

ОСНОВЫ ПРОГРАММНОГО КОНСТРУИРОВАНИЯ

Лекция № 1
4 сентября 2017



НЕФОРМАЛЬНАЯ ЧАСТЬ

- Я – Владимир Владимирович Парфиненко.
 - Бакалавр физики (ФФ), магистр математики (ММФ).
 - Профессиональный программист (Excelsior).
 - Регулярно чему-то учу (ОПК, ЛШ ФМШ).
- Контакт: vladimir.parfinenko@gmail.com

ФОРМАЛЬНАЯ ЧАСТЬ

О формате и оценках

УСЛОВИЯ РАБОТЫ

- Лекции: алгоритмы, структуры данных, практики программирования.
- Семинары: язык С, применение теоретических знаний на практике.
 - Второй семинарист: Илья Сергеевич Иванов.
- Самостоятельная работа: неотъемлемая часть практики.
- Материалы на сайте ork.afti.ru: база задач и лекции.

ФОРМА ОЦЕНКИ

- Курс ОПК является обязательным.
- Работа в семестре: оценивается двумя семинаристами, как-то коррелирует с баллами на орк.afti.ru.
- Проект: оценивается по множеству критериев на зачете.
- Теория: два вопроса по теории на зачете.
 - Бонус теория: теоретические задачки на лекциях.

СОДЕРЖАТЕЛЬНАЯ ЧАСТЬ

Крайне быстрое установление контекста
для последующей работы

КРАТКО О ПРОГРАММИРОВАНИИ

- Есть тупой кусок кремния (1), годный к повторяющимся, рутинным операциям.
- Есть пользователь (2).
- Программист (3) *отбирает* у пользователя (2) рутину, объясняя куску кремния (1), как ее выполнять.
- В основе труда программиста (3) лежит модель системы.
- Хорошая программа = хорошая модель.

ЧТО ЕСЛИ...

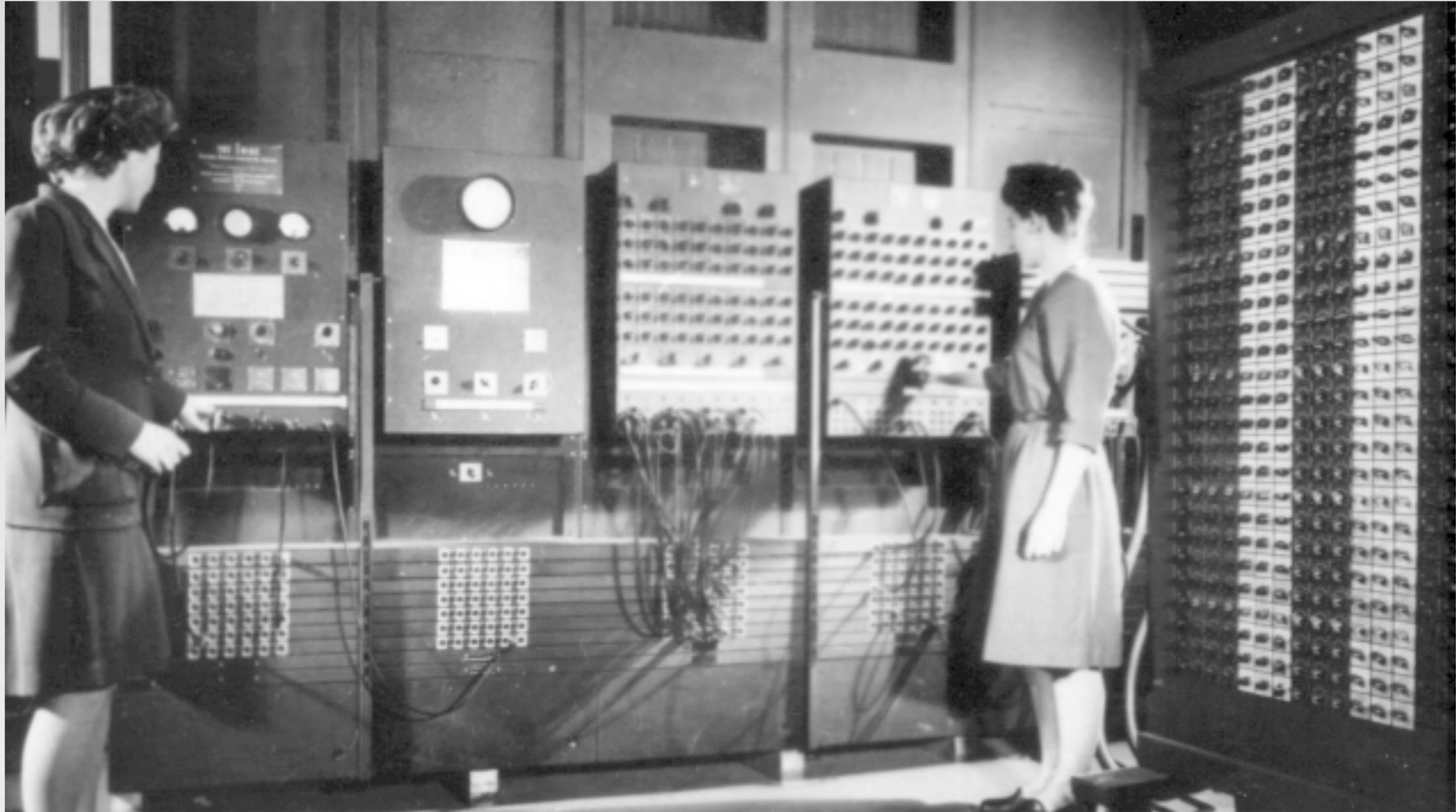
- Программист отобрал у пользователя не ту рутину...
- Программист плохо объяснил задачу тупому куску кремния...
- Программист понял пользователя неправильно...
- Модель у программиста в голове не соответствует действительности...
- Программист решил, что он умный и сам правильно знает, как пользователю жить...
- Пользователю никто не объяснил, как пользоваться программой...

