Оглавление

[Термины 2](#_Toc161426927)

[§1. Введение. 3](#_Toc161426928)

[§2. Типы данных. 4](#_Toc161426929)

[§3. Обработчики информации. СУБД. 5](#_Toc161426930)

[Дополнительные типы данных. Операции. 7](#_Toc161426931)

[Структура данных. 7](#_Toc161426932)

[СУБД. 9](#_Toc161426933)

[Жизненный цикл базы данных. 13](#_Toc161426934)

[§4. Excel. 14](#_Toc161426935)

[§5. Информация. 16](#_Toc161426936)

[Информационные издержки. 17](#_Toc161426937)

[Свойства информационных продуктов. 17](#_Toc161426938)

[Защита информационных продуктов. 17](#_Toc161426939)

[Основные тезисы. 19](#_Toc161426940)

[Графическое изображение. 19](#_Toc161426941)

[Затраты информационного продукта. 20](#_Toc161426942)

[Технические задания. 24](#_Toc161426943)

[§6. Виды программного обеспечения. 25](#_Toc161426944)

[§7. Системы обработки данных. Обработка данных на Excel. 30](#_Toc161426945)

# Термины

**Информация** – набор знаний о каком-то объекте, явлении, действии.

**Файл** – это место на некотором носителе информации, которое имеет своё имя.

**Реляция (с англ. relation)** – отношение, относительно.

Совокупность таблиц – **база данных**.

**Структура данных** – это схема представления информации для того, чтобы осуществить удобный и быстрый поиск необходимых данных.

**Клиент** – объект, который получает информацию и подаёт запросы.

**Сервер** – объект, который принимает запросы и передаёт информацию.

**Атрибут** – название столбца таблицы данных.

**Картеж** – конкретная запись строки.

**Информационный продукт** – это документированная информация, подготовленная в соответствии пользователя, представленная в виде товара.

Графическое изображение в виде 0 и 1 называется **битовая плоскость**.

# §1. Введение.

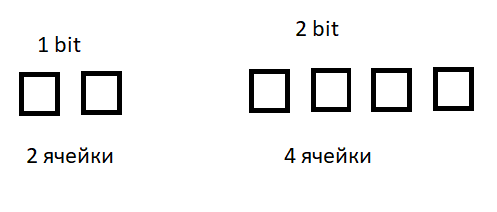
Информационные технологии – просматривают методы сбора, хранения и обработки информации.

**Информация** – набор знаний о каком-то объекте, явлении, действии.

**Элементарная информация:**

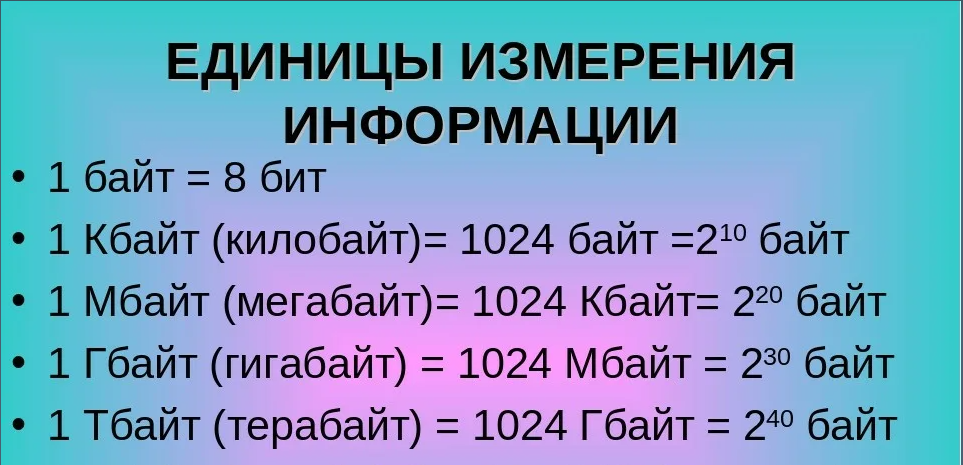
**Измерение информации:**

1 бит – это ячейка памяти, которая может принимать только 2 значения, либо 0, либо 1





8 bit – или ячейка памяти, которая принимает 256 значений – 1 byte(байт).



# §2. Типы данных.

1. Двоичный тип данных – Boolean.

Тип данных, который подобен биту, принимает значение да (1) или нет (0).

2. Целочисленный тип данных – Integer (Int).

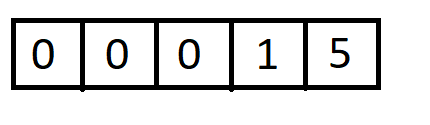
Обычное целое число в некоторых границах.

(Примечание из интернета) Целый тип int имеет размер 4 байта (32 бита). Минимальное значение -2 147 483 648, максимальное значение 2 147 483 647.

3. Действительный тип данных – Real (Float).

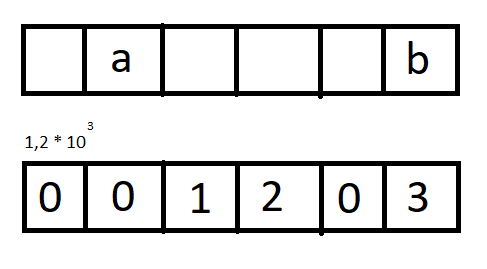
Представляет из себя число в виде десятичной дроби, записанный с фиксированной или плавающей точкой.

