

ПОВТОРЕНИЕ

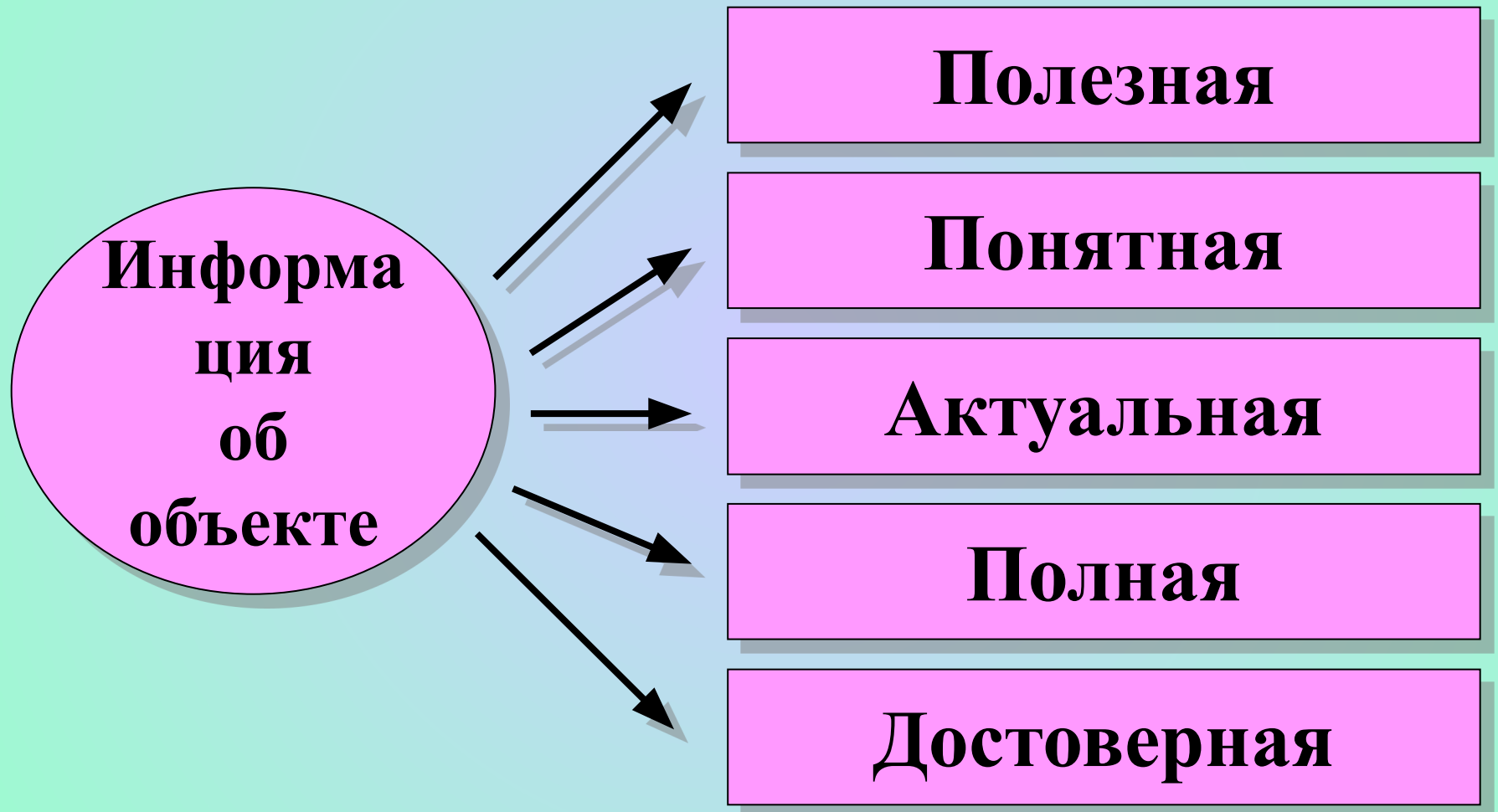
Человек и информация

Информатика – это наука, изучающая все аспекты получения, хранения, преобразования, передачи и использование информации с помощью компьютера.

