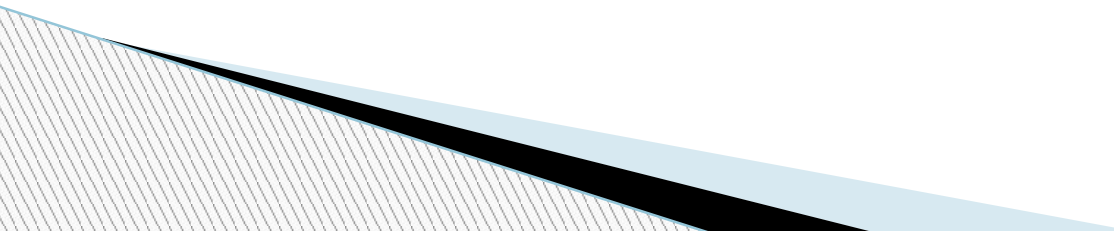


# Тема 4

## Программное обеспечение информационных технологий



# План лекции:

- Этапы решения задач с использованием компьютера.
  - Алгоритмы.
  - Классификация языков программирования.
  - Классификация программного обеспечения.
- 

***1 вопрос***  
***Этапы решения задач с  
использованием компьютера***



# Процесс решения любой задачи на компьютере можно разделить на:

- содержательная постановка (формулировка) задачи;
- математическая постановка (формулировка) задачи;
- выбор метода решения задачи;
- составление алгоритма решения задачи;
- составление программы на языке программирования;
- отладка программы;
- решение данной задачи по заданной программе.

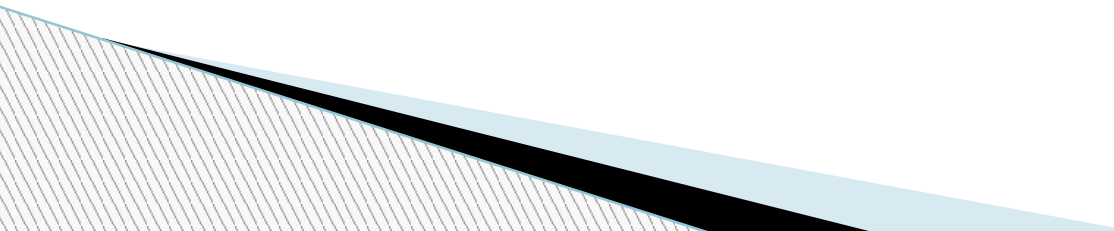
# Схема решения любой задачи

□ «МЕТОД» –  
«АЛГОРИТМ» –  
«ПРОГРАММА»

