фамилия\_Группа** (вкладка **Файл** → **Сохранить как...** → **Рабочий стол** → **Моя рабочая папка**).

11. Откройте Лист1. Переименуйте его в **ПЛТ**.

12. Задания 1–7 выполните в листе **ПЛТ**. Оформите исходные данные для каждой таблицы и решение. Пример см. табл. ниже.

**Таблица 1** – Пример оформления заданий

	<b>А</b>	<b>В</b>
<b>1</b>	<b>Задание №1</b>	
<b>2</b>	Срок	3 года
<b>3</b>	% годовых	10%
<b>4</b>	Сумма	200 000 руб.
<b>5</b>	Ежемесячный платёж	#формула

**Задание 1.** Рассчитать ежемесячный платёж по кредиту, если вы взяли кредит на 3 года под 10% годовых на сумму 200 000 руб.

**Задание 2.** Вы взяли ипотеку на 15 лет под 6% годовых на сумму 3 000 000 руб. Рассчитайте ежемесячный платеж.

**Задание 3.** Строительная компания взяла кредит на 5 лет под 10% годовых для покупки экскаватора стоимостью 2 500 000 руб. Какой будет ежемесячный платёж?

**Задание 4.** Компания планирует купить офис стоимостью 8 000 000 руб. и взять ипотеку на 15 лет под 7% годовых. Какой будет ежемесячный платёж?

**Задание 5.** Фирма взяла заём на 2 года под 12% годовых для закупки кирпича и цемента на 500 000 руб. Какой будет ежемесячный платёж?

**Задание 6.** Девелоперская компания<sup>1</sup> берёт кредит на 10 лет под 9% годовых для строительства многоквартирного дома. Сумма кредита – 50 000 000 руб. Какой будет ежемесячный платёж?

**Задание 7.** Компания арендует башенный кран с возможностью выкупа через 4 года. Стоимость крана – 3 000 000 руб., ставка 11% годовых. Какой будет ежемесячный платёж?

13. Выполните текущее сохранение.

14. Откройте Лист2. Переименуйте его в **ПС**.

15. Задания 8–10 выполните в листе **ПС**. Оформите исходные данные для каждой таблицы и решение.

**Функция ПС** (Приведённая стоимость) помогает узнать, сколько нужно вложить сейчас, чтобы получить нужную сумму в будущем.

*Синтаксис:*

---

<sup>1</sup> Компания, которая одновременно исполняет функции инвестора, проектного бюро, агента по торговле недвижимостью и застройщика.

=ПС(ставка; кол-во периодов; платеж; будущая стоимость; [тип])

где:

- **ставка** – процентная ставка за период (месяц, год и т. д.);
- **кол-во периодов** – сколько раз будет начисляться процент (месяцев, лет и т. д.);
- **платёж** – регулярные выплаты (если их нет, указываем 0);
- **будущая стоимость** – сумма, которую хотим получить в будущем (пишем с минусом, так как это доход);
- **тип** – 0 (по умолчанию – платежи в конце периода) или 1 (в начале периода).

*Функция ПС отвечает на вопрос:* сколько нужно вложить сейчас, чтобы через  $N$  лет получить нужную сумму?

Если банк даёт 6% годовых, то 100 000 руб., вложенные сегодня, через год станут 106 000 руб. Функция ПС поможет рассчитать, сколько нужно вложить, если у нас другие суммы, сроки и ставки.

**Пример 3.** Вклад на будущее обучение. Вы хотите накопить 500 000 руб. для оплаты учёбы ребёнка через 5 лет. Банк предлагает 8% годовых. Сколько нужно вложить сегодня?

*Формула:*

$$=ПС(8\%/12; 5*12; 0; -500000)$$

Ответ: 338 739,68 руб. Если вложить 338 739,68 руб. сегодня под 8% годовых, через 5 лет получится 500 000 руб.

**Задание 8.** Вы планируете через 10 лет купить квартиру за 10 000 000 руб. Банк предлагает 6% годовых. Сколько нужно вложить сейчас?

**Задание 9.** Капитал для выхода на пенсию. Вы хотите через 20 лет накопить 5 000 000 руб. на пенсию. Доходность ваших инвестиций – 10% годовых. Сколько нужно вложить сейчас?

**Задание 10.** Строительная компания хочет через 3 года купить экскаватор за 2 000 000 руб. Можно инвестировать деньги под 7% годовых. Сколько нужно вложить сейчас?

16. Выполните текущее сохранение.

## Практическая работа №39. Текстовые функции

**Цель:** познакомиться с текстовыми функциями в электронных таблицах, научиться их применять для обработки строковых данных, а также улучшить навыки работы с данными с помощью текстовых функций для анализа, изменения и форматирования текста.

### Ход выполнения практической работы

**Текст** – это любая совокупность буквенно-цифровых символов, которые не являются числовым значением, значением даты или времени или формулой.

Текстовые функции предназначены для работы с текстовыми строками: извлечения, замены, объединения, поиска и обработки символов в тексте. Ниже представлены наиболее популярные текстовых функции (см. таблицу ниже).

**Таблица 1 – Основные текстовые функции**


Функция	Описание
<b>ДЛСТР</b> (текст)	Возвращает количество символов в строке.
<b>ЗАМЕНИТЬ</b> (стар текст; начальная позиция; число знаков; нов текст)	Заменяет указанное число символов текстовой строки другой текстовой строкой.
<b>КОД</b> (текст)	Возвращает код ANSI для первого символа в тексте
<b>ЛЕВ</b> (текст; количество знаков)	Извлекает указанное количество символов с начала строки.
<b>ПОВТОР</b> (текст; число повторений)	Повторяет текст заданное число повторений
<b>ПОДСТАВИТЬ</b> (текст; стар текст; нов текст; номер вхождения)	Подставляет значение аргумента "нов текст" вместо значения аргумента "стар текст" в текстовой строке. Функция <b>ПОДСТАВИТЬ</b> используется, когда нужно заменить определенный текст в текстовой строке.
<b>ПРАВ</b> (текст; количество знаков)	Извлекает указанное количество символов с конца строки.
<b>ПРОПИСН</b> (текст)	Делает все буквы в тексте прописными.
<b>ПРОПНАЧ</b> (текст)	Первая буква в строке текста и все первые буквы, следующие за знаками, отличными от букв, преобразуются в прописные (верхний регистр). Все прочие буквы в тексте преобразуются в строчные (нижний регистр).
<b>ПСТР</b> (текст; начальная позиция; число знаков)	Извлекает символы из строки, начиная с указанной позиции.
<b>СЖПРОБЕЛЫ</b> (текст)	Удаляет из текста все пробелы, за исключением одиночных пробелов между словами. Функция <b>СЖПРОБЕЛЫ</b> используется для обработки текстов, полученных из других прикладных программ, если эти тексты могут содержать лишние пробелы.
<b>СИМВОЛ</b> (число)	Возвращает символ, который соответствует коду ANSI, указанному в аргументе число

<b>СТРОЧН</b> (текст)	Преобразует текст в нижний регистр (строчные буквы)
<b>СЦЕПИТЬ</b> (текст1; текст2; ...)	Используйте функцию СЦЕПИТЬ, относящуюся к текстовым функциям, для объединения двух или нескольких текстовых строк в одну.
<b>ТЕКСТ</b> (текст1; текст2; ...)	Преобразует числовое значение в текстовый формат.

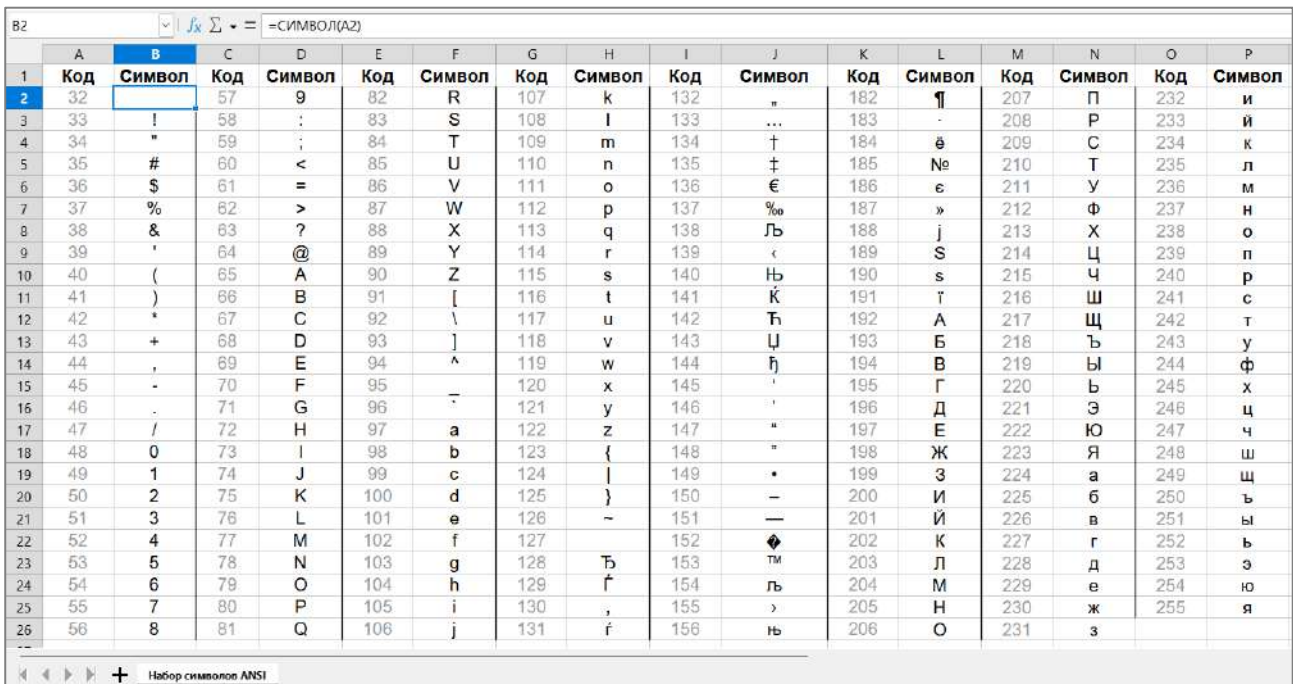
Каждый символ, который можно отобразить на экране, имеет свой числовой код. Например, код прописной буквы А в латинской раскладке 65, код амперсанда (&) – 38. Эти коды присвоены не только буквенно-цифровым символам, которые можно ввести с клавиатуры, но и «дополнительным» символам, которые можно отобразить путем ввода соответствующего кода.

