

Харьковский национальный университет имени В. Н. Каразина

Механико-математический факультет

Кафедра теоретической и прикладной информатики

Направление подготовки: информатика

Семестр 2

АЛГОРИТМЫ И СТРУКТУРЫ ДАННЫХ

**МОДЕЛИ ВЫЧИСЛЕНИЙ**

## ***Объект изучения:***

процесс построения и анализа алгоритмов  
и структур данных

## ***Предмет изучения:***

модели вычислений, структуры данных и  
методы анализа эффективности алгоритмов

## ***Цели изучения:***

познакомиться с концепцией модели вы-  
числений;

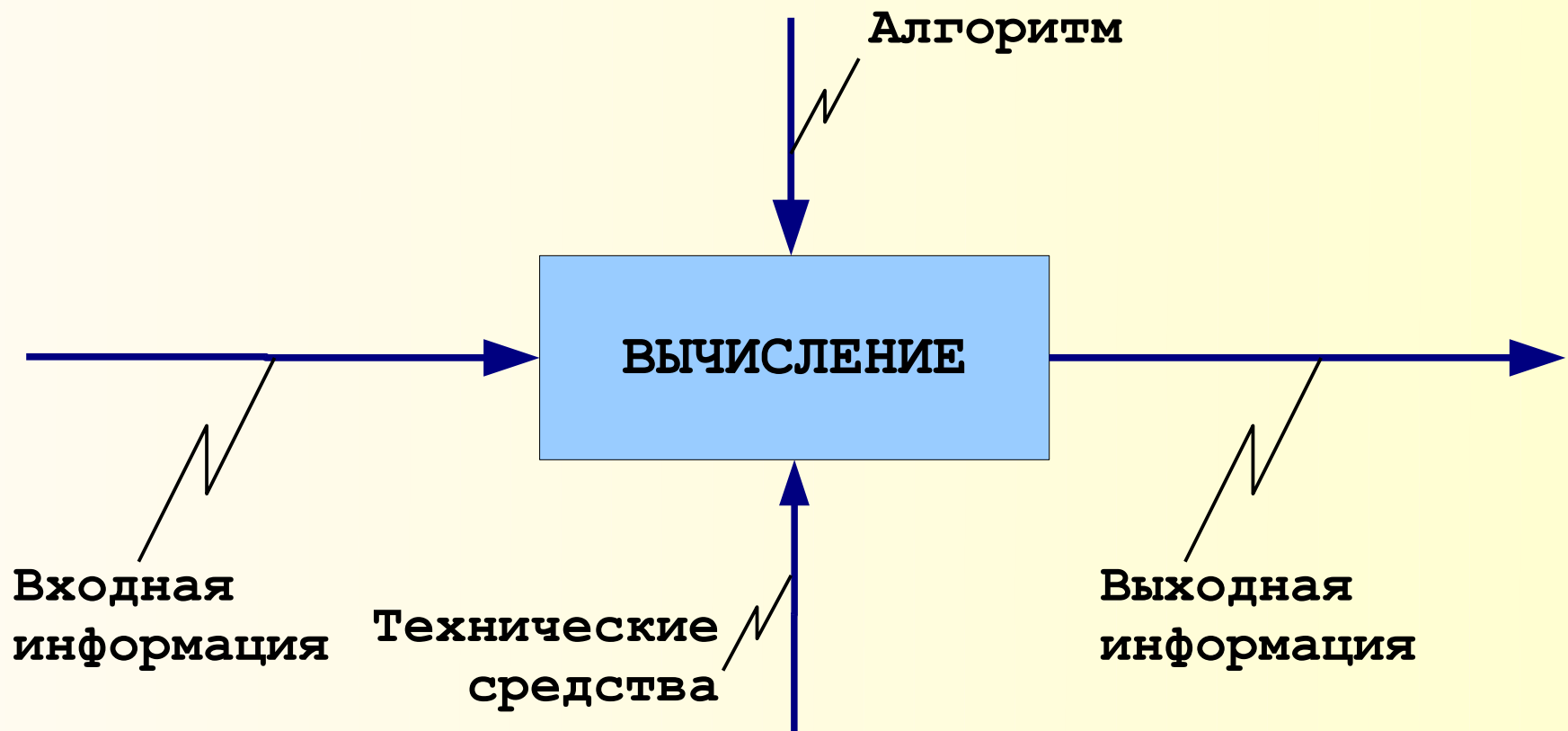
ввести метрики для оценки эффективности  
алгоритмов;

