



От жесткого к мягкому



- Что такое программа?
 - Уровень аппаратного обеспечения (процессор, память).
 - Программа как последовательность команд в памяти.
 - Циклы и передачи управления.
 - Процедуры и процедурное программирование.
 - Объекты и объектное программирование.
 - Функциональное программирование...



Что такое программа?

```
if (top !== self) {
  function calcWidth() {
    var wW = 0;
    if (typeof window.innerWidth === 'number') {
      wW = window.innerWidth;
    } else if (document.documentElement && documentElement.clientWidth) {
      wW = document.documentElement.clientWidth;
    } else if (document.body && document.body.clientWidth) {
      wW = document.body.clientWidth;
    }
    if (sH = document.documentElement.scrollHeight) {
      var wH = window.innerHeight || document.documentElement.clientHeight || sH;
      var sW = !document.all && (sH > wH) ? sH : wH;
      menuWidth = wW - sW;
    }
  }
  calcWidth();
}
```



Программа с точки зрения компьютера



- Машинный код – это набор инструкций (команд) процессора.
- У каждого процессора свои инструкции.
- Помимо процессора могут использоваться ячейки памяти и внешние устройства (со своими инструкциями).



Программа в памяти компьютера

Программа может обращаться к ячейкам памяти (помещать их на регистры процессора, копировать, записывать результат с регистра процессора и т. д.)



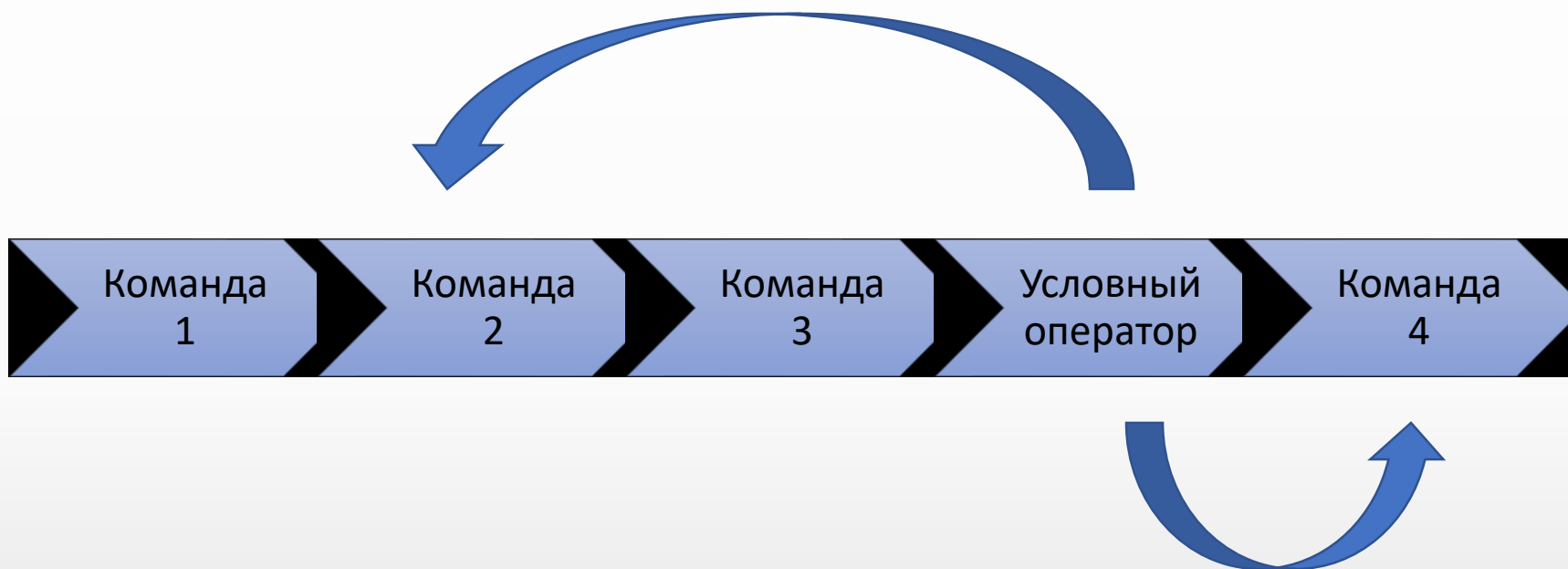
Программа размещается в памяти компьютера и может сама быть объектом работы программы.