Функция **СИМВОЛ()** позволяет определить символ, представленный в виде кода ANSI, с помощью функции:

=**СИМВОЛ**(число)

 **Задание 1.** Создайте таблицу данных и выполните задание с использованием текстовых функций.

1. На рабочем столе в папке Программы найдите и откройте LibreOffice Calc и сохраните файл в свою рабочую папку с именем **ПР39\_Фамилия\_Группа** (вкладка **Файл** → **Сохранить как...** → **Рабочий стол** → **Моя рабочая папка**).
2. Переименуйте Лист1 в **Набор символов ANSI**.
3. Создайте таблицу в соответствии с рисунком, представленным ниже.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	Код	Символ	Код	Символ	Код	Символ	Код	Символ	Код	Символ	Код	Символ	Код	Символ	Код	Символ
2	32		57	9	82	R	107	k	132	"	182	¶	207	П	232	и
3	33	!	58	:	83	S	108	l	133	...	183	·	208	Р	233	й
4	34	"	59	;	84	T	109	m	134	†	184	•	209	С	234	к
5	35	#	60	<	85	U	110	n	135	‡	185	№	210	Т	235	л
6	36	\$	61	=	86	V	111	o	136	€	186	€	211	У	236	м
7	37	%	62	>	87	W	112	p	137	‰	187	»	212	Ф	237	н
8	38	&	63	?	88	X	113	q	138	Љ	188	ј	213	Х	238	о
9	39	'	64	@	89	Y	114	r	139	Њ	189	ѕ	214	Ц	239	п
10	40	(	65	A	90	Z	115	s	140	Ќ	190	š	215	Ч	240	р
11	41	)	66	B	91	[	116	t	141	ќ	191	ţ	216	Ш	241	с
12	42	*	67	C	92	\	117	u	142	ќ	192	а	217	Щ	242	т
13	43	+	68	D	93	]	118	v	143	џ	193	б	218	Ъ	243	у
14	44	,	69	E	94	^	119	w	144	ђ	194	в	219	Ы	244	ф
15	45	-	70	F	95	_	120	x	145	'	195	Г	220	Ь	245	х
16	46	.	71	G	96	`	121	y	146	•	196	Д	221	Э	246	ц
17	47	/	72	H	97	a	122	z	147	"	197	Е	222	Ю	247	ч
18	48	0	73	I	98	b	123	{	148	"	198	Ж	223	Я	248	ш
19	49	1	74	J	99	c	124		149	•	199	З	224	а	249	щ
20	50	2	75	K	100	d	125	}	150	—	200	И	225	б	250	ъ
21	51	3	76	L	101	e	126	~	151	—	201	Й	226	в	251	ы
22	52	4	77	M	102	f	127		152	◆	202	К	227	г	252	ь
23	53	5	78	N	103	g	128	Ђ	153	™	203	Л	228	д	253	э
24	54	6	79	O	104	h	129	Ѓ	154	Љ	204	М	229	е	254	ю
25	55	7	80	P	105	i	130	,	155	›	205	Н	230	ж	255	я
26	56	8	81	Q	106	j	131	ѓ	156	њ	206	О	231	з		

**Рис. 36.** В этой таблице используется функция **СИМВОЛ()**, которая отображает все печатные символы набора ANSI

**Примечание:** на рисунке выше показана таблица, которая содержит некоторый набор символов ANSI (за исключением 31 непечатаемого символа, кроме того, код ANSI 32 обозначает символ пробела). В каждом случае символ был отображен с помощью применения функции **СИМВОЛ()** к значению в ячейке слева. Сам символ, который отображается из кода ANSI, зависит от шрифта в ячейке. Символы, показанные на рис. 36, – это символы, которые отображаются для нормальных текстовых шрифтов, например Arial. Если же вы примените к таблице шрифт типа Symbol или Wingdings, то увидите другой набор символов.

4. Выполните текущее сохранение документа.

Функция **КОДСИМВ()** является обратной функцией для функции **СИМВОЛ()**. Таким образом, функция **КОДСИМВ()** получает в качестве аргумента текстовый символ и возвращает значение кода ANSI:

Функция **СТРОЧН()** преобразует заданную строку в символы нижнего регистра (строчные буквы).

Функция **ПРОПИСН()** преобразует указанную строку в символы верхнего регистра (прописные буквы).

Функция **ПРОПНАЧ ()** преобразует заданную строку таким образом, что первый символ строки имеет верхний регистр, а все остальные символы заданы в нижнем регистре.



**Задание 2.** В таблице содержатся данные о студентах, но в столбце "ФИО" встречаются разные регистры букв. Используйте функции преобразования регистра, чтобы: привести все фамилии в верхний регистр; привести все имена в формат с заглавной первой буквой; привести все отчества в нижний регистр.

5. Перейдите на Лист2 и переименуйте его в **Задание 2**.

6. Создайте таблицу данных, в соответствии с таблицей ниже.

**Таблица 2** –Таблица данных для задания №2

	А	В	С	Д	Е
1	№	Фамилия	Имя	Отчество	Группа
2	1	иВанов	ПЕТР	АЛЕКСЕЕВИЧ	ИС-21
3	2	Петров	николай	сергеевич	ИС-21
4	3	СИДОРОВ	АНДРЕЙ	Васильевич	ИС-21
5	4	КУЗНЕЦОВ	Дмитрий	ПАВЛОВИЧ	ИС-21
6	5	Смирнов	ВАСИЛИЙ	ИВАНОВИЧ	ИС-22
7	6	ФЁДОРОВ	евгений	МАКСИМОВИЧ	ИС-22
8	7	БЕЛОВ	СЕРГЕЙ	Дмитриевич	ИС-22

7. В Столбце **F** приведите фамилии в верхний регистр используя текстовую функцию.

8. В Столбце **G** и **H** соответственно приведите имя и отчество к формату «Первая буква заглавная», используя текстовую функцию.

9. В Столбце **I** приведите группу в нижний регистр используя текстовую функцию.

10. Выполните текущее сохранение документа.



**Задание 3.** Создайте таблицу данных и выполните задание с использованием текстовых функций.

11. Перейдите на Лист3 и переименуйте его в **Задание 3**.

12. Создайте таблицу данных, в соответствии с таблицей ниже.

**Таблица 3** – Таблица данных для задания 3

	<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>
<b>1</b>	<b>ФИО</b>	<b>Должность</b>	<b>Телефон</b>	<b>Адрес</b>
<b>2</b>	Иванов И. И.	Менеджер продаж	89123456789	Москва, ул. Ленина, 1
<b>3</b>	Петров П. П.	Директор	89234567890	Санкт-Петербург, ул. Невский, 10
<b>4</b>	Сидоров С. С.	Разработчик	89345678901	Казань, ул. Баумана, 5
<b>5</b>	Козлова К. К.	Бухгалтер	89456789012	Екатеринбург, ул. Чапаева, 15
<b>6</b>	Алексеева А. А.	HR-менеджер	89567890123	Нижний Новгород, ул. Горького, 8

13. Используя функцию СЖПРОБЕЛЫ, уберите лишние пробелы в строках из столбца ФИО в столбце **E**.

14. Используя функцию ЛЕВ, извлеките первые 6 символов из столбца «Телефон» для получения кода города в столбце **F**.

15. Используя функцию ПРАВ, извлеките последние 4 символа из столбца «Телефон» для получения номера абонента в столбце **G**.

16. Используя функцию ЗАМЕНИТЬ, замените в столбце «Адрес» слово «ул.» на «улица» в столбце **H**.

17. Используя функцию СЦЕПИТЬ, объедините столбцы «ФИО» и «Должность» в одну строку через запятую в столбце **I**.

18. Используя функцию ДЛСТР, определите количество символов в строках столбца «Адрес» в столбце **J**.

19. Преобразуйте все буквы в столбце "ФИО" в верхний регистр в столбце **K**.

20. Выполните текущее сохранение документа.



**Задание 4.** Изучить работу функции ПОВТОР() и выполнить задание.

Более популярное применение функции **ПОВТОР()** это создание графиков из текстовых символов. В этом случае в качестве аргумента **число повторений** для функции **ПОВТОР()** используется число из ячейки, и путем повторения определенного символа это число отображается «графически».


Простой пример – самая стандартная гистограмма, отображающая частоту для определенной переменной. На рис. 37 показана текстовая гистограмма; в столбце **А** указаны интервалы, а в столбце **В** соответствующая им частота. Функция **ПОВТОР()** создает гистограмму в столбце **С**, повторяя символ вертикальной черты (|) в соответствии со значением частоты, как в данной формуле: **=ПОВТОР("|"; В4)**.

21. Перейдите на Лист4 и переименуйте его в **Задание 4**.
22. Создайте таблицу данных, в соответствии с рисунком ниже

C4			
	A	B	C
1	Результаты голосования — респонденты		
2			
3	<b>Женщины</b>	<b>Респонденты</b>	<b>Гистограмма</b>
4	18-34	75	
5	35-49	83	
6	50-64	79	
7	>65	55	
8			
9	<b>Мужчины</b>		
10	18-34	70	
11	35-49	73	
12	50-64	82	
13	>65	60	

**Рис. 37.** Функция **ПОВТОР()** для создания гистограммы

23. Выполните текущее сохранение документа.

 **Задание 5.** В таблице содержатся разрозненные данные о студентах: их фамилии, инициалы, номера групп, а также успеваемость. Требуется с помощью функции **СЦЕПИТЬ()** создать сводную строку с полной информацией о каждом студенте в следующем формате:

"Студент Фамилия И.О. (группа: XX-XX) имеет средний балл: X.XX."