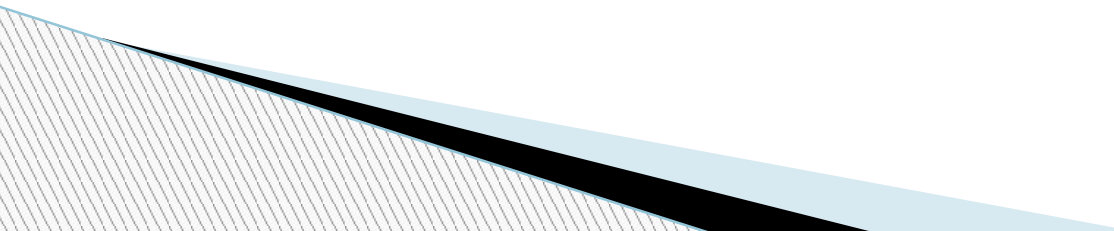


# **2 вопрос**

# **АЛГОРИТМЫ**



# *Понятие алгоритма*

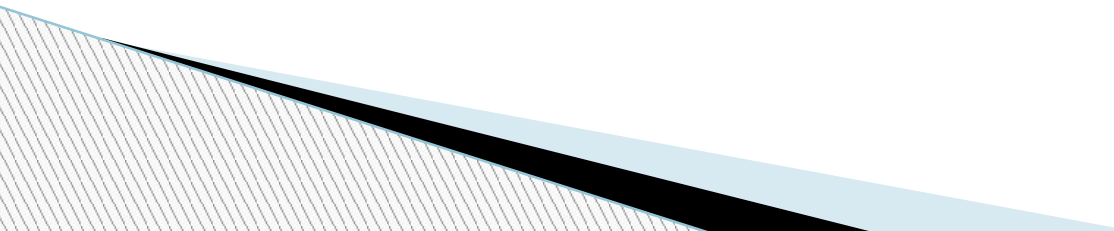
- ▣ **Алгоритм (общее понятие)** - это точное и понятное предписание исполнителю совершить последовательность действий, направленных на решение поставленной задачи.
  - ▣ **Алгоритм (компьютерный)** — точное предписание, определяющее вычислительный процесс, ведущий, от варьируемых начальных данных к искомому результату.
- 

# Реализация алгоритма

- ▣ **Исполнитель алгоритма** - это некоторая абстрактная или реальная (техническая, биологическая или биотехническая) система, способная выполнить действия, предписываемые алгоритмом.
- ▣ **Среда** (или обстановка) - это "место обитания" исполнителя.
- ▣ Каждый исполнитель может выполнять команды только из некоторого строго заданного списка - **системы команд** исполнителя.
- ▣ После вызова команды исполнитель совершает соответствующее **элементарное действие**.
- ▣ **Отказы** исполнителя возникают, если команда вызывается при недопустимом для нее состоянии среды.



# *Основные свойства алгоритмов:*

- Понятность
  - Дискретность
  - Определенность
  - Результативность
  - Массовость
- 

# ПОНЯТНОСТЬ

- для исполнителя - т.е. исполнитель алгоритма должен знать, как его выполнять.

# ДИСКРЕТНОСТЬ

- (прерывность, раздельность) - т.е. алгоритм должен представлять процесс решения задачи как последовательное выполнение простых или ранее определенных шагов.

# ОПРЕДЕЛЕННОСТЬ

- т.е. каждое правило алгоритма должно быть четким, однозначным и не оставлять места для разночтений.

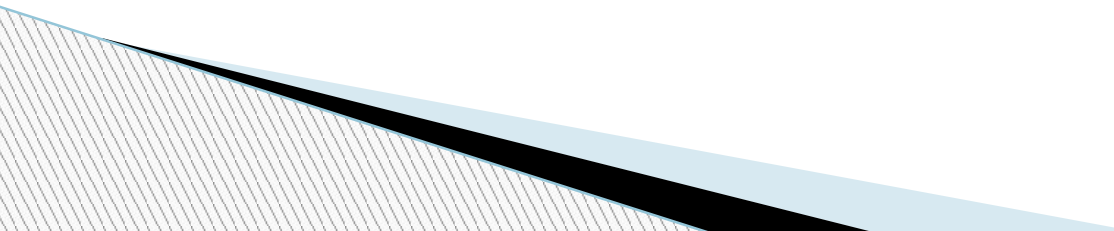
# РЕЗУЛЬТАТИВНОСТЬ

- (или конечность). Это свойство состоит в том, что алгоритм должен приводить к решению задачи за конечное число шагов.

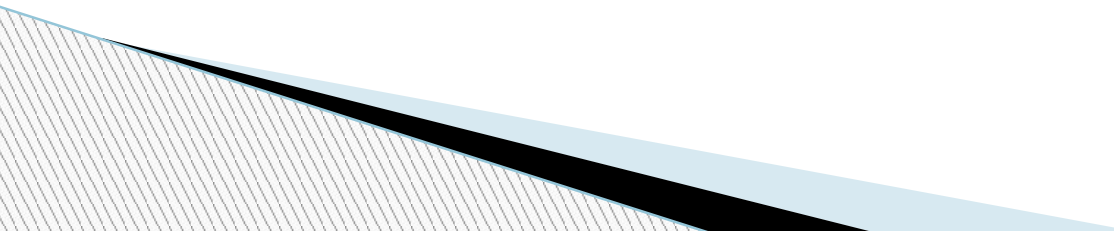
# МАССОВОСТЬ

- означает, что алгоритм решения задачи разрабатывается в общем виде, т.е. он должен быть применим для некоторого класса задач, различающихся лишь исходными данными.
- При этом исходные данные могут выбираться из некоторой области, которая называется **областью применимости алгоритма.**

