

# Обратная польская нотация (RPN)

Паначёв Максим Александрович  
/ старший преподаватель кафедры ВМиКН /  
15 апреля 2020 г.

# Машина фон Неймана

- Реальные вычислители, построенные согласно архитектурным принципам фон Неймана, следуют дополнительному соглашению: команды и их аргументы записываются в строго определённом порядке:

Command Code	Argument 1	Argument 2	...	Argument N
--------------	------------	------------	-----	------------

# Высокоуровневые языки

- Высокоуровневый язык программирования — язык программирования, разработанный для быстроты и удобства использования.
- Конструкции высокоуровневых языков, как правило, более приближены к конструкциям естественного языка современной математики.

# Высокоуровневые языки

- Одна из задач, которую приходится решать разработчикам языков высокого уровня – трансляция (преобразование) математических формул (выражений) в последовательность низкоуровневых команд – инструкций процессора.

$$y = A + B * \sin x^2$$

**MUL x, x** Умножить x на x

**SIN y, x** Вычислить  $\sin(x)$ , результат положить  
в y

**MUL y, B** Умножить y на B

**ADD y, A** Прибавить A к y

# Математические выражения

- Каждое математическое выражение можно представить в трех специфических видах записи:
  - Инфиксная нотация — это форма математических записей, которую использует большинство людей: левый операнд, знак действия, правый операнд. Например:  $3 + 4$  или  $3 + 4 * (2 - 1)$ .
  - Префиксная нотация — сначала пишется действие, затем левый операнд, затем правый операнд. Для выражения « $a+b$ » префиксная запись будет иметь вид: « $+ab$ ». Выражению « $a+b*(c+d)$ » соответствует запись « $+a*b+cd$ ».
  - Постфиксная нотация — сначала идёт левый операнд, потом правый операнд, затем знак операции. Для выражения « $a+b$ » постфиксная запись имеет вид « $ab+$ ». Выражению « $a+b*(c+d)$ » соответствует запись « $abcd+*+$ ».

# Математические выражения

- Порядок выполнения операций в префиксной и постфиксной нотациях однозначно задаётся порядком следования знаков операций в исходном выражении, поэтому отпадает необходимость использования скобок, введения приоритетов и ассоциативности операций.

