



Программирование, лекция 3



Кафедра ИУ7 МГТУ им. Н. Э. Баумана,
2021 год



Логический (булевский, булев) тип данных

Только 2 значения:

- True
- False

`bool([x])`

Приведение логического типа данных

- 0, 0.0 - False
- остальные числа - True

`bool(12.75)`

`bool((2*5-10)**14)`

`int(True)`

`float(False)`

Логические операции

and

or

not

Операции сравнения

<

<=

>

>=

==

!=

Приоритеты операций

1. Арифметические, побитовые (см. предыдущую лекцию)
2. Сравнения
3. Логические

Оператор присваивания

Оператор присваивания предназначен для связывания имен со значениями и для изменения атрибутов или элементов изменяемых объектов. Оператор присваивания связывает переменную с объектом.

Оператор присваивания обозначается '='.

Обычная (каноническая) форма:

`name = value`

Позиционное присваивание

name1, name2, ..., nameN = value1, value2, ..., valueN

Комбинированное присваивание

`+=`

`-=`

`*=`

`/=`

`//=`

`%=`

`**=`

`&=`

`|=`

`>>=`

`<<=`

Условный оператор

if выражение1:

 блок1

elif выражение2:

 блок2

elif выражение3:

 блок3

else:

 блок4

Условный оператор. Примеры

if $x > y$:

$m = x$

else:

$m = y$

if $x > y$: $m = x$

else: $m = y$

Условный оператор. Примеры. Неполный оператор условия

`m = x`

`if x < y:`

`m = y`

Условный оператор. Примеры. Полный оператор

```
if x < y:
```

```
    print('x меньше y')
```

```
elif x==y:
```

```
    print('x и y равны')
```

```
else:
```

```
    print('x больше y')
```

Тернарный оператор

`x if C else y`

Самый низкий приоритет среди всех операций языка

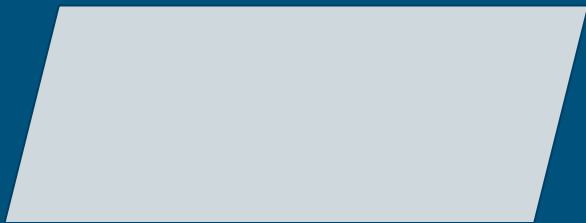
Схемы алгоритмов

ГОСТ 19.701-90 Схемы алгоритмов,
программ, данных и систем.

Условные обозначения и правила
выполнения

ISO 5807:1985

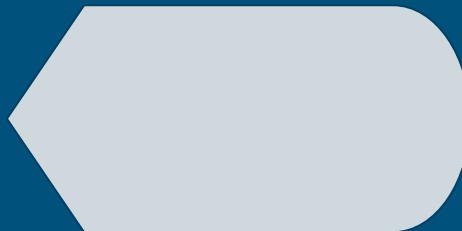
Схемы алгоритмов. Символы данных



Данные, носитель которых не определён
(не конкретизирован)

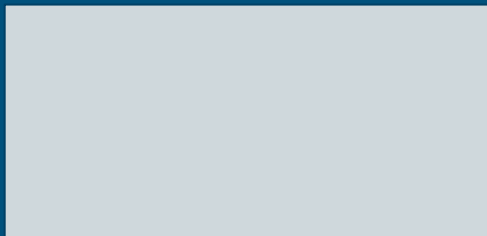


Ввод вручную

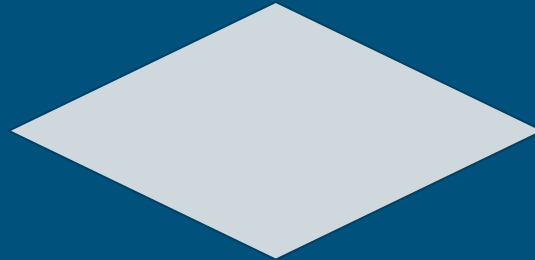


Отображающее устройство

Схемы алгоритмов. Символы процесса



Процесс (операция или группа операций по обработке данных любого вида)