Фиксированная точка – 1.5



Плавающая точка - 

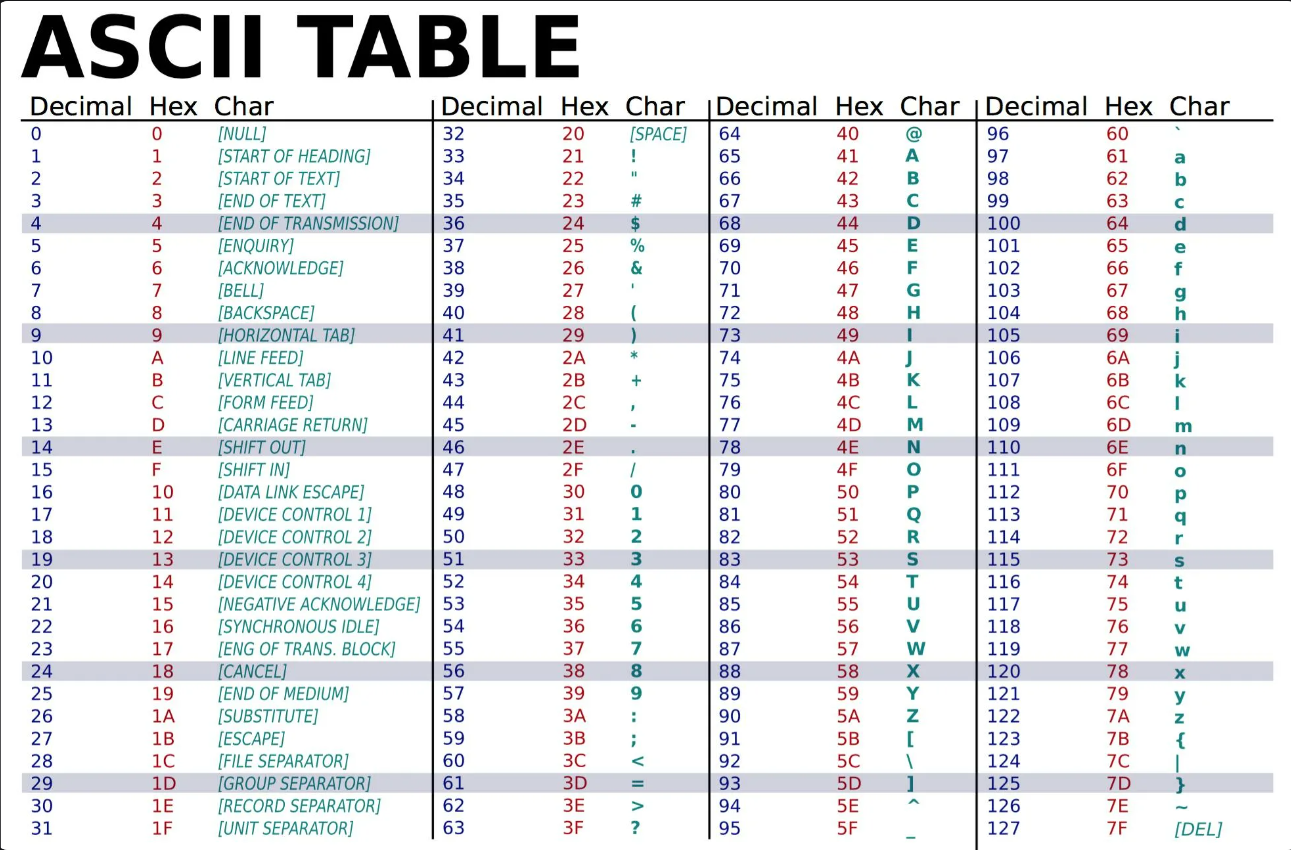
Где a – мантисс, b – степень



4. Строковый тип данных – набор текстовой информации (String). В этом случае каждый символ кодируется некоторым числом. Используется таблица ASCII -Американский стандартный код управления памяти. Таблица ASCII использует для кодирования 1 байт. Таблица состоит из двух частей:

1. Фиксирована. В ней все буквы латиницы, каждый символ имеет свой код. A – 65.

2. Может быть изменена. Для каждой страны она своя.



[Подробнее про ASCII таблицу.](https://grandidierite.github.io/ASCII-table/)

Но с течением времени таблица ASCII устарела и изжила себя. В наше время повсеместно используется система Unicode.

В системе Unicode каждый символ представляется в двух байтах. Первый байт – код таблицы (код страны), второй байт – код символа. Unicode использует для кодирования 2 байта.

# §3. Обработчики информации. СУБД.

Для обработки информации используются различные системы.

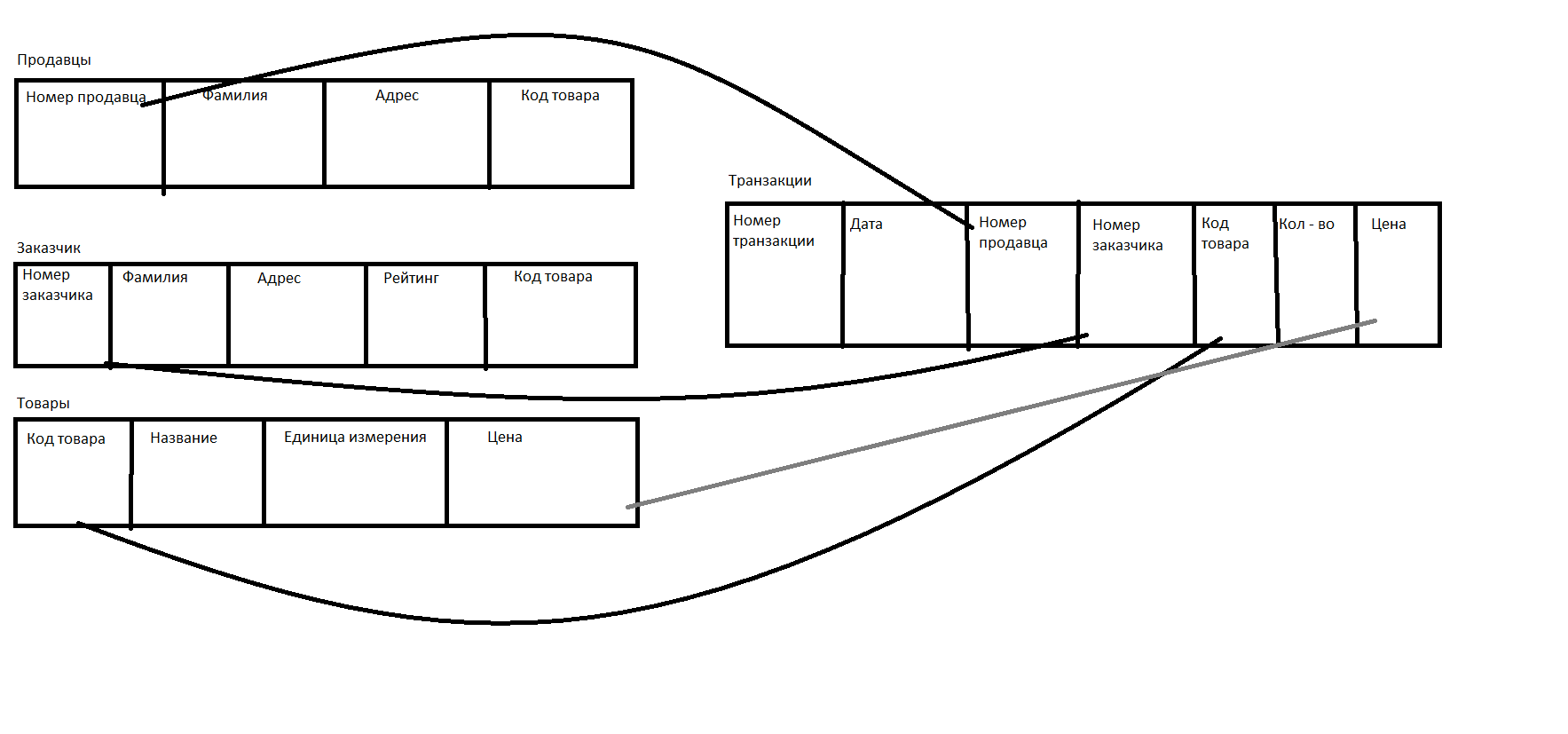
Специализированная система называется СУБД – система управления базами данных. В этом случае информация хранится внутри некоторых файлов, которые называются базами данных. Обычно базы данных имеют большой объём.

**Файл** – это место на некотором носителе информации, которое имеет своё имя.

На сегодняшний день чаще всего используются реляционные базы данных.

**Реляция (с англ. relation)** – отношение, относительно.

В этом случае база данных представляет из себя набор таблиц, связанных между собой некоторыми отношениями:



Связи бывают двух типов – первичные и вторичные.

Совокупность таблиц – база данных.

## Дополнительные типы данных. Операции.

**Дополнительные типы данных:**

1. Дата и время.

2. Ссылочные.

**Операции:**

Каждый тип данных имеет свои операции. Типы данных определяются теми операциями, которые над ними можно провести:

Логические операции (для [Boolean](#_§2._Типы_данных.)) – OR (или), And (и), Not (отрицание).

Целочисленные операции (для [Int](#_§2._Типы_данных.)) – сложение, вычитание, умножение, целочисленное деление.

Действительные операции (для [Float](#_§2._Типы_данных.)) – все арифметические операции.

В строковом ([String](#_§2._Типы_данных.)) типе возможны только следующие операции:

1. Контактенация – присоединение.

2. Выделение подстроки.

3. Поиск.

4. Сравнение двух строк.

В дата и время возможно прибавление и вычитание любого количества дней.

Ссылочный тип – построение различных структур данных.

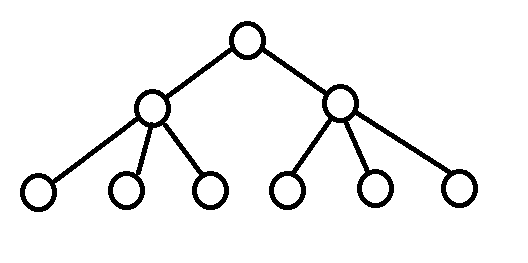
## Структура данных.

**Структура данных** – это схема представления информации для того, чтобы осуществить удобный и быстрый поиск необходимых данных.

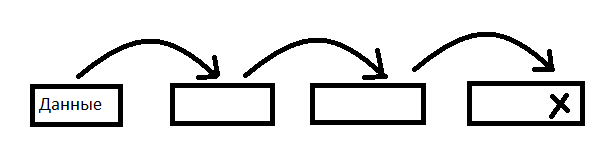
Самая простая структура данных – таблица.

1. Таблица состоит из строк и столбцов, при этом каждый столбец имеет своё название, а строки свой номер.

