

Программирование

Лекция 1

I. История развития вычислительной техники

1. Разностная машина (1822, 1843-1891)
Вычисление логарифмических и тригонометрических таблиц разностными методами.
2. Аналитическая машина (1834-1864)
Фактически построена в 1941 инженерами IBM

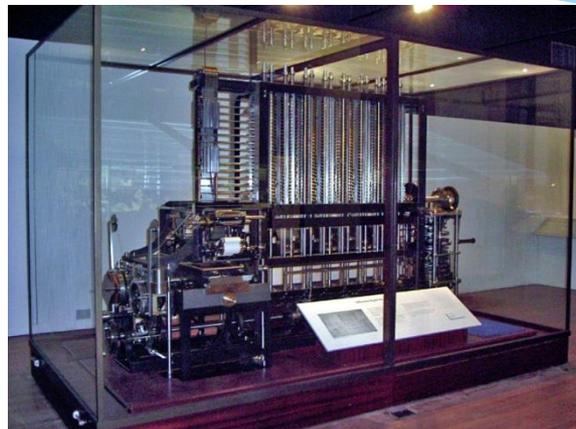


Чарльз Бэббидж
(1791-1871)

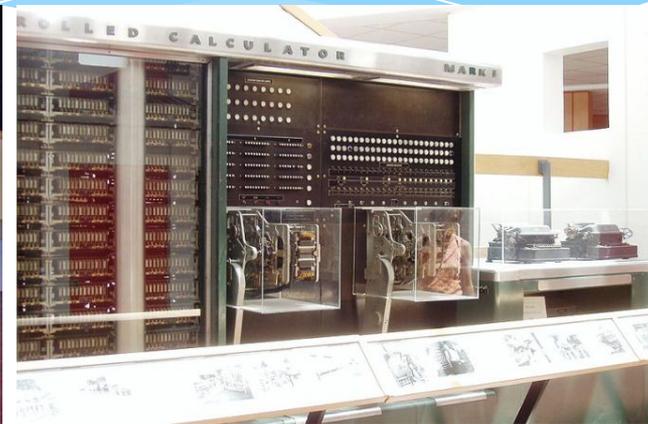
1.2 Вычислительные машины Бэббиджа



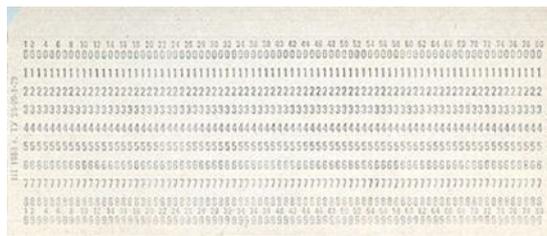
Малая вычислительная машина



Большая вычислительная машина



Аналитическая машина. Mark-1



Перфокарта.
Изобретена Жаккаром в 1808 году

1.3 Ада Байрон-Лавлайс

Первый программист!

Программы для аналитической машины Бэббиджа

Описаны основные программные конструкции которые используются по сей день.

В материалах Бэббиджа и комментариях Лавлейс намечены такие понятия, как подпрограмма и библиотека подпрограмм, модификация команд и регистр, которые стали употребляться только в 1950-х годах



Ада Лавлайс
(1815-1852)

1.4 Джон фон Нейман

Принцип использования двоичной системы счисления для представления данных и команд.

Принцип программного управления.

Программа состоит из набора команд, которые выполняются процессором друг за другом в определенной последовательности.

Принцип однородности памяти.

Как программы, так и данные хранятся в одной и той же.

Принцип адресуемости памяти.

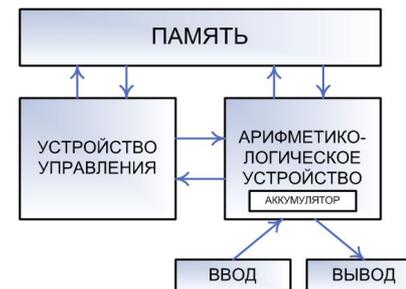
Структурно основная память состоит из пронумерованных ячеек; процессору в произвольный момент времени доступна любая ячейка.

Принцип последовательного программного управления

Все команды располагаются в памяти и выполняются последовательно, одна после завершения другой.

Принцип условного перехода.

