

ИСТОРИЯ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ

Индивидуальный проект

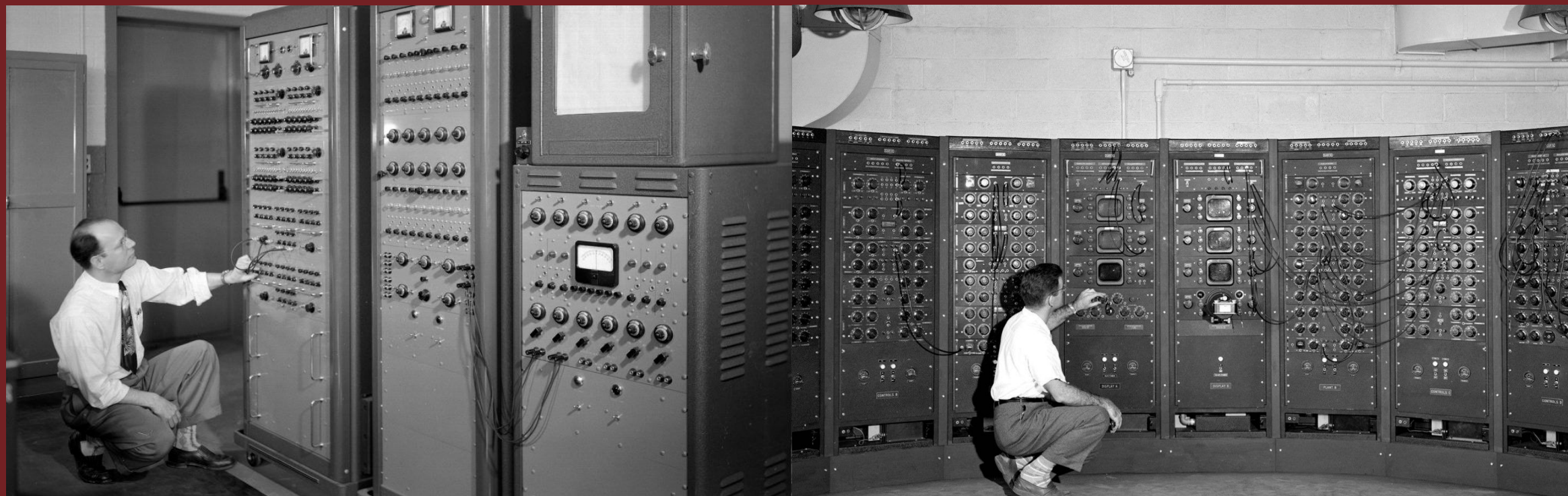
Автор: Скляр Максим Дмитриевич

*Руководитель проекта: Бабич Оксана
Дмитриевна*

2021г



КАК ВЫГЛЯДЕЛИ ПЕРВЫЕ ЭВМ :



