



# ЯЗЫКИ ПРОГРАММИРОВАНИЯ

РАБОТУ ПОДГОТОВИЛ УЧЕНИК 9 «Б» КЛАССА  
ИЗМАЙЛОВ РАМИЛЬ

# Содержание

- ▶ Причины изучения разных языков программирования.
- ▶ История языков программирования.
- ▶ Принцип работы языков программирования.
- ▶ Мертвые языки программирования и причины их «смерти».
- ▶ Вывод.

# История языков программирования.

Кондар Цузе



Планкалькюль



# Фортан (фрагменты кода)

```

File Edit View Project Debug Tools Window Help FullScreen
QS12R_wrapper.F
(Globals) %SUBROUTINE mexFunction(nlhs, plhs, nrhs, prhs)
40: + "Too many output arguments"
41: endif
42:
43: if( mxIsClass( prhs(1), 'function_handle') .ne. 1) then
44:   call mexErrMsgTxt('1st input argument must be a
45: + function_handle')
46: elseif( mxIsClass( prhs(2), 'single') .ne. 1) then
47:   call mexErrMsgTxt('2nd input argument must be a single')
48: elseif( mxIsClass( prhs(3), 'single') .ne. 1) then
49:   call mexErrMsgTxt('3rd input argument must be a single')
50: elseif( mxIsClass( prhs(4), 'single') .ne. 1) then
51:   call mexErrMsgTxt('4th input argument must be a single.')
52: elseif( mxIsClass( prhs(5), 'int32') .ne. 1) then
53:   call mexErrMsgTxt('5th input argument must be an int32')
54: endif
55: -----
56:
57: input(1) = prhs(1) ! копируем указатель на ссылку на функцию (хорошо, что не надо думать про тип, просто копируем как есть)
58:
59: classid = mxClassIDFromClassname('single') ! определяем класс аргумента вычисляемой функции
60: input(2) = mxCreateNumericArray(2, (/1,1/),classid,0) ! создаем указатель на аргумент, сюда потом будем записывать аргумент функции
61:
62: ! у указателя prhs(2) надо взять ещё указатель...
63: call mxCopyPtrToReal4( mxGetPr(prhs(2)), A, 1) ! WARNING: переменная д/б REAL4
64: call mxCopyPtrToReal4( mxGetPr(prhs(3)), B, 1) ! WARNING: переменная д/б REAL4
65: call mxCopyPtrToReal4( mxGetPr(prhs(4)), err, 1) ! WARNING: переменная д/б REAL4
66: call mxCopyPtrToInteger4( mxGetPr(prhs(5)), N, 1) ! WARNING: переменная д/б INT
67:
68: call QS12R(RESULT, A, B, ourfunc, err, N, IERR) ! Выполнение функции, ради которой всё и затевалось
69:
70: write(line,*) 'A number of nodes = ',N ! Нам интересно итоговое количество узлов для заданной точности
71: k=mexPrintf(line//achar(13))
72:
73: plhs(1) = mxCreateNumericArray(2, (/1,1/),classid,0) ! В plhs(1) нет объекта, надо его создать
74: call mxCopyReal4ToPtr(RESULT, mxGetPr(plhs(1)), 1) ! Запишем результат на выход. Не забываем про соответствие типов
75:
76: ! не забываем за собой убирать
77: call mxDestroyArray(input(2))
78:
79:

```

```

FTN 5.1+642 87/02/02. 10.44.20 PAGE 18
SUBROUTINE RCKELI 74/175 OPT=0,ROUND= A/ S/ M/-D,-DS
DO=-LONG/-DT,ARG=-COMMON/-FIXED,CS= USER
/-FIXED,DB= TB/ SB/ SL/ ER/-ID/ PMD/-ST,-AL,PL=5000
FTNS,I=CHRLGS,L,LO,DB,PN,PW=90,PS=80.