познакомиться с базовыми алгоритмами и  
структурами данных;

выяснить роль структур данных для повы-  
шения эффективности алгоритма

**Информатика (information and computing science)** — наука об информации, методах ее накопления, хранения, обработки и передачи при помощи технических средств

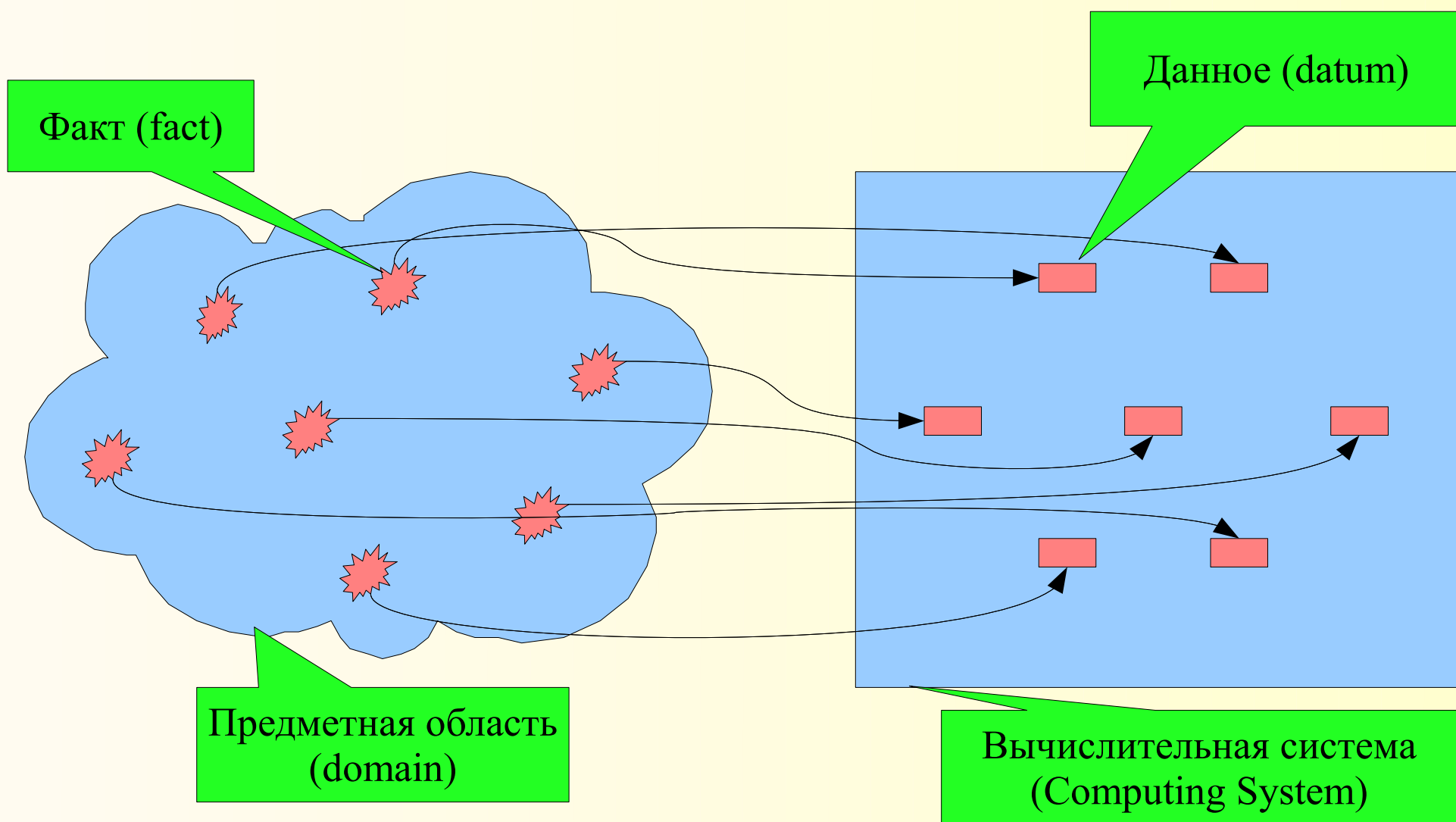
**Вычисление (computing)** — целенаправленный процесс преобразования информации