# *Способы описания алгоритмов*

- Словесно-формульное описание.
  - Графическое описание.
  - Описание на алгоритмическом языке (псевдокод).
  - Программная форма описания.
- 

# *Словесно-формульное описание алгоритма*

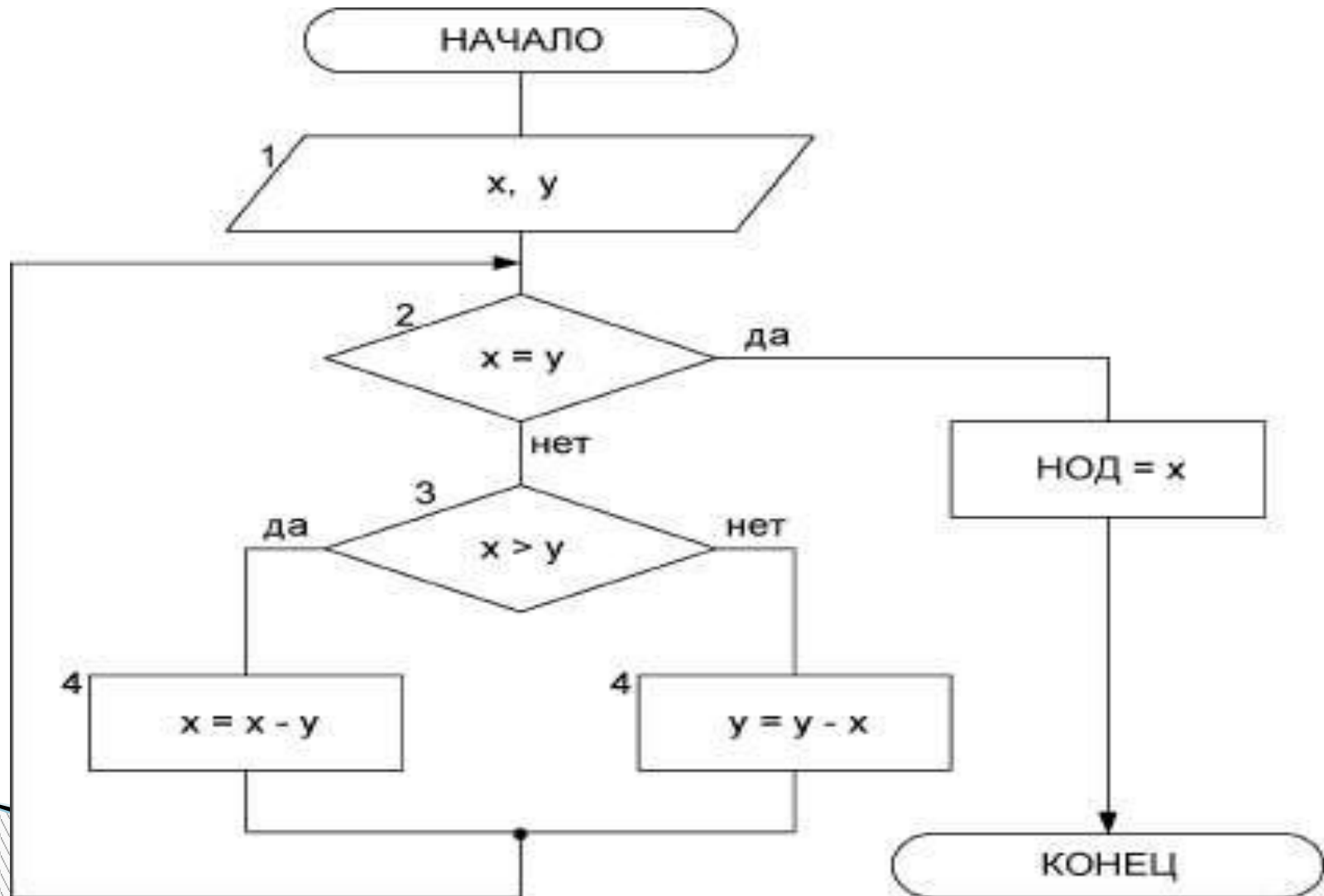
- т. е. описание алгоритма с помощью слов и формул - последовательных этапов обработки данных на естественном языке (например, на русском).
- 



# *Графическое описание алгоритма*

- т. е. описание с помощью схем алгоритмов, является более компактным и наглядным по сравнению с словесно-формульным.