2. Дерево – под деревом подразумевается структура, где имеется некоторый элемент – корень дерева. У каждого элемента может быть несколько дочерних элементов. Но у каждого элемента имеется только один родитель. Связь указывает кто кому является.

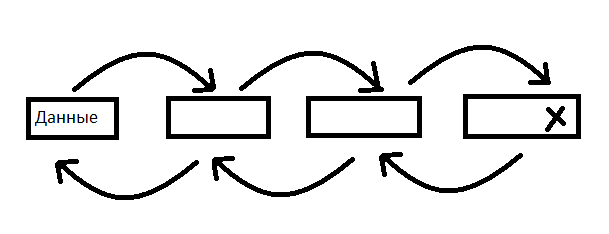


3. Списки – списком называется набор данных, когда вместе с данными хранится указатель на следующий набор данных.



Главное свойство списков – возможность удобного добавления и удаления данных. Позволяют добавлять и удалять данные в ходе работы.

Списки бывают однонаправленные (как на рисунке выше) и двунаправленные:



Минус однонаправленных заключается в том, что если потеряется ссылка, то потеряются и данные.

## СУБД.

СУБД (системы управлением базами данных) – они бывают открытыми и закрытыми.

Открытые – позволяют открывать и изменять систему и базу данных.

Закрытые – не позволяют вносить изменения в саму базу данных, и в систему обработки.

История:

Раньше на больших вычислительных машинах была система R. В это же время был разработан язык SQL – язык запросов – лежит в основе всех БД (баз данных) систем в наше время.

Системы Informix и Oracle были двумя конкурирующими системами, но система Informix не выдержала конкуренции.

Oracle – используется во всех коммерческих компаниях.

Многие системы работают по принципу Клиент -> сервер.

**Клиент** – объект, который получает информацию и подаёт запросы.

**Сервер** – объект, который принимает запросы и передаёт информацию.

Были попытки ввести в систему **утончённого клиента**, но она не получила своего развития.

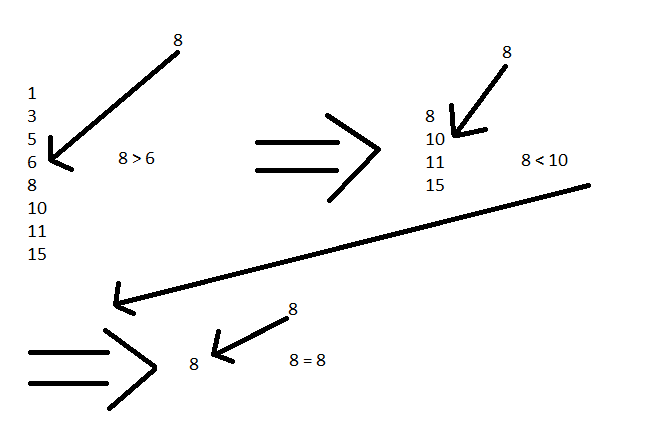
Данные в БД (базах данных) обычно представлены в виде таблиц, и к ним добавляются файлы индексов.

Файл индексов представляет из себя некоторый ключ и номер данных в исходной таблице. Это необходимо для того, чтобы осуществлять быстрый поиск информации.

Если рассматривать совокупность исходной таблицы и индексированного файла, то можно считать, что данные этой таблицы отсортированы по некоторому показателю.

Данные называются отсортированы, когда они располагаются в некотором порядке.

Отсортированные данные позволяют осуществить быстрый поиск (двоичный поиск). Данные сравниваются по величине со средним элементом и в дальнейшем рассматривается половина находящееся меньше этого числа или больше.



Количество запросов определяется по формуле:

K – количество, n – запрос (как на рисунке запрос 8)

K = 

n = 1000 => k = 10

Построение индексных файлов позволяет сэкономить место. Поэтому вся база данных состоит из набора данных, то есть большое количество таблиц. Все эти таблицы связанны некоторыми отношениями, и к этим таблицам приложены индексные файлы.

Такая система позволяет осуществлять быстрый поиск необходимой информации.

Таблица может быть представлена в виде дерева. Но чаще всего такие данные представляются в виде списка.

Все таблицы можно связать. Для связи между таблицами используются ключи.

Ключ – некоторое значение позволяющее определить запись или набор записей, значение которой равно данному ключу.

Ключ может состоять из комбинации полей.

Все записи в такой базе данных должны быть между собой согласованны. То есть, по заданному ключу, в новой таблице, должна существовать запись. Запись не обязательно должна быть уникальной.

**Атрибут** – название столбца таблицы данных.

**Картеж** – конкретная запись строки.

Объединением отношений является отношение всех картежей, которые присутствуют в обоих отношениях.

Объединение обозначается:

При этом все дублирующие кортежи игнорируются.



Пересечением двух отношений является отношение, которое присутствует в обоих отношениях.

Обозначается:



Декартово произведение двух отношений является отношение представляющее из себя объединение всех заголовков, а значениями являются все возможные картежи.

Обозначается:

R x S

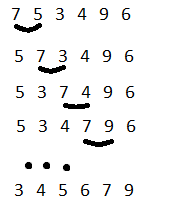
Для построения индексного файла используются различные виды сортировки. **Под сортировкой подразумевается расположение данных в определённом порядке:**

1. Пузырьковая сортировка.

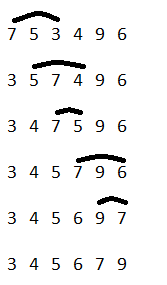
2. Сортировка вставками.

3. Быстрая сортировка.

1. Сравниваются два соседних элемента и их располагают в нужном порядке. Проводя несколько раз, получается отсортированный массив.



2. На каждом шаге находится минимальный элемент и меняется местами с соответствующим элементом.



3. Заключается в следующем. Выбирается некоторый элемент, и все элементы, которые меньше его, записываются в первую часть массива, а все что больше, во вторую часть. Потом части сортируются и объединяются.

Когда данные отсортированы, мы можем найти нужные данные используя метод [двоичного поиска](#ДвоичныйПоиск).

## Жизненный цикл базы данных.

1. Планирование базы данных:

Заключается в постановке задачи, написания технического задания. Разработка инструкции. Проектирование базы данных. Выбор СУБД

2. Проектирование базы данных:

Выбор таблиц. Выбор отношений между таблицами. Выбор систем запросов.

3. Создание базы данных – физическое.

4. Тестирование.

5. Исправление ошибок.

6. Функционирование системы.

Все СУБД функционируют на основе запросов. Запросом называется команда, которая даётся программе, работающей с базой данных, при выполнении которой, получается таблица, соответствующая данному запросу. Эта таблица может быть сохранена в БД или предоставлена поиску.

# §4. Excel.

Excel представляет из себя набор таблиц, связанных между собой.

Каждый набор таблиц называется книгой.

Книга – состоит из листов.

Лист представляет из себя электронную таблицу.

Таблица состоит из строк и столбцов.

Каждый столбец имеет название, состоящее из латинских букв:



После того как заканчиваются буквы алфавита:

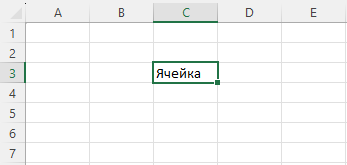


И так до IV.

Каждая строка имеет свой номер:



На пересечении столбца и строки находится ячейка:



Каждая ячейка имеет адрес, который состоит из:

Названия книги ! имя листа ! название столбца -> номер строки.

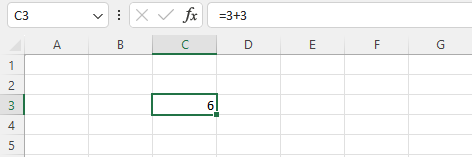
Но обычно первую половину адреса игнорируют, поэтому адрес ячейки — это название столбца + номер строки. Например A11, B23, IV55.

Кроме того, в каждой ячейке можно задавать имя, которое в дальнейшем использовать.

В ячейке могут содержаться данные следующих типов: Числовые, строковые, даты и времени, формулы. Для преобразования одного типа в другой, используется большой набор стандартных функций.

В каждую ячейку можно вписать формулы.