# Дерево выражения

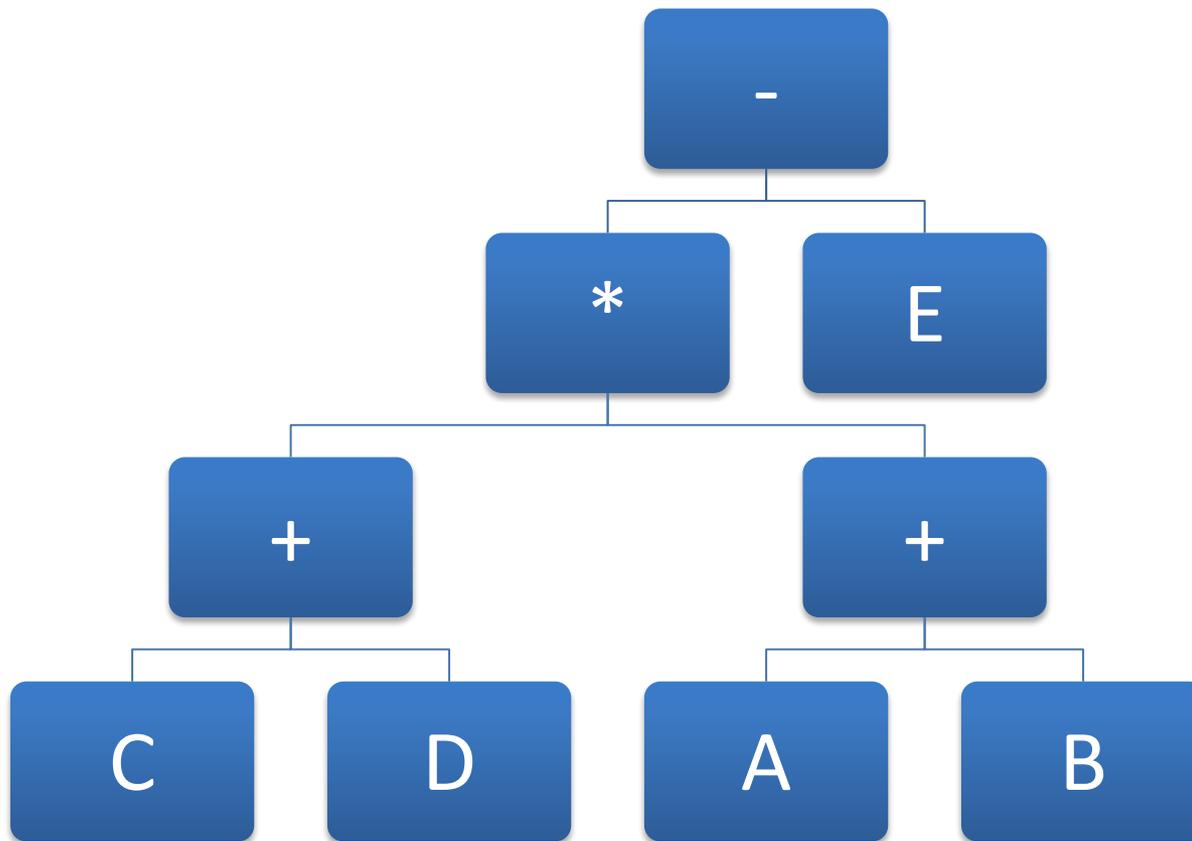
- Пусть задано простое арифметическое выражение вида:

$$(A + B) * (C + D) - E$$

- Представим это выражение в виде дерева, в котором узлам соответствуют операции, а ветвям – операнды.

# Дерево выражения

$(A + B) * (C + D) - E$

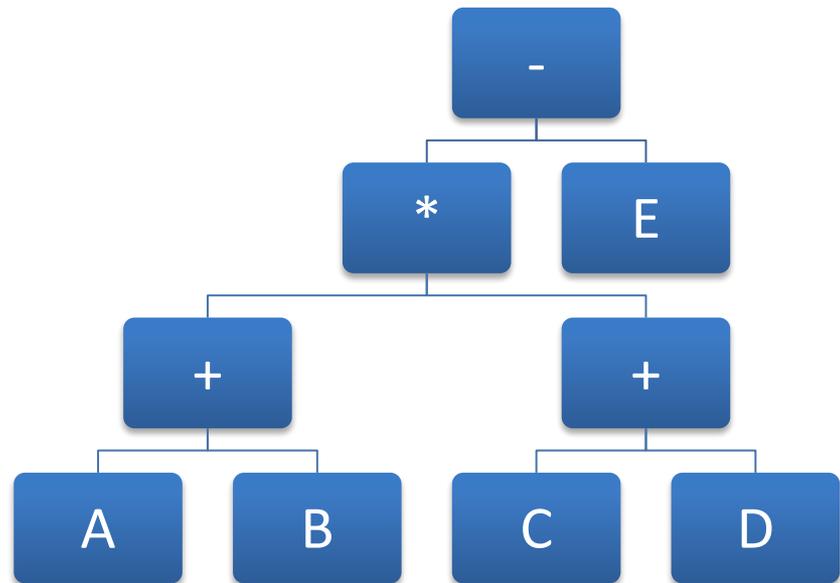


# Обратная польская нотация

- Наибольшее распространение при решении практических задач получил метод трансляции выражений с помощью обратной польской записи, которую предложил польский математик Ян Лукашевич.
- Обратная польская нотация (RPN, англ. Reverse Polish Notation) — форма бесскобочной записи математических выражений согласно постфиксной нотации.

