Stack client example: Postfix expression evaluation

Infix. Standard way of writing arithmetic expressions, using parentheses for precedence.

Example. (1+((2+3)*(4*5))) = (1+(5*20)) = 101

Postfix. Write operator *after* operands (instead of in between them).

Example. 1 2 3 + 4 5 * * + Remarkable fact. No parentheses are needed!

Jan Łukasiewicz 1878–1956

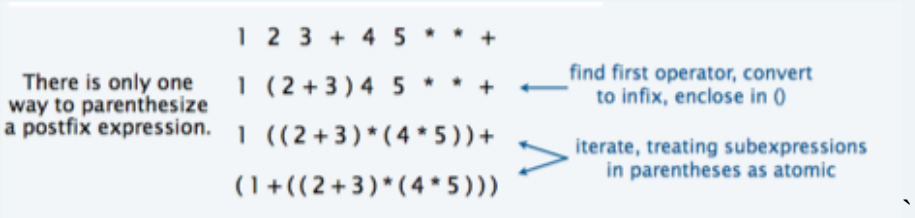
HP-35 (1972)
First handheld calculator. "Enter" means "push". No parentheses.

Made slide rules obsolete (!)

16

also called "reverse Polish" notation (RPN)

There is only one way to parenthesize a postfix expression.



makes slide rule obsolete!... but that was a long long time ago!!!

Next. With a stack, postfix expressions are easy to evaluate.

```
public class Postfix
 {
   public static void main(String[] args)
   {
     Stack<Double> stack = new Stack<Double>();
     while (!StdIn.isEmpty())
     {

    String token = StdIn.readString();
    if (token.equals("*"))
          stack.push(stack.pop() * stack.pop());
   else if (token.equals("+"))
         stack.push(stack.pop() + stack.pop());
   else if (token.equals("-"))
        stack.push(- stack.pop() + stack.pop());
  else if (token.equals("/"))
        stack.push((1.0/stack.pop()) * stack.pop());
  else if (token.equals("sqrt"))
        stack.push(Math.sqrt(stack.pop()));
  else
    stack.push(Double.parseDouble(token));
  }
    StdOut.println(stack.pop());
  }

}
```