ЧТО ПРЕДСТАВЛЯЛ ИЗ СЕБЯ МАШИННЫЙ КОД :

```
4C1C 7F 4E9E 1C7A4E 4E7C5A 1C4E 4E4F 7C1C 9E 1C4E4D4C9E 7A7A3A5C
5B 7F 5E5B4C4C1C5B5A6D5E9E4C 4F9B7C 9E1C 9B4C5E 5A5B 1C9B7C9E
6C4D4E 7C 9B5B 5E9B 1C3A4E9B 3A 7F3A 6D4C 7F9E3B4D 9B3A7C
9E 5E5E 4C4E 7A4C5C6C7C4C7C7A 7A 9C 9B 4D 5B5B 5B4E 4D5E
5C5E5A4C 6C5E7F 4D4E7A 4C9B9B3B 9B 6C4C9E5A7A5C5E5A 5B6D 9B9B7F
5A3B1C5E9E 5B7A 5B5A5A5A 5E 3A4C4C7A 5B7A9B 5E7C 7A4E7A
5B6D 1C 7C4C9B 5E 7A3B 7C5E 1C5B 6D 5A1C 4D5A5B 7A
5B9E6D4E5E1C9E 1C9C1C5E4C 5E7C9B 9B5B4C 4E 5B 7A9E7A9E1C7F9E
3B 4C9C5E7A4C1C5B6D4D5B1C5B 4C7C 5E4C1C1C5A4C4E 5E6C4D4C4C 4C 9E9B
5A 3B4D 4C 6C7F1C 1C4C4E7C5E4C 9B 4D5B9B5E 1C7A4D4C5C7A5A7F
9C 4C 4C9E4E4D 9B 4E7A 1C7A7C9E 5A5B9B4E7C 7A 4C9C1C4E6C5B5E
9B4C9E 7F9B 3B 5E 4D 4D6D 1C9B 9B 5B5E9B5B5B
1C 6C5B1C 9E4F7F 3B 7A 5A4C3A 9E1C4E3B4E 9E7A7F 1C7C 3A9E 7F4C
1C1C6C 4C 9E5E5C5B7F 7C 9E4E1C9E9B 7F 4C9B4F9B 3B3B9E6D7C3B4C5E3A
1C 5E7A7C4D 4D4C5C5B6D5A9E 6D7F 9E 3B 7C 4C4C4E4E7A1C 4C3A3B4E7C
3B4C 3B 1C5E 7F3A 7F9B4C 4C5E4C 9B 7F 6D3B7F 4C 5E4C4C5E
9B9E5E 4C1C5B5E 3A 5A7F 5A5E7A4D 7A 7C7F3B6C 5A6D1C1C
5B9E 5B4C9C9E5E 4D3B 6D 4E 1C7A1C 3B 6C 4E 5C5B4F9B 6C3A 5A
4C7A5B4E 7F 7A3B 9B4C 1C 7A9B5A4E9E 7C1C6D9B 7A3B9E5E4E7F
4C7F 3B5B 5C6C5E9B7F 1C9B3B 4F 9B9B6D9B 3B6D4C3A 3B4C9E9B1C 7C
5B4D1C 5A4C7A5E7F 4E6D5A5E 5E7C 3B3B5B 7A6C 5E 7C1C4C4F4E4C6C
9E9B5A3B5B 9B9B1C1C5E4C5B 3A4D4F5A4F5E 4E 3B 5B3B 9C4C 7F9B
3B 7F5B5B7C9B4E 9B4C7A5E9B9B4D4E 7F6C 1C 7F 3B 5B 4E6C4E 5E
1C9E5E5E 4E4D4C6C5E9B 9C4C7C 4C7C9B 7F 4C9E 4F5E4C4E 5A
5E4C7C7A 4E1C5B1C6D4C 7F3B4C 9B5B6D9B3A 9E4F 4E 4E5A5E 7C
1C 9B5C 4E4C 7C 3B5B5B4D3B 9E9C7A4C 4C4C7C 9B9B4C4C
9B 4C 7C5B5A4E4C4E3B 5A 4F6C5E4E9B9E4F9C7F 3B7A4F 5A6C5E 5B1C 3A
9B5C1C1C 4D5B5E7C7A4C 5E4C 5A9E 9E 6D3B9B 4E 6C 6C5E 3B5E7C9B
5B4C 7F 6D 1C7F 3A 6D 9E3A 4E 5C9B3B7A 5E9E4E4C 4C7F
6C9E 5C 7C4C 4C 7F1C9E 9C 5B6C4C4C1C4C4E5E3B 4C4E 5A4C5E6C5E7F
3A1C 7F9E 5C 4C4C 3B4E7A3B 3A1C5E 7A3B4E4C5E 1C6D6D7F
4D 5E5B6C 7A4C6D1C 9E 4E 7F1C5B9B 7C 6C 5B 9B3B 4C5A4E4E
3A5A7A 9B1C3B 1C 7A 3B 5E 5A1C4F4F 1C6D9E 9B 9B7C4E1C4C5B9B
5E7F5B 6C 6C4C4C4C5E7C3A7F4C5E6C 5A 5A 9B 4F 4C4C5B4E4C1C5A5E 5E
3A5E 6D4C 3B 4D4F 5B6C 4C7F6D 6C 5E 3A4C 5E 5A4C
```