...ТО

Пользователь всегда прав!



P. S. Пользователь есть всегда и у любой программы!



ИСТОРИЯ ПРОГРАММИРОВАНИЯ

Первые компьютеры (ENIAC, ...)



МАШИННЫЕ КОДЫ

0 и 1 хороши для кремния, но не для программистов

ПЕРВЫЙ ЯЗЫК ПРОГРАММИРОВАНИЯ

```
        .TITLE    HELLO WORLD
        .MCALL    .TTYOUT, .EXIT
HELLO:: MOV      #MSG, R1 ; STARTING ADDRESS OF STRING
1$:     MOVB     (R1)+, R0 ; FETCH NEXT CHARACTER
        BEQ     DONE      ; IF ZERO, EXIT LOOP
        .TTYOUT                ; OTHERWISE PRINT IT
        BR     1$           ; REPEAT LOOP
DONE:   .EXIT

MSG:    .ASCIZ  /Hello, world!/
        .END    HELLO
```

Язык низкого уровня — язык Ассемблера

БАЗОВАЯ ИДЕЯ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ

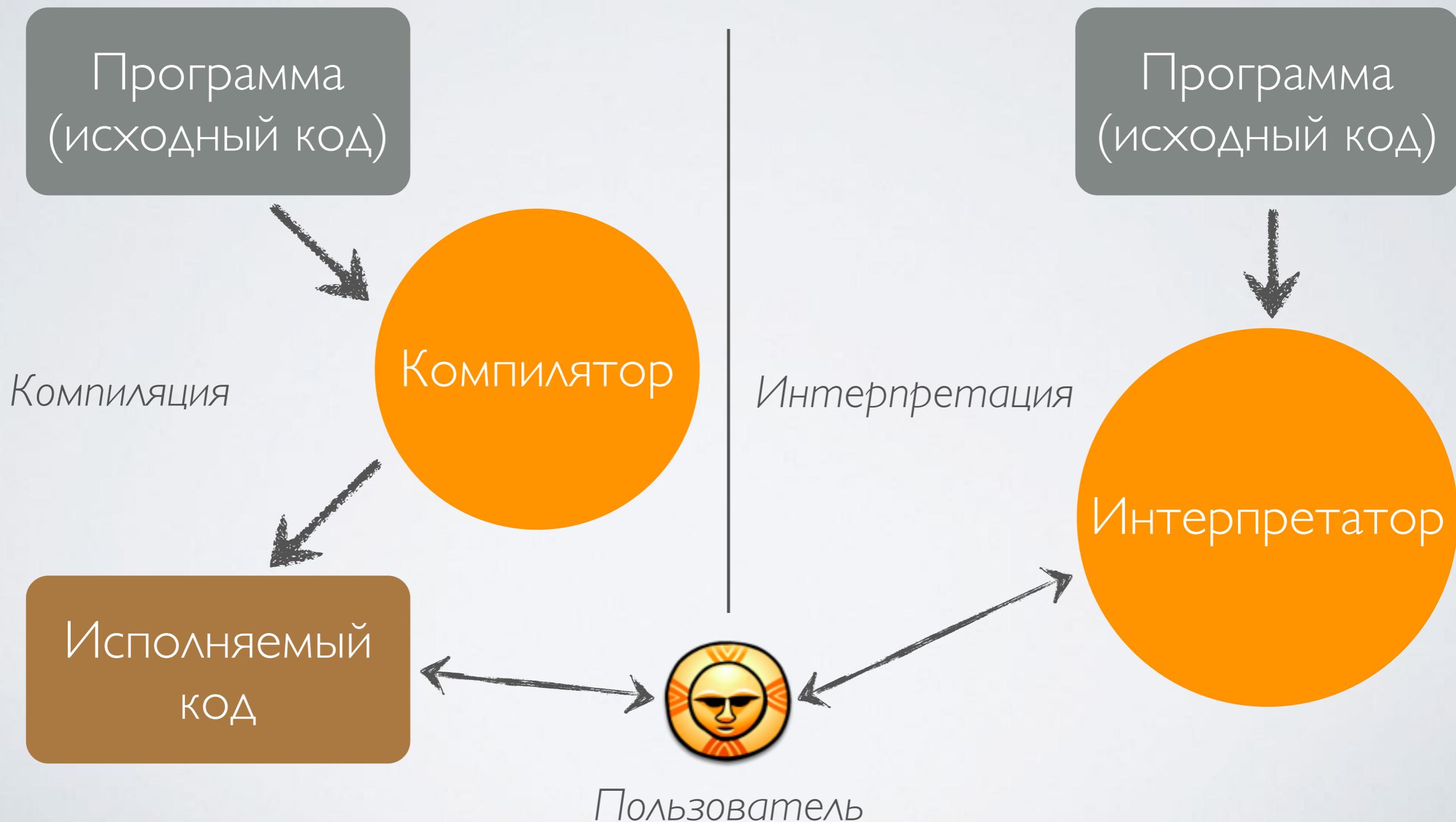


Машинные коды

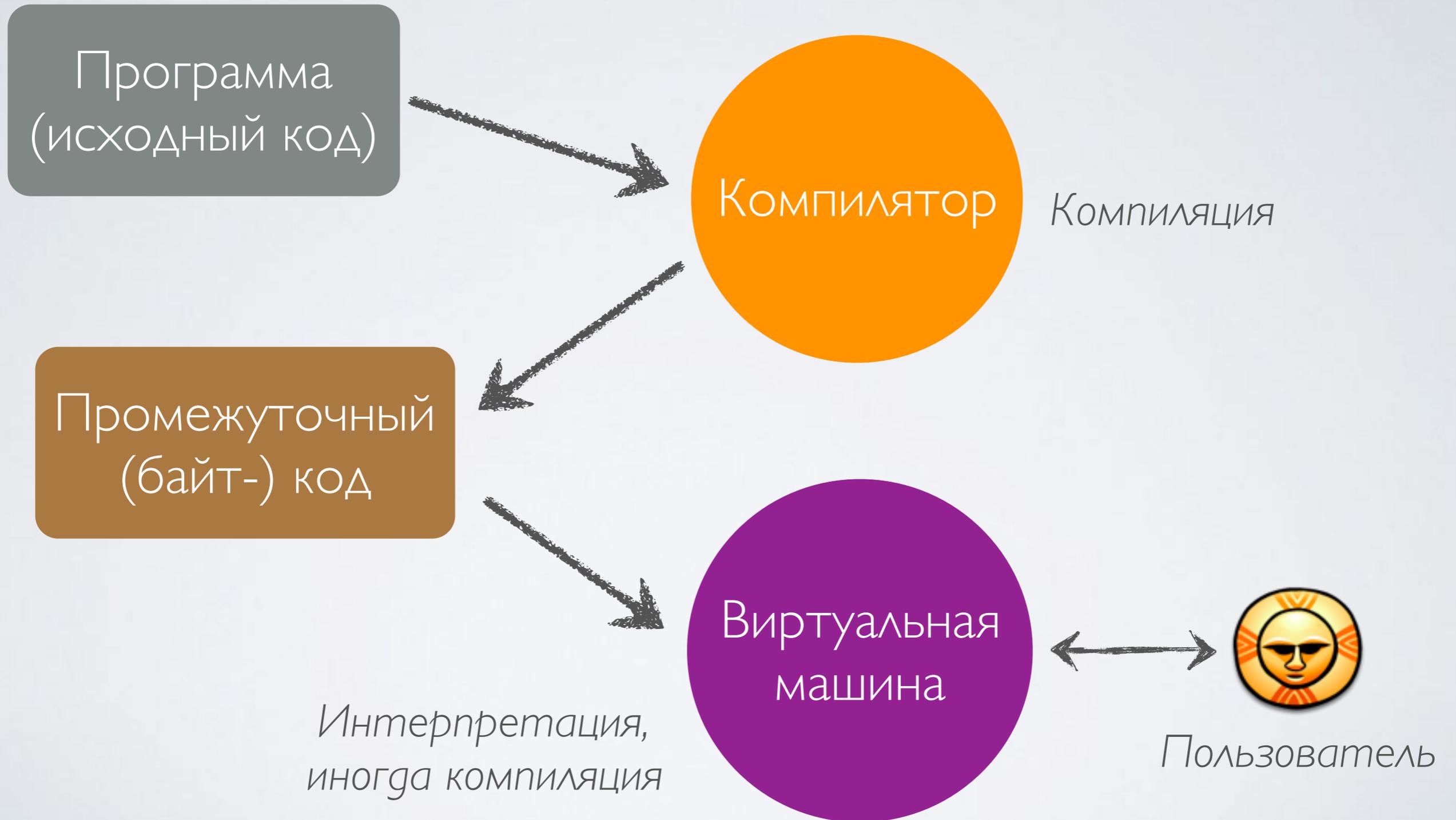


Использование ЯП

СПОСОБЫ ИСПОЛНЕНИЯ



