

Основы программного конструирования

ЛЕКЦИЯ №1

6 ФЕВРАЛЯ 2023



Знакомимся

- Борин Владислав Михайлович
- Лекции: Понедельник 14:30 316 (т.е. тут).
- Почта для связи: V.M.Borin@inp.nsk.su
- Для особо экстренных вариантов: +7-903-933-33-66

Как работаем?

- Лекции – Получаем новые знания, задаем вопросы.
- Лабы – Осваиваем полученные теоретические знания. (до конца апреля)
- Самостоятельная работа – Дома решаем задачи, развиваемся, на лабах разбираемся в сложностях и сдаем.
- Проект – все что вы захотите, но достаточного уровня сложности. (Май)

Основной ресурс: <http://opk.afti.ru/>



План на ближайшие год

- Первый семестр: алгоритмы, структуры данных, практики программирования.
- Базовый язык – Python
- Второй семестр (структурное программирование): примерно все тоже самое, но сложнее (будем вникать в суть того, как это устроено).
- Базовый язык – C.



Лабы

Аудитория 212

Гемузов Артем Сергеевич	22312
Иванова Александра Алексеевна	22305
Квинт Лукас Эрик	22313
Климов Богдан Алексеевич	22313
Макаров Данил Андреевич	22305
Попов Дмитрий Константинович	22313
Саликов Григорий Дмитриевич	22313
Салтымаков Степан Романович	22312
Сычкова Дарина Сергеевна	22313
Фурцев Матвей Геннадьевич	22305
Хрисанов Константин Владимирович	22312
Шатров Артемий Евгеньевич	22305
Шашкина Елизавета Анатольевна	22312

- Лабы: Четверг 12:40 (2 пары)
- 2 подгруппы (терминальные классы 212, 306)
- + Все те, кого нет в списке идут в 212.



Лабы

Аудитория 306

Антонов Максим Андреевич	22313
Астафьева Алиса Алексеевна	22305
Бороздин Павел Александрович	22312
Герлингер Евгений Алексеевич	22305
Дрижак Иван Васильевич	22313
Лотарев Артур Русланович	22305
Малиновский Владислав Евгеньевич	22312
Машков Михаил Андреевич	22312
Останин Фёдор Вадимович	22313
Паньшин Евгений Олегович	22305
Погуляев Михаил Игоревич	22305
Позднякова Мария Анатольевна	22312
Русанов Илья Константинович	22313
Труфанов Иван Сергеевич	22305
Шешенин Георгий Михайлович	22312

- Лабы: Четверг 12:40 (2 пары)
- 2 подгруппы (терминальные классы 212, 306)



Итог вашей работы – экзамен

предварительная шкала

- 0-50 баллов за семестр. Оценка семинариста не имеет строгой привязки к баллам на сайте!
- 0-50 баллов за теорию (Меньше 20 – экзамен не сдан)
- Проект: 0-100 баллов (Меньше 40 – экзамен не сдан)

Итоговая оценка выставляется по сумме набранных баллов:

- ✓ 170...200 – отлично.
- ✓ 140...169 – хорошо.
- ✓ 110...139 – удовлетворительно.
- ✓ меньше 110 – экзамен не сдан ☹



Баллы за проект

Итоговый балл за проект складывается из 5 критериев, каждый в диапазоне 0...20 баллов.

- Корректность работы (падает ли программа, как часто? Корректно ли работает?).
- Функциональность (Реализация всего чего обещали, могут попросить чего-то доделать, если кажется что слишком просто).
- UI и документированность (Легко ли этим пользоваться без ваших пояснений?)
- Качество кода (Слишком много всего).
- Общая оценка программы (субъективная оценка проекта, без каких-либо критериев).

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.