Решение: один вход, несколько альтернативных выходов

Схемы алгоритмов. Специальные символы

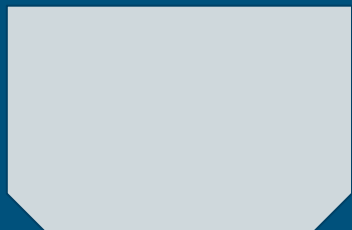
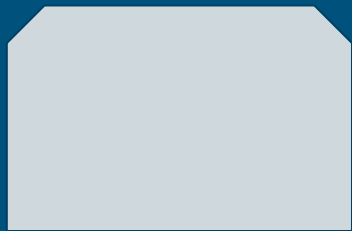


Терминатор (начало
или конец программы)

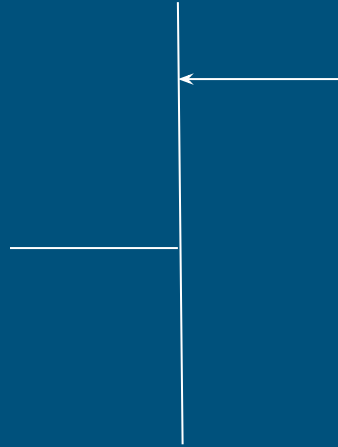
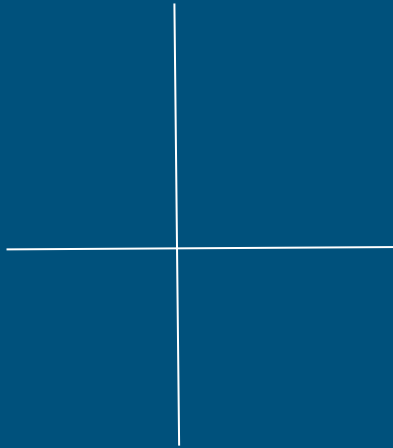


Соединитель - переход в
другую часть схемы.
Используется для обрыва
линии и продолжения её в
другом месте

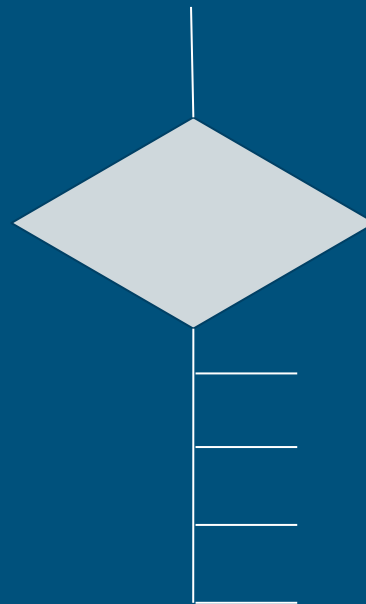
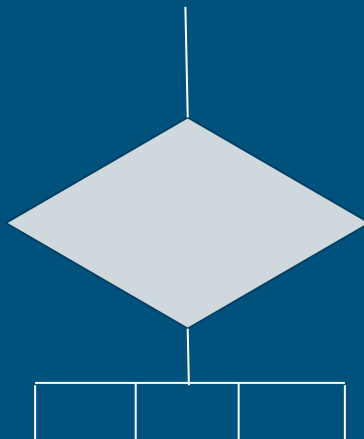
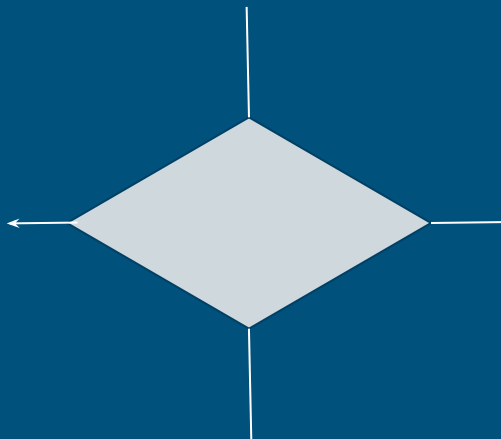
Схемы алгоритмов. Граница цикла



Схемы алгоритмов. Линии



Схемы алгоритмов. Специальные условные обозначения



Схемы алгоритмов. Применение символов

- схема данных
- схема программы
- схема работы системы
- схема взаимодействия программ
- схема ресурсов системы

Другие стандарты схем

- IDEF (I-CAM DEFinition или Integrated DEFinition) - для моделирования широкого спектра сложных систем в различных разрезах
- UML (Unified Modeling Language) - открытый стандарт, использующий графические обозначения для создания абстрактной модели системы
- ARIS (Architecture of Integrated Information Systems) - методология для моделирования бизнес-процессов организаций