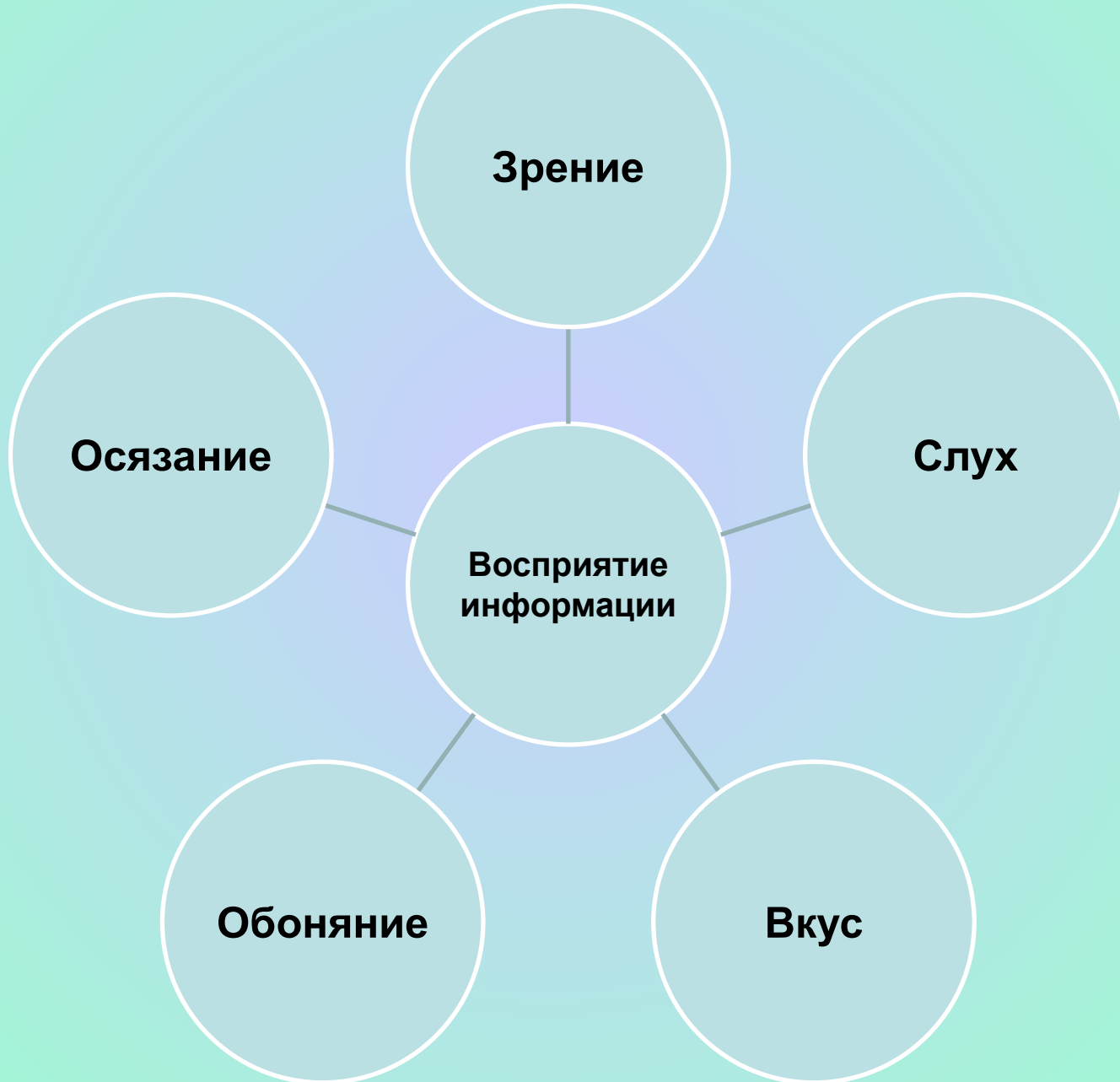
Компьютер – универсальный прибор для обработки информации.

