```
function $NextWord()$
    $state \leftarrow s_0$
    $lexeme \leftarrow''$
    $clear(stack)$
    $push(\langle bad, bad\rangle)$
    while $state \neq s_e$ do
        $NextChar(char)$
        $InputPos \leftarrow InputPos + 1$
        $lexeme \leftarrow lexeme + char$
        if $Failed[state, InputPos]$ then
            break    ▷尽早结束，避免平方回滚
        end if
        if $state \in S_A$ then
            $clear(stack)$
        end if
        $push(\langle state, InputPos\rangle)$
        $cat \leftarrow CharCat[char]$
        $state \leftarrow \delta[state, cat]$
    end while
    while $state \notin S_A$ and $state \neq bad$ do
        $Failed[state, InputPos] \leftarrow true$
        $\langle state, InputPos\rangle \leftarrow pop()$
        $truncate(lexeme)$
        $RollBack()$
    end while
    if $state \in S_A$ then
        return $TokenType[state]$
    else
        return $bad$
    end if
end function

function $InitializeScanner()$
    $InputPos \leftarrow 0$
    for each state $s$ in the DFA do
        for $i = 0$ to $|input\ stream|$ do
            $Failed[s, i] \leftarrow false$
        end for
    end for
end function
```