# Трансляция выражений

- Для представления записи выражения в RPN (постфиксной нотации) необходимо совершить левый обход полученного дерева:  
 $AB + CD + * E -$



# Обратная польская нотация

- Из-за отсутствия скобок обратная польская запись короче инфиксной, а вычисление выражения происходит естественным образом – в порядке следования операций.
- Например, вычисление выражения  $AB + CD + * E$  – может быть проведено следующим образом:

# Обратная польская нотация

Вычисление выражения  $AB + CD + * E$ :

№ п/п	Анализируемая строка	Действие
0	$AB + CD + * E -$	$X = A + B$
1	$XC D + * E -$	$Y = C + D$
2	$XY * E -$	$X = X * Y$
3	$XE -$	$X = X - E$
4	$X$	Вычисление

Здесь  $X$  и  $Y$  – вспомогательные переменные.

# Алгоритм Дейкстры

- В ноябре 1961 г. Эдсгер Вибе Дейкстра опубликовал алгоритм для преобразования выражений из инфиксной нотации в RPN.

# Приоритеты операций

Зададим приоритеты используемых операций:

Операция	Приоритет
(	0
)	1
+, -	2
*, /	3

# Алгоритм Дейкстры

- Базовые правила:
  - Исходная строка символов просматривается слева направо.
  - Операнды сразу переписываются в выходную строку.
  - Знаки операций заносятся в стек.

# Алгоритм Дейкстры

- Правила работы со стеком:
  - если стек пуст, операция из входной строки переписывается в стек;
  - текущая операция выталкивает из стека все операции с большим или равным приоритетом в выходную строку;
  - открывающая скобка всегда проталкивается в стек;
  - закрывающая круглая скобка выталкивает в выходную строку все операции из стека до ближайшей открывающей скобки. Сами скобки в выходную строку не попадают, уничтожая друг друга.

# Алгоритм Дейкстры. Пример.

Индекс текущего символа	0	1	2	3	4	5	6	7	8	9	10	11	12	
Входная строка	(	A	+	B	)	*	(	C	+	D	)	-	E	
Состояние стека	(	(	+	+		*	(	(	+	+	*	-	-	
Выходная строка		A		B	+			C		D	+	*	E	-

# Контрольные вопросы

1. Какая запись представляет собой обратную польскую нотацию выражения « $A+B*(C-D)$ »?

- $A+B*C-D$
- $ABCD-*+$
- $+A*B-CD$

2. Кто из учёных ввёл понятие обратной польской нотации?

- Эдсгер Дейкстра
- Джон фон Нейман
- Ян Лукашевич

# Постановка задачи

Написать программу, преобразующую инфиксную нотацию в RPN и обратно. При этом, программа должна сама определять нотацию по виду входной строки. Нотацией «по умолчанию» считается инфиксная.

Примеры работы программы (знак □ указывает, что далее следует результат выполнения программы):

dijkstra.js a+b/(c-d) □ abcd-/+

dijkstra.js abcd-/+ □ a+b/(c-d)

Запуск программы без аргументов выводит сообщение с информацией о программе и вариантах запуска.

# Постановка задачи

Кроме того, программа должна уметь обрабатывать типовые ошибки, указывая при выводе конкретный тип (или код) ошибки:

- 1) пропуск скобки: «(a+b» или «a+(b+c)», выражения «((a+b))», «a/(b)», «(((a)))» считаются корректными;
- 2) пропуск операнда (оператора): «a+», «a/()», «ab»;
- 3) повтор операнда (оператора): «aa+b», «a++b», «i++»;
- 4) некорректный ввод: «привет!», «\$#@» и т.п.

# Постановка задачи

Вызов для тех, кому изи:

- 1) добавить в алгоритм оператор возведения в степень и унарный минус;
- 2) повтор буквенных операндов воспринимать как их умножение;
- 3) вычислять то, что можно вычислить: « $(a+b)^*(2+3)$ »  
 « $ab+5^*$ », « $ab+43^{**}$ »  « $(a+b)^*12$ »