История программирования

Концепции, языки, тенденции

Марк Шевченко

Московский Клуб Программистов

https://markshevchenko.pro

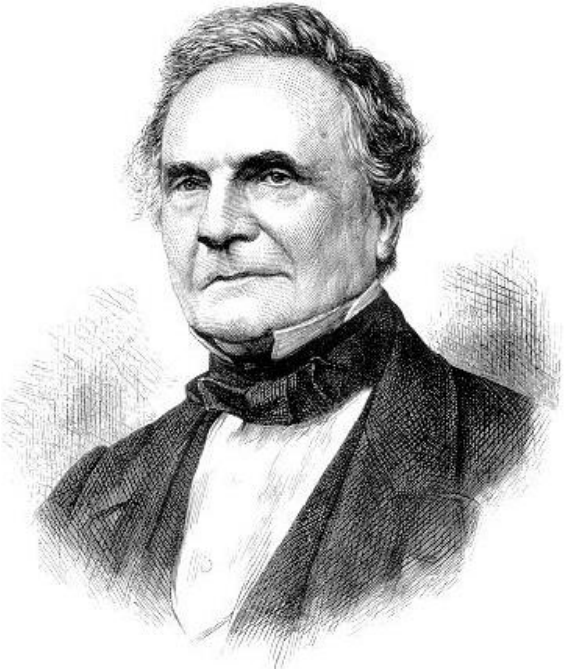
https://prog.msk.ru


Side Effect

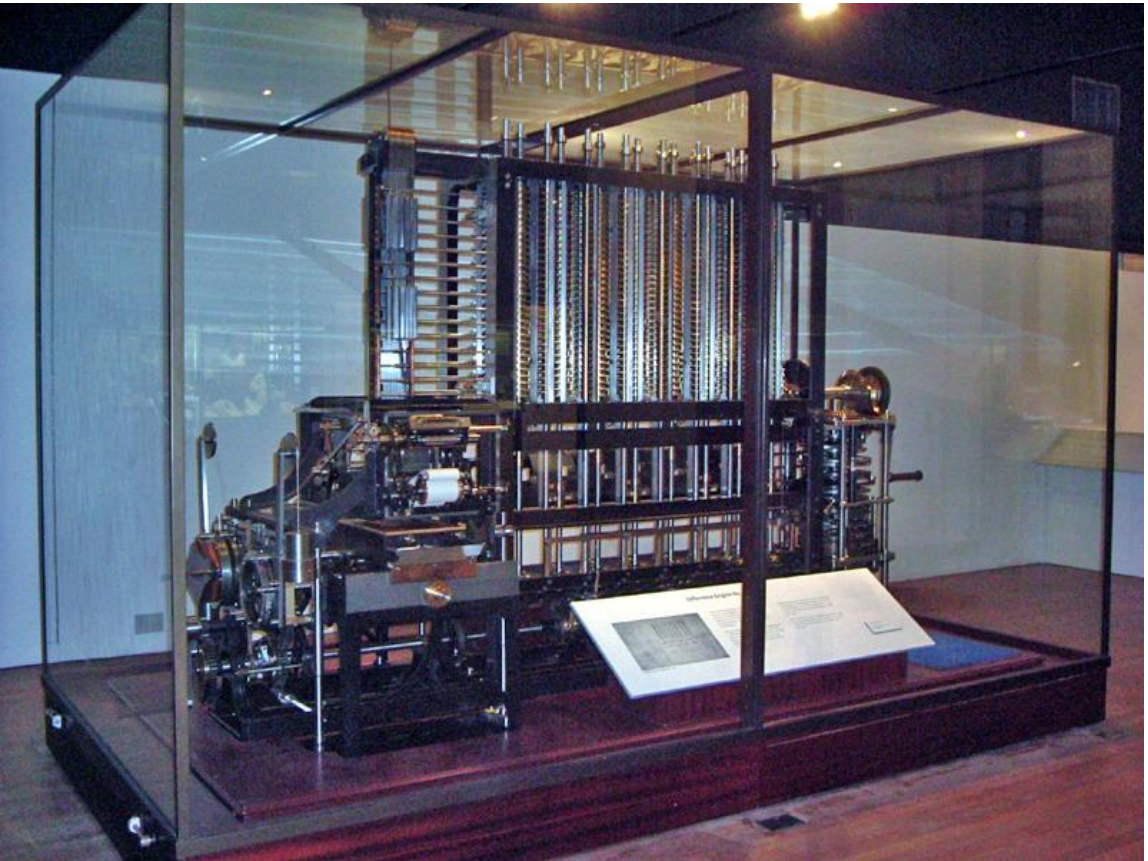
Машина Чарльза Бэббиджа

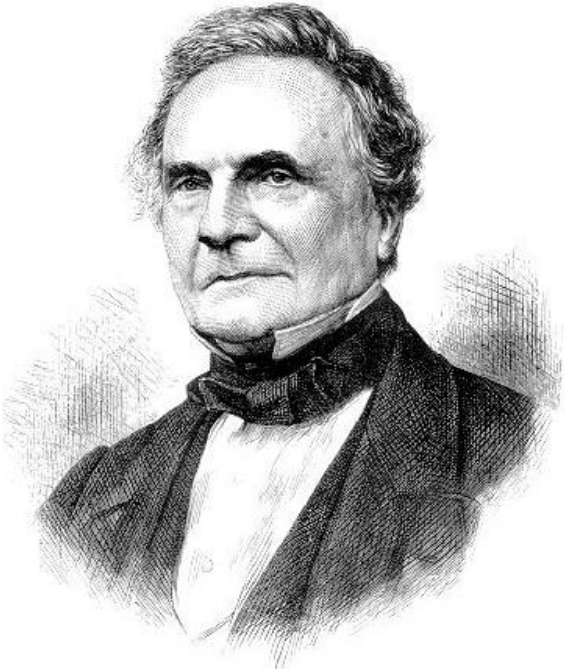
Машина Чарльза Бэббиджа

Машина Чарльза Бэббиджа

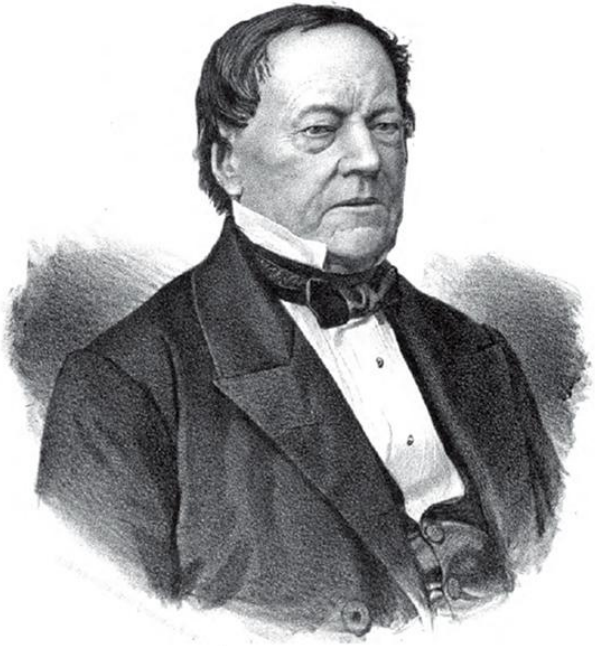


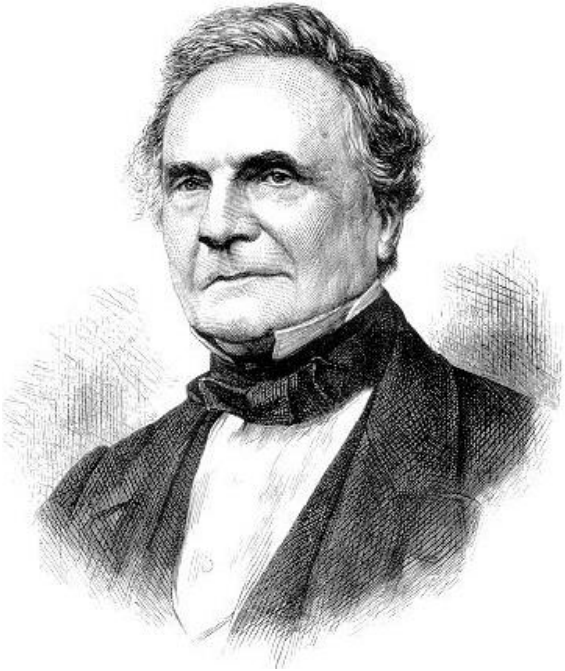
Машина Чарльза Бэббиджа





Машина Чарльза Бэббиджа





Метод конечных разностей

Машина Чарльза Бэббиджа

