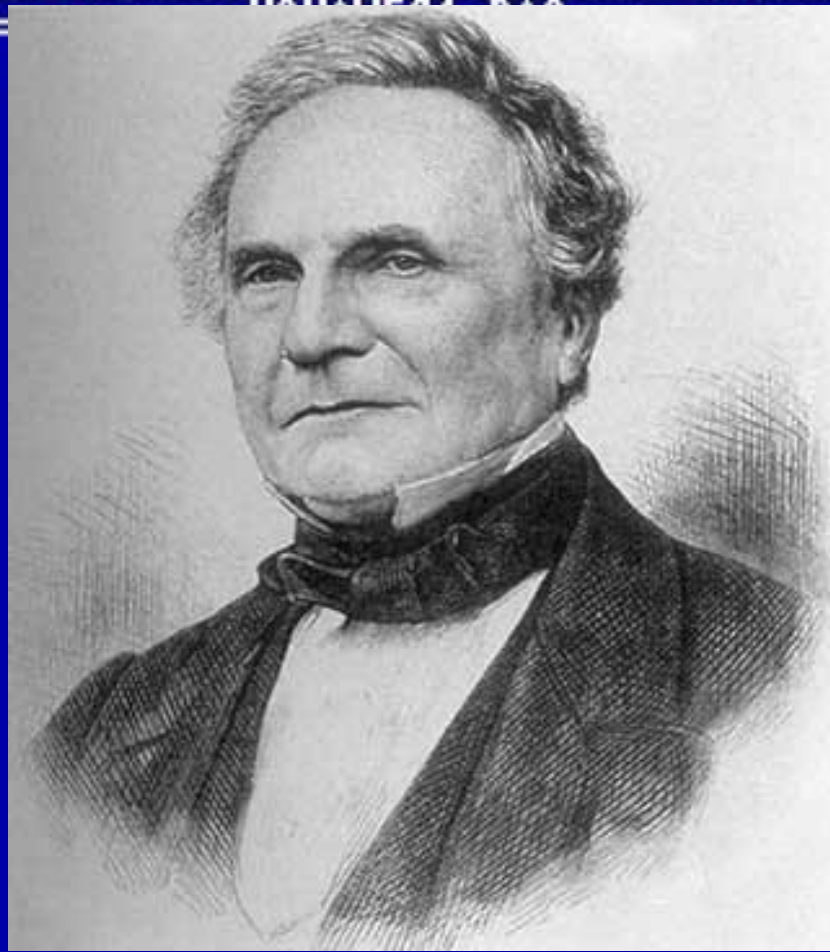


История развития языков программирования



Чарльз Бэббидж





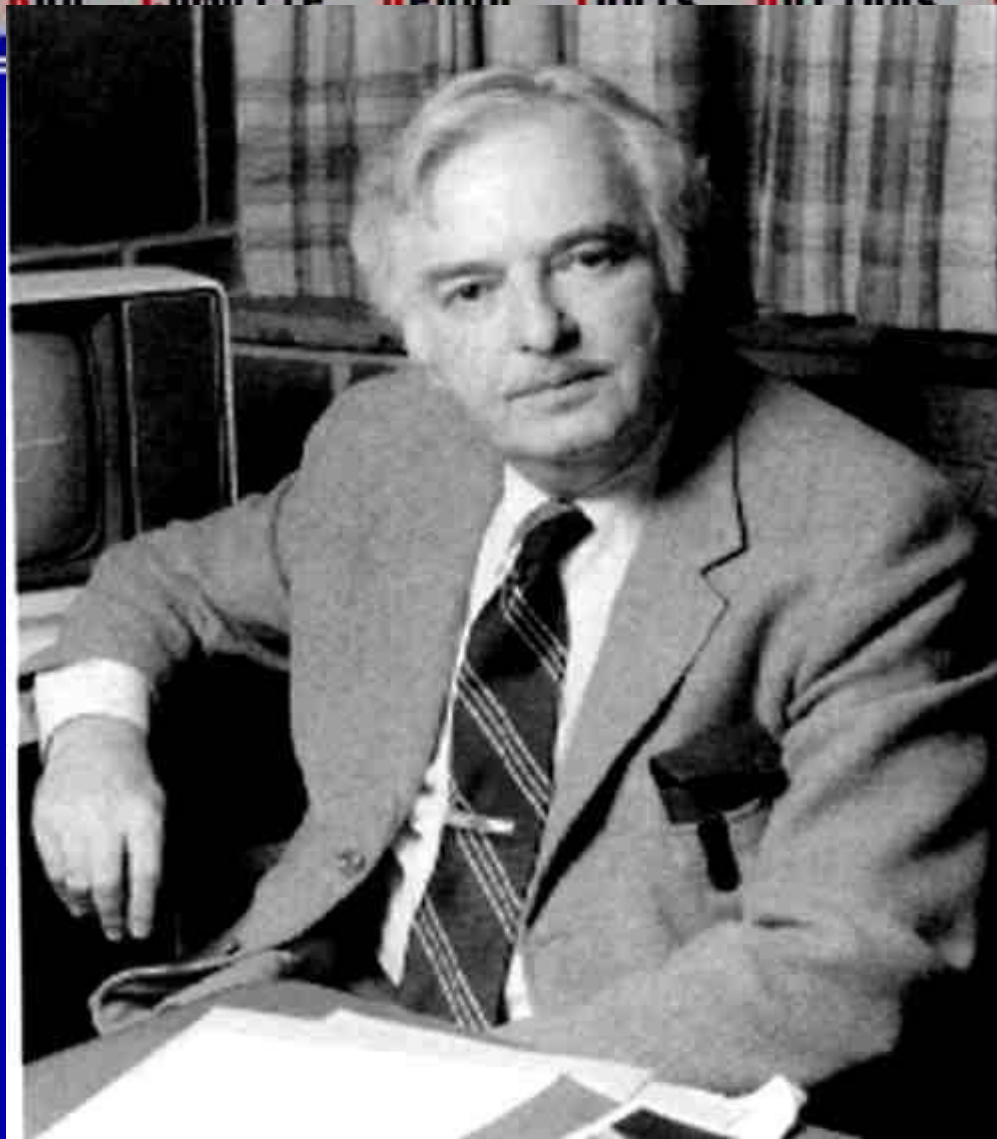
Джон Мокли



Марк-1

1:1

F1 Help F2 Save F3 Open Alt+F9 Compile F9 Make Alt+F10 Local menu



Джон Джордж Кемени

1:1

F1 Help F2 Save F3 Open Alt+F9 Compile F9 Make Alt+F10 Local menu

```

10 REM Диаграмма
20 SCREEN 9
30 COLOR 14, 1
40 DATA 11500, 2300, 9700, 5100, 12400, 8200
45 DATA "Токио", "Гамбург", "Москва", "Бангкок", "Мехико", "Париж"
50 LINE (40, 300)-(550, 300): LINE (40, 300)-(40, 20)
60 FOR k = 1 TO 6
70 READ m(k)
80 NEXT k
90 a = m(1)
100 FOR k = 2 TO 6
110 IF m(k) > a THEN a = m(k)
120 NEXT k
130 FOR k = 1 TO 6
140 READ name$(k)
150 NEXT k
160 FOR k = 1 TO 6
170 n = 10 * k + 3