# Алгоритм нахождения НОД двух натуральных чисел



# *Описание алгоритма на алгоритмическом языке (алгоязыке)*

- ▣ **Алгоритмический язык** — это средство для записи алгоритмов в аналитическом виде, промежуточном между записью алгоритма на естественном (человеческом) языке и записью на языке ЭВМ (языке программирования).

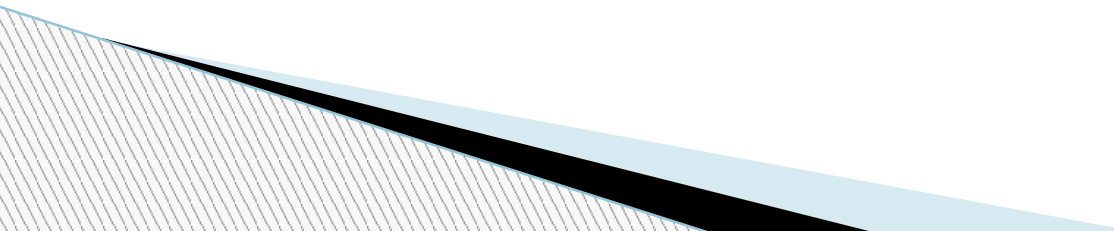
# Запись алгоритма расчета зарплаты на алгоритмическом языке:

```
алг ЗАРПЛАТА(цел ST, вещ ZP)
  арг ST
  рез ZP
нач
  если ST<5
    то ZP:=150
  иначе
    если ST<=15
      то ZP:=180
    иначе ZP:=180+(ST-15)10
  все
все
кон
```

# *Программная форма*

- ▣ **Программная форма** представляет собой тексты программ, написанных на различных языках программирования.

# *Виды алгоритмов*

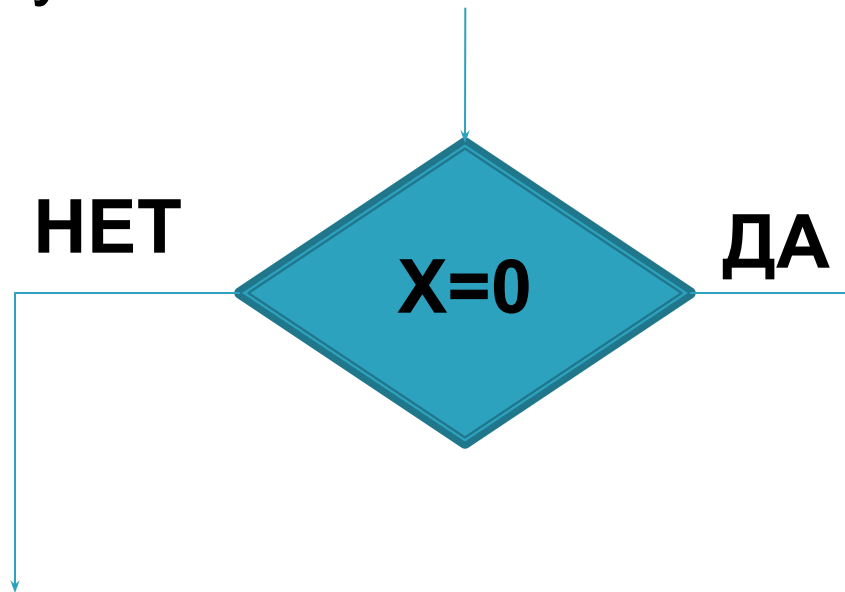
- ▣ **линейной** структуры,
  - ▣ **разветвляющейся** структуры,
  - ▣ **циклической** структуры.
- 

# Линейные алгоритмы

- ▣ **Линейным** называется алгоритм, в котором все этапы решения задач выполняются строго последовательно.