24. Перейдите на Лист5 и переименуйте его в **Задание 5**.
25. Создайте таблицу данных, в соответствии с таблицей ниже.

**Таблица 4** –Таблица данных для задания 5

	<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>E</b>	<b>F</b>	<b>G</b>
<b>1</b>	<b>№</b>	<b>Фамилия</b>	<b>Имя</b>	<b>Отчество</b>	<b>Группа</b>	<b>Средний балл</b>	<b>Итоговый отчет</b>
<b>2</b>	1	Иванов	Петр	Алексеевич	ИС-21	4,5	?
<b>3</b>	2	Петров	Николай	Сергеевич	ИС-21	3,8	?
<b>4</b>	3	Сидоров	Андрей	Васильевич	ИС-22	4,2	?
<b>5</b>	4	Кузнецов	Дмитрий	Павлович	ИС-22	4,9	?
<b>6</b>	5	Смирнов	Василий	Иванович	ИС-23	3,6	?
<b>7</b>	6	Фёдоров	Евгений	Максимович	ИС-23	4,1	?
<b>8</b>	7	Белов	Сергей	Дмитриевич	ИС-24	3,9	?

26. Для объединения данных из нескольких ячеек в одну строку используется функция **СЦЕПИТЬ()**.

27. Чтобы взять первую букву текста в ячейке, используйте функцию **ЛЕВ()**.

28. Чтобы после инициалов автоматически ставилась точка, добавьте её в формулу: **=СЦЕПИТЬ(ЛЕВ(B2;1); ".")**.

29. Выполните текущее сохранение документа.

## Практическая работа №40-41. Построение диаграмм для иллюстрации статистических данных. Построение графиков функций

**Цель работы:** изучить возможные форматы ячеек и способы автозаполнения; освоить принципы использования формул, выполнения вычислений, построения простейших диаграмм.

### Ход выполнения практической работы

1. На рабочем столе в папке **Программы** найдите программу LibreOffice Calc и откройте её.

2. Сохраните файл в свою рабочую папку с именем **ПР40-41\_Фамилия\_Группа** (вкладка **Файл** → **Сохранить как...** → **Рабочий стол** → **Моя рабочая папка**).



**Задание 1.** Построение объёмного графика полусферы.

#### Порядок построения полусферы:

1. Формула сферы  $R^2 = x^2 + y^2 + z^2$ . Следует выразить переменную  $z = \sqrt{R^2 - x^2 - y^2}$ .

2. В первой строке, начиная с ячейки **B1**, нужно внести значения переменной  $x$  в диапазоне от **-3** до **3** с шагом **0,5**.

3. В **столбце А**, начиная с ячейки **A2**, следует ввести значения переменной  $y$  в том же диапазоне от **-3** до **3** с тем же шагом **0,5**.

4. В ячейке **B16** задаём значение радиуса 16

5. В ячейку **B2** вводим формулу переменной  $z$ :

$$=\text{КОРЕНЬ}(\$B\$16^2-B\$1^2-\$A2^2)$$

6. При этом:

– ссылка на ячейку, содержащую значение радиуса, является абсолютной, т. к. радиус остается неизменным **\$B\$16** (используйте функциональную клавишу **F4**);

– ссылка на ячейку, содержащую значение переменной  $y$  является **смешанной B\$1** (относительной по столбцу, и **абсолютной по строке**);

– ссылка на ячейку, содержащую значение переменной  $x$  так же, является **смешанной \$A2** (**абсолютной по столбцу**, и относительной по строке);

7. Далее используя автозаполнение, копируем формулу в другие ячейки диапазона (рис. 38).

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	X/Y	-3,00	-2,50	-2,00	-1,50	-1,00	-0,50	0,00	0,50	1,00	1,50	2,00	2,50	3,00
2	-3,00	15,43	15,52	15,59	15,64	15,68	15,71	15,72	15,71	15,68	15,64	15,59	15,52	15,43
3	-2,50	15,52	15,60	15,68	15,73	15,77	15,80	15,80	15,80	15,77	15,73	15,68	15,60	15,52
4	-2,00	15,59	15,68	15,75	15,80	15,84	15,87	15,87	15,87	15,84	15,80	15,75	15,68	15,59
5	-1,50	15,64	15,73	15,80	15,86	15,90	15,92	15,93	15,92	15,90	15,86	15,80	15,73	15,64
6	-1,00	15,68	15,77	15,84	15,90	15,94	15,96	15,97	15,96	15,94	15,90	15,84	15,77	15,68
7	-0,50	15,71	15,80	15,87	15,92	15,96	15,98	15,99	15,98	15,96	15,92	15,87	15,80	15,71
8	0,00	15,72	15,80	15,87	15,93	15,97	15,99	16,00	15,99	15,97	15,93	15,87	15,80	15,72
9	0,50	15,71	15,80	15,87	15,92	15,96	15,98	15,99	15,98	15,96	15,92	15,87	15,80	15,71
10	1,00	15,68	15,77	15,84	15,90	15,94	15,96	15,97	15,96	15,94	15,90	15,84	15,77	15,68
11	1,50	15,64	15,73	15,80	15,86	15,90	15,92	15,93	15,92	15,90	15,86	15,80	15,73	15,64
12	2,00	15,59	15,68	15,75	15,80	15,84	15,87	15,87	15,87	15,84	15,80	15,75	15,68	15,59
13	2,50	15,52	15,60	15,68	15,73	15,77	15,80	15,80	15,80	15,77	15,73	15,68	15,60	15,52
14	3,00	15,43	15,52	15,59	15,64	15,68	15,71	15,72	15,71	15,68	15,64	15,59	15,52	15,43
15														
16	R =	16,00												

Рис. 38. Создание матрицы полусферы

8. Выделяя получившуюся матрицу (диапазон ячеек **B2:N14**), создаём поверхностную диаграмму (вкладка **Вставка** → **Диаграмма**). Шаг 1 (рис. 39).

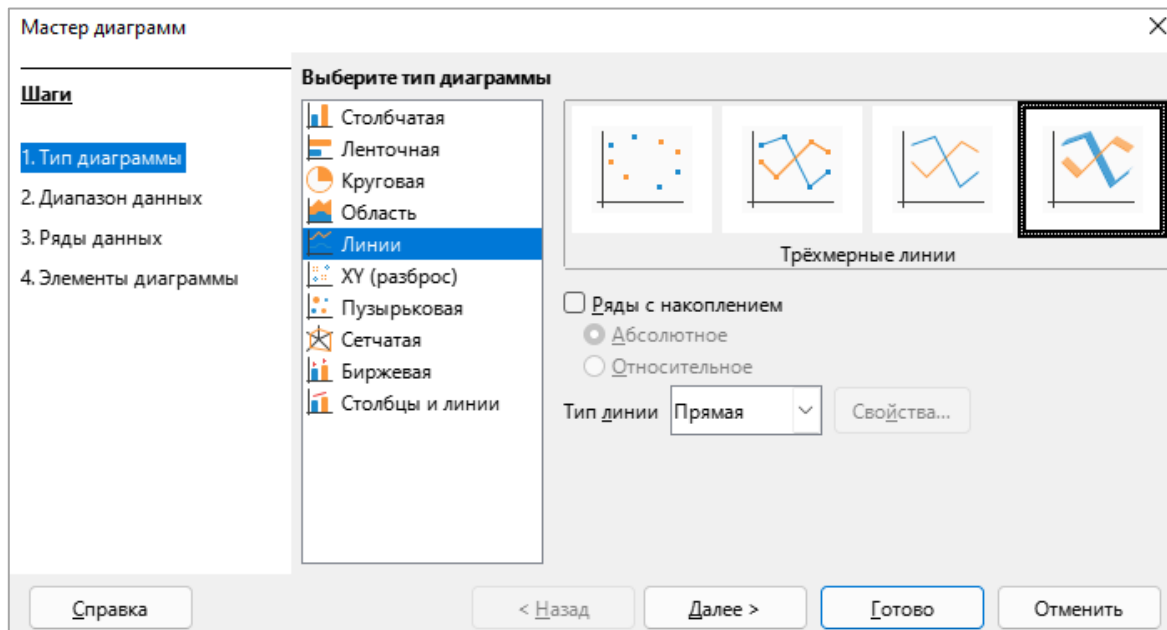
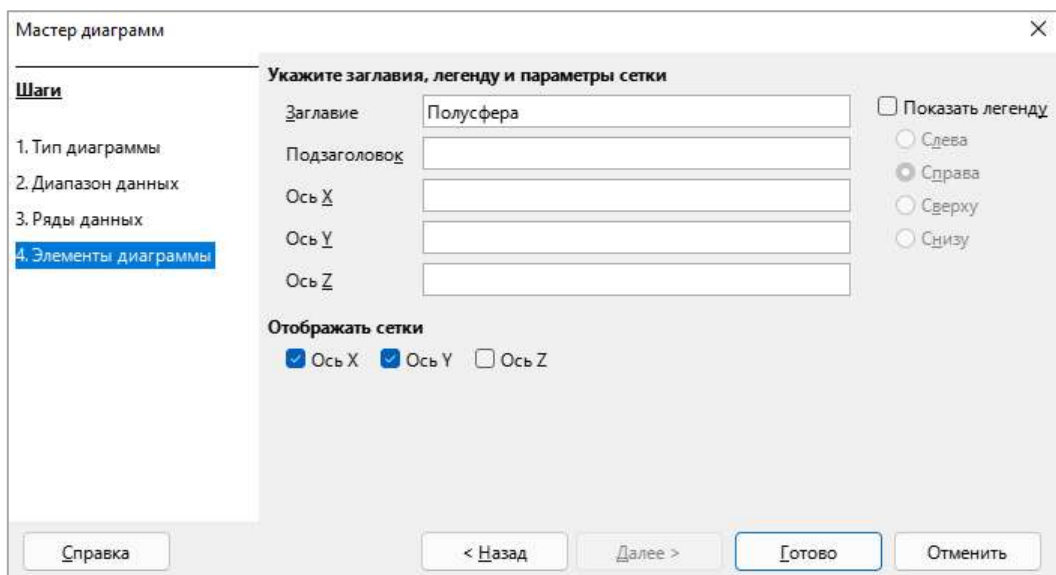