Отображение на экране может отличаться от внутреннего представления данных:



Формулы начинаются со знака равно, и содержат в себе адреса ячеек, разделённые знаками операций, а также стандартными функциями.

Используется всего 5 операций: + - \* / ^(степень)

При использовании данных, формулы можно копировать или протягивать. При этом используется абсолютная или относительная адресация. Абсолютная адресация - $. Фиксирует столбец или строку.

[Работа первого урока.](It%20в%20менеджменте%201%20урок.xlsx) (если файл конспектов скачан в виде архива)

# §5. Информация.

По методу использования, информация бывает двух видов: **явная и неявная**.

**Явная** информация может быть формализована, она может быть представлена в общедоступном виде и располагаться на одном из носителей информации.

**Носители информации:**

1. Механические – доска, камень.

2. Бумажные – книги.

3. Магнитные – жёсткий диск.

4. Электронные – флешка.

5. Оптические – CD диск.

Неявная информация – это совокупность обработанной информации, представленная как единое целое и имеющая практическое применение: изучение языков (пример).

Неявная информация не может быть хранима на вышеперечисленных носителях.

**Кроме того, информацию можно подразделить на:**

1. Универсальная.

2. Рассеянная.

3. Смешенная.

1. Под универсальной информацией подразумевается совокупность знаний, которые могут быть использованы в каких-либо действиях.

2. Рассеянная информация – это информация, которая может быть освоена в течении некоторого времени: изучение языков и др.

3. Смешанная информация – сочетание явной и неявной информации: создание бизнеса/онлайн торговли.

## Информационные издержки.

**Для обеспечения использования информации необходимы информационные издержки. Они включают в себя:**

1. Затраты на приобретение информационного продукта.

2. Затраты на обучение и приобретение опыта.

3. Затраты на сбор информации в организации.

4. Затраты на сбор информации о количестве продукции, сопутствующих условиях.

5. Затраты на услуги консультантов.

На современном этапе передача информации осуществляется по различным каналам связи, что значительно удешевляет поиск информации и её использование.

## Свойства информационных продуктов.

**Основные Свойства информационных продуктов:**

1. Использованная информация не уничтожается, поэтому информационный продукт может быть использован самым широким пользователем.

2. Информационный продукт используется субъективно.

3. Информационные продукты имеют ценность, которая познаётся на опыте.

4. Информация может быть первичной, вторичной и т.д. (то, что говорили на лекции – первичная информация, этот файл – вторичная)

5. Простота и дешевизна копирования продукта, порождает необходимость защиты интеллектуальной собственности. Информация также должна быть защищена различными системами шифрования.

## Защита информационных продуктов.

**Любой информационный продукт должен быть защищён. Для шифрования информации используются различные способы:**

1. Подстановочный – замена букв.

На современном этапе подстановочные шифры практически не используются. Поэтому разрабатываются различные схемы шифрования.

2. Простейшая схема шифрования на сегодняшний день – XOR.

Она заключена на использовании операций XOR. Иногда её называют “исключающее или”

|  |  |  |
| --- | --- | --- |
| **X** | **Y** | **X xor Y** |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

Любая информация на современном носителе представляется в двоичном виде. Поэтому для шифрования используется двоичная операция. Операция XOR – двоичная операция. Для того чтобы зашифровать информацию используется ключ.

**Пример:**

Текст – 010 111 000 011

Ключ – 010 010 010 010

Используя таблицу XOR, получаем следующее:

0 (из текста) и 0 (из ключа) = 0

1 и 1 = 0

0 и 0 = 0

1 и 0 = 1

1 и 1 = 1

И т.д получаем:

000 101 010 001

Основное преимущество XOR, если к шифротексту применить ключ, получим исходный текст.

Но это схема шифрований не обладает хорошей защищенностью.

Во время Войны (1941), для связи между военными кораблями использовалась система шифрования Enigma. Кроме того, на современном этапе используются более современные системы шифрования. Эти системы заключаются в том, что математические операции асинхронны. На основании этого основаны все современные шифры.

Асинхронность – это когда действие в одну сторону проходит легко, а в обратную очень сложно.

## Основные тезисы.

1. **Информационный продукт** – это документированная информация, подготовленная в соответствии пользователя, представленная в виде товара.

Если продукт не предоставляется потребителю, то предоставляется доступ к нему, но при этом говорят о том, что производится информационная услуга.

2. Любую информацию можно рассмотреть в трёх ракурсах:

2.1. Синтаксическом.

2.2. Семантическом.

2.3. Прагматическим.

2.1. Синтаксический ракурс – рассматривает кодирование информации, при передаче, хранении и доступе. Любая текстовая информация представима в виде битов. Некоторая графическая информация также представима в виде битов.

Любое графическое изображение можно предоставить в виде 0 и 1.

Графическое изображение. В виде 0 и 1 называется **битовая плоскость**.

На компьютерах любое изображение на сегодняшний день использует RGB представление. Каждый цвет кодируется тремя цифрами составляющие красного, зелёного и синего. Наиболее часто используются величины 0 – 255.

Система кодирования следующая – используется 16тиричная система исчисления:



Для использования единичной цифры (например, 3) используется пол байта, для десятичной цифры используется 1 байт (например, 10).

Для использования одной составляющей цвета используется 1 байт.

Для того чтобы использовать 1 ячейку цвета, используется 3 байта или 24 бита.

Вся графическая информация представляется в виде пикселей.

**Коды некоторых основных цветов:**

Красный: FF0000

Зелёный: 00FF00

Синий: 0000FF

Белый: FFFFFF

Черный: 000000

Таким образом можно получить  различных цветов.

## Затраты информационного продукта.

**Затраты на реализацию информационного продукта:**

1. Издержки производства:

1.1. Разработка содержания продукта.

1.2. Разработка оформления продукта.

1.3. Тестирование.

2. Издержки тиражирования.

3. Издержки распространения.

3.1. Маркетинг.

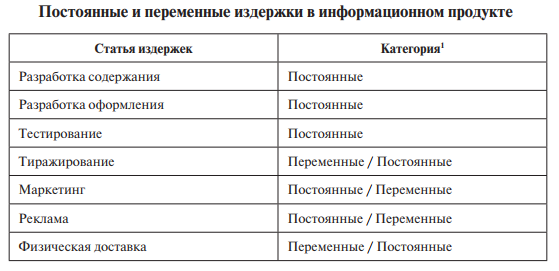
3.2. Реклама.

3.3. Доставка продуктов потребителю.

**Издержки бывают:**

1. Постоянные – не зависят от количества копий.

2. Переменные – зависят от количества копий.



Все издержки в информационно продукте – невозвратные.

Обычно переменные издержки существенно ниже постоянных. Поэтому при большом объёме продаж, общие затраты издержек на единицу информационного продукта примерно равен (формула):

Затраты на создание / число проданных копий.

Поэтому объём продаж становится фактором успеха информационного продукта.

Кроме того, борьба за объём продаж ведётся любыми законными средствами, в том числе средствами ценовой политики.

Примером может послужить борьба между Oracle и Informix. Oracle выиграл за счёт того что был дешевле и доступнее.

**Ценовая политика заключается в следующем:**

1. Каждому покупателю продукт продаётся по особой цене, учитывающей эластичность спроса.

2. Цена зависит от количества приобретённого товара.

Кроме того, цена различается для различных групп покупателей.

**У информационных продуктов существуют версии. Каждая следующая версия допускает:**

1. Исправление ошибок предыдущей версии.

2. Включение дополнительных возможностей.

3. Изменение оформления.

Обычно продукты выпускаются внутри одной версии в трёх видах: Basic, Standard, Premium.

Basic – содержит основные функциональные возможности и обычно поставляется бесплатно.

Standard – содержит более развёрнутые возможности и обычно поставляется условно-бесплатно, то есть имеет период бесплатного пользования.

Premium – содержит все возможные функциональные возможности и обычно поставляется на платной основе.

Кроме того, существуют бета версии, в этом случае поставщик объявляет, что не несёт ответственность за использование этого продукта.

**Кроме того, информационные продукты подразделяются по цене. Создание версий также основано на:**

1. Сегментация клиентов.