x	$p(x)=2x^2-3x+2$	$\Delta_1(x)=p(x+1)-p(x)$	$\Delta_2(x)=\Delta_1(x+1)-\Delta_1(x)$
0	2	-1	4
1	1	3	
2	4		
3			

Машина Чарльза Бэббиджа

x	$p(x)=2x^2-3x+2$	$\Delta_1(x)=p(x+1)-p(x)$	$\Delta_2(x)=\Delta_1(x+1)-\Delta_1(x)$
0	2	-1	4
1	1	3	4
2	4		
3			

Машина Чарльза Бэббиджа

x	$p(x)=2x^2-3x+2$	$\Delta_1(x)=p(x+1)-p(x)$	$\Delta_2(x)=\Delta_1(x+1)-\Delta_1(x)$
0	2	-1	4
1	1	3	4
2	4	3+4	
3			

Машина Чарльза Бэббиджа

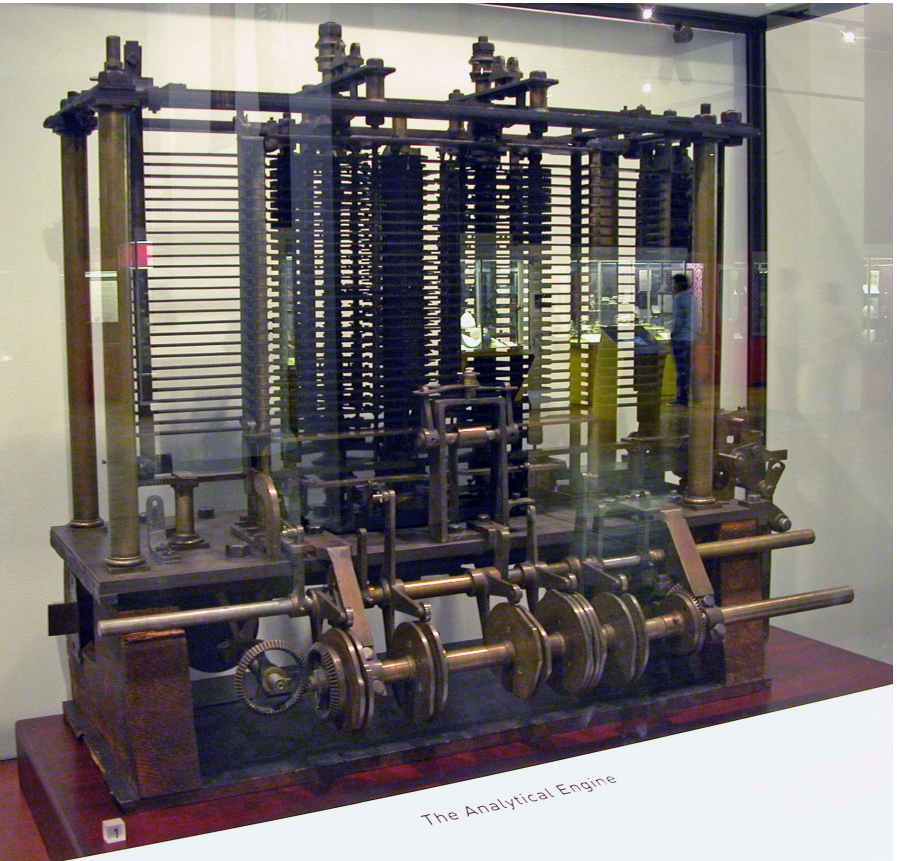
x	$p(x)=2x^2-3x+2$	$\Delta_1(x)=p(x+1)-p(x)$	$\Delta_2(x)=\Delta_1(x+1)-\Delta_1(x)$
0	2	-1	4
1	1	3	4
2	4	7	
3	4+7		