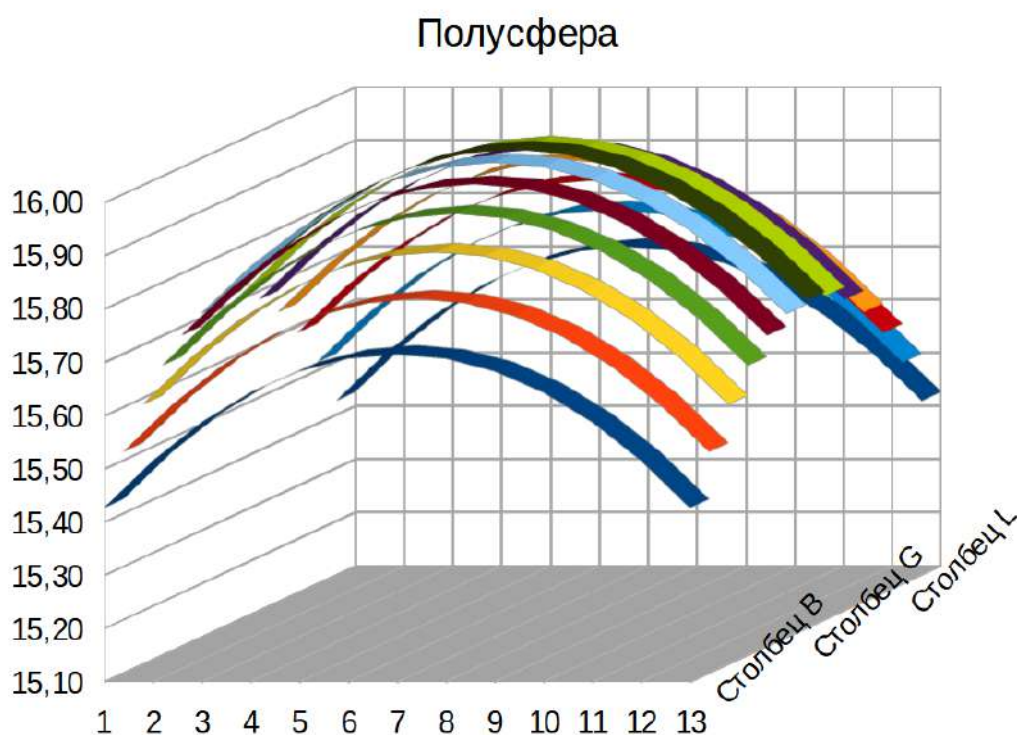
Рис. 39. Шаг 1

9. Перейдите к шагу 4 и настройте его как показано на рис. 40.




**Рис. 40.** Шаг 4

10. Получаем изображение полусферы. Сравните полученный результат с рис. 41.



**Рис. 41.** Полусфера

11. Переименуйте ярлычок Лист1 на **Задание 1**. Выполните текущее сохранение.

 **Задание 2.** Создание таблицы «Итоговые результаты за 1 семестр».

12. Перейдите на Лист2, переименуйте его в **Задание 2**.

13. На **Листе2** заполните таблицу: номера групп; статистика по оценкам (рис. 42).

	A	B	C	D	E	F	G	H
1	<b>Итоговые результаты за 1 семестр</b>							
2	по строительному отделению							
3	Экзамен по математике							
4	<b>Статистика по оценкам</b>						<b>Кол-во студентов в группах</b>	<b>Процент успеваемости в группах</b>
5	<b>Группа</b>	<b>5 (отлично)</b>	<b>4 (хорошо)</b>	<b>3 (удовл.)</b>	<b>2 (неудовл.)</b>	<b>Неявки</b>		
6	<b>C-211</b>	6	8	9	4	3		
7	<b>C-212</b>	5	12	10	3	1		
8	<b>C-213</b>	8	10	8	4	0		
9	<b>C-214</b>	10	4	9	5	2		
10	<b>C-215</b>	3	7	18	4	0		
11	<b>САД-211</b>	9	8	9	3	1		
12	<b>АРХ-211</b>	11	15	5	0	0		
13	<b>Итого:</b>							
14								
15								
16								

**Рис. 42.** Образец заполнения таблицы

14. Рассчитаем в ячейке **G6** количество студентов в первой группе. Для этого в указанную ячейку запишем формулу **=СУММ(B6:F6)**. После нажатия клавиши **Enter** в ячейке **G6** будет выполнен расчёт количества студентов.

15. Для распространения формулы ячейки **G6** на соседние ячейки, наведите мышь на правый нижний угол **G6** до получения чёрного крестика, нажмите левую кнопку мыши, и «протяните» ее до ячейки **G12**. Произойдет распространение формулы на соседние ячейки и автоматический расчёт количества студентов в остальных группах.

16. Процент успеваемости в группах рассчитывается в ячейке **H6** по формуле:

$$\text{Процент успеваемости} = \frac{\text{Количество 5, 4 и 3}}{\text{Количество студентов в группе}}$$

17. Полученный результат необходимо, с помощью команды **Формат** → **Ячейки** → **Числа**, представить в процентном формате с двумя знаками после запятой.

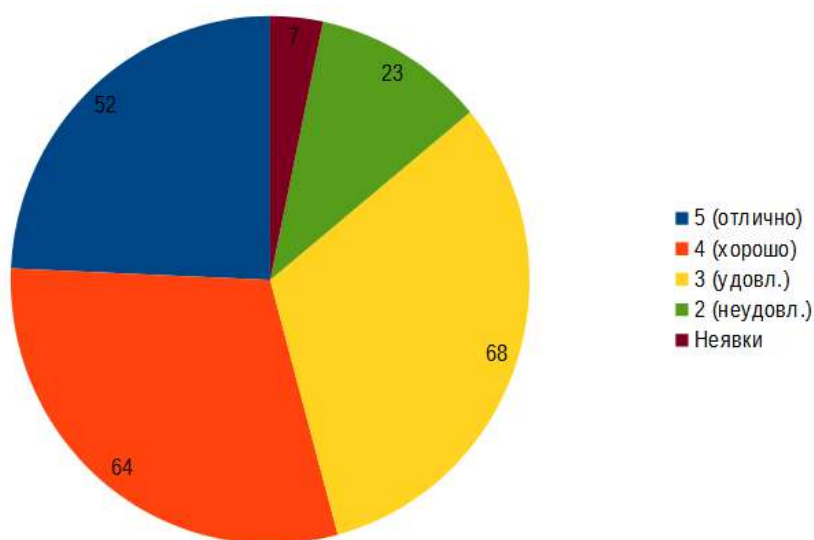
18. Построим две диаграммы, опираясь на данные заполненной выше таблицы. Выделите мышью диапазон ячеек с номерами групп **A6:A12**, затем, удерживая клавишу **Ctrl**, выделите несмежный диапазон ячеек с количеством студентов **G6:G12**.

- Выполните команду **Вставка** → **Диаграмма**, Тип диаграммы: **Столбчатая**.
- Нажмите кнопку **Далее**.
- Выберите опции: **Ряды данных в столбцах** и **Подписи в первом столбце**.
- Нажмите кнопку **Далее**.
- Нажмите кнопку **Далее** еще раз.
- Укажите заглавие: «**Количество студентов в группах**», ось X: «**Номера групп**», ось Y: «**Количество**», выберите опцию «**Не показывать легенду**».
- Нажмите кнопку «**Готово**». На листе появится диаграмма следующего вида (рис. 43):



**Рис. 43.** Столбчатая диаграмма

19. Выделите диапазоны ячеек с названиями оценок **B5:F5**, затем, удерживая клавишу **Ctrl**, выделите несмежный диапазон ячеек с итоговыми данными **B13:F13**. Постройте круговую диаграмму, учитывая, что ряды данных находятся в строках, а первая строка является подписью данных. На листе должна появиться диаграмма, как на рис. 44.



**Рис. 44.** Круговая диаграмма

20. Выполните текущее сохранение.



**Задание 3.** Построить графики посещаемости занятий.

21. Перейдите на Лист3, переименуйте его в **Задание 3**.

22. На **листе 3** заполнить и оформить таблицу по образцу (см. рис. 45).

23. Вычислить сумму часов посещений, а также среднее число посещений, используя функции **СУММ()** и **СРЗНАЧ()**.

	A	B	C	D	E	F	G	H
1	<b>Отчёт о посещаемости занятий студентами группы С-2022</b>							
2	<b>№ п/п</b>	<b>Фамилия, имя</b>	<b>Февраль</b>	<b>Март</b>	<b>Апрель</b>	<b>Май</b>	<b>Июнь</b>	<b>Всего</b>
3	1	Кириллов В.	120	168	140	150	70	648
4	2	Карпов К.	100	130	120	140	60	550
5	3	Матвеев А.	90	110	120	110	60	490
6	4	Ивакин И.	110	140	160	120	30	560
7	5	Дмитриев П.	70	60	100	90	80	400
8								
9		Среднее число посещений	98	122	128	122	60	530

**Рис. 45.** Данные о посещении занятий студентами

24. Для построения графиков выделить диапазон ячеек **B2:G7**, который содержит данные о посещаемости занятий студентами, а также названия месяцев и столбец с фамилиями.

25. С помощью команд **Вставка** → **Диаграмма** → **Линии**, построить график (рис. 46). При построении диаграммы учесть, что ряды данных указаны в строках.