# Разветвляющиеся алгоритмы

- ▣ **Разветвляющимся** называют такой алгоритм, в котором выбирается один из нескольких возможных путей (вариантов) вычислительного процесса.
- ▣ Каждый путь называется **ветвью алгоритма**.

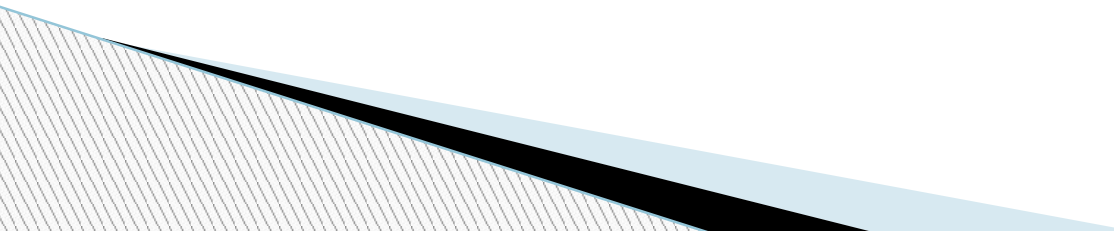




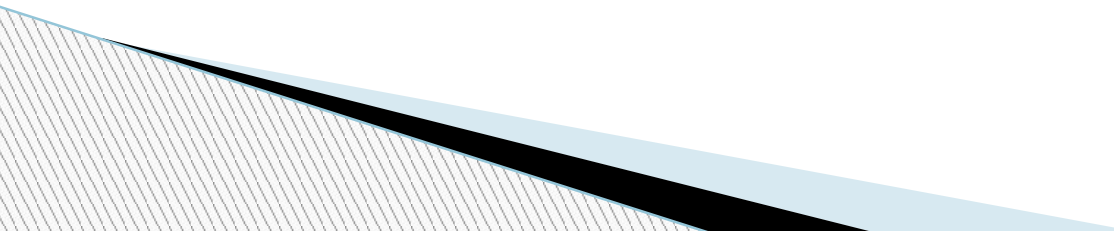
# Циклические алгоритмы

- **Циклическим** называют алгоритм, в котором получение результата обеспечивается многократным выполнением одних и тех же операций.
- Основой циклического алгоритма являются операции, многократное выполнение которых дает искомый результат.
- Такие операции называются **рабочими циклами**.

**3 вопрос**  
**Классификация языков**  
**программирования**



# Классификация языков программирования

- Машинно-ориентированные и машинно-независимые языки.
  - Процедурные языки.
  - Декларативные языки.
  - Объектно-ориентированные языки.
- 

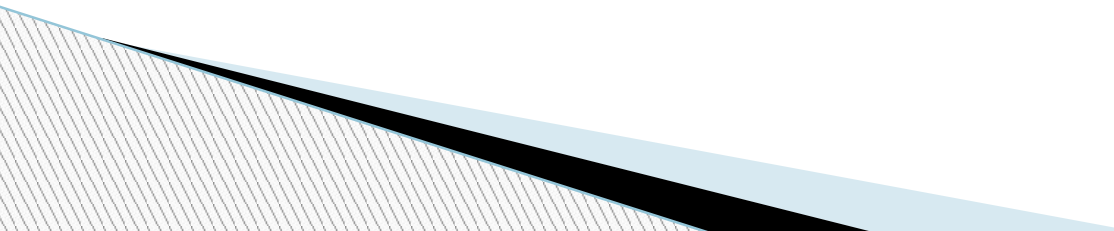
# Машинно-ориентированные языки

- ▣ **Машинно-ориентированные языки** - это языки, средства которых существенно зависят от особенностей конкретной ЭВМ.
- ▣ Пример: язык АССЕМБЛЕР

# Машинно-независимые языки

- ▣ **Машинно-независимые языки** - это средства описания алгоритмов решения задач и информации, подлежащей обработке, которые не требуют от программиста знания особенностей функционирования конкретной ЭВМ.
- ▣ Эти языки называются также еще **языками высокого уровня.**

# Классификация машинно-независимых языков

- Процедурные
  - Декларативные
  - Объектно-ориентированные
- 

# Процедурные языки

- ▣ **Процедурные языки** (называемые также директивными или императивными) - это языки, определяющие, как вычислять результат для какой-нибудь задачи в соответствии с заданным алгоритмом.
- ▣ К ним относятся: Алгол, Фортран, Бейсик, Паскаль, Си.