Машина Чарльза Бэббиджа

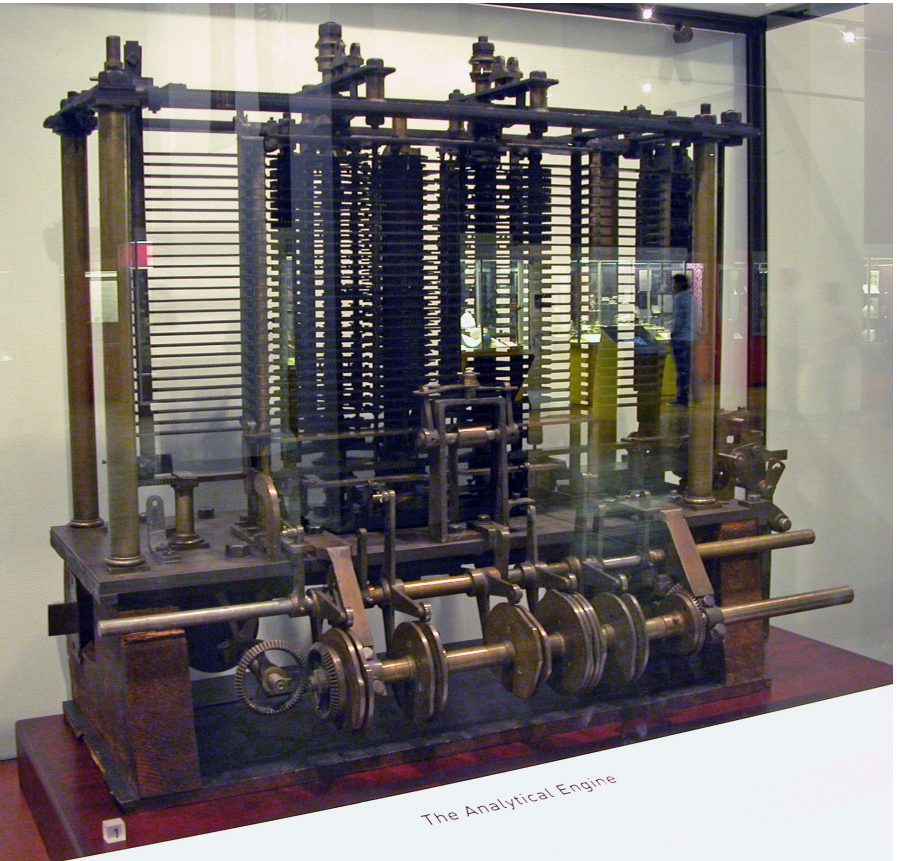
x	$p(x)=2x^2-3x+2$	$\Delta_1(x)=p(x+1)-p(x)$	$\Delta_2(x)=\Delta_1(x+1)-\Delta_1(x)$
0	2	-1	4
1	1	3	4
2	4	7	
3	11		

