

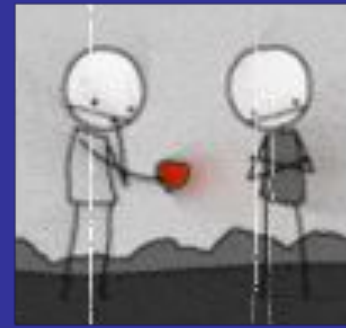


# Компьютерная модель и исполнители





# Модели задач



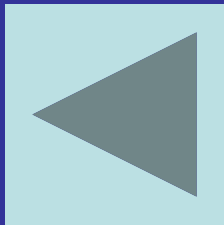
С моделями задач вы имеете дело ежедневно, ежечасно и даже ежеминутно. Но до сих пор вы, возможно, об этом не задумывались, поскольку **построение моделей для человека** так же **естественно**, как ходьба, или умение пользоваться ножом или вилкой. Разница же в том, что последнему вас долго учили, и теперь вы хорошо умеете это делать. А вот строить модели мы все, скорее всего, учимся стихийно, сами того не подозревая.



# МОДЕЛЬ



**Модель** – упрощенное представление о реальном объекте, процессе или явлении. Модели строят для познания окружающего нас мира, происходящих в нем процессов, а также для прогнозирования событий.



# Компьютерная модель



– модель задачи, составленная в расчете на исполнителя, имитированного на ЭВМ (реализована средствами программной и инструментальной среды компьютера)

# Исполнитель

- тот, кто будет преобразовывать исходные данные в результаты.



Разумеется, модель должна учитывать возможности исполнителя. То есть, приступая к созданию модели, нужно сразу сориентироваться на исполнителя, которому предстоит с ней работать.

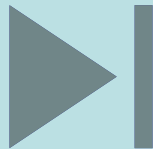
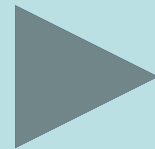
# Исполнитель алгоритма

**Исполнителя** можно представить себе как некоторое *устройство управления*, соединенное с *набором инструментов*.

*Устройство управления* понимает алгоритмы и организует их выполнение, командуя соответствующими инструментами.



**Инструменты** производят действия, выполняя команды управляющего устройства.



**Возможности исполнителя ВСЕГДА** ограничены. Поэтому прежде чем составлять алгоритм решения задачи, нужно узнать какие действия исполнитель может выполнить. Эти действия называются **система команд исполнителя (СКИ)**. При составлении алгоритмов можно использовать только эти команды.

**Требования к исполнителю:**

- четко выполнять каждое действие, входящее в алгоритм!



# Имитация исполнителя с помощью ЭВМ

***Разные классы задач требуют разные наборы допустимых действий – разных исполнителей.***  
Обычно набор допустимых действий явно или неявно указывается в формулировке задачи.

Чтобы для решения того или иного класса задач можно было применить ЭВМ, необходимо симитировать на ЭВМ нужный набор допустимых действий. Это ***называется имитация исполнителя с помощью ЭВМ.***

Один и тот же ***исполнитель может быть симитирован на ЭВМ многими способами.***  
При этом содержание действий остается неизменным, а их названия могут быть разными.



# Компьютерная модель и пользователь ЭВМ



При создании компьютерной модели в исполнителей могут быть прикладные программные среды – **текстовый и графический редакторы, электронная таблица, информационно-поисковая система**, как в отдельности, так и в сочетании.

Для того, чтобы создать модель в этих средах **пользователь ЭВМ** должен быть знаком со средой и командами, которые доступны данным исполнителям.

Кроме того, при создании модели пользователю необходимо продумать и выполнить определенную последовательность действий необходимых для решения поставленной задачи.

# Язык программирования и программист



Последовательность действий необходимая и достаточная для решения задачи называется **алгоритмом**.

Специальный язык для записи алгоритма в «понятной» исполнителю форме называется **языком программирования (ЯП)**.

Алгоритм, записанный на языке программирования, называется **программой**.

Специалиста, использующего языки программирования для создания компьютерной модели называют **программистом**.

# Программирование

Раздел информатики, цель которого –

разработать  
средства для  
подготовки задач к  
решению на ЭВМ

И

создать средства  
программного  
обеспечения, с помощью  
которых реализуется  
вычислительный процесс  
на ЭВМ и обмен ЭВМ  
информацией с внешним  
миром.

# Программное обеспечение

Сообщив алгоритм исполнителю-ЭВМ, человек уже не вмешивается в его работу.

Исполнитель-ЭВМ сам организует автоматическое выполнение шагов алгоритма и выдает результаты пользователю данной программы.

***Все программное обеспечение создано программистами. Пользователю ЭВМ необходимо это учитывать!***

# Задачи для исследования:

1. Можно ли создать исполнителя, который все может?
2. Почему так много языков программирования? Может быть стоит создать один – универсальный?