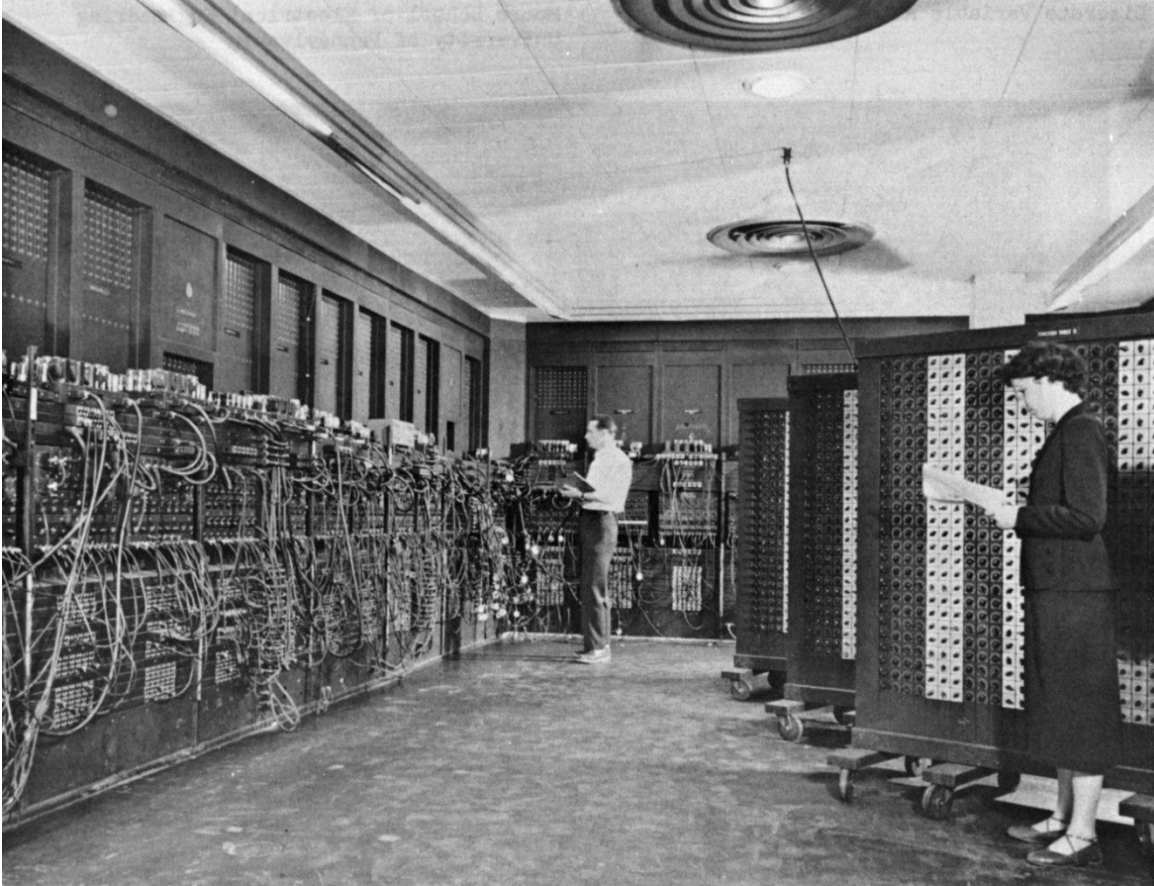


Что если...

- 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 самый умный и сам правильно знает, как пользователю жить...
- Пользователю никто не объяснил, как пользоваться программой...



Первые компьютеры



Electronic Numerical Integrator and Computer (ENIAC)

30 тонн массы.

Потреблял 174 кВт.

Количество электронных ламп – 17468

Вычислительная мощность – 357
умножений / 5000 сложений в секунду

Тактовая частота 100 кГц



Машинные коды

Очень сложно и очень ресурсоемко.