Аналитическая машина

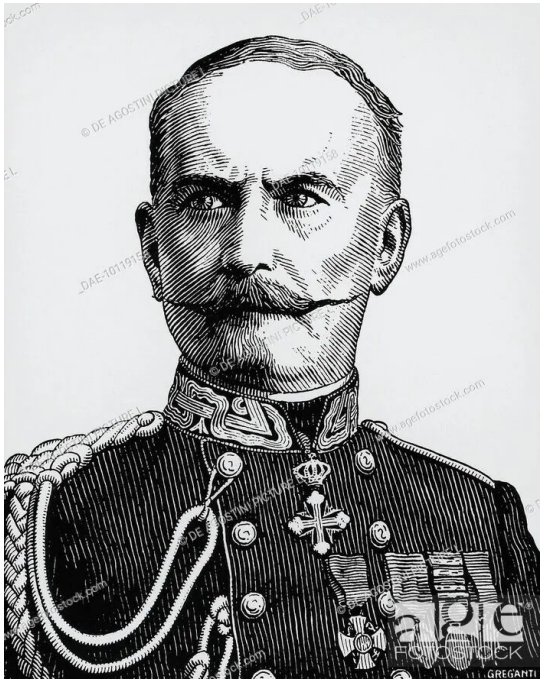
Машина Чарльза Бэббиджа



Машина Чарльза Бэббиджа



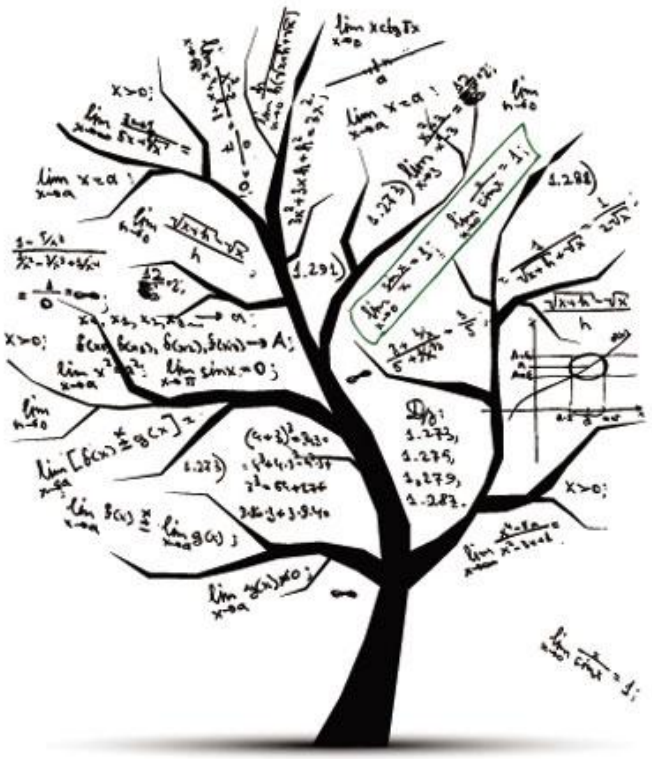
Машина Чарльза Бэббиджа



Машина Чарльза Бэббиджа

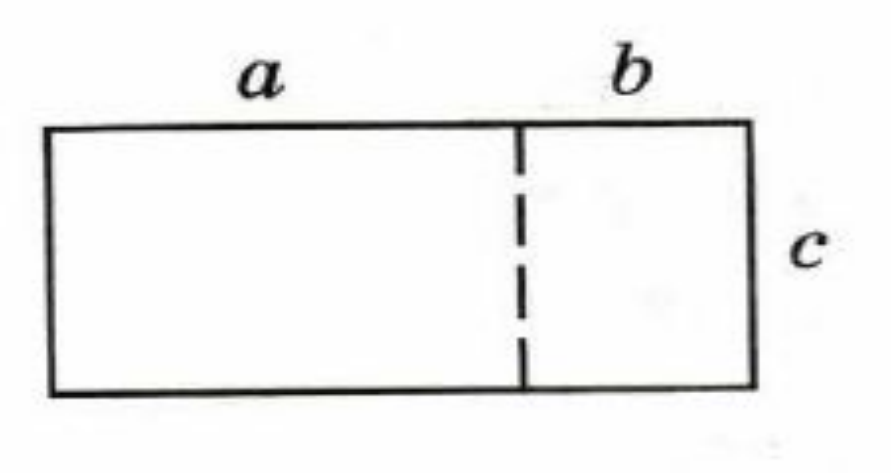
Кризис оснований математики

История математики.

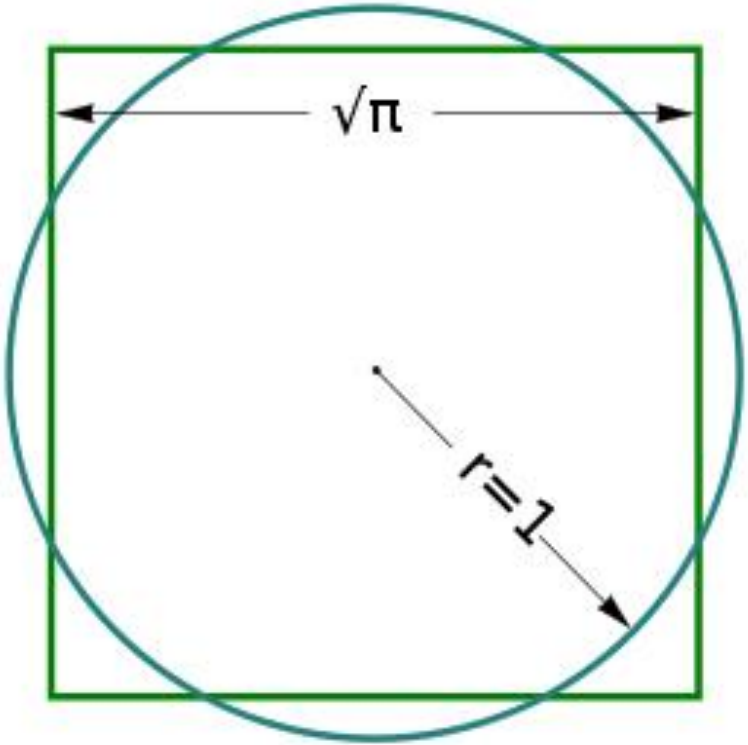


Дистрибутивный закон

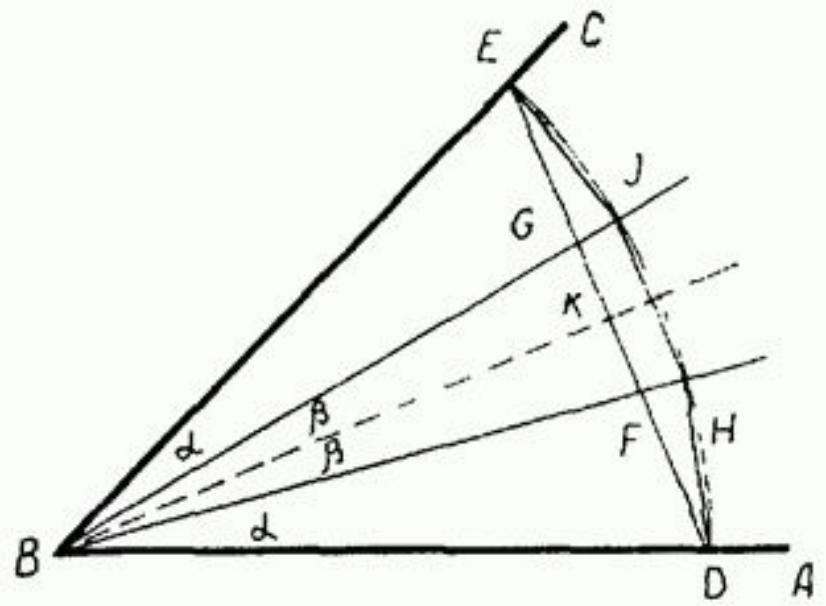
$$(a + b) \times c = a \times c + b \times c$$



Квадратура круга



Трисекция угла



Существует ли «общий язык математики»?

Строгость

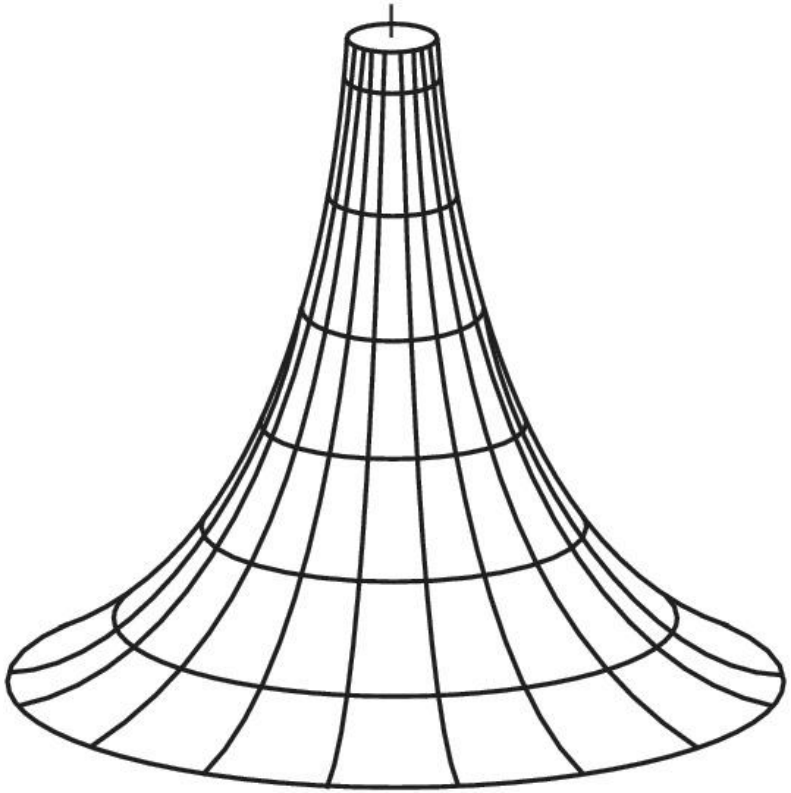
Сумма любого сходящегося ряда непрерывных функций непрерывна.