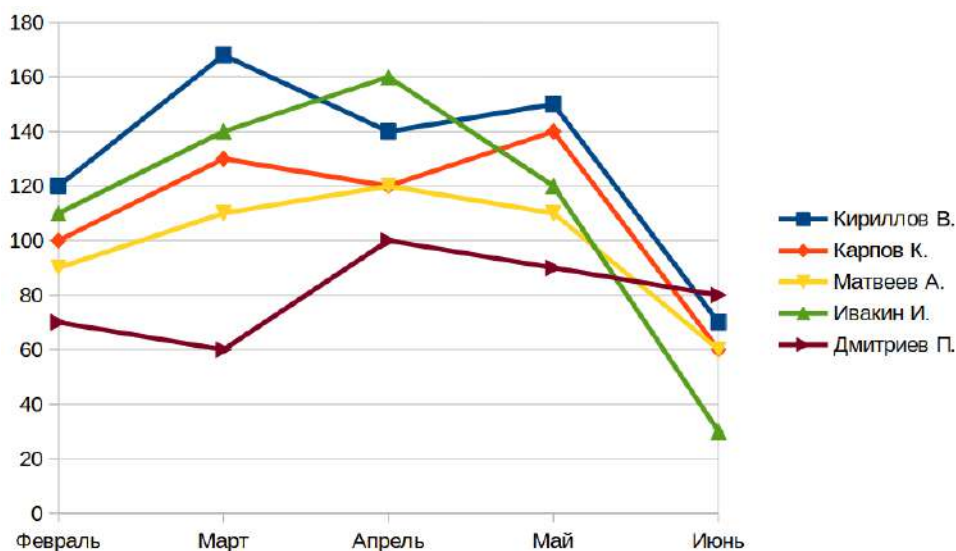


Рис. 46. Графики посещения занятий студентами

26. Выделить названия месяцев вместе с пустой ячейкой **B2**, а также данные о посещении занятий студентами **Кирилловым** и **Матвеевым**. Построить вторую диаграмму (рис. 47). Построить аналогичную диаграмму для **Карпова** и **Дмитриева** (рис. 47).

27. Выполните текущее сохранение.

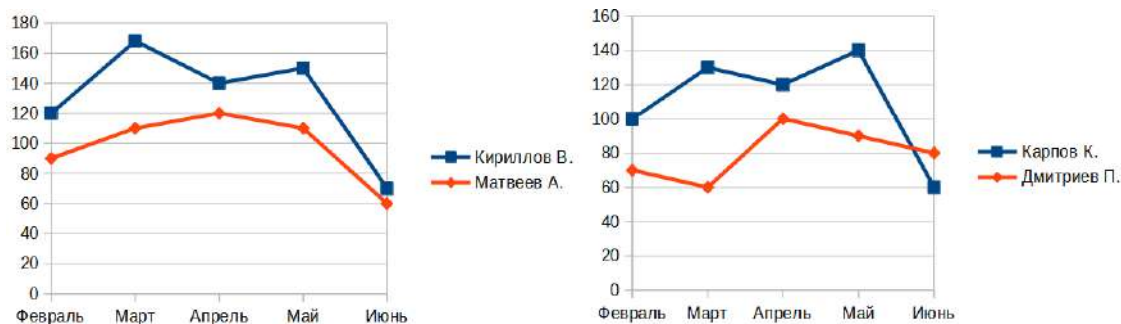


Рис. 47. Графики посещения занятий студентами




**Задание 4. Вычислить значения функций.**

28. Перейдите на Лист 4, переименуйте его в **Задание 4**.

Вычислить значения функций:

1.  $A^2 + B^2$ ;
2.  $(A + B)^2$ ;
3.  $\cos A \sin B$ ;
4.  $\sqrt{\frac{A}{B}}$ ;
5.  $\ln|A - B|$ ;
6.  $Be^{-A}$ ;
7.  $\cos^2 B + A$ ;
8.  $\cos(B^2) + A$ .

**Функции LibreOffice Calc** – это специальные, заранее созданные формулы, в которые пользователь должен ввести только аргументы. Для выбора необходимой функции используется мастер функций (кнопка , либо команда **Вставка** → **Функция**), который позволяет грамотно ввести требуемые параметры. В случае если в качестве параметра выступает массив значений, используют двоеточие, которое также называют оператором диапазона. Например, запись **A1:D4** определяет диапазон ячеек, расположенный между ячейками **A1** и **D4**.

29. Перейти на **Лист4**, ввести значения аргументов *A* и *B* в ячейки **B3:B4** (Таблица ). Ввести номера и формулы (столбцы **D** и **E**). В столбце **E** показана структура записи формул в электронных таблицах.

30. В столбце **E** вычислите значения функций. При этом в формулах вместо *A* и *B* необходимо указывать ссылки на ячейки, содержащие значения аргументов *A* и *B*. Например, формула ячейки **E3** будет иметь вид: **=B3^2+B4^2**.

**Таблица 1** – Вычисление значений функций

	<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>E</b>	<b>F</b>
<b>1</b>						
<b>2</b>				№	формулы	
<b>3</b>	A=	4		1	ввести формулу	41
<b>4</b>	B =	5		2	ввести формулу	81
<b>5</b>				3	ввести формулу	0,626795
<b>6</b>				4	ввести формулу	0,894427
<b>7</b>				5	ввести формулу	0
<b>8</b>				6	ввести формулу	0,091578
<b>9</b>				7	ввести формулу	4,080464
<b>10</b>				8	ввести формулу	4,991203

31. Выполните текущее сохранение



**Задание 5. Самостоятельная работа**

32. Перейдите на Лист 5, переименуйте его в **Задание 5**.

33. Вычислить на новом листе значение функции:

$$y = \operatorname{arctg}(e^{-2x}) + \frac{\ln(3+x)}{x^3+4}$$

в точках,  $x = -2$ ,  $x = 2$ .

Данная функция достаточно громоздкая, поэтому для удобства разобьём её на несколько составляющих:

$$y_1 = e^{-2x}$$

$$y_2 = \operatorname{arctg}(e^{-2x})$$

$$y_3 = \ln(3 + x)$$

$$y_4 = x^3 + 4$$

Принцип составления формул приведён на рис. 48. Для вычисления значения функции  $y$  в точке  $x = 2$ , формулы достаточно скопировать (распространить).

	A	B	C	D	E	F	G
1		x	y1	y2	y3	y4	y
2		-2	=EXP(-2*B2)	=ATAN(C2)	=LN(3+B2)	=B2^3+4	=D2+E2/F2
3		2					

**Рис. 48.** Вычисление значений функции  $y(x)$

34. Постройте график функции. Переименуйте ярлычок «Лист5» на **Задание 5**. Выполните текущее сохранение.

35. Вычислите на новом листе значение функции  $y$  в соответствии с заданием варианта, приведенным в таблице 2.

**Таблица 2** – Задания для самостоятельной работы

№	$y(x)$	$x$	№	$y(x)$	$x$
1	$y = \sqrt{\frac{2x^3 + \operatorname{arctg}(x)}{x + 2 \ln(x) }}$	$x = 1,$ $x = 3$	8	$y = \frac{x^3 + 4\operatorname{tg}(x)}{\sqrt{5x^3 + 4}}$	$x = 0,$ $x = 7$
2	$y = \frac{\sqrt{13x + e^{-x}}}{7x^2 + 7x + 14\ln(x)}$	$x = 2,$ $x = 5$	9	$y = \frac{\sin(x^3) - e^{2x}}{\sqrt{4x^3 + 1}}$	$x = 1,$ $x = 4$
3	$y = \frac{\ln(4x^2 + 2) + \sin^2(x)}{e^{2x-3}}$	$x = -2,$ $x = 1$	10	$y = \frac{\cos^3(x) + e^{-x}}{\sqrt{ 6 - 2x^3 }}$	$x = 0,$ $x = 2$
4	$y = \frac{(4x^2 + 2)^3 + \operatorname{tg}(x)}{\sqrt{ 2x - 3 }}$	$x = -1,$ $x = 1$	11	$y = \frac{3e^{\sin(x)} + x^3}{\ln(x) + 2\cos(2x)}$	$x = 1,$ $x = 4$
5	$y = \frac{\ln\sqrt{x^2 + 7} + x^4}{e^{2x} + \operatorname{tg}(2x)}$	$x = -3,$ $x = 0$	12	$y = \frac{e^{\sin(4+x)} \sin(x^2 + 9)}{\sqrt{ 4x^3 + 3 }}$	$x = -3,$ $x = 1$
6	$y = \frac{\sin\sqrt{2x + e^{-x}}}{\ln x  + x^4}$	$x = -1,$ $x = 2$	13	$y = \frac{\sin\sqrt{x^2 + 9}}{e^{\sin(4+x)}(4x^3 + 3)}$	$x = -2,$ $x = 0$
7	$y = \frac{2x^2 - 5x + \sqrt{x^3}}{\cos(2x^2 + 1)}$	$x = 1,$ $x = 4$	14	$y = \sin^2(x - 1) + \frac{\sqrt{x^2 + 5}}{\ln(2x^2)}$	$x = -1,$ $x = 3$

### Контрольные вопросы

1. Как выполняется ввод данных в текущую ячейку с использованием и без использования строки формул?

2. Как выполняется редактирование данных, введенных ранее в текущую ячейку?

3. С какого символа начинается ввод формулы в текущую ячейку?

4. Какие знаки действий и специальные символы используются при вводе формул?
5. Какими способами можно копировать формулы?
6. В какой последовательности происходит построение диаграммы?


## Практическая работа №42. Моделирование в электронных таблицах (на примерах задач из профессиональной области)

**Цель работы:** первоначальное знакомство с интерфейсом табличного процессора; практическое освоение процесса моделирования на примере конкретных задач; научиться вводить формулы для расчётов в табличном процессоре.

### Ход выполнения практической работы

1. На Рабочем столе в папке своего преподавателя найдите файл ПР42\_Заготовка. Откройте файл при помощи программы **LibreOffice Calc**.

2. Сохраните файл в свою рабочую папку с именем **ПР42\_Фамилия\_Группа** (вкладка **Файл** → **Сохранить как...** → **Рабочий стол** → **Моя рабочая папка**).