```

```

1 SUBROUTINE RCKELI(N,A,Y,X)
2 PARAMETER (IA=50)
3 REAL A(IA,IA), Y(IA), X(IA)
4 INTEGER N
5 *
6 *****
7 *
8 * SUBROUTINE ZUR RUECKWAERTSELIMINATION, D.H. ZUR LOESUNG VON
9 * R*X=Y
10 *
11 * ERKLAERUNG DER FORMALEN PARAMETER:
12 *
13 * NAME TYP DIMENSION ERKLAERUNG
14 * -----
15 * N INTEGER 0 DIMENSION DER MATRIX A (E)
16 * A REAL 2 DREICKSZERLEGUNG DER MATRIX A
17 * (UEBERSPEICHERT) (E)
18 * Y REAL 1 VEKTOR, DER DIE LOESUNG VON
19 * L*Y=B ENTHAEHLT (E)
20 * X REAL 1 VEKTOR ZUR AUFNAHME DER LOESUNG VON
21 * R*X=Y (A)
22 *
23 *
24 *
25 *
26 * E: EINGABEPARAMETER
27 * A: AUSGABEPARAMETER
28 *
29 * -----
30 *
31 * KONSTANTEN ('PARAMETER' IN FORTRAN):
32 *
33 * IA: MAXIMALE DIMENSION DER MATRIX A
34 *
35 * -----
36 X(N)=Y(N)/A(N,N)
37 DO 10 K=N-1,1,-1
38 S=0.
39 DO 20 J=K+1,N
40 S=S+A(K,J)*X(J)
41 10 X(K)=(Y(K)-S)/A(K,K)
42 RETURN
43 END

```

```

--VARIABLE MAP--(LO=A/R)
-NAME---ADDRESS ---BLOCK-----PROPERTIES-----TYPE-----SIZE---REFERENCES-

```

NAME	ADDRESS	BLOCK	PROPERTIES	TYPE	SIZE	REFERENCES
A	2	DUMMY-ARG		REAL	2500	1 3 36
J	2018			INTEGER		39/C 40 40
K	176B			INTEGER		37/C 39/C 40 40
N	1	DUMMY-ARG		INTEGER		41 41 41
S	200B			REAL		41 36 36 36
X	4	DUMMY-ARG		REAL	50	37/C 39/C 38/S 40 40/S
Y	3	DUMMY-ARG		REAL	50	41 3 36/S

```

--SYMBOLIC CONSTANTS--(LO=A/R)
-NAME---TYPE-----VALUE-----REFERENCES-

```

NAME	TYPE	VALUE	REFERENCES
IA	INTEGER	50	2/S 3 3 3 3

# Smalltalk

## (фрагменты кода)

```
select: aBlock
| newCollection |
newCollection := self species new.
self do: [:each |
    (aBlock value: each)
    ifTrue: [newCollection add: each]].
^newCollection
```

```
| rectangles aPoint|
rectangles := OrderedCollection
    with: (Rectangle left: 0 right: 10 top: 100 bottom: 200)
    with: (Rectangle left: 10 right: 10 top: 110 bottom: 210).
aPoint := Point x: 20 y: 20.
collisions := rectangles select: [:aRect | aRect containsPoint:
aPoint].
```