2. Различимость версий отдельными группами клиентов.

3. Выделение характеристик, которые могут служить барьерами в использовании.

Пример:

Продукт R-studio – используется для получения различных характеристик статистических данных. Имеются версии R-studio для студентов, врачей, госучреждений.

**Очень важным для программного продукта является его сопровождение.**

**Сопровождение включает в себя:**

1. Консультация по использованию.

2. Исправление неточностей.

3. Замена устаревших версий и установка дополнительных возможностей.

Обычно сопровождение оговаривается в условиях поставки продукта.

**История:**

Американское правительство подало иск на Microsoft за монопольное использование, и потребовала открыть исходный код продукта. Microsoft разделили исходный код на 100 частей, и отправили в 100 стран мира. В итоге никто не хотел делиться своими частями.

**Программные продукты имеют 2 формы представления:**

1. Исходный текст.

2. Двоичный/программный код.

1. Исходный текст может быть изменён пользователем и возможен к использованию в других программных продуктах.

2. Представляет из себя набор команд для компьютера в виде набора битов.

Двоичный код практически невозможно изменить, и из двоичного кода практически невозможно восстановить исходный код.

**Программные продукты подразделяются на:**

1. Операционные системы.

2. Офисные программы.

3. Специализированные программы – для узких целей.

4. Обучающие программы.

5. Обслуживающие программы – утилиты.

## Технические задания.

**Для того чтобы создать информационный продукт, вначале составляется техническое задание (ТЗ). ТЗ содержит в себе:**

1. Цели использования информационного продукта.

2. Условия, при которых этот продукт используется, описание дополнительных программных продуктов, необходимых для использования данного продукта.

3. Инструкция по использованию продукта.

4. Инструкция по внесению исправлений.

5. Инструкция по действиям в исключительные ситуации.

Домашнее задание №1: Написать ТЗ на разработку программного продукта Начисления зарплаты.

# §6. Виды программного обеспечения.

По использованию и созданию программ обеспечения подразделяются на 3 вида:

1. Трансляторы.

2. Интерпретаторы.

3. Интегрированные системы.

Программы делятся на 2 вида:

1. Исходный текст.

2. Двоичный код.

Пример: необходимо сложить 2 числа в памяти компьютера.

На исходном тексте запись принимает вид: c=a+b

Где а, b, c – ячейки памяти, записанные как a, b, c

Такая запись представима в исходном тексте.

В памяти компьютера это записывается как:

03 00001F 0014 – c

03 000027 0015 - =

05 00001F 0027 – a

08 00001F 0027 - +

06 00001F 0027 – b

Для перевода исходного текста в память компьютера, который нужно выполнить, используются специальные программы:

1. Трансляторы.

2. Интерпретаторы.

3. Интегрированные системы.

1. Трансляторы переводят весь исходный текст в двоичный код и сохраняет в виде файла. Преимущество: то, что проделано один раз, можно использовать многократно. Недостаток: при наличии ошибок нужно заново транслировать программу и заменять старую на новую.

2. Интерпретаторы переводят каждую команду в двоичный код и сразу исполняет, после этого рассматривает следующую команду. Преимущество: они позволяют сразу обнаружить ошибки.

3. На сегодняшний день, чаще используются интегрированные системы. Которые сочетают преимущества и интегратора и интерпретатора. Они позволяют как интегрировать программу, то есть выполнять по шагу, так и строить двоичный код. Примером таких программ служат продукты фирмы Borland, такие как: Delphi, Си. Также продукты Microsoft – Visual studio.

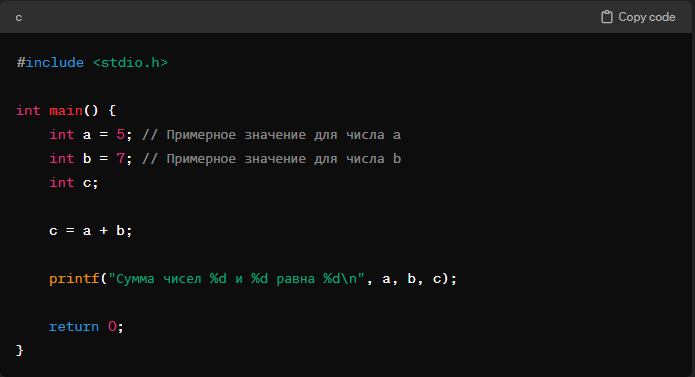
Программы на языках подразделяются на:

1. Программы высокого уровня.

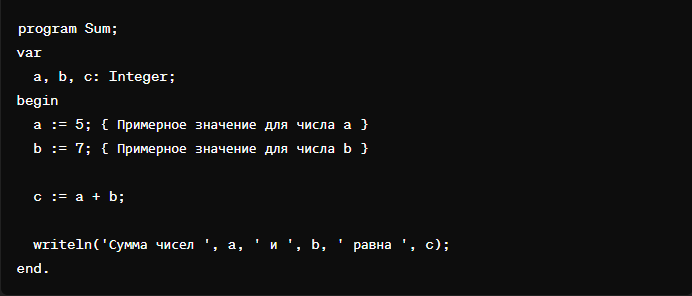
2. Программы низкого уровня.

1. Программы на языке высокого уровня записываются в виде команд более понятных человеку. Примеры:

Язык СИ:



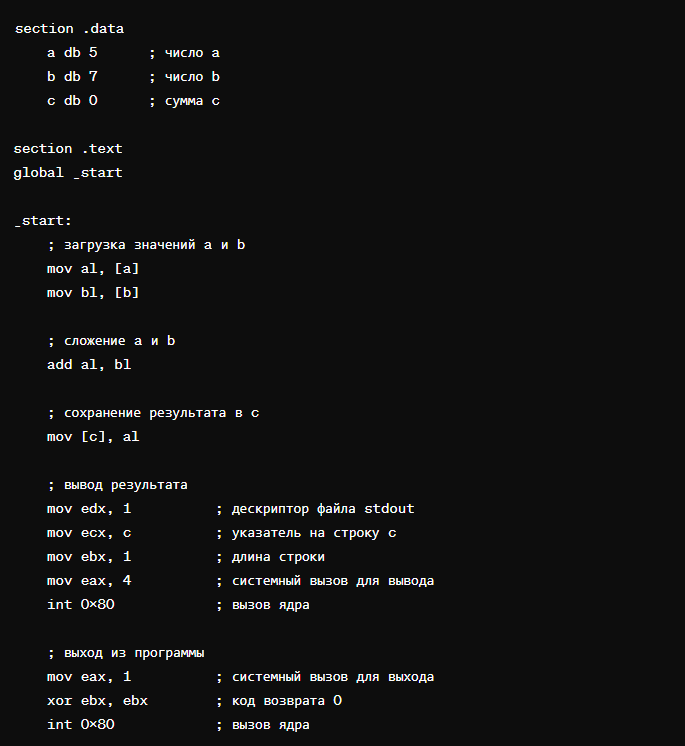
Pascal:



Все программы сохраняются в виде текстового файла и потом могут использоваться многократно. Однако для интерпретатора необходимо наличие самого интерпретатора. Для транслятора его наличие неважно.

Примером интерпретатора служит Python. Для того чтобы его использовать, нужно его установить.

2. Языком низкого уровня (машиноориентированный язык) является язык Ассемблера, где в этом языке каждая команда представима в виде некоторой мнемоники. Сложение двух числе на языке Ассемблера будут иметь вид:



Программа на языке Ассемблера изначально имеет большой объём, однако при трансляции и выполнении имеет большую скорость.

Создание интерпретатора или интегрированной системы на языке Ассемблера, представляет из себя значительные трудности. Программа на языке Ассемблера использует все ресурсы компьютера, но эти же ресурсы нужно использовать для самой интегрированной системы или работы интерпретатора.

Большую сложность также представляет из себя дисассемблирование. Процесс восстановления ассемблерного кода из двоичного.

Для написания программ используется алгоритмы. Алгоритмом называется описание набора элементарных операций, позволяющий решить данную задачу.

Алгоритм начисления налога:

Налог начисляется как 12% от всех начислений на человека.