**Прикладная информатика (information technique)** — наука о методах построения вычислительных систем

**Вычислительная система (Computing System)** — комплекс артефактов (алгоритмов, программ, данных, методик, инструкций, технических средств и т. п.), предназначенный для выполнения необходимых пользователю преобразований информации в рамках специфицированной предметной области

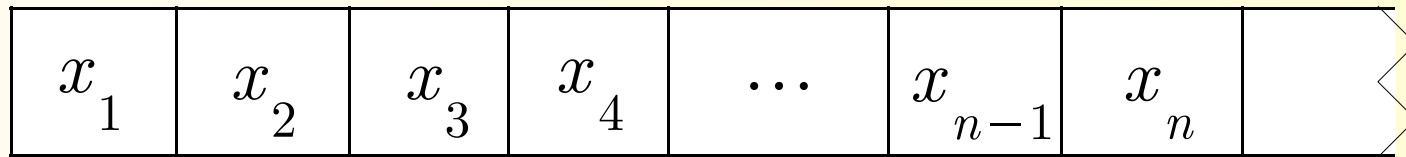
**Предметная область (domain)** — определенный аспект фрагмента реального мира, который должен быть отображен в базе данных вычислительной системы



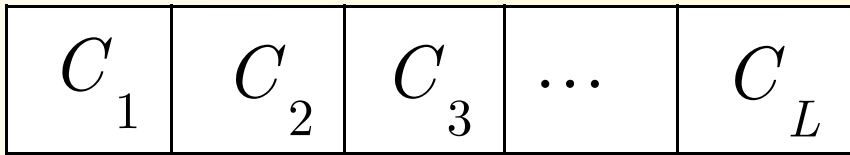
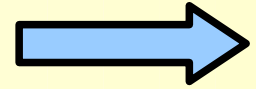
**Данные (data)** — факты или идеи, относящиеся к предметной области, выраженные средствами формальной системы, обеспечивающей возможность их хранения, обработки или передачи

**Алгоритм** — описание последовательности элементарных преобразований данных, проведение которых обеспечивает достижение поставленной цели – решение задачи

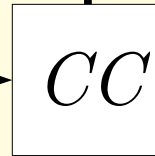
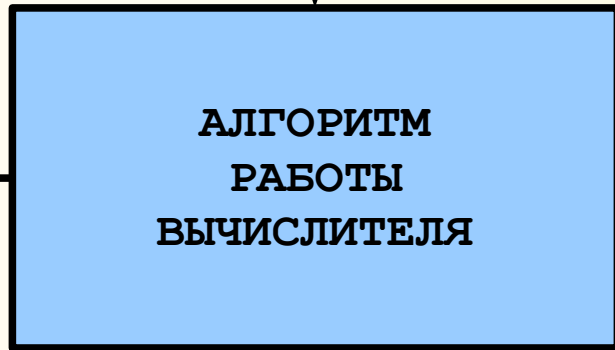
Алгоритм исполняет вычислитель. При этом требуется справедливости следующего условия: для выполнения алгоритма вычислителю не нужно использовать ничего, кроме входных данных и описания алгоритма. В этом смысле алгоритм является явной и полной спецификацией вычисления