КАК ВЫГЛЯДЕЛИ КОДЫ НА АССЕМБЛЕРЕ?

```

; --- I категория      ; ZZ
CMP  EAX, 0x841F0F66  ; 3D
TEST EAX, 0x841F0F66 ; A9
; --- II категория
PUSH 0x841F0F66      ; 68
MOV  EAX, 0x841F0F66 ; B8
MOV  ECX, 0x841F0F66 ; B9
MOV  EDX, 0x841F0F66 ; BA
MOV  EBX, 0x841F0F66 ; BB
MOV  ESP, 0x841F0F66 ; BC
MOV  EBP, 0x841F0F66 ; BD
MOV  ESI, 0x841F0F66 ; BE
MOV  EDI, 0x841F0F66 ; BF
; --- III категория
ADD  EAX, 0x841F0F66 ; 05
OR   EAX, 0x841F0F66 ; 0D
ADC  EAX, 0x841F0F66 ; 15
SBB  EAX, 0x841F0F66 ; 1D
AND  EAX, 0x841F0F66 ; 25
SUB  EAX, 0x841F0F66 ; 2D
XOR  EAX, 0x841F0F66 ; 35
; --- IV категория
MOV  AL, BYTE PTR [0x841F0F66] ; A0
MOV  EAX, DWORD PTR [0x841F0F66] ; A1
MOV  BYTE PTR [0x841F0F66], AL ; A2
MOV  DWORD PTR [0x841F0F66], EAX ; A3
CALL 0x841F0F66 ; E8
JMP  0x841F0F66 ; E9

```

```

                                MEMORY
0000 d6 22 26 92 0f 0f 2f 6f  ."&.../o
0008 e9 f5 e5 60 69 01 73 ff  ...`l.s.
0010 09 da 22 00 1b 00 44 4d  .."....DM
0018 7a b3 20 09 cd 31 00 e1  z. .l..
0020 f1 c9 c3 16 00 1b c5 01  .....
0028 ac ff cd 31 00 c1 c3 18  ...j....
0030 00 04 10 22 79 d6 6f fe  ..."y.o.
0038 18 da 43 00 d6 18 fe 18  ..C.....
0040 d2 3c 00 21 62 00 87 4f  <.lb..0
0048 09 7e 23 66 6f e9 3e 0e  ."#fo.>.

                                M Z H PE N C IFF=DI
                                MNEMONIC      STACK
>0000 SUB 34                000b 26976 6960  l'  A=255 (11111111) A'=000 (0
0002 LD H,146             0009 58869 e5f5  ..  F=255 (11111111) F'=000 (0
0004 RRCA                 0007 59759 e96f  .o  B=000 (00000000) B'=000 (0
0005 RRCA                 0005 12047 2f0f  /.  C=000 (00000000) C'=000 (0
0006 CPL                 0003 03986 0f92  ..  D=000 (00000000) D'=000 (0
0007 LD L,A              0001 09762 2622  &"  E=000 (00000000) E'=000 (0
0008 JP (HL)             >ffff 54784 d600  ..  H=000 (00000000) H'=000 (0
0009 PUSH AF              fffd 00000 0000  ..  L=000 (00000000) L'=000 (0
000a PUSH HL              fffb 00000 0000  ..
000b LD H,B              fff9 00000 0000  ..  I=x00 (00000000) R=x00 (0

```

FOLLOW

КАК ВЫГЛЯДЕЛИ КОДЫ НА ФОРТРАНЕ:

```
5 100 FORMAT(I5)
6     IF (K.LE.O.OR K.GT.TAMMÁX) STOP
7     READ *,(A(I),I=1,K)
8     PRINT *,(A(I),I=1,K)
9     PRINT * ,, 'SUMA=', SUM(A,K)
10    GO TO 10
11 99  PRINT * ,, "Todo listo"
12    STOP
13    END
14 SUBPROGRAMA DE SUMATORIA EN C
15 FUNCTION SUM(V,N)
16 REAL :: V(N) ! Declaración de estilo nuevo
17 SUM = 0.0
18 DO 20 I = 1,N
19     SUM = SUM + V(I)
```

A Program In FORTRAN

```
C FORTRAN EXAMPLE PROGRAM
C INPUT: AN INTEGER, LIST_LEN, WHERE LIST_LEN IS LESS
C        THAN 100, FOLLOWS BY LIST_LEN-INTEGGER VALUES
C OUTPUT: THE NUMBER OF INPUT VALUES THAT ARE GREATER
C        THAN THE AVERAGE OF ALL INPUT VALUES
C        DIMENSION INTLST(99)
C        IMPLICIT INTEGER(A, C, R, S)
C        RESULT = 0
C        SUM = 0
C        READ(5,501) LSTLEN
501 FORMAT(I3)
C        IF (LSTLEN) 106, 106, 101
C        IF (LSTLEN - 100) 101, 106, 106
101 CONTINUE
C READ INPUT DATA INTO AN ARRAY AND COMPUTE ITS SUM
DO 102 COUNTR = 1, LSTLEN
```

КАК ВЫГЛЯДЕЛИ КОДЫ НА ALGOL 68:

ALGOL 68

```
INT i, j, k, m, n; REAL a, b, x, y; BOOL p, q, overflow; CHAR c;
  FORMAT f; BYTES r; STRING s; BITS t; COMPL w, z; REF REAL xx, yy;
  UNION(INT, REAL) uir; PROC VOID task1, task2;
[ 1: n] REAL x1, y1; FLEX [1:n] REAL a1; [1: m, 1: n] REAL x2;
  [1: n, 1: n] REAL y2; [1: n] INT i1; [1: m, 1: n] INT i2; [1: n] COMPL z1;
PROC x or y = REF REAL: IF random < .5 THEN x ELSE y FI;
PROC ncos = (INT i)REAL: cos(2 * pi * i / n);
PROC nsin = (INT i)REAL: sin(2 * pi * i / n);
PROC finish = VOID: GO TO stop;
MODE BOOK = STRUCT(STRING text, REF BOOK next);
BOOK draft;
princeton: grenoble: st pierre de chartreuse: kootwijk: warsaw:
  zandvoort: amsterdam: tirrenia: north berwick: munich:
  finish
```

Draft Revised Report on the Algorithmic Language ALGOL

528-120

ROWS of MODE'•}.

The value "built" from a multiple value #V•, for some
a multiple value #W• determined as follows:
se A: #'ROWS'• is #'row'•:
▪ #W• is composed of
 (i) {one element} #V•,
 (ii) a descriptor #(1, fixed, (1, 1, 1))•;
se B: #'ROWS'• is some #'row ROWS2'•:
▪ #W• is composed of
 (i) the elements of #V•,
 (ii) a descriptor
#(c, fixed, (1, 1, d), (l1, u1, d1), ..., (ln, un, dn))•,
 where #d = (u1 - l1 + 1) #d1•, and
 where #(c, s, (l1, u1, d1), ..., (ln, un, dn))• is the descr
#V•.

КАК ВЫГЛЯДЕЛИ КОДЫ НА COBOL :

COBOL example (Contd.)

```
000312 01  SUMMARY-LINE.
000313      05 FILLER PIC X(7) VALUE SPACES.
000314      05 TOTAL-READ PIC 99.
000315      05 FILLER PIC X VALUE SPACE.
000316      05 FILLER PIC X(17) VALUE 'Records were read'.
000318      05 FILLER PIC X(53) VALUE SPACES.
000319
000320 PROCEDURE DIVISION.
000330 PREPARE-SENIOR-REPORT.
000340      OPEN INPUT STUDENT-FILE.
000350          OUTPUT PRINT-FILE.
000351      MOVE ZERO TO RECORDS-WRITTEN.
000360      READ STUDENT-FILE
000370          AT END MOVE 'NO' TO DATA-REMAINS-SWITCH.
000380      END-READ
```

```
000120      MOVE REPORT-TITLE TO CRTOUT.
000130      PERFORM WRITE-CRTOUT.
000140      ADD 1 TO LINE-COUNT.
```

(2) SELECTION

```
001000      IF TEST = "M"
001010*         THEN
001020             MOVE 1 TO TEST-CASE
001030         ELSE
001040             MOVE 2 TO TEST-CASE
001050*         ENDIF
```

(3) DO-WHILE

```
004010      PERFORM VALUE-PRINT
004020          UNTIL COUNT > 10.
004030
*
*
```

ВЫВОД:

Языки программирования начинают оказывать на нашу жизнь довольно сильное влияние. Ни один день не проходит без использования компьютерных или мобильных приложений, которые созданы благодаря языкам программирования. Прошло то время, когда работать с одной и той же версией языка программирования можно было десятилетиями. В наше время особенность работы программиста состоит в постоянном изучении новых языков и технологий. Курсы по программированию могут быть хорошим трамплином, но основной опыт программисты получают в процессе работы, как бы учась и работая одновременно. Начиная со школьного возраста стоит начинать вникать и осваивать эту тему, ведь за этим стоит будущее.