1. Взять зарплату человека.

2. Прибавить надбавку.

3. Прибавить больничный.

4. Прибавить премию.

5. Вычислить 12% от всей суммы.

6. Из основной суммы вычислить аванс и налог.

7. Полученная сумма к выдаче.

Любой алгоритм переводится в программу на языке высокого уровня, затем транслируется в двоичном коде.

Алгоритмы бывают трёх видов:

1. Линейные.

2. Разветвляющие.

3. Цикличные.

1. Линейный алгоритм состоит из команд, которые выполняются последовательно друг за другом.

2. Разветвляющийся алгоритм состоит из команд, часть из которых выполняется в зависимости от какого-либо условия.

3. Цикличный алгоритм состоит из команд, которые выполняются многократно, до тех пор, пока выполнено некоторое условие.

Алгоритмы состоят из участков, которые описаны выше.

ДЗ: составить алгоритм учёта затрат строительной фирмы.

# §7. Системы обработки данных. Обработка данных на Excel.

Для обработки данных в Excel имеется большой набор стандартных функций, которые разделены на категории:

1. Математическая.

2. Логическая.

3. Статистическая.

4. Текстовая.

5. Массивы и ссылки.

6. Работа с базой данных.

7. Дата и время.

8. Работа с ошибками.

9. Финансовая.

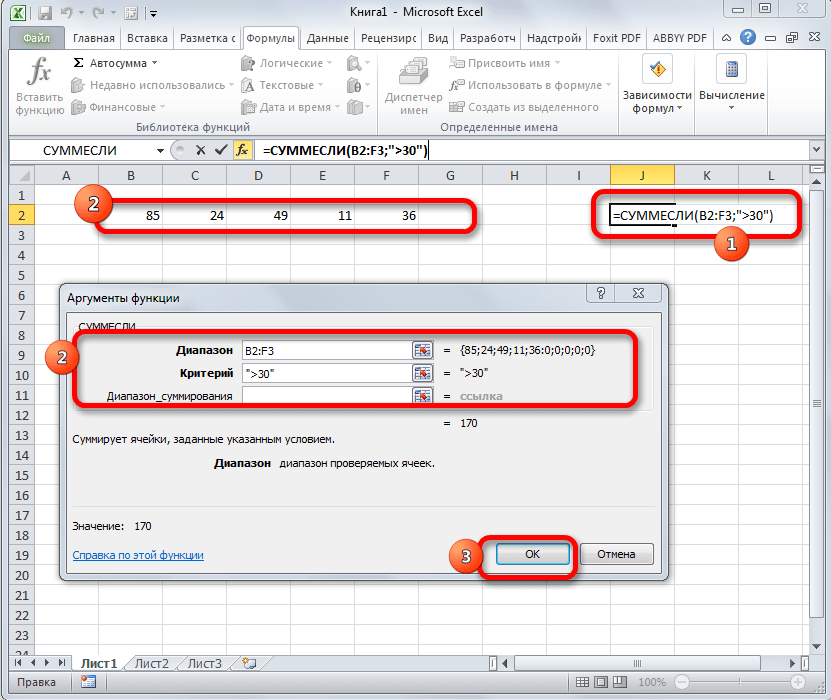
Помимо этих категорий, имеется полный перечень всех функций.

1. К математическим функциям относятся все элементы математических функций: exp, log, sin, cos…

А также стандартные функции:

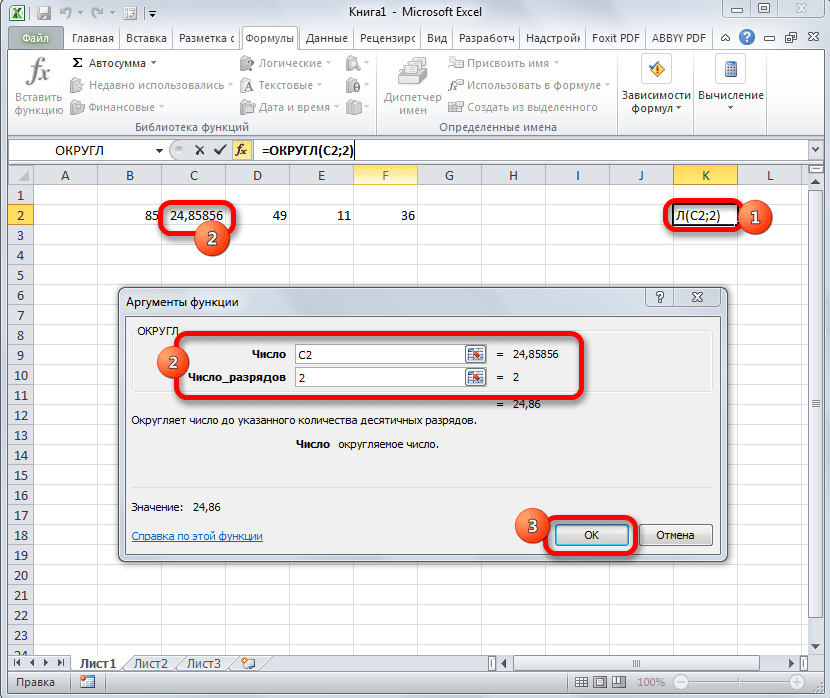
**=СУММ(число1;число2;…)**

В окне аргументов в поля следует вводить ссылки на ячейки с данными или на диапазоны. Оператор складывает содержимое и выводит общую сумму в отдельную ячейку.



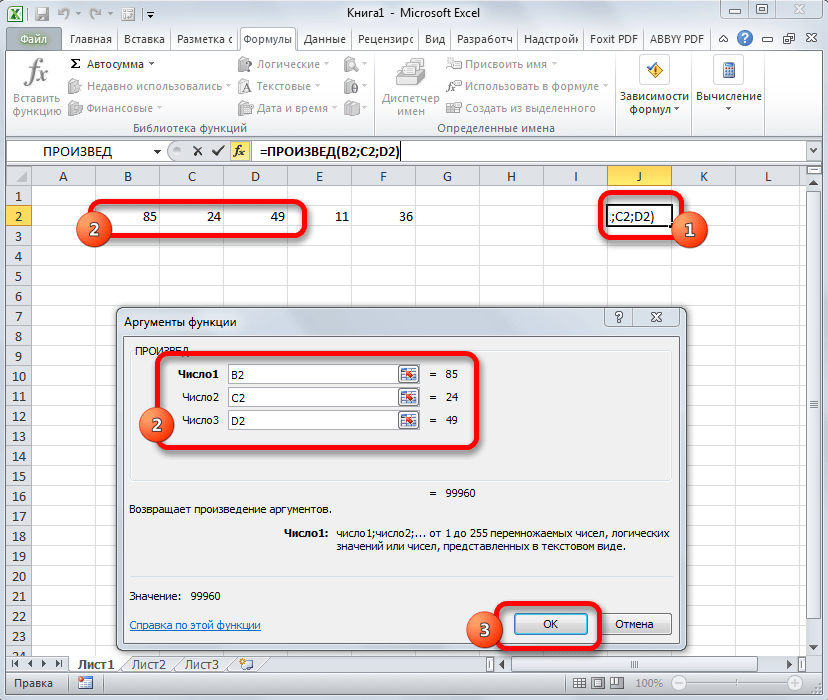
**=ОКРУГЛ(число;число\_разрядов)**

Кроме того, в Экселе существуют такие функции, как **ОКРУГЛВВЕРХ** и **ОКРУГЛВНИЗ**, которые соответственно округляют числа до ближайшего большего и меньшего по модулю.