 **Задание 1. Обои и комната.** Выполнить имитационное<sup>2</sup> моделирование посредством табличного процессора.

#### 1 ЭТАП. ПОСТАНОВКА ЗАДАЧИ

**Описание задачи.** В магазине продаются обои. Наименования, длина и ширина рулона известны. Для удобства обслуживания надо составить таблицу, которая позволит определить необходимое количество рулонов для оклейки любой комнаты.

**Цель моделирования:** помочь покупателям быстро определять необходимое количество рулонов обоев.

**Формализация задачи:** формализуем задачи быстро определять необходимое количество рулонов обоев.

Уточняющий вопрос	Ответ
Что моделируется?	Система, состоящая из двух объектов: комнаты и обоев
Форма комнаты?	Прямоугольная
Что известно о комнате?	Размеры комнаты задаются высотой ( $h$ ), длиной ( $a$ ) и шириной ( $b$ )
Как учитывается не оклеиваемая поверхность?	15% площади стен комнаты занимают окна и двери. Можно рассчитать процент оклеиваемой поверхности. Для этого нужно знать размеры и количество окон и дверей

<sup>2</sup> **Имитационное моделирование (симуляция)** – это распространенная разновидность аналогового моделирования, реализуемого с помощью набора математических средств, специальных компьютерных программ-симуляторов и особых *ИТ*, позволяющих создавать в памяти компьютера процессы-аналоги, с помощью которых можно провести целенаправленное исследование структуры и функций реальной системы в режиме ее «имитации», осуществить оптимизацию некоторых ее параметров.

Что известно об обоях?	Наименования, длина и ширина
Какая часть рулона уходит на обрезки?	10% площади рулона
Надо ли покупать рулоны про запас?	Да, желательно 1 рулон
Можно ли купить часть рулона?	Нет, Количество рулонов должно быть целым
Что надо определить?	Необходимое количество рулонов обоев

## 2 ЭТАП. РАЗРАБОТКА МОДЕЛЕЙ

**Компьютерная модель.** Для моделирования выберем среду табличного процессора *LibreOffice Calc*. В этой среде информационная и математическая модели объединяются в таблицу, которая содержит три области:

- исходные данные;
- промежуточные расчёты;
- результаты.

**Таблица 1** – Расчётные формулы для задачи 1

	Ячейка	Формула
<b>Формула 1</b>	<b>B9</b>	=2*(\$B\$6+\$B\$7)*\$B\$5*(1-\$B\$8)
<b>Формула 2</b>	<b>D14</b>	=(1-\$B\$12)*B14*C14
<b>Формула 3</b>	<b>E14</b>	=ЦЕЛОЕ(\$B\$9/D14)+1

### АЛГОРИТМ ВВОДА ФОРМУЛЫ

*Введём формулу 1 в ячейку B9.*

1. Нажмите ЛКМ по ячейке **B9**. Поставьте знак равенства (=).
2. Далее последовательно введите число и арифметический знак, и левую круглую скобку, представленные в формуле (2\*()).

3. Далее в формуле идёт **абсолютная ссылка**<sup>3</sup>, с которой более подробно мы познакомимся в следующих практических работах. Сейчас научимся правильно и быстро её вводить. Для того, чтобы ввести **\$B\$6** нужно нажать ЛКМ по ячейке **B6** (вводить с клавиатуры ничего не нужно!), после нажать на функциональную клавишу на клавиатуре **F4** (это одна клавиша, а не две по отдельности!), после чего должны появиться знаки доллара.

4. Введите арифметический знак сложения (+) и аналогичным образом введите абсолютную ссылку с ячейкой **B7** как показано в пункте 3, после чего не забудьте поставить правую закрывающую скобку.

<sup>3</sup> Абсолютная ссылка обозначается знаком \$ в координатах строки и столбца, например **\$A\$1**. **Знак доллара** фиксирует ссылку на данную ячейку, так что она остается неизменной независимо от того, куда смещается формула.

5. Самостоятельно введите оставшуюся часть формулы ( $*B5*(1-B8)$ ), используя клавиатуру и ЛКМ.

6. Введите формулу 2 в ячейку **D14**.

7. Введите формулу 3 в ячейку **E14**.

8. Используйте маркер автозаполнения для того, чтобы быстро заполнить ячейки. Для этого выделите ячейки **D14:E14**. В выделенном диапазоне найдите в правом нижнем углу чёрный квадрат (маркер автозаполнения) (рис. 49) и зажав ЛКМ, протяните вниз до 19-й строки включительно.



Площадь рулона	Количество рулонов
4,725	8

Рис. 49. Маркер автозаполнения

**Примечание.** Функция **ЦЕЛОЕ()** округляет до ближайшего целого числа, меньшего чем заданное. Но, поскольку количество рулонов нельзя округлять в меньшую сторону, то к значению функции прибавляем 1 для округления в большую сторону и получаем 1 запасной рулон.

### 3 ЭТАП. КОМПЬЮТЕРНЫЙ ЭКСПЕРИМЕНТ

**Тестирование.** Провести тестовый расчёт компьютерной модели по данным, приведённым в таблице.

**Эксперимент 1.** Провести расчёт количества рулонов обоев для помещений вашей квартиры.

**Эксперимент 2.** Изменить данные некоторых образцов обоев и проследить за пересчётом результатов.

**Эксперимент 3.** Добавить строки с образцами и дополнить модель расчётом по новым образцам.

#### **Проведение исследований:**

1. Введите в таблицу тестовые данные и сравните результаты тестового расчёта с результатами, приведёнными на рисунке 50.

2. Поочерёдно введите размеры комнат вашей квартиры.

3. Проведите другие виды расчётов согласно плану.


9	Площадь стен	35,36			
10					
11	Обои			Промежуточные расчёты	
12	Обрезки	10,00%			РЕЗУЛЬТАТЫ
13	Наименования	Длина	Ширина	Площадь рулона	Количество рулонов
14	Образец 1	10,5	0,5	4,725	8
15	Образец 2	10,5	0,6	5,67	7
16	Образец 3	10,5	0,7	6,615	6
17	Образец 4	13	0,5	5,85	7
18	Образец 5	13	0,6	7,02	6
19	Образец 6	13	0,7	8,19	5

Рис. 50. Пример расчёта

#### 4 ЭТАП. АНАЛИЗ РЕЗУЛЬТАТОВ

По данным таблицы можно определить количество рулонов каждого образца обоев для любой комнаты.

Выполните текущее сохранение документа.

 **Задание 2. Сберкасса.** Выполнить имитационное моделирование посредством табличного процессора.

#### 1 ЭТАП. ПОСТАНОВКА ЗАДАЧИ

**Описание задачи.** За 2 часа до обеденного перерыва 40 бабушек встали в очередь за пенсией. Кассирша обслуживает клиента в среднем одну минуту.

Первая бабушка «мучила» кассиршу вопросами 9 мин. 15 с. Каждая следующая бабушка, частично «мотая на ус» ответы, адресованные предыдущим бабушкам, «мучает» кассиршу на 10 с меньше. Построить модель ситуации и исследовать ее.

**Цель моделирования:** исследовать ситуацию с разных углов зрения (задания типа «что будет, если ...», «как сделать, чтобы...»), сделать выводы и дать свои рекомендации по улучшению обслуживания.

**Формализация задачи:** чтобы понять суть задачи, ответим на уточняющие вопросы.

Уточняющий вопрос	Ответ
Что моделируется?	Система, состоящая из двух простых объектов: кассир и очередь
Чем характеризуется кассир?	Временем обслуживания одного клиента
Что известно об очереди?	Очередь состоит из 40 бабушек
Чем характеризуется одна бабушка?	Временем на расспросы
Чем характеризуется сберкасса (среда, в которой разворачиваются моделируемые события)?	Временем до обеденного перерыва, временем до закрытия кассы

Что надо определить?

Время обслуживания каждой бабушки. Время, затраченное каждой бабушкой на получение денег.

## 2 ЭТАП. РАЗРАБОТКА МОДЕЛИ

**Компьютерная модель.** Объединим информационную и математическую модели, воспользовавшись средой табличного процессора. Для этого задачи промежуточные данные и результаты представляются в виде единого блока, так как в этом блоке содержится вся необходимая информация для анализа и выводов.

### Расчётные формулы:

	Ячейка	Формула
Формула 1	<b>A14</b>	=A13+1
Формула 2	<b>B13</b>	=\$B\$5+\$B\$7
Формула 3	<b>B14</b>	=B13-\$B\$8
Формула 4	<b>C14</b>	= C13+B13

**Примечание.** В операциях участвуют данные типа **Время**. Результат выполнения такой операции будет записан в ячейке в числовой форме. Чтобы перевести числовой формат в тип **Время**, следует использовать **вкладку Формат** → **Формат чисел** → **Время**.

## 3 ЭТАП. КОМПЬЮТЕРНЫЙ ЭКСПЕРИМЕНТ

### План эксперимента

**Тестирование:** провести правильность ввода формул. Произвести расчёты.

**Эксперимент 1.** Изменяя значения ячеек **B5**, **B7**, **B8**, исследовать, влияние этих характеристик на скорость движения очереди.

### Проведение исследования

1. Введите в таблицу контрольные исходные данные и скопируйте расчётные формулы в две-три строки.
2. Заполните формулами ячейки на 52 строки.
3. Определите по таблице (столбец **C**), сколько времени понадобится кассиру, чтобы обслужить всю очередь.
4. Найдите в таблице строку, которая соответствует наступлению обеденного перерыва.

## 4 ЭТАП. АНАЛИЗ РЕЗУЛЬТАТОВ МОДЕЛИРОВАНИЯ

По результатам моделирования в электронных таблицах ответить на следующие вопросы:

- 1) Как долго будет «мучить» кассиршу сороковая бабушка?

2) Хватит ли на обслуживание всех бабушек дообеденного времени (2 часа)?

3) Если не хватит времени, то какую по счёту бабушку успеют обслужить до перерыва?

4) Хватит ли времени на обслуживание всей очереди до конца рабочего дня, если после перерыва касса будет открыта ещё 4 часа?