ВИРТУАЛЬНЫЕ МАШИНЫ



КОМПИЛИРУЕМЫЕ ИЛИ ИНТЕРПРЕТИРУЕМЫЕ

**THE
C
PROGRAMMING
LANGUAGE**

Класс языка	процедурный
Тип исполнения	компилируемый
Появился в	1972
Автор	Деннис Ритчи, Кен Томпсон
Расширение файлов	<code>.c</code> - для файлов кода, <code>.h</code> - для заголовочных файлов
Выпуск	C11 (декабрь 2011)
Система типов	статическая слабая

**THE
C
PROGRAMMING
LANGUAGE**

The C Programming Language^[1] (often referred to as "K&R"), the seminal book on C

Paradigm	Imperative (procedural), structured
Designed by	Dennis Ritchie
Developer	Dennis Ritchie & Bell Labs (creators); ANSI X3J11 (ANSI C); ISO/IEC JTC1/SC22/WG14 (ISO C)
First appeared	1972; 45 years ago ^[2]
Stable release	C11 / December 2011; 5 years ago
Typing discipline	Static, weak, manifest, nominal

ПАРАДИГМЫ ПРОГРАММИРОВАНИЯ

- Императивная. Программа – это и есть алгоритм. В каждый момент есть глобальное состояние (содержимое памяти), явно доступное для изменения. *Языки: Pascal, C, ...*
- Функциональная. Программа – это функция. Нет явного состояния. *Языки: LISP, Haskell, Microsoft Excel (!).*
- Логическая. Программа – это набор предикатов и правил вывода. *Языки: Prolog, SQL.*

ИМПЕРАТИВНЫЙ С

```
int fact(int n) {  
    int result = 1;  
    for (int i = 2; i <= n; i++) {  
        result *= i;  
    }  
    return result;  
}
```

ФУНКЦИОНАЛЬНЫЙ HASKELL

```
fact 1 = 1  
fact n = n * fact (n - 1)
```

ФУНКЦИОНАЛЬНЫЙ EXCEL

	A	B
1	n	fact(n)
2	1	1
3	2	2
4	3	6
5	4	24
6	5	=B5*A6
7	6	720
8	7	5040
9	8	40320
10	9	362880
11	10	3628800

ЛОГИЧЕСКИЙ PROLOG

```
fact(1, 1):-!. .
```

```
fact(N, A):-  
    M = N - 1,  
    fact(M, B),  
    A = B * N.
```

ЯЗЫКИ ВЫСОКОГО УРОВНЯ

35 FORTRAN (1950-е гг.)

34 LISP (1950-е гг.)

??? Basic (1964 г.)

??? Pascal (1970 г.)