При выполнении программы, ячейки памяти, где она содержится вызываются последовательно и команды из этих ячеек выполняются.



Условный оператор

Передача управления



Передача управления

Условный оператор на основании той или иной ячейки памяти передает управление той или иной команде, также размещенной в памяти.



Ассемблер

```
mov ax,1           ; В EAX будет факториал, на это число мы будем умножать все остальные.  
mov cx,[digit]     ; В ECX число факториал которого нужно найти.  
factCycle: mul cx   ; умножаем еах на есх  
dec cx            ; вычитаем из есх единицу  
cmp cx,1          ; проверяем не равен ли ECX единице  
jne factCycle  
; JNE= JUMP if NOT EQUAL - переход если неравно. если ECX != 1 - переходим на следующую итерацию.
```

Да, на ассемблере можно написать все!

Нет, мы не будем этого делать и вам не советуем.



Ассемблер, да не тот

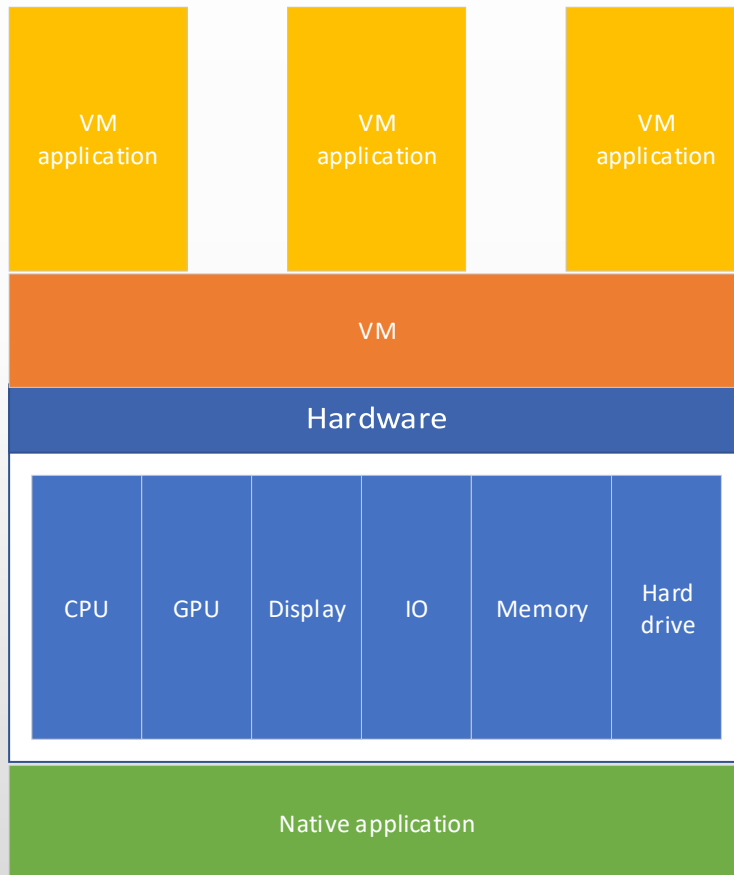
```
L0
    LINENUMBER 44 L0
    LCONST_1
    LSTORE 1

L1
    LINENUMBER 45 L1
    ICONST_1
    ISTORE 3
    ILOAD 0
    ISTORE 4
    ILOAD 3
    ILOAD 4
    IF_ICMPGT L2

L3
    LINENUMBER 46 L3
    LLOAD 1
    ILOAD 3
    I2L
    LMUL
    LSTORE 1
    ...
    ...
```