Принцип изначально предложенный Адой Лавлейс



Машина фон
Неймана

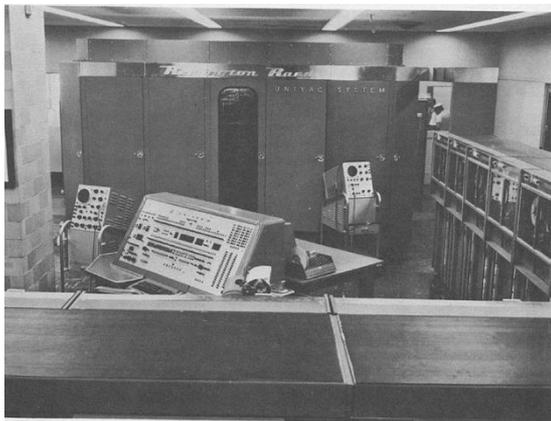


Джон фон Нейман
(1903-1957)

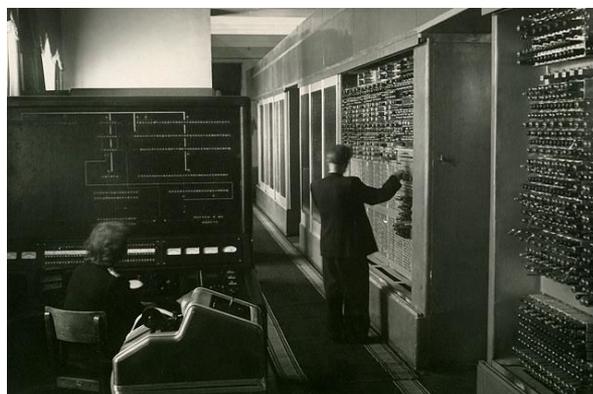
II. Первое поколение вычислительных машин 1950-1960

Особенности:

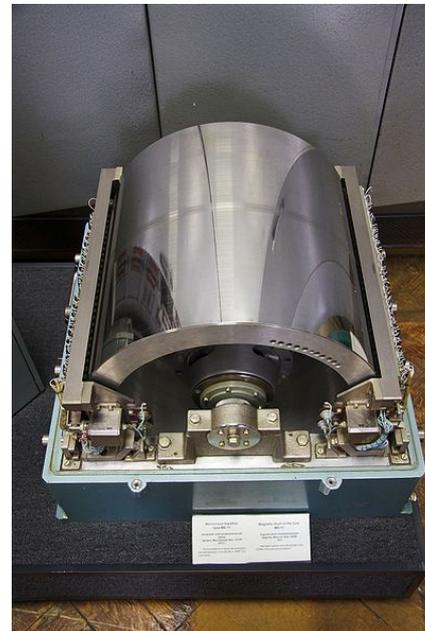
Электронные лампы, ЭЛТ, магнитные барабаны и ленты



UNIVAC - 1



БЭСМ - 1



Магнитный барабан

Второе поколение вычислительных машин 1960-1970

Особенности:

Полупроводники и магнитные элементы. Блочный принцип построения. Первые дисплеи (1964). Внешние накопители на жестких и гибких дисках



PDP-1



БЭСМ-6

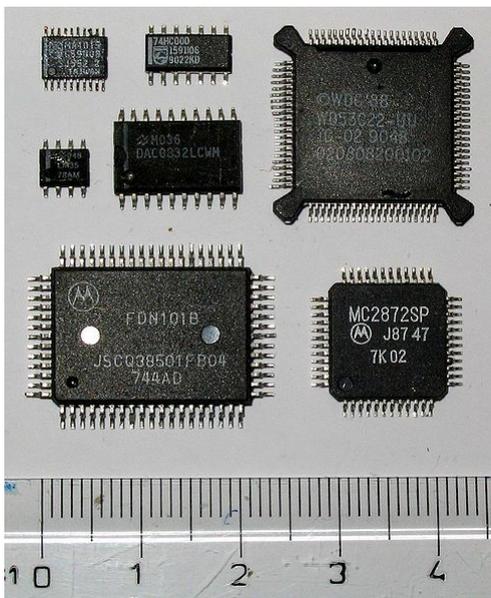


Внешние накопители

Третье поколение вычислительных машин 1970-1980

Особенности:

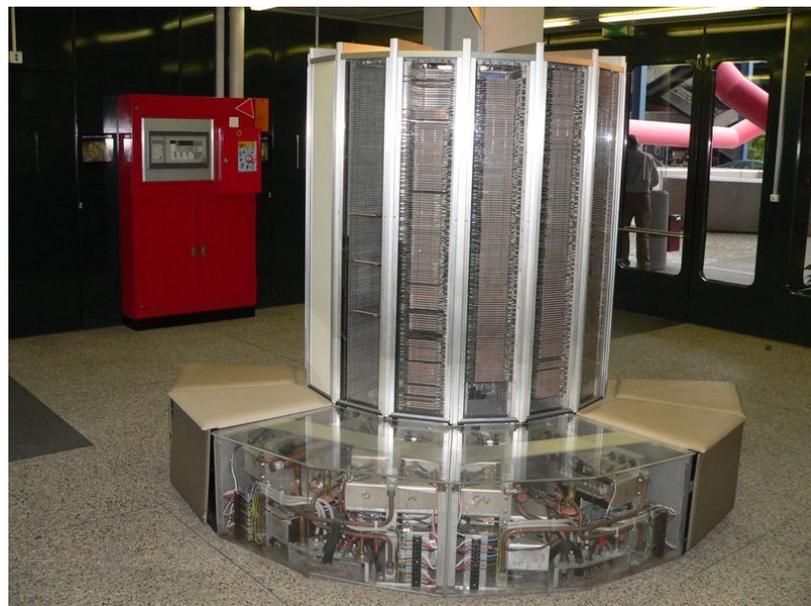
Интегральные схемы, ферритовые элементы хранения данных, графические интерфейсы, первые персональные компьютеры (Apple I, 1976)



Интегральные схемы



Xerox Alto, первый граф. интерфейс



Первый суперкомпьютер Cray-1

Четвертое поколение вычислительных машин 1980-1990

Особенности:

Большие интегральные схемы, развитие персональных компьютеров



ZX Spectrum



IBM 5150



Macintosh 128K

Пятое поколение вычислительных машин 1990-наше время

Особенности:

Повышение производительности, параллелизм, развитие сетей, программного обеспечения и интерфейсов



III. Языки Программирования

Язык программирования определяет набор лексических, синтаксических и семантических правил, используемых при составлении компьютерной программы.

Классификации языков:

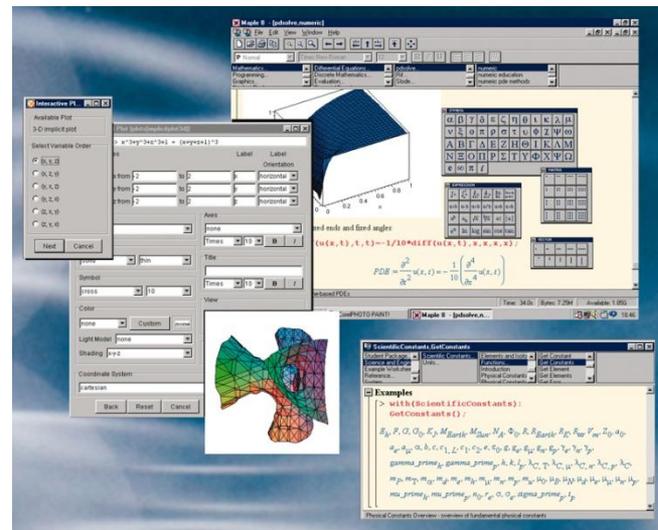
- Машинно-ориентированные
- Низкоуровневые
- Высокоуровневые
- Сверхвысокоуровневые
- Специализированные

- Интерпретируемые
- Компилируемые
- Смешанные

IV. Пакет символьных вычислений Maple

Программный пакет, система компьютерной алгебры. Является продуктом компании Waterloo Maple Inc., которая с 1982 года выпускает и продвигает на рынке программные продукты, ориентированные на сложные математические вычисления, визуализацию данных и моделирование.

Система Maple предназначена для символьных вычислений, хотя имеет ряд средств и для численного решения дифференциальных уравнений и нахождения интегралов. Обладает развитыми графическими средствами. Имеет собственный язык программирования.