2 C (1972 г.)

3 C++ (1983 г.)

12 Delphi/Object Pascal (1986 г.)

9 Perl (1987 г.)

5 Python (1991 г.)

10 Ruby (1993 г.)

1 Java (1995 г.)

8 JavaScript (1995 г.)

7 PHP (1995 г.)

4 C# (2001 г.)

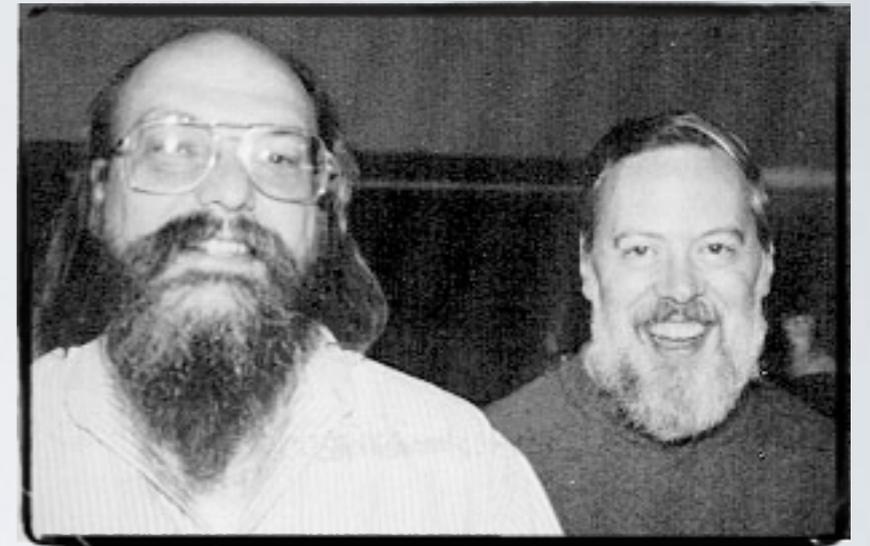
6 Visual Basic .NET (2001 г.)

28 Scala (2003 г.)

11 Swift (2014 г.)

...

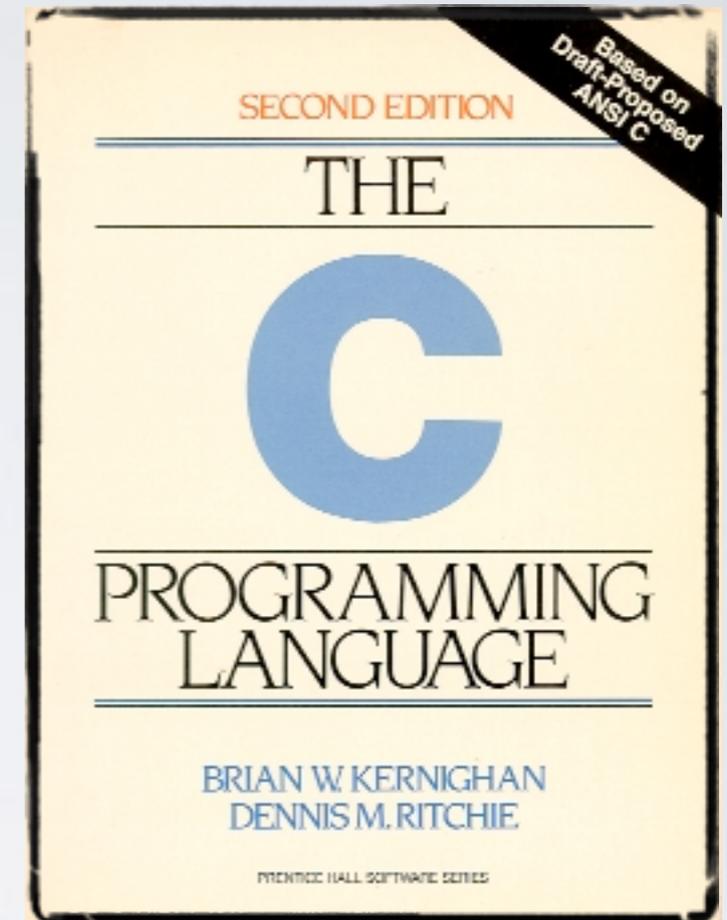
ЯЗЫК С



- 1972 г., Деннис Ричи (Bell Labs).
- «Переносимый ассемблер» для системных целей:
 - Операционные системы: ядра, системные библиотеки.
 - Компиляторы, виртуальные машины, ...
- Дает широкие возможности, но требует аккуратности.

ИСТОРИЯ

- 1972 г. – рождение.
- 1978 г. – выход книги, «K&R» C.
 - 1983 г. – отпочковался C++.
- 1989 г. – стандарт ANSI C или C89.
- 1999 г. – стандарт C99 (не совместим с C++!).
- 2011 г. – стандарт C11.



КОМПИЛЯТОРЫ

- GNU Compiler Collection.
- Microsoft Visual C++.
- Intel C++ Compiler.
- Clang + Low Level Virtual Machine.



HELLO_WORLD.C

```
#include <stdio.h>
```

```
int main(void) {  
    printf("Hello world!\n");  
    return 0;  
}
```



КОНЕЦ ПЕРВОЙ ЛЕКЦИИ