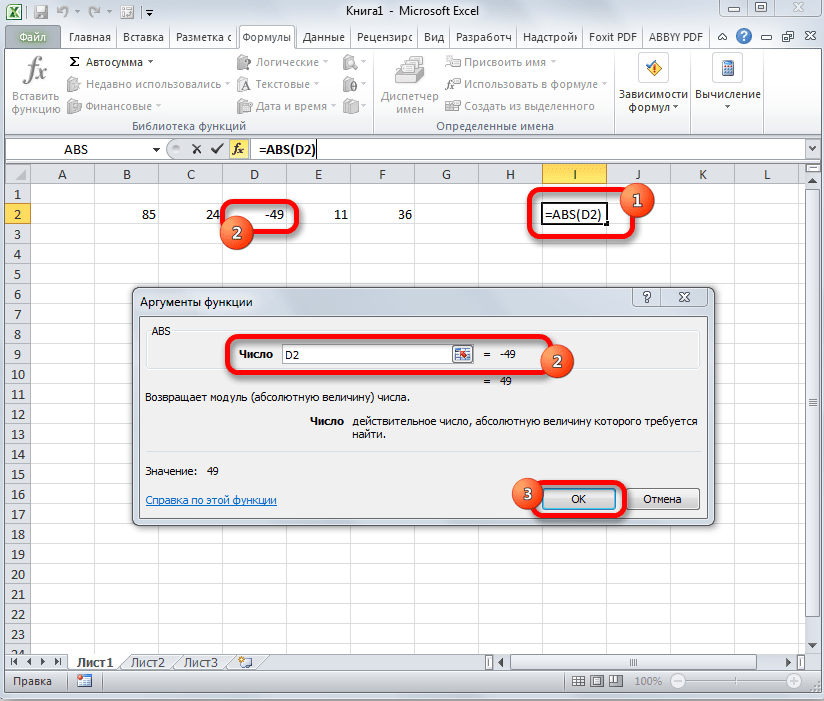
**=ПРОИЗВЕД(число;число;…)**

Задачей оператора **ПРИЗВЕД** является умножение отдельных чисел или тех, которые расположены в ячейках листа. Аргументами этой функции являются ссылки на ячейки, в которых содержатся данные для перемножения. Всего может быть использовано до 255 таких ссылок. Результат умножения выводится в отдельную ячейку.



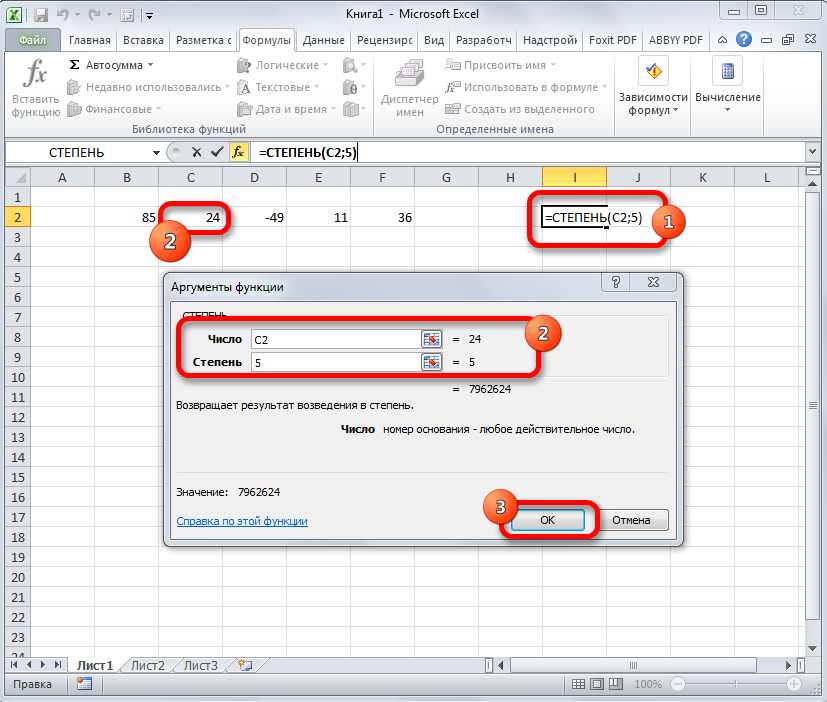
**=ABS(число)**

С помощью математической формулы **ABS** производится расчет числа по модулю. У этого оператора один аргумент – **«Число»**, то есть, ссылка на ячейку, содержащую числовые данные. Диапазон в роли аргумента выступать не может.



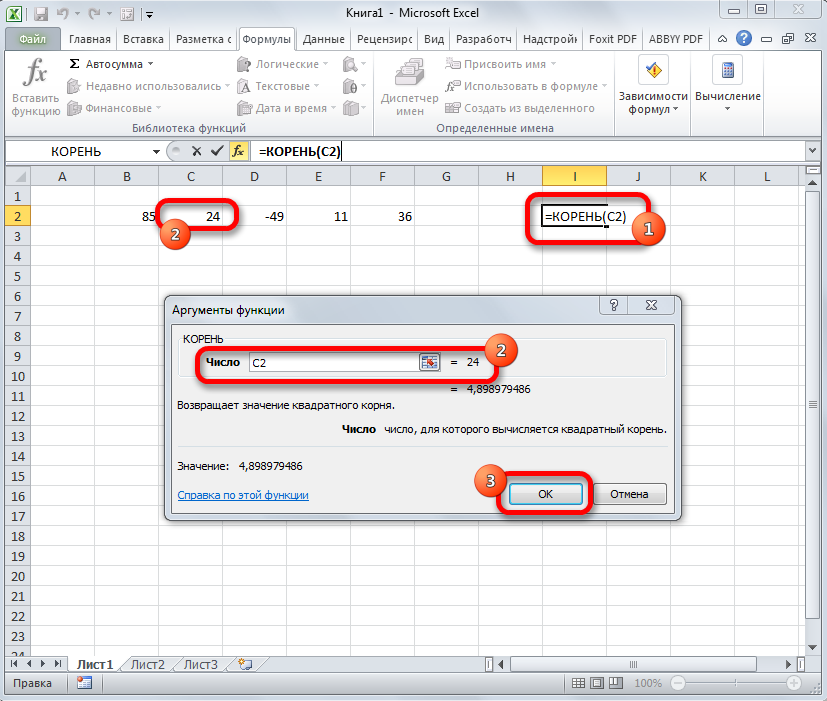
**=СТЕПЕНЬ(число;степень)**

Из названия понятно, что задачей оператора **СТЕПЕНЬ** является возведение числа в заданную степень. У данной функции два аргумента: **«Число»** и **«Степень»**. Первый из них может быть указан в виде ссылки на ячейку, содержащую числовую величину. Второй аргумент указывается степень возведения.



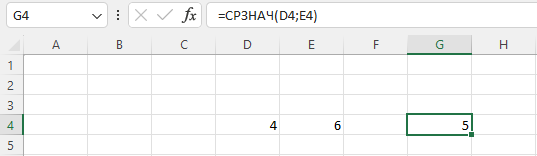
**=КОРЕНЬ(число)**

Задачей функции КОРЕНЬ является извлечение квадратного корня. Данный оператор имеет только один аргумент – «Число». В его роли может выступать ссылка на ячейку, содержащую данные.



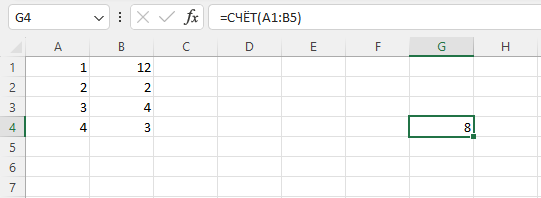
**=СРЗНАЧ(число1;число2;…)**

Вычисляет среднеарифметическое всех объединений.



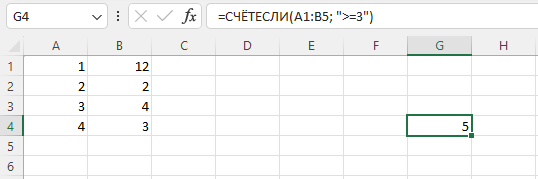
**=CЧЁТ(начало диапазона:конец диапазона)**

Вычисляет количество заполненных ячеек в заданном диапазоне.