Машинный код "Минск-22"

Код передн. ест. тм	Значение кода	Печат на узкой ленте
.	0	0
.	1	1
.	2	2
.	3	3
.	4	4
.	5	5
.	6	6
.	7	7
.	8	8
.	9	9
.	+ восьмёрка.	+
.	- восьмёрка.	-





Языки низкого уровня

Проще и удобнее!

```
        .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
```



Базовая идея языков программирования



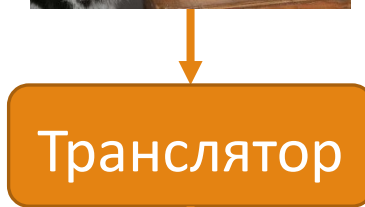
011001
110101



Машинные коды



```
program cool  
begin  
    ...  
end
```



Транслятор

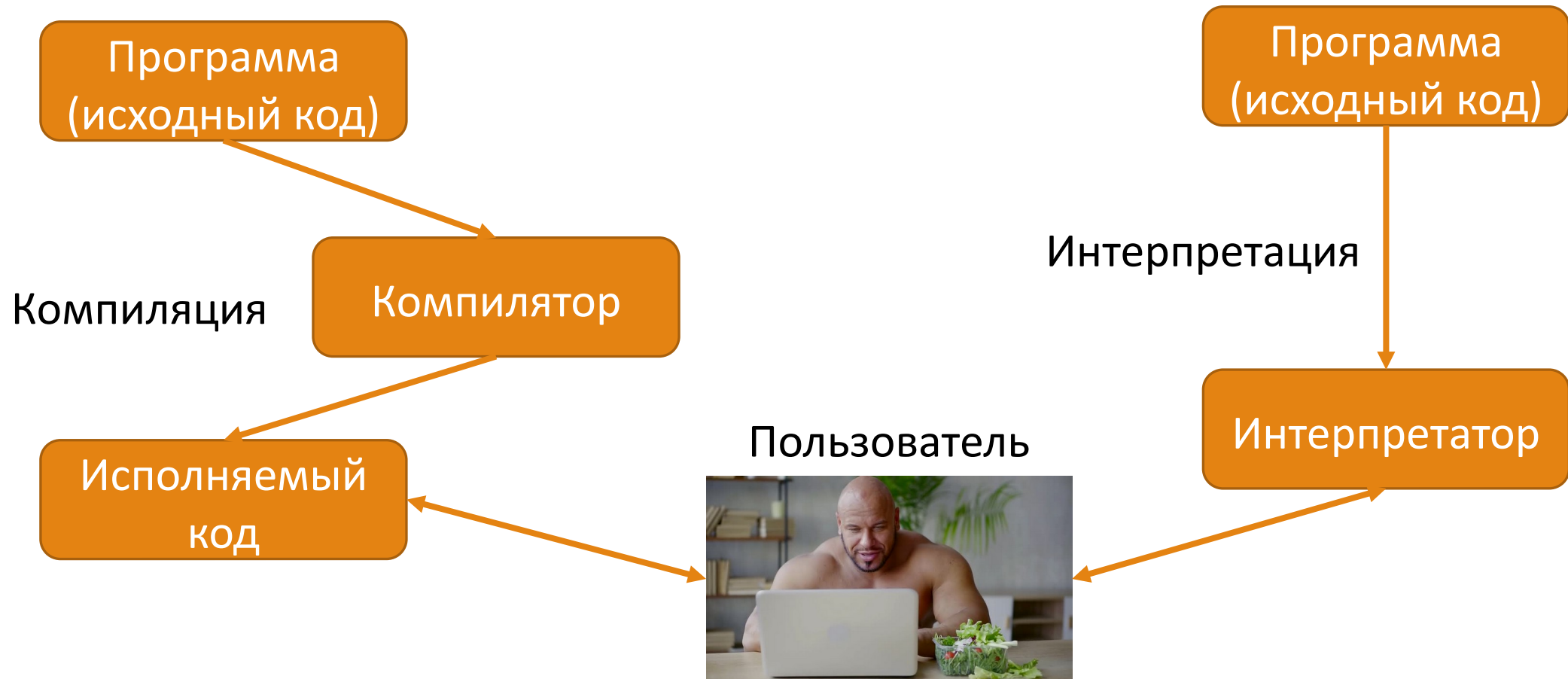