Две среды разработки в одном пакете



Maple 13



Classic Worksheet Maple 13

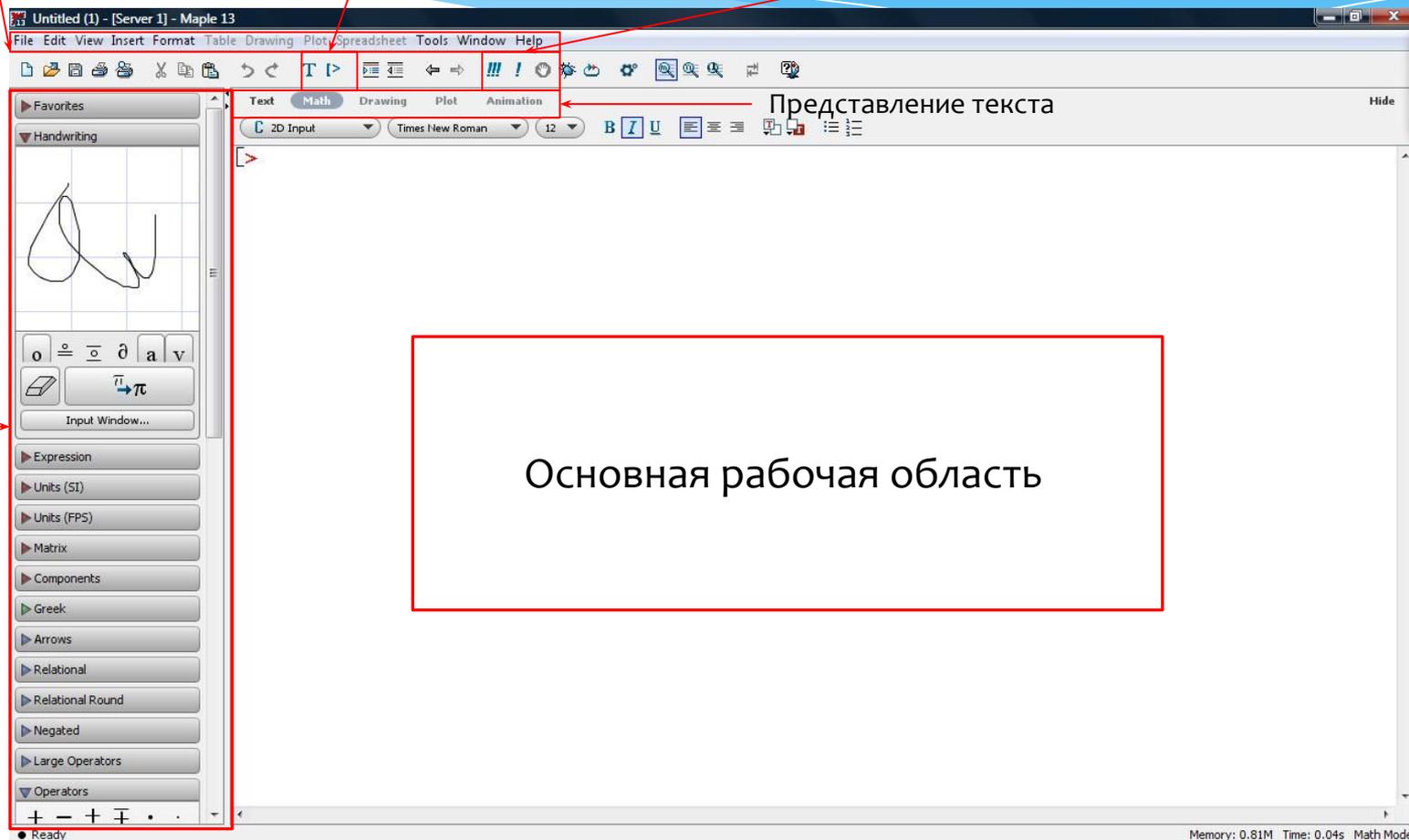
IV.1 Пользовательский интерфейс

Главное меню

Управление вводом команд

Управление процессом выполнения

Панель инструментов



Основная рабочая область

IV.2 Встроенная справочная система

В Maple встроенная справочная система содержит описание и примеры для всех команд и функций.