Информация – это знания, которые мы получаем из различных источников

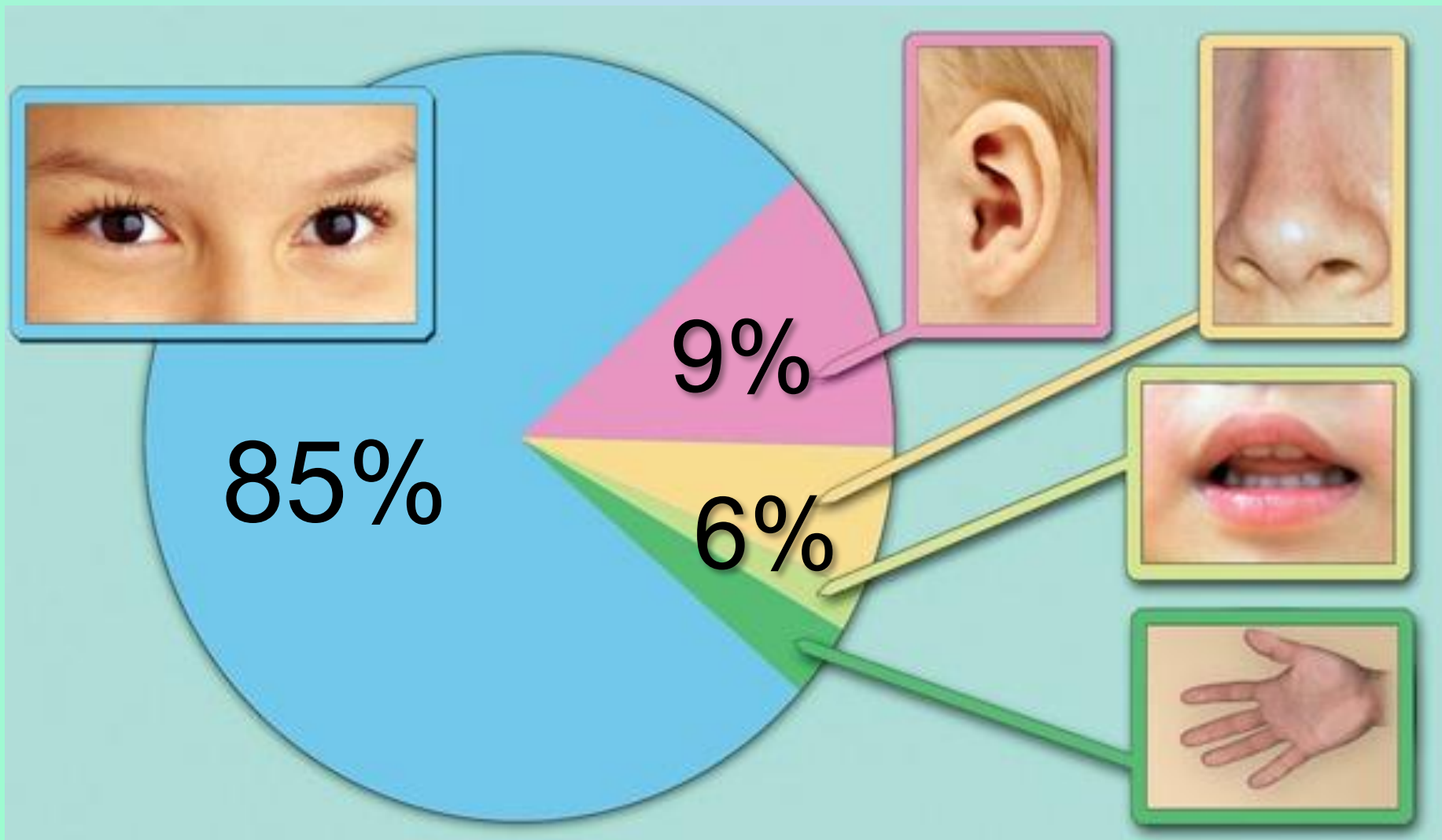
Свойства информации



Восприятие информации



Как мы воспринимаем информацию



Обмен информацией

```
graph TD; A[Обмен информацией] --> B[Язык]; B --> C[Естественный]; B --> D[Формальный];
```

Язык

Естественный
(разговорный: русский,
английский, китайский,
французский и др.)