# Классификация процедурных языков

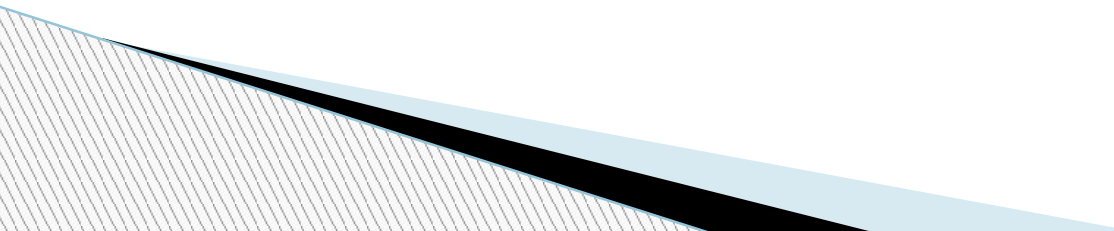
- ▣ Структурные
  - ▣ Неструктурные
- 



# Неструктурное программирование

- ▣ **Неструктурное программирование** допускает использование в явном виде команды безусловного перехода (в большинстве языков GOTO).
- ▣ Типичными представителями неструктурных языков являются ранние версии **Бейсика** и **Фортрана**.

# Структурное программирование

- В структурном программировании задача разбивается на большое число мелких подзадач, каждая из которых решается своей процедурой или функцией.
  - Программирование без GO TO.
  - Пример: Паскаль, СИ
- 

# Декларативные языки

- **Декларативные языки** - это языки, оперирующие с помощью задания данных и отношений между ними. Вместо алгоритмов в таких языках используются правила логического вывода, которые позволяют системе находить нестандартные, заранее не определенные решения.
- Декларативные языки называют также еще **языками искусственного интеллекта.**

# Классификация декларативных языков

- ▣ функциональные (аппликативные)
  - ▣ логические
- 

# Функциональные языки

- В основе **функциональных языков** лежит понятие функции как "черного ящика", имеющего несколько параметров (аргументов) на входе и один результат на выходе.
- В таких языках отсутствуют операторы: все действия, в том числе и управляющие конструкции, выполняются при помощи вызовов функций.
- Одним из первых функциональных языков стал **Лисп**, созданный американским ученым Джоном Маккарти в 1957 году.

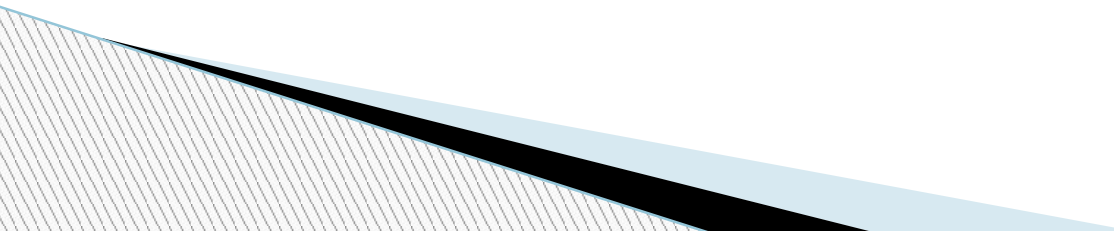
# *Логическое программирование*

- ▣ **Логическое программирование** представляет собой попытку возложить на программиста только постановку задачи, а поиски путей ее решения предоставить транслятору.
- ▣ Логические языки (например, **Пролог**) имеют специальные конструкции для описания объектов и связей (отношений) между ними.