Входные  
данные

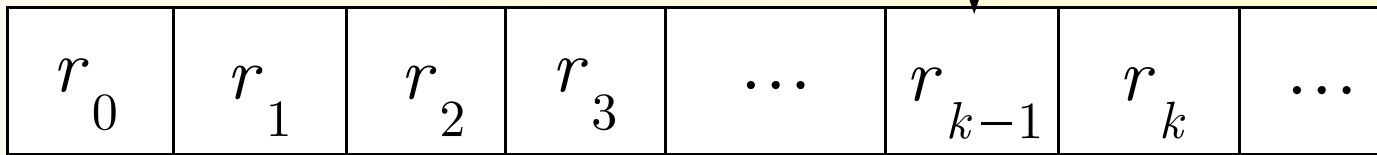


Программа

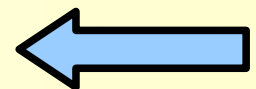


АЛГОРИТМ  
РАБОТЫ  
ВЫЧИСЛИТЕЛЯ

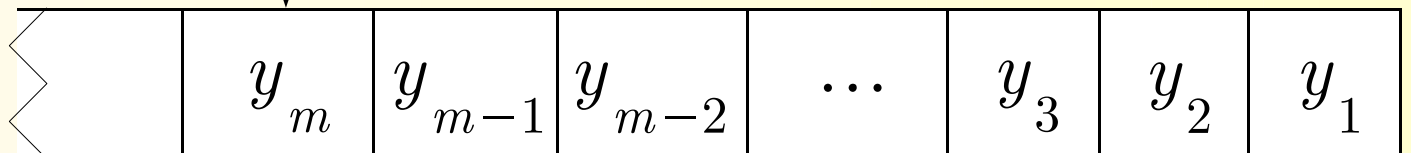
СС



Память



Выходные  
данные



# КОМАНДЫ ВЫЧИСЛИТЕЛЯ

КОМАНДА	СЕМАНТИКА КОМАНДЫ
LOAD a	$r[0] := v(a); CC := (CC) + 1$
STORE i	$r[i] := r[0]; CC := (CC) + 1$
STORE @i	$r[r[i]] := r[0]; CC := (CC) + 1$
ADD a	$r[0] := r[0] + v(a); CC := (CC) + 1$
SUB a	$r[0] := r[0] - v(a); CC := (CC) + 1$
MULT a	$r[0] := r[0] * v(a); CC := (CC) + 1$
DIV a	$r[0] := \max\{k   k * v(a) \leq r[0]\};$ $CC := (CC) + 1$
READ i	$r[i] := x[in]; in := in + 1;$ $CC := (CC) + 1$
READ @i	$r[r[i]] := x[in]; in := in + 1;$ $CC := (CC) + 1$
WRITE a	$y[out] := v(a); out := out + 1;$ $CC := (CC) + 1$
JUMP b	$CC := b$
JCTZ b	$CC := (r[0] > 0) ? b : (CC) + 1$
JZERO b	$CC := (r[0] = 0) ? b : (CC) + 1$
HALT	Завершение работы

**Адресация :**

=i

i

@i

**Целевой**

**адрес :**

$v(=i) := i$

$v(i) :=$

$(r[i])$

$v(@i) :=$

$(r[r[i]])$



# АЛГОРИТМ РАБОТЫ ВЫЧИСЛИТЕЛЯ

```
compute := proc (C — программа)
  CC := 1; in := 1; out := 1;
  while true do
    if (CC) > L or (CC) < 1 then
      return «Error: invalid address»
    elif C[(CC)] = 'HALT' then
      return «Normal termination»
    elif C[(CC)] isn't recognised then
      return «Error: unrecognized command»
    else
      execute_command(C[(CC)])
    end if
  end do
end proc
```