Формальный
(язык профессии или
области знаний:
формулы,
ноты, жесты и др.)

Информационные процессы

```
graph TD; A[Информационные процессы] --- B[Процесс хранения информации]; A --- C[Процесс передачи информации]; A --- D[Процесс обработки информации];
```

Процесс
хранения
информации

Процесс
передачи
информации

Процесс
обработки
информации

Подходы к измерению информации

Содержательный

Количество информации, заключённое в сообщении, определяется объёмом знаний, который несёт это сообщение, т. е. сообщение содержит информацию для человека, если содержащиеся в нем сведения являются новыми, понятными и пополняют его знания.

Алфавитный

Позволяет определить количество информации, заключенной в тексте. Он является объективным, т.е. не зависящим от субъекта (человека), воспринимающего текст.

$$2^i = N, \quad I = K \cdot i$$

Единицы измерения объема информации

Название	Условное обозначение	Соотношение с другими единицами
Килобит	Кбит	1 Кбит = 1024 бит = 2^{10} бит \approx 1000 бит
Мегабит	Мбит	1 Мбит = 1024 Кбит = 2^{20} бит \approx 1 000 000 бит
Гигабит	Гбит	1 Гбит = 1024 Мбит = 2^{30} бит \approx 1 000 000 000 бит
Килобайт	Кбайт (Кб)	1 Кбайт = 1024 байт = 2^{10} байт \approx 1000 байт
Мегабайт	Мбайт (Мб)	1 Мбайт = 1024 Кбайт = 2^{20} байт \approx 1 000 000 байт
Гигабайт	Гбайт (Гб)	1 Гбайт = 1024 Мбайт = 2^{30} байт \approx 1 000 000 000 байт
Терабайт	Тбайт (Тб)	1 Тбайт = 1024 Гбайт = 2^{40} байт \approx 1 000 000 000 000 байт

Система счисления – способ записи чисел с помощью цифр



ПОЗИЦИОННЫЕ

Величина,
обозначаемая цифрой
в записи числа,
зависит от её позиции
(арабская, двоичная,
шестнадцатеричная)

НЕПОЗИЦИОННЫЕ

Величина,
обозначаемая цифрой
в записи числа, не
зависит от её позиции
(римская)

Файлы и файловая система

Файл — это определенное количество информации, имеющее имя и хранящееся в долговременной памяти.

Файловая система - это система хранения файлов и организации каталогов.

Имя файла

```
graph TD; A[Имя файла] --> B[Собственно имя файла]; A --> C[Расширение];
```

Собственно имя

файла

Задаёт

пользователь

(название файла)

Например:

Информатика

Расширение

Задаётся

программой

Например: .docx

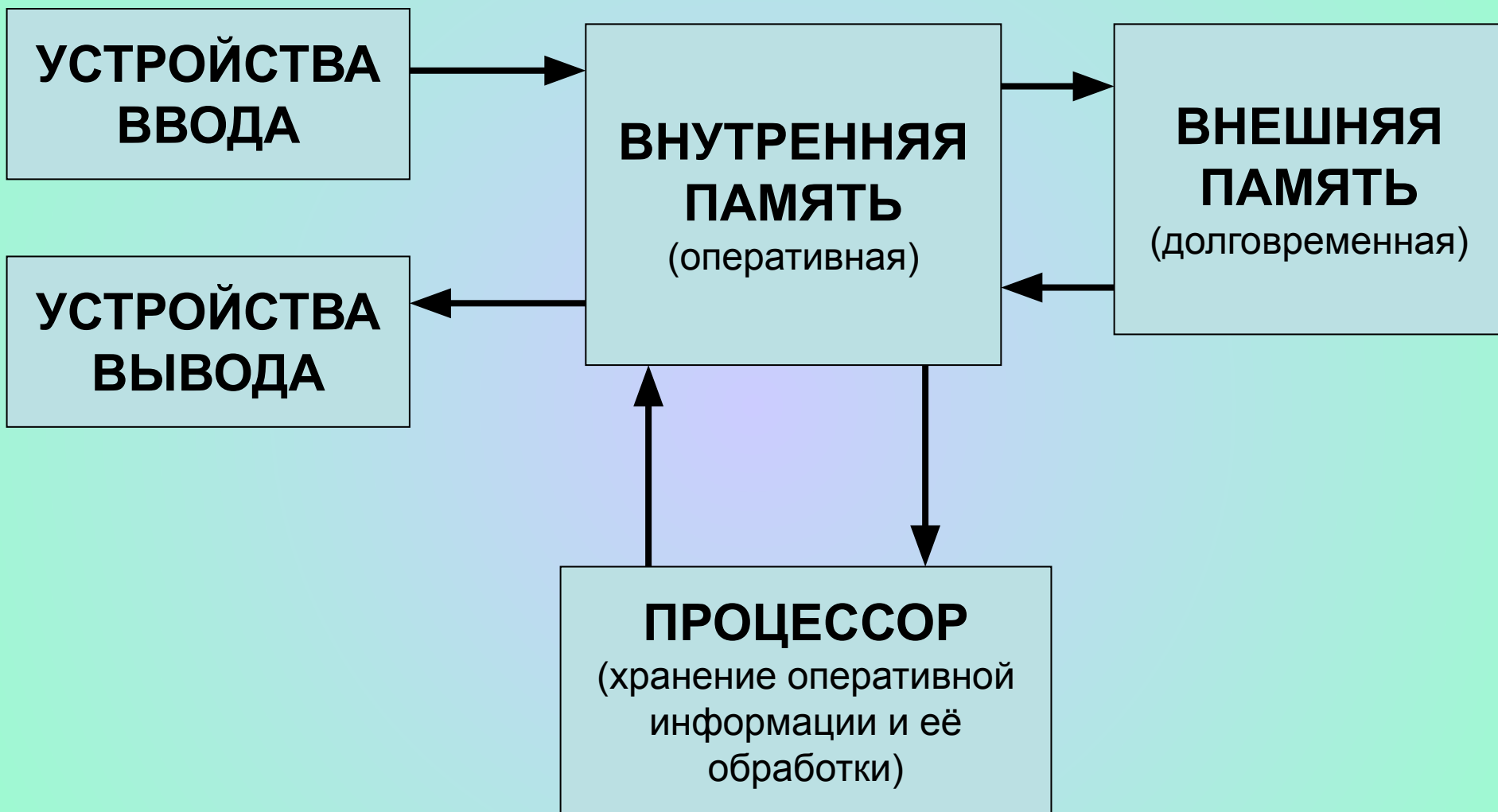
Тип файла	Расширение
Исполняемые программы	.exe, .com
Текстовые файлы	.txt, .rtf, .doc, .docx
Файлы, содержащие ЭТ	.xls
Графические файлы	.bmp, .gif, .jpg, .png, .pds
Web-страницы	.htm, .html
Звуковые файлы	.wav, .mp3, .midi, .kar, .ogg
Видеофайлы	.avi, .mpeg
Сжатые файлы	.rar, .zip
Код (текст) программы на языках программирования	.bas, .pas

Архитектура ЭВМ

это описание устройства и принципов работы компьютера, достаточное для пользователя и программиста.



В состав компьютера входят:



ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ (ПО)

Это совокупность программ, хранящихся в долговременной памяти.