```

Basic

Immediate



THE

C

PROGRAMMING
LANGUAGE

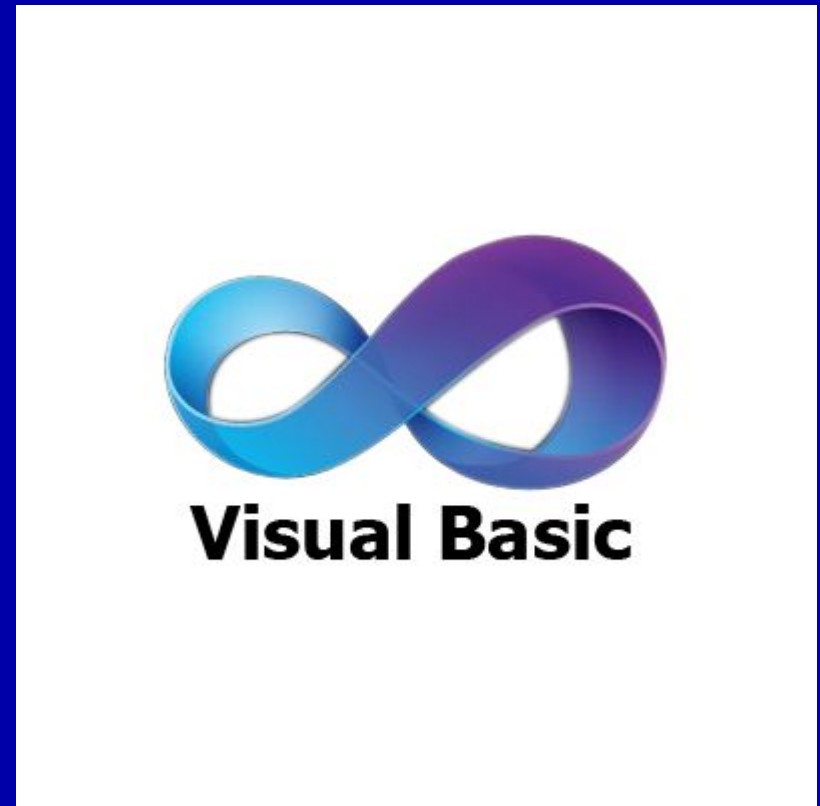






Паскаль.Н.В

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <unistd.h>
4 #include <sys/types.h>
5 #include <sys/stat.h>
6
7 int main(int argc, char **argv)
8 {
9     int memory[32768], space[32768];
10    int fd, len, buf, spointer, npointer;
11    int kl = 0;
12
13    if (argc <= 1)
14    {
15        printf("Usage: %s <brainfuck file>\n", argv[0]);
16        exit(1);
17    }
18
19    /* Open brainfuck file */
20    if ((fd = open(argv[1], O_RDONLY)) < 0) exit(1);
21
22    /* Read the file byte by byte */
23    len = spointer = 1;
24    while (len > 0)
25    {
26        len = read(fd, &buf, 1);
27        space[spointer] = buf;
28        spointer++;
29    }
30
31    close(fd);
32
33    for (npointer = 0; npointer < 32768; npointer++) memory[npointer] = 0;
34
35    len = spointer;
36    spointer = npointer = 0;
37
38    for (spointer = 0; spointer < len; spointer++)
39    {
40        switch(space[spointer])
41        {
42            /* Increment pointer value */
43            case '+':
44                memory[npointer]++;
45                break;
46            /* Decrement pointer value */
47            case '-':
48                memory[npointer]--;
49                break;
50            /* Increment pointer */
51            case '>':
52                npointer++;
53                break;
54            /* Decrement pointer */
55            case '<':
56                npointer--;
57                break;
58            /* Print current pointer value */
59            case '.':
60                putchar(memory[npointer]);
61                break;
62            /* Read value and store in current pointer */
63            case ',':
64                memory[npointer] = getchar();
65                break;
66            /* Start loop */
67            case '[':
68                if (memory[npointer] == 0)
69                {
70                    /* Find matching ] */
71                    spointer++;
72                    while (kl > 0 || space[spointer] != ']')
73                    {
74                        if (space[spointer] == '[') kl++;
75                        if (space[spointer] == ']') kl--;
76                        /* Go in right direction */
77                        spointer++;
78                    }
79                }
80                break;
81            /* End loop */
82            case ']':
83                if (memory[npointer] != 0)
84                {
85                    /* Find matching [ */
86                    spointer--;
87                    while (kl < 0 || space[spointer] != '[')
88                    {
89                        if (space[spointer] == '[') kl--;
90                        if (space[spointer] == ']') kl++;
91                        /* Go left */
92                        spointer--;
93                    }
94                }
95                break;
96        }
97    }
98
99    putchar('\n');
100    return (0);
101 }
```



Всем спасибо за внимание

1:1