Строгость

Сумма любого сходящегося ряда непрерывных функций непрерывна.

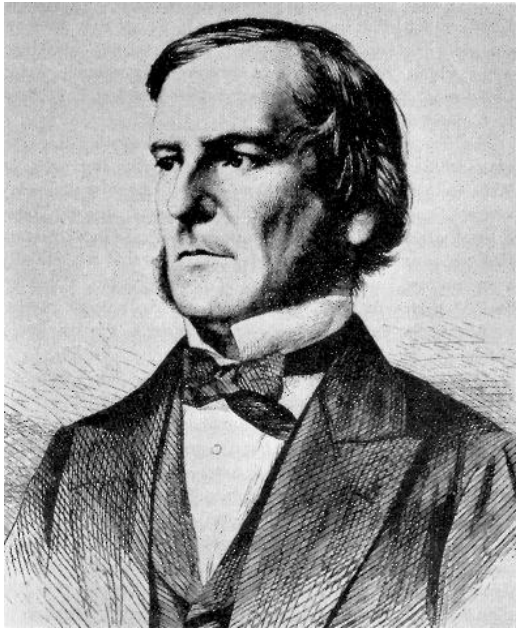
$$f(x) = \sin x - \frac{1}{2}\sin 2x + \frac{1}{3}\sin 3x - \frac{1}{4}\sin 4x \cdots$$

Формализация

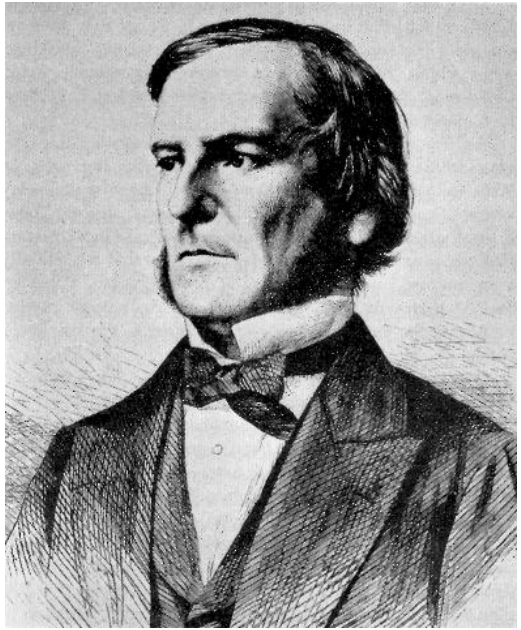


Систематизация

1) $x \vee y = y \vee x, \ x \wedge y = y \wedge x;$
2) $x \vee (y \vee z) = (x \vee y) \vee z,$
   $x \wedge (y \wedge z) = (x \wedge y) \wedge z;$
3) $(x \wedge y) \vee y = y, \ (x \vee y) \wedge y = y;$
4) $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z),$
   $x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z);$
5) $(x \wedge \bar{x}) \vee y = y, \ (x \vee \bar{x}) \wedge y = y.$



Систематизация

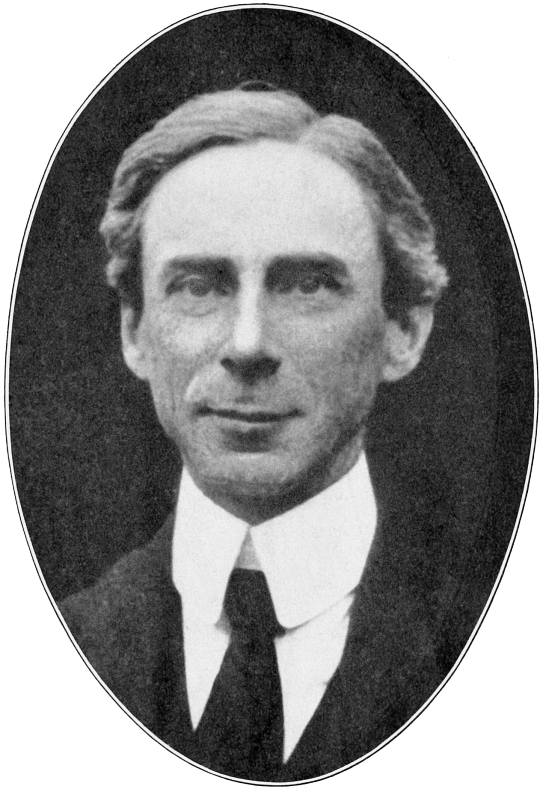


Систематизация

(1) $0 \in N$;
(2) $x \in N \longrightarrow Sx \in N$;
(3) $x \in N \longrightarrow Sx \neq 0$;
(4) $x \in N \wedge y \in N \wedge Sx = Sy \longrightarrow x = y$;
(5) $0 \in M \wedge \forall x \ (x \in M \longrightarrow Sx \in M) \longrightarrow N \subseteq M$

Актуальная бесконечность

Парадокс Рассела



Предположим, у нас есть множество всех множеств, которые не являются собственными элементами. Является ли это множество собственным элементом?