ПО

```
graph TD; PO[ПО] --- A[ПРИКЛАДНОЕ]; PO --- B[СИСТЕМНОЕ]; PO --- C[СИСТЕМЫ ПРОГРАММИРОВАНИЯ];
```

ПРИКЛАДНОЕ
(программы для работы с текстом, графикой и др.)

СИСТЕМНОЕ
(операционные системы, сервисные программы)

СИСТЕМЫ ПРОГРАММИРОВАНИЯ
(языки программирования)

Алгоритмы

Алгоритм – описание последовательности действий (план), строгое исполнение которых приводит к решению поставленной задачи за конечное число шагов.

Исполнитель – это объект, умеющий выполнять определенный набор действий. Исполнителем может быть человек, робот, животное, компьютер.

Система команд исполнителя (СКИ) – это все команды, которые исполнитель умеет выполнять.

Среда исполнителя – обстановка, в которой функционирует исполнитель.

Свойства алгоритмов

1. **Дискретность** - алгоритм должен состоять из конкретных действий, следующих в определенном порядке.
2. **Детерминированность** - любое действие должно быть строго и недвусмысленно определено в каждом случае.
3. **Конечность** - каждое действие и алгоритм в целом должны иметь возможность завершения.
4. **Массовость** - один и тот же алгоритм можно использовать с разными исходными данными.
5. **Результативность** - отсутствие ошибок, алгоритм должен приводить к правильному результату для всех допустимых входных значениях.

Способ описания алгоритмов

- Словесные
- Графические (блок-схемы)
- Программные

Виды алгоритмов

Линейный - алгоритм, в котором все предписания (шаги) выполняются так, как записаны, без изменения порядка следования, строго друг за другом.

Разветвляющийся - алгоритм, в котором выполнение того или иного действия (шага) зависит от выполнения или не выполнения какого-либо условия.

Циклический - алгоритм, в котором некоторая последовательность действий повторяется несколько раз

Язык программирования Pascal

Структура программы:

```
program <имя программы>;  
const <постоянные значения>;  
var <используемые переменные>;  
begin <начало программы>  
  <оператор 1>;  
  <оператор 2>;  
  ...  
  <оператор n>  
end.
```

Типы данных:

Целые числа - **integer**
Вещественные - **real**
Символы - **char**
Строки - **string**
Логические переменные - **boolean**

Оператор вывода текста на экран:

writeln (выражение);

Оператор ввода значений:

readln (переменная1, , переменнаяN)

Пример простой программы:

Задача: Найти сумму 2х целых, вводимых с клавиатуры, чисел

```
program summa;
```

```
var a,b,c: integer;
```

```
begin
```

```
Writeln ('введите 2 целых числа');
```

```
Readln (a,b);
```

```
c:=a+b;
```

```
Writeln ('сумма равна: ', c);
```

```
end.
```

Условный оператор

- Полное условие:

```
if <условие> then  
    <команды>
```

```
else <команды>;
```

- Не полное условие:

```
if <условие> then  
  
<команды>;
```

- Несколько условий:

```
if <условие> then  
    <команды>
```

```
else
```

```
if <условие> then  
    <команды>
```

```
else <команды>;
```

Пример задачи с условием

Задача 1. ВВЕСТИ ЧИСЛО. ЕСЛИ ОНО ПОЛОЖИТЕЛЬНОЕ, ВЫЧЕСТЬ 30, ИНАЧЕ ПРИБАВИТЬ 70

```
program zadanie1;  
var  
  a: integer;  
begin  
  writeln('Введите число');  
  readln(a);  
  if a >= 0 then  
    a := a - 30  
  else  
    a := a + 70;  
  writeln('a = ', a);  
end.
```


Пример задачи с условием

Задача 2. Ввести целое число и определить четное оно или нет.

```
program zadanie2;  
var  
  a: integer;  
begin  
  writeln ('Введите целое число ');  
  readln (a);  
  if a mod 2 = 0 then  
    writeln ('Число четное')  
  else  
    writeln ('Число не четное ');  
end.
```