5) Как влияет на время обслуживания очереди уменьшение времени расспросов? Можно и сокращать это время и к чему это приведёт?

Выполните текущее сохранение документа.



**Задание 3. Память студента.** Выполнить имитационное моделирование посредством табличного процессора.

## 1 ЭТАП. ПОСТАНОВКА ЗАДАЧИ

**Описание задачи.** Студент Ставропольского строительного техникума учит стихотворение из 40 строк. Чтобы запомнить первую строчку, ему понадобилась всего 1 мин. На каждую следующую он тратит на 10% времени больше. Стихотворение держится в памяти студента не дольше 3 часов, а до техникума бежать 15 мин. Как организовать заучивание стихотворения?

**Цель моделирования:** предлагается исследовать ситуацию и предложить реальные способы ускорения процесса заучивания стихотворения.

**Формализация задачи:** ответим на уточняющие вопросы по условию задачи.

Уточняющий вопрос	Ответ
Что моделируется?	Система, состоящая из двух объектов: студента и стихотворения
Чем характеризуется студент?	Временем заучивания первой строки, процентом снижения прилежания с каждой строкой. Временем удерживания в памяти любой информации
Что известно о стихотворении?	Стихотворение состоит из 40 строк
Чем характеризуется среда?	Временем бега от дома к техникуму
Что надо определить?	Время заучивания каждой строки. Суммарное время на заучивания всех строк стихотворения. Время заучивания стихотворения

## 2 ЭТАП. РАЗРАБОТКА МОДЕЛИ

**Компьютерная модель.** Объединим информационную и математическую модели, воспользовавшись средой табличного процессора. Для этой задачи промежуточные данные и результаты представляются в виде единого блока, так как в этом блоке содержится вся необходимая информация для анализа и выводов.

**Расчётные формулы:**

	Ячейка	Формула
Формула 1	A13	=A12+1
Формула 2	B12	=\$B\$5
Формула 3	B13	=B12*(1+\$B\$6)
Формула 4	C 12	=B12
Формула 5	C13	=C12+B13

**Примечание.** В операциях участвуют данные типа Время. Результат выполнения такой операции будет записан в ячейке в числовой форме. Чтобы перевести числовой формат в тип Время, следует использовать вкладку **Формат** → **Формат чисел** → **Время**.

### 3 ЭТАП. КОМПЬЮТЕРНЫЙ ЭКСПЕРИМЕНТ

#### План эксперимента

**Тестирование:** провести правильность ввода формул.

1. Рассчитать в электронной таблице время заучивания каждой строки стихотворения и суммарное время заучивания. Исследовать полученные результаты.

2. Изменяя значения в ячейках **B5** и **B6**, исследовать влияние старательности студента на общее время заучивания стихотворения.

#### Проведение исследования

**Тестирование.** Ввести в таблицу контрольные исходные данные и скопируйте расчётные формулы в две-три строки.

#### Эксперимент 1. Исследование процесса заучивания стихотворения.

Скопируйте формулы в нижестоящие ячейки, в общей сложности на 51 строку. По расчётным данным дайте ответ на следующие вопросы:

– Сколько времени понадобилось бы мальчику, чтобы выучить всё стихотворение?

– Из скольких строк (максимум) должно состоять стихотворение, если преподаватель студента утверждает, что его знаний хватает не больше чем на 3 часа, а до техникума бежать 15 минут? Каким по счёту может быть пара литературы, чтобы знания «не выветрились» из головы?

– Сколько строк можно успеть выучить до начала «мультика», если до него осталось 45 минут?

**Эксперимент 2. Исследование влияния старательности на время заучивания стихотворения.** Изменяя значения в ячейках **B5** и **B6**, исследовать влияние старательности студента на общее время заучивания стихотворения. По расчётным данным дайте ответы на следующие вопросы:

– Как влияет время заучивания 1-й строки на общее время запоминания стихотворения?

– Как влияет концентрация внимания (уменьшение D%) на общее время запоминания стихотворения?

#### 4 ЭТАП. АНАЛИЗ РЕЗУЛЬТАТОВ МОДЕЛИРОВАНИЯ

По результатам, рассчитанным в электронной таблице, ответьте на следующие вопросы:

1) Что нужно изменить в модели заучивания стихов, чтобы просчитать её для любых других усидчивых и нерадивых<sup>4</sup> студентов. На какой части таблицы это отразится?

2) Что нужно изменить в модели заучивания стихов, чтобы просчитать её для стихотворения любой длины?

Выполните текущее сохранение документа.



**Задание 4. Расчёт кривой падения электрика.** Выполнить имитационное моделирование посредством табличного процессора.

#### 1 ЭТАП. ПОСТАНОВКА ЗАДАЧИ

**Описание задачи.** Электрик Петров приставил к стене лестницу и, поднявшись вверх, остановился на одной из ступенек. В это время концы лестницы начали скользить вдоль стены и пола. Провести исследование, по какой кривой будет падать вниз электрик Петров.

**Цель моделирования:** исследовать движение лестницы и траекторию падения электрика.

**Формализация задачи:** проведём формализацию задачи в виде ответов на вопросы.

Уточняющий вопрос	Ответ
Что моделируется?	Объект «лестница», который представляет собой систему, состоящую из ступенек
Какие параметры лестницы известны?	Длина, количество ступенек, угол, образуемый лестницей и стеной
Как расположены ступеньки на лестнице?	На одинаковом расстоянии
Как движется лестница?	Концы скользят вдоль стены и пола
Что надо определить?	Кривую, по которой будет падать электрик
Чем характеризуется точка кривой?	Координатами $x$ и $y$ , которые определяются в некоторой заданной системе координат и связаны с углом отклонения лестницы

#### 2 ЭТАП. РАЗРАБОТКА МОДЕЛЕЙ

**Компьютерная модель.** Для построения компьютерной модели выберем среду табличного процессора *LibreOffice Calc*. В этой среде информационная и математическая модели объединяются в таблицу, которая содержит три области:

– исходные данные;

<sup>4</sup> Небрежно относящийся к своим обязанностям, небрежный.

- промежуточные расчёты;
- результаты.

**Примечание.** При использовании встроенных функций **SIN()** и **COS()** важно помнить, что угол должен выражен в радианах<sup>5</sup>.

**Расчётные формулы:**

	Ячейка	Формула
Формула 1	C9	=C\$4/C\$5
Формула 2	C10	=C\$6*C\$9
Формула 3	C11	=C\$4-C\$10
Формула 4	A13	0
Формула 5	A14	= A13+ C\$7
Формула 6	B13	=A13/180*ПИ()
Формула 7	C13	=C\$11*SIN(B13)
Формула 8	D13	=C\$10*COS(B13)

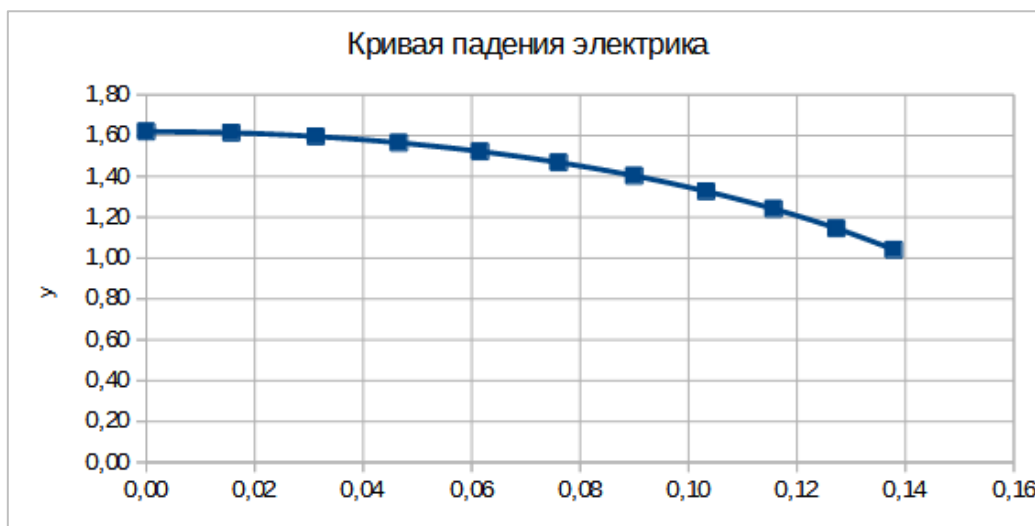
**3 ЭТАП. ПОСТРОЕНИЕ КРИВОЙ ПАДЕНИЯ ЭЛЕКТРИКА**

Выделить диапазон ячеек **C12:D23**, нажмите вкладку **Вставка** → **Диаграмма**:

- **Шаг 1:** Тип диаграммы → XY (разброс); Тип линии → Сглаженная → нажмите Далее.
  - **Шаг 2:** Диапазон данных: выберите Ряды данных в столбцах и Подписи в первой строке → нажмите Далее.
  - **Шаг 3:** Ряды данных → нажмите Далее.
  - **Шаг 4:** Элементы диаграммы: введите заглавие, ось X, ось Y, включите отображение сетки и уберите легенду → нажмите Готово.
- Сравните результат с рисунком ниже.

---

<sup>5</sup> **Радианом** называют угол, образованный дугой окружности длиной равной радиусу этой окружности. Длина окружности равна:  $l = 2 \cdot \pi \cdot r$ , где  $r$  – радиус этой окружности. Тогда, разделив на радиус, получаем, что полный угол в радианах равен:  $\frac{l}{r} = \frac{2 \cdot \pi \cdot r}{r} = 2 \cdot \pi$  радиан. В градусах этот же угол равен, как известно, 360°.



**Рис. 51.** Кривая падения электрика

#### 4 ЭТАП. КОМПЬЮТЕРНЫЙ ЭКСПЕРИМЕНТ

##### План эксперимента

**Тестирование:** провести правильность ввода формул.

1. Исследовать вид кривой падения электрика в зависимости от номера ступеньки.
2. Исследовать вид кривой в зависимости от номера ступеньки.