ЮГО-СЕВЕРНАЯ

Формализм

- Полнота
- Независимость
- Непротиворечивость
- Разрешимость

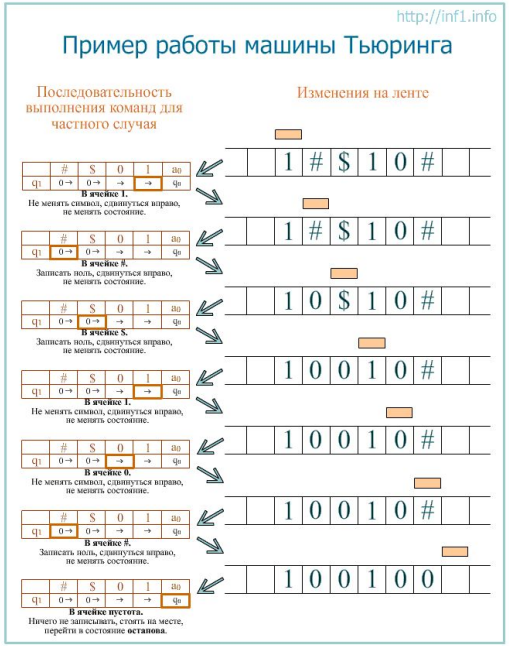
Теоремы о неполноте и неразрешимости

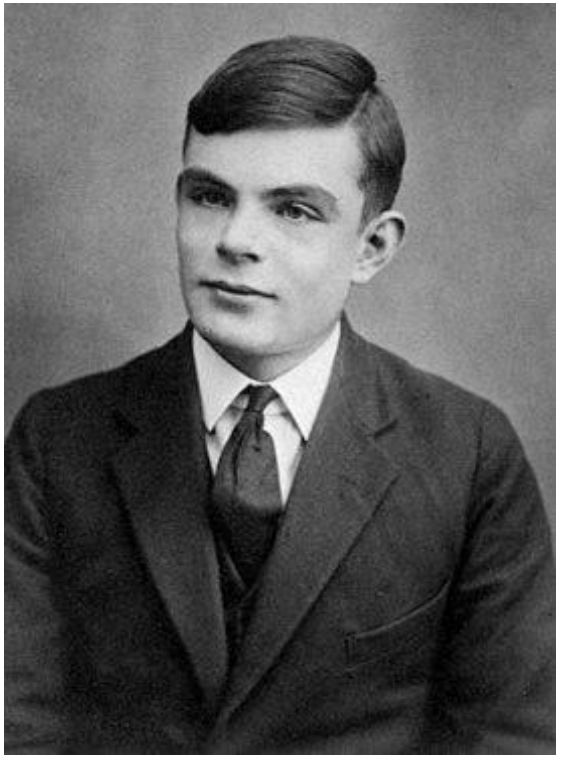
- В арифметике есть истинные недоказуемые утверждения.
- Одно из недоказуемых утверждений — непротиворечивость арифметики.

Теоремы о неполноте и неразрешимости

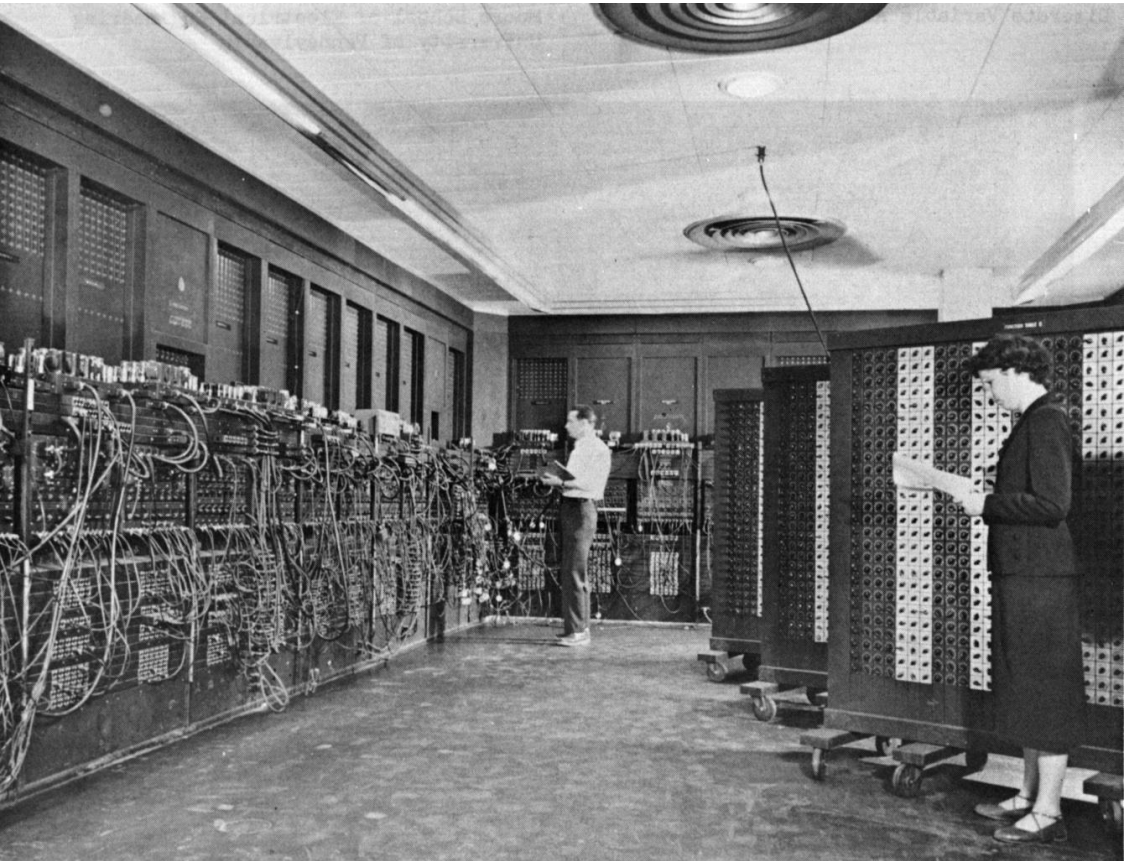
$$
\begin{aligned}
m+n &= \lambda f\ x.m\ f\ (n\ f\ x) = \\
&= \lambda f\ x.(\lambda f\ x.f^m\ x)\ f\ (n\ f\ x) = \\
&= \lambda f\ x.(\lambda x.f^m\ x)\ (n\ f\ x) = \\
&= \lambda f\ x.f^m\ (n\ f\ x) = \\
&= \lambda f\ x.f^m\ ((\lambda f\ x.f^n\ x)\ f\ x) = \\
&= \lambda f\ x.f^m\ ((\lambda x.f^n\ x)\ x) = \\
&= \lambda f\ x.f^m\ (f^n\ x) = \\
&= \lambda f\ x.f^{m+n}\ x
\end{aligned}
$$