011001
110101

Использование ЯП

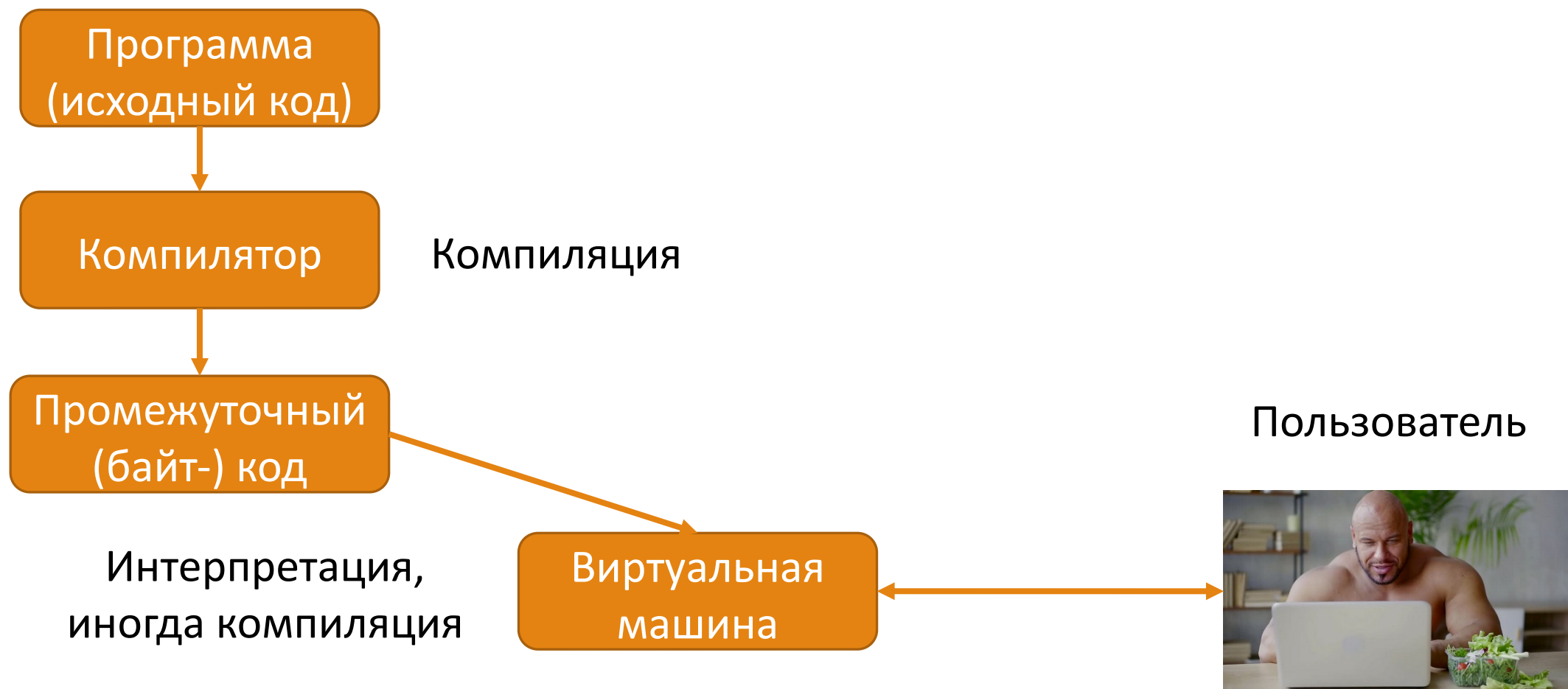


Способы исполнения





Виртуальные машины



Парадигмы программирования

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



Императивный C

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



Функциональный EXCEL и HASKELL

n	fact(n)
1	1
2	2
3	6
4	24
5	120
6	=B9*C8
7	5040
8	40320
9	362880
10	3628800

```
fac :: Integer -> Integer
fac 0 = 1
fac n | n > 0 = n * fac (n - 1)
```