# Объектно-ориентированные ЯЗЫКИ

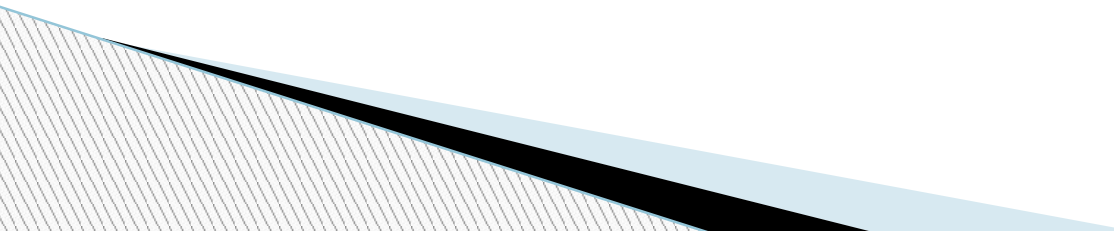
- ▣ **Объектно-ориентированные языки** представляют собой отображение объектов реального мира, их свойств (атрибутов) и связей между ними при помощи специальных структур данных.
- ▣ Пример: C++, Java, Visual Basic.

**4 вопрос**  
**Классификация программного**  
**обеспечения**



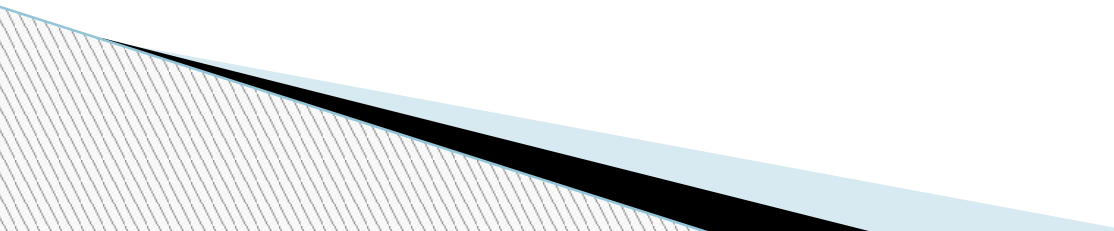


# Классификация программного обеспечения

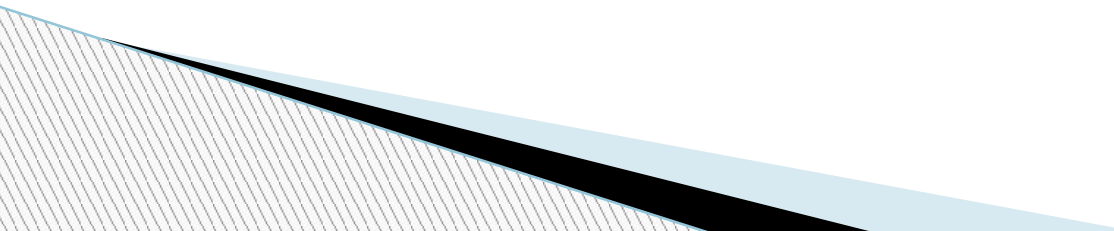
- Системные программы.
  - Инструментальные программы.
  - Прикладные программы.
- 

# *Системные программы*

**▣ Системные программы** — предназначены для эксплуатации и технического обслуживания ЭВМ, управления и организации вычислительного процесса при решении любой конкретной задачи на ЭВМ и т. д.



# Классификация системных программ

- Операционные системы.
  - Оболочки операционных систем.
  - Драйверы устройств.
  - Программы-утилиты.
  - Антивирусные программы.
- 

# Инструментальные программы

- ▣ **Инструментальные программы** – это программы-инструменты для создания другого программного обеспечения.