# Java

## (фрагменты кода)

```
7  /**
8   *
9   * @author <i>Irina</i>
10  */
11  public class Student extends Teacher {
12  // fields
13     private int id;
14     private String name;
15     private int age;
16     private int unusedField; //unused field
17
18     public Student(int id, int age, String name) {
19         this.id = id;
20         this.name = name;
21         this.age = age;
22     }
23     @Deprecated
24     public int getAge() {
25         return age;
26     }
```

```
        this.time = 0;
        return 2;
    }

    // Step to the next time in the simulation
    this.time += tInc;

    final double degreeToRadianConv = 3.14 / 180.0;
    // First calculate the direction cosines for the cannon orientation.
    // Projection b of cannon length L,
    // b = L times cos (90 degrees - alpha)
    b = L * Math.cos((90.0 - Alpha) * degreeToRadianConv); // projection of barrel
                                                    // onto x-z plane

    // Cannon length, xyz. Lx = b * cos(gamma)
    Lx = b * Math.cos(Gamma * degreeToRadianConv); // x-component of barrel length
    Ly = L * Math.cos(Alpha * degreeToRadianConv); // y-component of barrel length
    Lz = b * Math.sin(Gamma * degreeToRadianConv); // z-component of barrel length

    cosX = Lx / L;
    cosY = Ly / L;
    cosZ = Lz / L;

    // These are the x and z coordinates of the very end of the cannon
    // barrel
    // we'll use these as the initial x and z displacements
    xe = L * Math.cos((90 - Alpha) * degreeToRadianConv) * Math.cos(Gamma * degreeToRadianConv);
    ze = L * Math.cos((90 - Alpha) * degreeToRadianConv) * Math.sin(Gamma * degreeToRadianConv);

    // Now we can calculate the position vector at this time
    // x = vx = (vm times cos(theta x)) * t
    s.i = Vm * cosX * time + xe;
    s.j = (Yb + (L * Math.cos(Alpha * degreeToRadianConv))) + (Vm * cosY * time) - (0.5 * g * (time * time));
    s.k = Vm * cosZ * time + ze;
    // Cutoff the simulation if it's taking too long
```

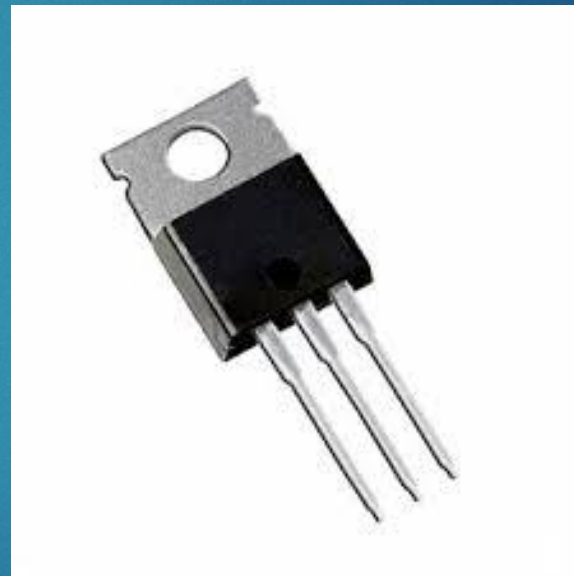
# C++

(фрагмент кода)

```
1  #include<iostream>
2  using namespace std;
3
4  int main() {
5  cout<<"Enter The Size Of Array:  ";
6  int size;
7  cin>>size;
8
9
10 int array[size], key,i;
11
12 // Taking Input In Array
13 for(int j=0;j<size;j++){
14 cout<<"Enter "<<j<<" Element: ";
15 cin>>array[j];
16 }
17
18 //Your Entered Array Is
19 for(int a=0;a<size;a++){
20     cout<<"array[ "<<a<<" ] = ";
21     cout<<array[a]<<endl;
22 }
```