##### Проведение исследования

**Эксперимент 1.** Скопируйте формулы в нижестоящие ячейки.

**Эксперимент 2.** Исследование вида кривой

Постройте диаграммы кривых для различных номеров ступенек, для этого:

– Введите в ячейку **С6** номер ступеньки. Проследите за пересчётом значений в таблице и изменениями на диаграмме.

Выполните текущее сохранение документа.



**Задание 5. Продукты для похода.** Выполнить имитационное моделирование посредством табличного процессора.

На пятом листе книги, самостоятельно создайте таблицу и выполните необходимые расчёты, используя формулы.

Для организации похода надо построить модель расчёта нормы продуктов для группы туристов. Известна норма каждого продукта на 1 человека на 1 день, количество человек и количество дней похода.

Рассчитать необходимое количество продуктов на весь поход для 1 человека и для всей группы. **Провести расчёты для разных исходных значений количества дней и туристов.** Нормы продуктов приведены в таблице.

<b>Продукты для похода</b>			
Кол-во человек	15		
Кол-во дней	6		
Название	1 чел / день	15 человек / 1 день	15 человек / 6 дней
Вермишель, г	55	Рассчитать по формуле	Рассчитать по формуле
Рис, г	20		
Пшено, г	20		
Гречка, г	20		
Картофель, г	200		
Колбаса, г	40		
Сыр	50		
Масло, г	35		
Суп, пакет (г)	330		
Хлеб, г	330		
Булка, г	330		
Чай, г	5		

Выполните текущее сохранение документа.

## Список информационных источников

1. PEP 8 (Python Enhancement Proposal) – руководство по стилю кода для Python, описывающее стандарты разработки на этом языке URL: <https://peps.python.org/pep-0008/>.
2. Базы данных LibreOffice Base// Math Документация LibreOffice. URL: <https://libreoffice.readthedocs.io/ru/latest/base.html>.
3. Босова, Л. Л. Информатика. Базовый уровень. 10–11 классы. Компьютерный практикум / Л. Л. Босова, А. Ю. Босова, И. Д. Куклина и др. – М. : БИНОМ. Лаборатория знаний, 2021. – 144 с. : ил.
4. Демидова М. Операторы Python: для чего они нужны и какими бывают //Skillbox Media | Код. [Ссылка на статью](#).
5. Натансон И. П., Сидоров Ю. В. Функция // Большая российская энциклопедия: научно-образовательный портал – URL: <https://bigenc.ru/c/funktsiia-fc637f/?v=10538176>.
6. Табличный процессор LibreOffice Calc // Документация Краткое руководство. URL: <https://libreoffice.readthedocs.io/ru/latest/calc.html>.
7. Уоррен М. А. Применение электронных таблиц для решения задач математического моделирования в школьном курсе информатики : дис. – 2023.

## Справочная информация. Сочетания клавиш

Действие	Сочетание клавиш	Описание
<b>Работа с документом</b>		
Открыть окно «Поиск и замена»	Ctrl+F	Открыть диалоговое окно Поиск и замена, чтобы начать поиск символа / слова / фразы в редактируемом документе
Открыть окно 'Поиск и замена' с полем замены	Ctrl+H	Открыть диалоговое окно Поиск и замена с полем замены, чтобы заменить одно или более вхождений найденных символов.
Сохранить документ	Ctrl+S	Сохранить все изменения в редактируемом документе. Активный файл будет сохранен с текущим именем, в том же местоположении и формате
<b>Навигация</b>		
Перейти в начало строки	Home	Установить курсор в начале редактируемой строки
Перейти в начало документа	Ctrl+Home	Установить курсор в самом начале редактируемого документа
Перейти в конец строки	End	Установить курсор в конце редактируемой строки
Перейти в конец документа	Ctrl+End	Установить курсор в самом конце редактируемого документа
Увеличить	Ctrl++	Увеличить масштаб редактируемого документа
Уменьшить	Ctrl+-	Уменьшить масштаб редактируемого документа
Перейти на один символ влево	←	Переместить курсор на один символ влево
Перейти на один символ вправо	→	Переместить курсор на один символ вправо
Перейти в начало слова или на одно слово влево	Ctrl + ←	Переместить курсор в начало слова или на одно слово влево.
Перейти на одно слово вправо	Ctrl + →	Переместить курсор на одно слово вправо
Перейти на одну строку вверх	↑	Переместить курсор на одну строку вверх
Перейти на одну строку вниз	↓	Переместить курсор на одну строку вниз
<b>Написание</b>		
Закончить абзац	↵ Enter	Завершить текущий абзац и начать новый
Удалить	← Backspace, Delete	Удалить один символ слева (Backspace) или справа (Delete) от курсора

Действие	Сочетание клавиш	Описание
Создать неразрываемый пробел	Ctrl+⇧ Shift+␣ Пробел	Создать между символами пробел, который нельзя использовать для начала новой строки
Создать неразрываемый дефис	Ctrl+⇧ Shift+Hyphen (минус)	Создать между символами дефис, который нельзя использовать для начала новой строк
<b>Отмена и повтор</b>		
Отменить	Ctrl+Z	Отменить последнее выполненное действие
Повторить	Ctrl+Y	Повторить последнее отмененное действие
<b>Вырезание, копирование и вставка</b>		
Вырезать	Ctrl+X	Удалить выделенный фрагмент текста и отправить его в буфер обмена компьютера. Скопированный текст можно затем вставить в другое место этого же документа, в другой документ или в какую-то другую программу
Копировать	Ctrl+C	Отправить выделенный фрагмент текста в буфер обмена компьютера. Скопированный текст можно затем вставить в другое место этого же документа, в другой документ или в какую-то другую программу
Вставить	Ctrl+V	Вставить ранее скопированный текст из буфера обмена компьютера в текущей позиции курсора. Текст может быть ранее скопирован из того же самого документа, из другого документа или из какой-то другой программы
Вставить гиперссылку	Ctrl+K	Вставить гиперссылку, которую можно использовать для перехода по веб-адресу
Копировать форматирование	Ctrl+⇧ Shift+C	Скопировать форматирование из выделенного фрагмента редактируемого текста. Скопированное форматирование можно затем применить к другому тексту в этом же документе.
Применить форматирование	Ctrl+⇧ Shift+V	Применить ранее скопированное форматирование к тексту редактируемого документа.
<b>Выделение текста</b>		
Выделить все	Ctrl+A	Выделить весь текст документа вместе с таблицами и изображениями
Выделить фрагмент	⇧ Shift+→ ←	Выделить текст посимвольно
Выделить с позиции курсора до начала строки	⇧ Shift+Home	Выделить фрагмент текста с позиции курсора до начала текущей строки

Действие	Сочетание клавиш	Описание
Выделить с позиции курсора до конца строки	⇧ Shift+End	Выделить фрагмент текста с позиции курсора до конца текущей строки
Выделить до конца слова	Ctrl+⇧ Shift+→	Выделить фрагмент текста с позиции курсора до конца слова
Выделить до начала слова	Ctrl+⇧ Shift+←	Выделить фрагмент текста с позиции курсора до начала слова.
Выделить одну строку выше	⇧ Shift+↑	Выделить одну строку выше (курсор находится в начале строки).
Выделить одну строку ниже	⇧ Shift+↓	Выделить одну строку ниже (курсор находится в начале строки).
Выделить страницу вверх	⇧ Shift+Page Up	Выделить часть страницы с позиции курсора до верхней части экрана
Выделить страницу вниз	⇧ Shift+Page Down	Выделить часть страницы с позиции курсора до нижней части экрана.
<b>Оформление текста</b>		
Жирный шрифт	Ctrl+B	Сделать шрифт в выделенном фрагменте текста жирным, придав ему большую насыщенность
Курсив	Ctrl+I	Сделать шрифт в выделенном фрагменте текста курсивным, придав ему наклон вправо
Подчеркнутый шрифт	Ctrl+U	Подчеркнуть выделенный фрагмент текста чертой, проведенной под буквами
Зачеркнутый шрифт	Ctrl+5	Зачеркнуть выделенный фрагмент текста чертой, проведенной по буквам
Подстрочные знаки	Ctrl+.	Сделать выделенный фрагмент текста мельче и поместить его в нижней части строки, например, как в химических формулах
Надстрочные знаки	Ctrl+,	Сделать выделенный фрагмент текста мельче и поместить его в верхней части строки, например, как в дробях
Стиль Заголовок 1	Alt+1	Применить к выделенному фрагменту текста стиль Заголовок 1
Стиль Заголовок 2	Alt+2	Применить к выделенному фрагменту текста стиль Заголовок 2
Вставка разрыва страницы	Ctrl+⇧ Enter	Вставить разрыв страницы в текущей позиции курсора
Добавить номер страницы	Ctrl+⇧ Shift+P	Добавить номер текущей страницы в текущей позиции курсора
Непечатаемые символы	Ctrl+⇧ Shift+Num8	Показать или скрыть непечатаемые символы
Удалить один символ слева	← Backspace	Удалить один символ слева от курсора
Удалить один символ справа	Delete	Удалить один символ справа от курсора

Методические указания по выполнению практических работ

**Данилова Мария Игоревна**  
**Антошина Людмила Александровна**

## ИНФОРМАТИКА

### Часть 2

Методические указания по выполнению практических работ

для студентов 1 курса

07.02.01 Архитектура

08.02.01 Строительство и эксплуатация зданий и сооружений

08.02.08 Монтаж и эксплуатация оборудования и систем газоснабжения

08.02.12 Строительство и эксплуатация автомобильных дорог, аэродромов и  
городских путей сообщения

08.02.13 Монтаж и эксплуатация внутренних сантехнических устройств,  
кондиционирования воздуха и вентиляции

21.02.19 Землеустройство

очной и заочной формы обучения

Компьютерная вёрстка М. И. Данилова

Библиотека

ГБПОУ «Ставропольский строительный техникум»  
355035, г. Ставрополь, ул. Комсомольская, 73