КАК ВЫГЛЯДЕЛИ КОДЫ НА LIPS:

Язык программирования: LISP

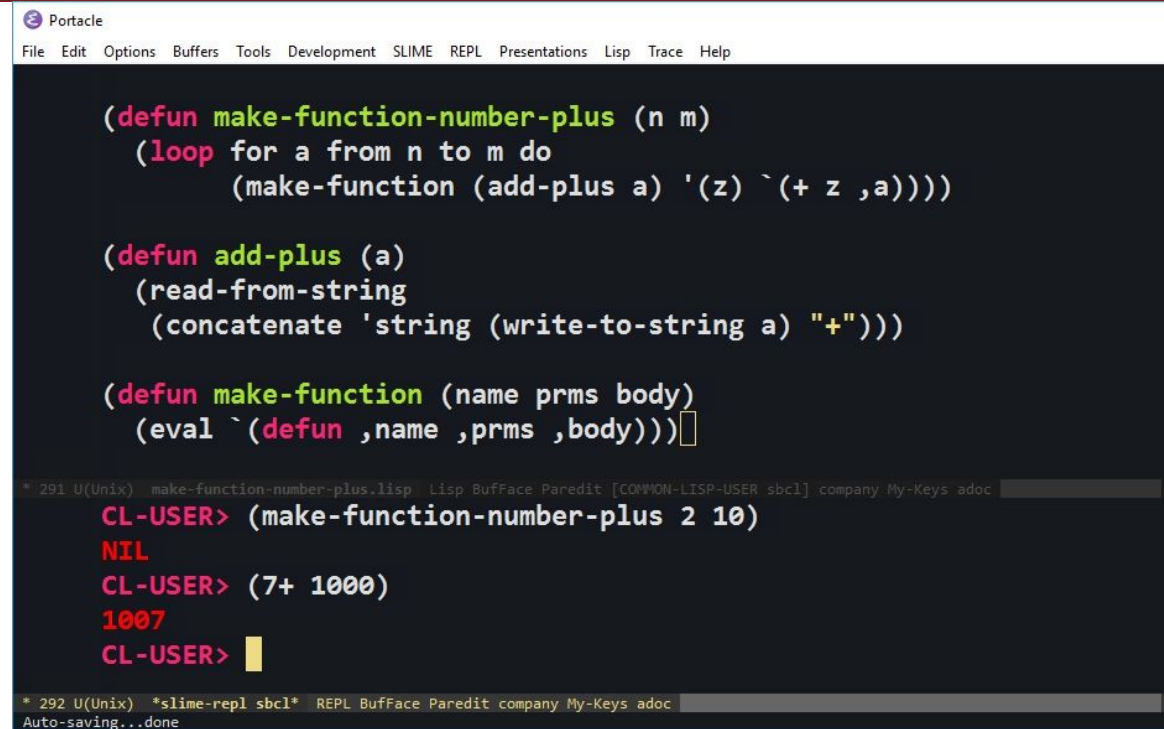
Программа 1:

```
(defun factorial (n)
  (if (= n 0)
      1
      (* n (factorial (- n 1)))))
```

Программа 2:

```
(defun fact-iter (result counter)
  (if (= counter 0)
      result
      (fact-iter (* counter result)
                  (- counter 1))))

(defun factorial (n)
  (fact-iter 1 n))
```



The screenshot shows a Lisp REPL window titled "Portacle". The menu bar includes "File", "Edit", "Options", "Buffers", "Tools", "Development", "SLIME", "REPL", "Presentations", "Lisp", "Trace", and "Help". The main area displays the following code and execution results:

```
(defun make-function-number-plus (n m)
  (loop for a from n to m do
    (make-function (add-plus a) '(z) `(+ z ,a))))

(defun add-plus (a)
  (read-from-string
   (concatenate 'string (write-to-string a) "+")))

(defun make-function (name prms body)
  (eval `(defun ,name ,prms ,body)))
```

Execution results:

```
* 291 U(Unix) make-function-number-plus.lisp: Lisp BufFace Paredit [COMMON-LISP-USER sbcl] company My-Keys adoc
CL-USER> (make-function-number-plus 2 10)
NIL
CL-USER> (7+ 1000)
1007
CL-USER> 
```

At the bottom, the status bar shows: "* 292 U(Unix) *slime-repl sbcl* REPL BufFace Paredit company My-Keys adoc Auto-saving...done"

СПИСОК ИСПОЛЬЗУЕМЫХ МАТЕРИАЛОВ:

Интернет ресурсы:

1. https://ru.wikipedia.org/wiki/Язык_программирования
2. https://ru.wikipedia.org/wiki/История_языков_программирования

Книги:

3. Campbell-Kelly M. From Airline Reservations to Sonic the Hedgehog: A History of the Software Industry // MIT Press, 2003
4. Campbell-Kelly M. A History of the Software Industry // MIT Press, 2003
5. Bergin T. J., Gibson. R. G. History of Programming Languages // ACM Press, 1996