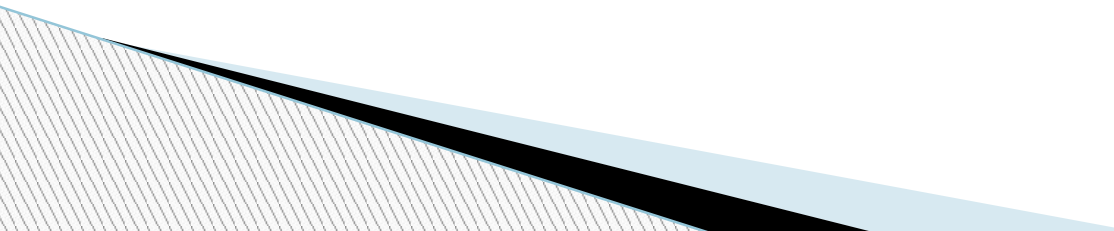
# Программирование

- ▣ **Системы программирования (среда программирования)** — это пакет программ, предназначенный для разработки и эксплуатации программ на конкретном языке программирования.
- ▣ **Язык программирования** — это инструмент для создания компьютерных программ.
- ▣ **Трансляторы** – это программа, обеспечивающая перевод *исходной программы* пользователя (программа на языке программирования) на язык ЭВМ (машинный язык).

# Прикладные программы (приложения)

- ▣ **Прикладные программы (приложения)** – это все остальные компьютерные программы, не относящиеся к системным или инструментальным.

# Классификация прикладного ПО

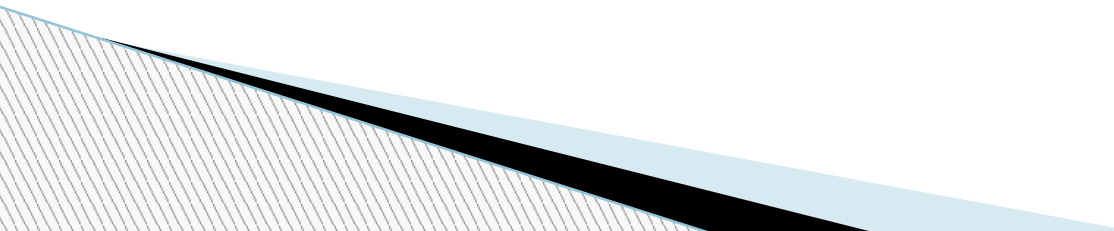
- Офисные программы
  - Мультимедийные программы
  - Учебные и обучающие программы
  - Программы для работы в компьютерных сетях
  - Игры
  - Другие программы
- 

# *Офисные программы*

- ▣ **Офисные программы** — это программы, используемые для создания, обработки и представления различных документов в повседневной деятельности.



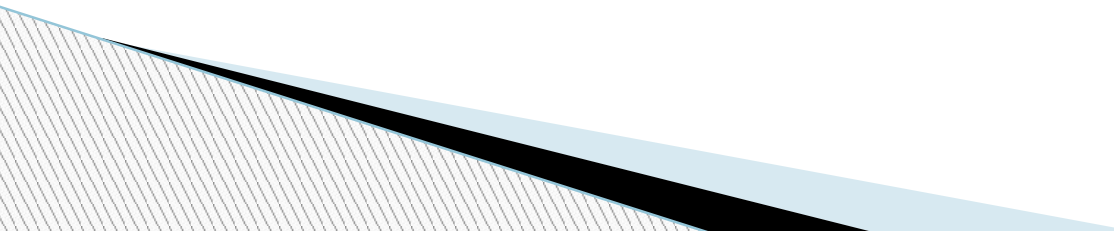
# Классификация офисных программ

- Программы для работы с текстами (Текстовые редакторы).
  - Электронные таблицы (Табличные процессоры).
  - Системы управления базами данных (СУБД).
  - Интегрированные среды.
  - Другие офисные программы.
- 

# Мультимедийные программы

- ▣ **Мультимедийные программы** – это компьютерные программы, предназначенные для создания и обработки графической, аудио-, видео- и др. информации.

# Классификация мультимедийных программ

- Программы для работы с видео
  - Программы для работы со звуком
  - Программы для работы с графикой
  - Виртуальная реальность и другие программы
  - И т.д.
- 

# Компьютерные программы для работы в вычислительных сетях

- ▣ **Компьютерные программы для работы в вычислительных сетях** – это программы для соединения и организации сеанса связи с удаленным пользователем, программы навигации в сетях (браузеры), работы с электронной почтой, работы с удаленными файлами (менеджеры зачек и др.), поиска информации в сетях и т.д.