Логический PROLOG

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

`fact(N, A):-`

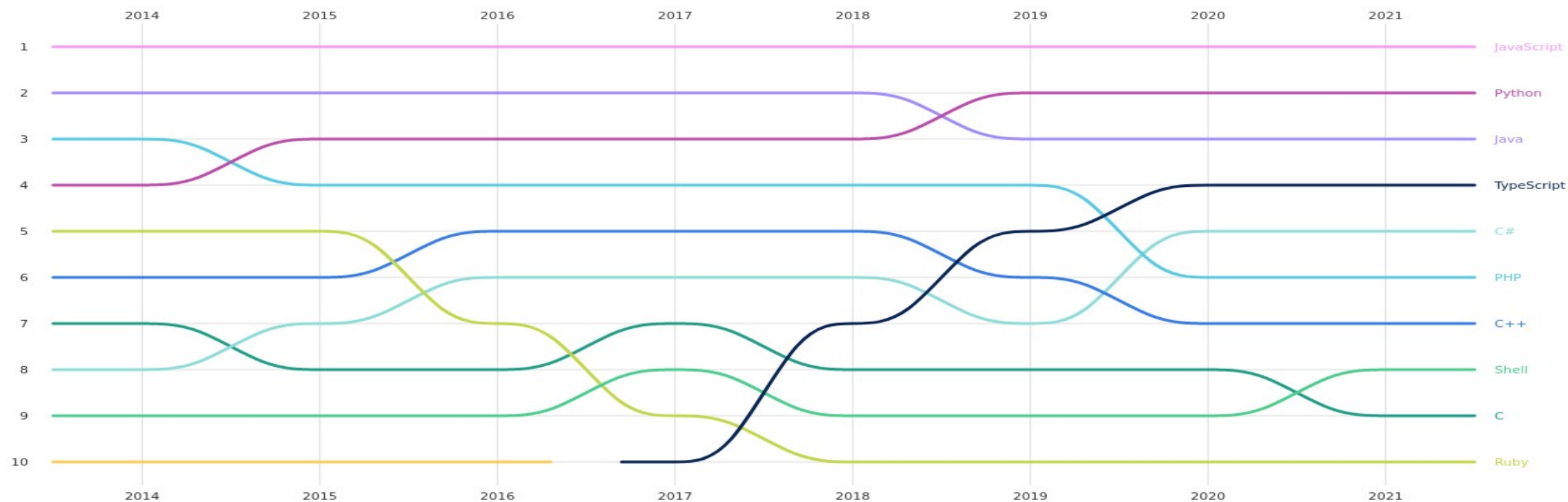
$M = N - 1,$

`fact(M, B),`

$A = B * N.$

Языки высокого уровня

Top languages over the years



Язык PYTHON

1991 г. Гвидо Ван Россум.

Упор на читаемость и простоту кода

**PYTHON ОТЛИЧНЫЙ
ЯЗЫК ДЛЯ ИЗУЧЕНИЯ
ПРОГРАММИРОВАНИЯ.
НЕТ ОПЕРАТОРНЫХ
СКОБОК, НЕТ
ФУНКЦИИ MAIN.**



**В ПРОГРАММЕ МОГУТ
БЫТЬ ОШИБКИ ИЗ-ЗА
ЛИШНЕГО ПРОБЕЛА.**



HELLO.C

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

```
int main()  
{  
    printf("Hello world!");  
}
```



HELLO.PY

```
print("Hello world!")
```




Тенденции

- Языки программирования становятся все мощнее за счет усложнения абстракций.
- Тупые куски кремния не становятся умнее, но усложняются, поэтому «объяснять» им все сложнее.
- Трансляторы становятся все сложнее.
- Ожидания пользователей растут.
- Новые технологии появляются ежедневно – программист обязан постоянно учиться
- Есть базовые вещи (алгоритмы, структуры данных, ...)

На сегодня все

**ЭТОТ БАЛКОН ВЫГЛЯДИТ, КАК ЧАСТЬ
МОЕГО КОДА**

1001MEM.RU

**Я НЕ ЗНАЮ, ЗАЧЕМ ЭТО НУЖНО,
НО УДАЛЯТЬ КАК-ТО СРЕМНО**