Теоремы о неполноте и неразрешимости





Компьютеры

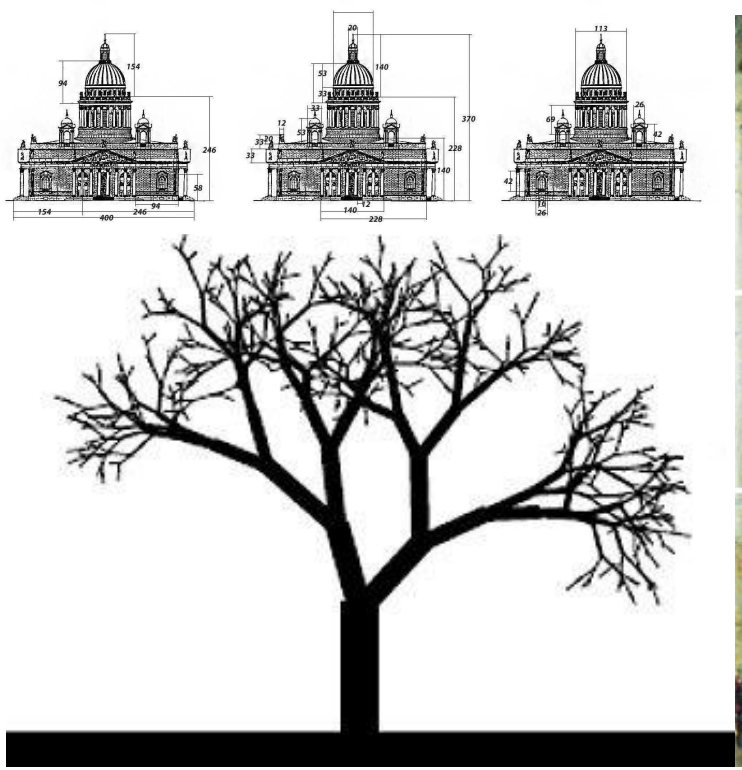


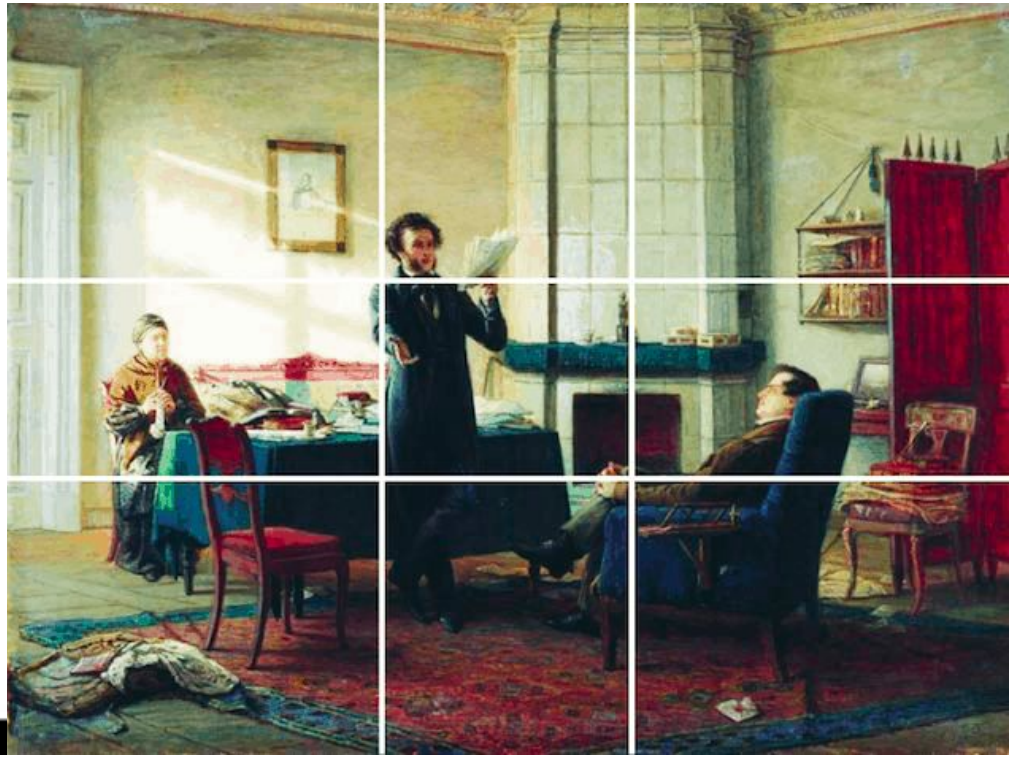
- Десятичная система
- Коммутация

Компьютеры

- Однородность памяти
- Адресность
- Программное управление

Золотое сечение





Числа Фибоначчи

0, 1, 1, 2, 3, 5, 8, 13, 21, 34,
55, 89, 144, 233, 377, 610, 987,
1597, 2584, 4181, …

Язык Ассемблера

```
@format = private constant [3 x i8] c"%d\0A"
declare i32 @printf(i8*, ...)

define i32 @main() {
entry:
  %a = alloca i32, align 4
  store i32 0, i32* %a
  %b = alloca i32, align 4
  store i32 1, i32* %b

  %i = alloca i32, align 4
  store i32 0, i32* %i

  br label %l1
l1:
```

Язык Ассемблера

```
  %0 = load i32* %a, align 4
  %1 = call i32 (i8*, ...)* @printf(i8* getelementptr inbounds
            ([3 x i8]* @format, i32 0, i32 0), i32 %0)
  %2 = load i32* %b, align 4
  %3 = add i32 %0, %2
  store i32 %3, i32* %b
  store i32 %2, i32* %a

  %4 = load i32* %i, align 4
  %5 = add i32 %4, 1
  store i32 %5, i32* %i

  %6 = icmp eq i32 %5, 20
  br i1 %6, label %l2, label %l1

l2:
  ret i32 0
}
```

Fortran

```
program fibonacci
    integer a, b, t
    a = 0
    b = 1
    i = 1

10  print *, a
    t = b
    b = a + b
    a = t

    i = i + 1
    if (i .le. 20) goto 10
end program fibonacci
```