# Принцип работы

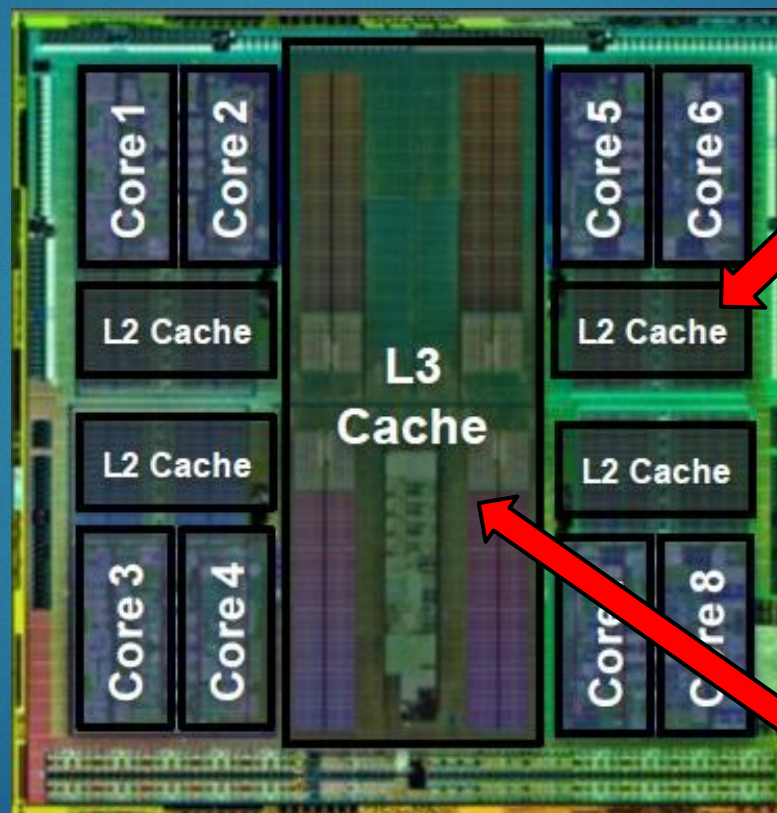
## Транзистор





# Память в процессоре

Кэш-память 1-го уровня  
расположена в ядре



Кэш-память 2-го  
уровня

Кэш-память 3-го  
уровня

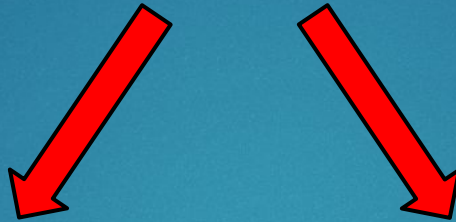
## Оперативная память



## Жесткий диск



# Языки программирования



Низкоуровневые  
(меньше функций,  
но больше  
производительность)

Высокоуровневые  
(Большое разнообразие  
в функциях, но и  
большие затраты  
ресурсов компьютера)



# Мертвые языки программирования

# COBOL

```
IDENTIFICATION DIVISION.  
PROGRAM-ID. SAMPLE.  
  
DATA DIVISION.  
WORKING-STORAGE SECTION.  
  
    77 fib1 pic 999.  
    77 fib2 pic 999.  
    77 fib3 pic 999.  
    77 i pic 99.  
    77 fibst pic XXX.  
    77 res pic X(64).  
  
PROCEDURE DIVISION.  
    move 0 to i  
    move 0 to fib1  
    move 1 to fib2  
    move "" to res  
    perform until i greater than 15  
        add fib1 to fib2 giving fib3  
        move fib2 to fib1  
        move fib3 to fib2  
        move fib1 to fibst  
        string res DELIMITED BY SPACE  
            fibst DELIMITED BY SIZE  
            "," DELIMITED BY SIZE into res  
        add 1 to i  
    end-perform.  
    display res "..."  
    stop run.
```

# C++

```
1 #include <iostream>  
2 using namespace std;  
3 int Fib(int i)  
4 {  
5     int value = 0;  
6     if(i < 1) return 0;  
7     if(i == 1) return 1;  
8     return Fib(i-1) + Fib(i - 2);  
9 }  
10 int main()  
11 {  
12     int i = 0;  
13     while(i < 47)  
14     {  
15         cout << Fib(i) << endl;  
16         i++;  
17     }  
18     return 0;  
19 }
```

# Befunge

```
vv < <
```

```
2
```

```
^ v<
```

```
v13v4
```

```
^ ^
```

```
> >?> ?>5^
```

```
v v
```

```
v97v6
```

```
v v<
```

```
8
```

```
. > > ^
```

```
^<
```







# Вывод и используемая литература

- ▶ <https://xakep.ru/>
- ▶ <https://map.hexlet.io>
- ▶ <https://ru.wikipedia.org>

Спасибо за внимание