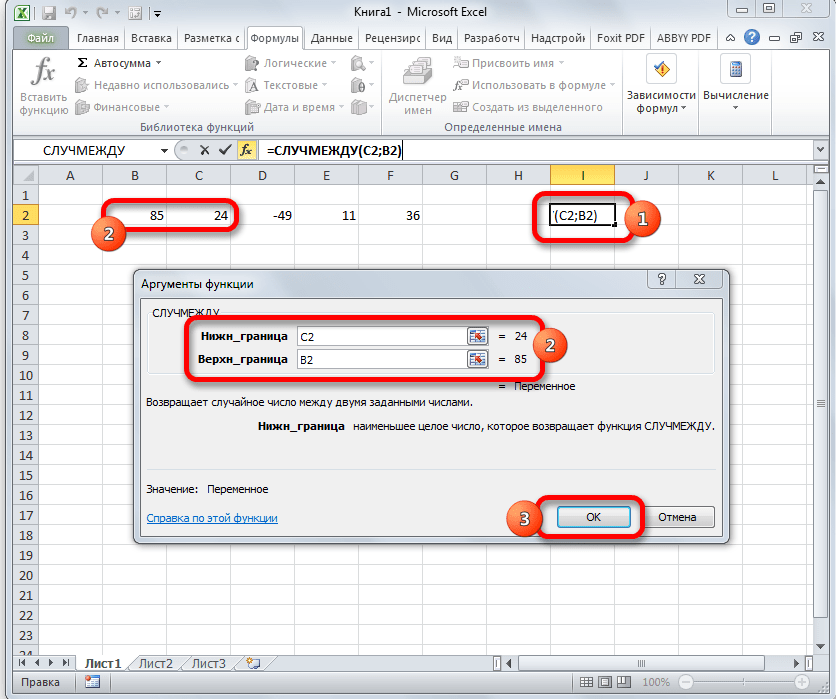
**=CЧЁТ(начало диапазона:конец диапазона)**

Вычисляет количество заполненных ячеек в заданном диапазоне, с определённым критерием.



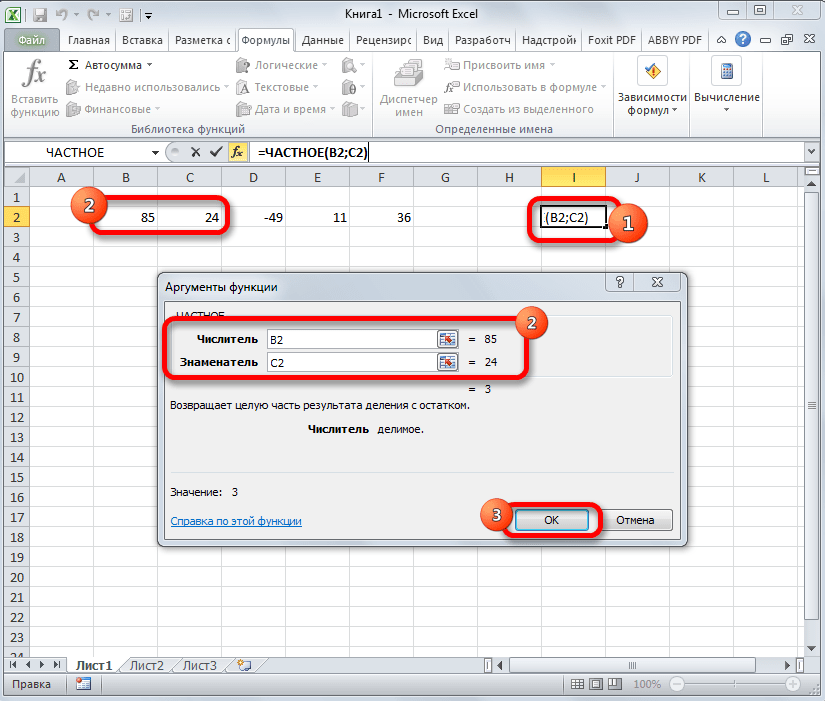
**=СЛУЧМЕЖДУ(Нижн\_граница;Верхн\_граница)**

Довольно специфическая задача у формулы СЛУЧМЕЖДУ. Она состоит в том, чтобы выводить в указанную ячейку любое случайное число, находящееся между двумя заданными числами. Из описания функционала данного оператора понятно, что его аргументами является верхняя и нижняя границы интервала.



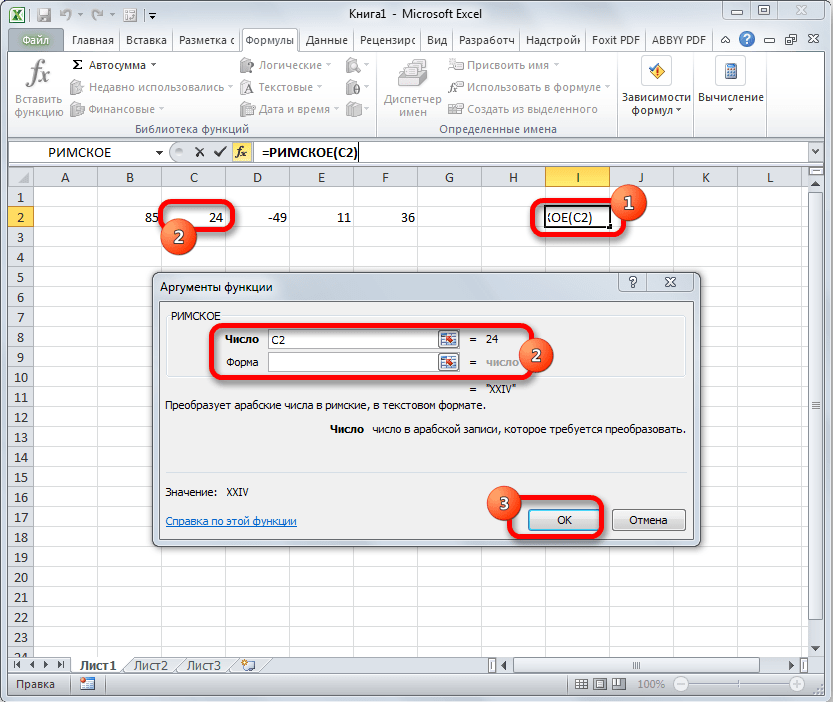
**=ЧАСТНОЕ(Числитель;Знаменатель)**

Оператор **ЧАСТНОЕ** применяется для деления чисел. Но в результатах деления он выводит только четное число, округленное к меньшему по модулю. Аргументами этой формулы являются ссылки на ячейки, содержащие делимое и делитель.



**=РИМСКОЕ(Число;Форма)**

Данная функция позволяет преобразовать арабские числа, которыми по умолчанию оперирует Excel, в римские. У этого оператора два аргумента: ссылка на ячейку с преобразуемым числом и форма. Второй аргумент не является обязательным.



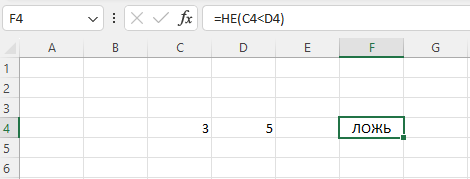
2. Логические функции:

Не, и, или, если.

Функции могут быть вложены друг в друга.

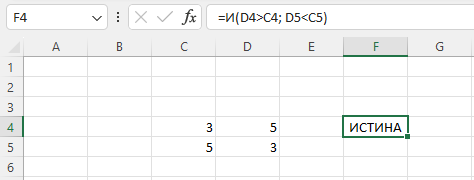
**=НЕ(C4<D4)**

Функция Не означает отрицание условия. Содержит логическое выражение со знаком сравнения.



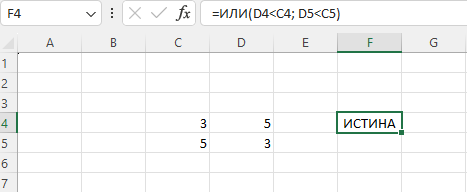
**=И(D4>C4; D5<C5)**

Функция И – содержит несколько аргументов. В качестве результата будет истина, если все результаты истина, иначе ложь.



**=ИЛИ(D4<C4; D5<C5)**

Функция ИЛИ – полностью идентична И, но только результат будет истина, если хоть 1 аргумент истина.



**=Если(лог. выр; истина; ложь)**

Функция если содержит 3 аргумента:

1. Условие выражения.

2. Действие в случае, если выражение истина.

3. Действие если ложь.