LISP

```
(define (fibonacci n)
  (define (build-fibonacci m a b l)
    (cond
      ((= m 0) l)
      (else (build-fibonacci (- m 1) b (+ a b) (cons a l)))))
  (build-fibonacci n 0 1 '()))

(print (fibonacci 20))
```

```
(4181 2584 1597 987 610 377 233 144 89 55 34 21 13 8 5 3 2 1 1 0)
```

Pascal

```pascal
program Fibonacci;

var
  A, B, T: Integer;
  I: Integer;

begin
  A := 0;
  B := 1;

  for I := 1 to 20 do
  begin
    WriteLn(A);
    T := B;
    B := A + B;
    A := T;
  end;
end.
```

(

```
program Fibonacci;

var
  A, B, T: Integer;
  I: Integer;

begin
  A := 0;
  B := 1;

  for I := 1 to 20 do
  begin
    WriteLn(A);
    T := B;
    B := A + B;
    A := T;
  end;
end.
```

```
#include <stdio.h>

void main()
{
    int a = 0, b = 1, t;
    for (int i = 0; i < 20; i++)
    {
        printf("%d\n", a);

        t = b;
        b = a + b;
        a = t;
    }
}
```

Python

```python
import itertools

def fibonacci():
    a, b = 0, 1
    while True:
        yield a
        a, b = b, a + b

for f in itertools.islice(fibonacci(), 20):
    print(f)
```

Python

```
import itertools

def fibonacci():
    a, b = 0, 1
    while True:
        yield a
        a, b = b, a + b

for f in itertools.islice(fibonacci(), 20):
    print(f)
```

```
a, b = 0, 1
for i in range(20):
    print(a)
    a, b = b, a + b
```

Prolog

```
fibonacci(1, [0]).
fibonacci(2, [1,0]).
fibonacci(N, [R,A,B|Cs]) :-
    N > 2,
    N1 is N - 1,
    fibonacci(N1,[A,B|Cs]),
    R is A + B.
```

```
?- fibonacci(20, X).
X = [4181, 2584, 1597, 987, 610, 377, 233, 144, 89|...] ;
false.
```

ML

```
let rec fibonacci n =
  match n with
  | 1 -> [0]
  | 2 -> [1; 0]
  | _ -> match fibonacci (n - 1) with
         | a::b::cs -> (a + b)::a::b::cs
         | _        -> []

printfn "%A" (fibonacci 20)
```

Полиморфизм

```
type
  IntegerBinaryNode = record
    Value: Integer;
    Left: ^IntegerBinaryNode;
    Right: ^IntegerBinaryNode;
  end;

  StringBinaryNode = record
    Value: String;
    Left: ^StringBinaryNode;
    Right: ^StringBinaryNode;
  end;
```

Полиморфизм

```
type 'a binary_tree = Leaf
                    | Node of 'a * 'a binary_tree * 'a binary_tree


let rec contains x t = match t with
                     | Leaf -> false
                     | Node (y, left, right) -> x = y
                                             || contains x left
                                             || contains x right
```

Полиморфизм

```
public class BinaryNode<T>
{
    public T Value { get; set; }

    public BinaryNode<T> Left { get; set; }

    public BinaryNode<T> Right { get; set; }

    public bool Contains(T x)
    {
        return x == Value
            || (Left != null && Left.Contains(x))
            || (Right != null && Right.Contains(x));
    }
}
```

SQL

```sql
WITH Fibonacci AS
(
  SELECT 1 AS Number, 0 AS A, 1 AS B
   UNION ALL
  SELECT Number + 1 AS Number, B AS A, (A + B) AS B
    FROM Fibonacci
   WHERE Number < 20
)
SELECT Number, A FROM Fibonacci
```

SmallTalk

```
a := 0.
b := 1.
20 timesRepeat: [
   a displayNl.
   t := b.
   b := a + b.
   a := t.
].
```

Java

```java
public class Fibonacci {
  public static void main(String []args) {
    int[] fibonacci = new int[20];
    fibonacci[0] = 0;
    fibonacci[1] = 1;

    for (int i = 2; i < fibonacci.length; i++)
      fibonacci[i] = fibonacci[i - 1] + fibonacci[i - 2];

    System.out.println(java.util.Arrays.toString(fibonacci));
  }
}
```

Erlang

```erlang
-module(main).
-export([start/0, fibonacci/3]).

fibonacci(1, A, _) ->
    io:format("~B~n", [A]);

fibonacci(N, A, B) ->
    spawn(?MODULE, fibonacci, [N - 1, B, A + B]),
    io:format("~B~n", [A]).

start() ->
    spawn(?MODULE, fibonacci, [20, 0, 1]).
```

Haskell

```haskell
module Main where

fibonaccies = 0 : 1 : zipWith (+) fibonaccies (tail fibonaccies)

main = print $ take 20 fibonaccies
```

Лаборатория

Тенденции

- Основные ошибки

Тенденции

- Основные ошибки
- Производительность программиста

Тенденции

- Основные ошибки
- Производительность программиста
- Производительность программ

Тенденции

- Основные ошибки
- Производительность программиста
- Производительность программ
- Теория

Тенденции

- Основные ошибки
- Производительность программиста
- Производительность программ
- Теория
- Разнообразие языков

Тенденции

- Основные ошибки
- Производительность программиста
- Производительность программ
- Теория
- Разнообразие языков
- Разнообразие парадигм

Ссылки

- Что на самом деле делала программа Ады Лавлейс? https://habr.com/ru/post/422169/
- Клайн, Морис. Утрата определённости.
- Петцольд, Чарльз. Читаем Тьюринга.
- Бритфус, Сергей. Кризис оснований математики. http://opentextnn.ru/old/man/index.html@id=873
- Бэкус, Дж. Тьюринговская лекция (перевод на русский).
- http://rkka21.ru/docs/turing-award/jb1977r.pdf
- Online Compiler:
- https://kripken.github.io/llvm.js/demo.html
- https://www.tutorialspoint.com/compile_fortran_online.php
- https://repl.it
- Hindley R. Principal Type Scheme.
- http://www.users.waitrose.com/~hindley/SomePapers_PDFs/1969PrincTypScm,B.pdf
- Пирс, Бенджамин. Типы в языках программирования.

Итого

- Машина Чарльза Бэббиджа
- Кризис оснований математики
- Архитектура фон Неймана
- Примеры
- Тенденции
- Ссылки

--

Марк Шевченко,
@markshevchenko,
https://markshevchenko.pro