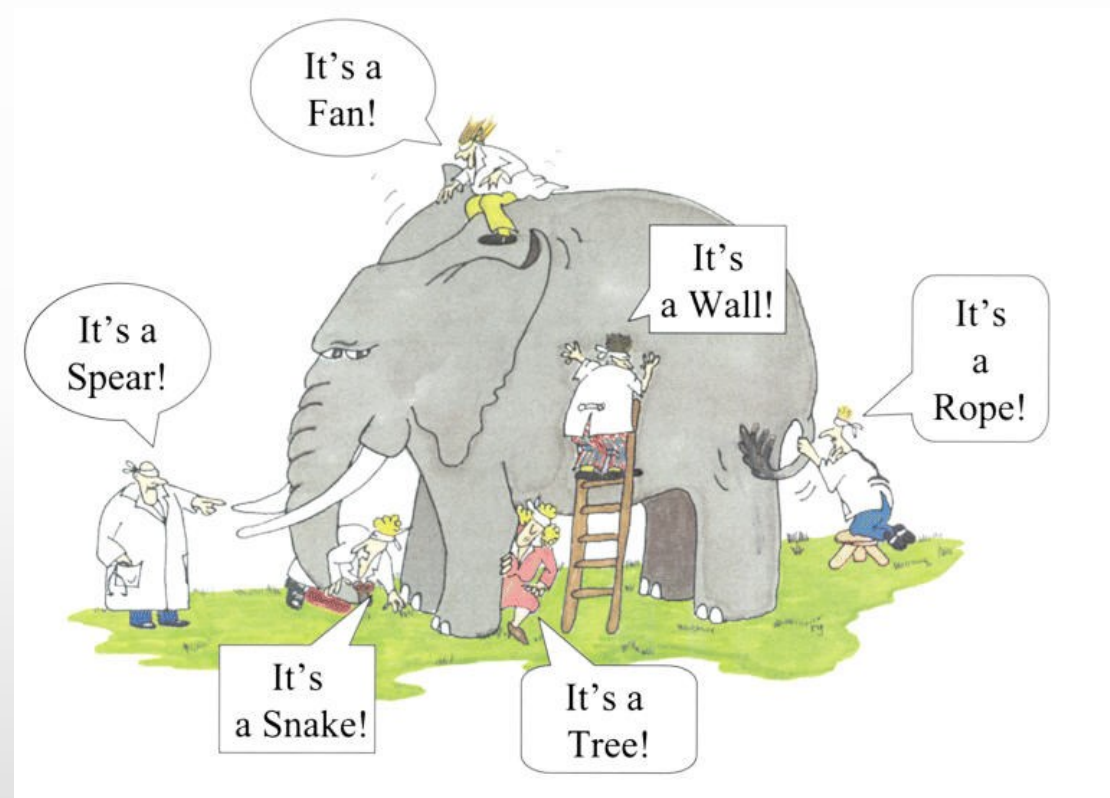
Вот это оригинал:

```
fun Int.factorial(): Long{
    var prod = 1L
    for(i in 1..(this)){
        prod *= i
    }
    return prod
}
```

- Абстрагирование функций ОС
- Дополнительные функции (контроль памяти, хранение, передача данных)
- Изоляция приложений
- Оптимизация работы приложений

Парадигмы





Процедурное программирование

- Процедуры (функции) – это куски программы, предназначенные для многократного использования.
- Процедуры имеют набор параметров и выходное значение.

```
long int factorial(int n)
{
    if (n<=1)
        return(1);
    else
        n=n*factorial(n-1);
    return(n);
}
```



Объектное программирование

- Объект – сущность, определяемая набором своих методов (поведений).
- Использование объекта никак (в идеале) не завязано на его реализацию и определяется только декларацией его поведений.

```
public class MyNumber extends Integer {  
    pubic long factorial() {  
        return SomeMathProcessor.factorial(this);  
    }  
}
```

Так программу писать не надо



Функциональное программирование

```
factorial :: Integer -> Integer
factorial 1 = 1
factorial n = n * factorial (n - 1)
```

Чисто функциональная программа (в понимании Haskell):

- Нет состояния. Любая программа – последовательность функций.
- Нет переменных, их нельзя «испортить». Нет проблем с параллелизацией.
- Программа – это математическая формула.