Вызов может осуществляться:

- Через меню Help в Главном Меню
- С помощью горячих клавиш

Ctrl+F1 (Ctrl+F2, в поздних версиях) – контекстная справка. Помощь по команде на которой стоит курсор

F1 – Меню поиска в подсистема помощи

F2 – Быстрая справка

IV.3 Основные объекты и типы данных

Основные объекты:

- Числа (1, 2.5, 10e2)
- Константы (π , I, infinity,...)
- Строки ("string", "name")
- Имена (A, B1, func, arr)

Типы данных:

- Выражения
 - x^2
 - $\cos(x) + \sin(x)^2$
- Уравнения/неравенства
 - $x^2 + 2x + 1 = 0$
 - $y > -2$
- Диапазоны/интервалы
 - 1 .. 10
 - A .. K
- Списки, множества, последовательности
 - $[[1,2,3],[4,5,6]]; \{1,2,"cos(x)","456"\}$

IV.4 Среда Maple как научный калькулятор

Каждая команда завершается символами «;» (точка с запятой) или «:» (двоеточие)

- Точка с запятой показывает результат выполнения команды сразу после нее
- Двоеточие выполняет команду, но не выводит результат на экран

```
> 1;
1
> a + b;
a + b
> z - 1/x;
z - 1/x
>
```

Операции:

- Сложение (+): $a+b$;
- Вычитание (-): $b-c$;
- Умножение (*): $x*y$;
- Деление (/): $1/x$;
- Возведение в степень (^): x^2 ;
- Остаток от деления (mod): $10 \bmod 3$;
- Присваивание (:=): $a:=1$;

Операции. Примеры

> 1 + 1;

2

> 1 - 2;

-1

> 2 · cos(x²) + sin(2 · x);

2 cos(x²) + sin(2 x)

> $\frac{1}{(x + 1)^2}$;

$\frac{1}{(x + 1)^2}$

> 10 mod 3;

1

> /

IV.6 Переменные и правила именования

Переменная в языке программирования — поименованная либо адресуемая иным способом область памяти, имя или адрес которой можно использовать для осуществления доступа к данным, находящимся в переменной (по данному адресу).

В Maple переменная это строка символов (имя) начинающееся с буквы.

Требования к имени переменной:

- Первым символом должны быть буква
- Больше и маленькие буквы различаются: Переменная **a** не одно и тоже что переменная **A**
- В имени переменной допускаются: латинский буквы, цифры и символ подчеркивания ()
- Не должно совпадать с каким-либо ключевым словом или именем команды

Значение в переменную записывается с помощью операции присваивания

Переменные. Примеры

```
> a := 1;
```

```
a := 1
```

```
> b := a + c;
```

```
b := 1 + c
```

```
> z1 := Pi - x2;
```

```
z := π - x2
```

```
> z0;
```

```
z0
```

```
> myfunc;
```

```
myfunc
```

```
> lvar := 1;
```

```
Error, invalid left hand side of assignment
```

```
>
```

IV.7 Функции, команды и библиотеки

Для решения задач в языках программирования используются различные управляющие конструкции и функции (команды). Под функцией понимается именованная подпрограмма выполняющая какие-либо операции. Как и в математике, в функции могут быть переданы входные параметры. Каждая функция возвращает значение(я).

Синтаксис вызова стандартной команды следующий:

Имя_функции(*par_1, par_2, ..., par_n*);

Здесь **имя_функции** это имя вызываемой функции, а *par_1, ..., par_n* – передаваемые параметры.

- Параметры могут быть значениями, именами переменных, выражениями, списками и т.д.
- Количество параметров зависит от конкретной функции.
- Параметры бывают обязательные и дополнительные

Т.к. функций в Maple очень много. Для удобства использования они разбиты на библиотеки. Библиотеки подключаются с помощью команды `with(<имя_библиотеки>)`

Функции. Примеры.

> `exp(x)`; # экспонента

$$e^x$$

> `sqrt(x)`; # квадратный корень

$$\sqrt{x}$$

> `abs(x)`; # модуль

$$|x|$$

> `ln(x)`; # натуральный логарифм

$$\ln(x)$$

> `log[y](x)`; # логарифм от x по основанию y

$$\frac{\ln(x)}{\ln(y)}$$

> `sin(x)`;

$$\sin(x)$$

> `cos(x)`;

$$\cos(x)$$

> `tan(x)`;

$$\tan(x)$$

> `cot(x)`;

$$\cot(x)$$

> `arcsin(x)`;

$$\arcsin(x)$$

> `arctan(x)`;

$$\arctan(x)$$

>

Литература

* В. Говорухин, Б. Цибулин

КОМПЬЮТЕР В МАТЕМАТИЧЕСКОМ ИССЛЕДОВАНИИ.

Часть I

* А.В.Матросов

МАРЛЕ 6. РЕШЕНИЕ ЗАДАЧ ВЫСШЕЙ МАТЕМАТИКИ И
МЕХАНИКИ

Сайт: www.spacephys.ru