

معرفی زبان تسلنگ

مستند حاضر زبان ساده‌ی تسلنگ^۱ را معرفی می‌کند. در گام‌های تمرین عملی درس طراحی کامپایلر، بخش‌هایی از یک مترجم برای این زبان نوشته می‌شوند. قواعد این زبان در ادامه‌ی این مستند بیان می‌شوند.

- ۱ زبان تسلنگ دارای یک نوع داده‌ی اصلی می‌باشد: کلمه (word) و لیستی از کلمه‌ها (list). هر کلمه یک مقدار عددی علامت‌دار را در خود نگه می‌دارد.
- ۲ برنامه‌های این زبان در یک فایل نوشته می‌شوند که شامل تعدادی تابع می‌باشد. در این زبان متغیرهای سراسری (Global) وجود ندارند.
- ۳ خط اول هر تابع، نام تابع و پارامترهای آن را مشخص می‌کند.
- ۴ بدنه‌ی هر تابع بین دو علامت «{» و «}» قرار می‌گیرد و شامل تعدادی عبارت (Statement) می‌باشد.
- ۵ شباهت زیادی بین ساختار عبارت‌ها و اولویت عملگرها در زبان تسلنگ و زبان C وجود دارد.
- ۶ هر بلوک (Block) در این زبان نیز بین دو علامت «{» و «}» قرار می‌گیرد.
- ۷ در هر بلوک می‌توان متغیر تعریف نمود و بلوک‌ها می‌توانند تو در تو (Nested) باشند. حوزه‌ی (Scope) هر متغیر مشابه زبان C تعریف می‌گردد.

^۱ The Simple Language (of NIT Compiler Course III)

۸ متغیرهایی محلی هر بلوک با استفاده از کلمه‌ی کلیدی «list» یا «word» و به شکل زیر تعریف می‌شوند:

```
word w;          // w is a variable of type word
list l;          // l is a variable of type list with 10 elements
```

۹ مقدار خروجی یک تابع با استفاده از کلمه‌ی کلیدی «return» مشخص می‌شود و با اجرای عبارتی که با این کلمه شروع می‌شود، اجرای تابع خاتمه می‌یابد.

۱۰ مثالی از تعریف یک تابع در ادامه نشان داده می‌شود.

```
word sum(word a, word b)
{
    word ret;
    ret = a + b;
    return sum;
}
```

۱۱ همان طور که در مثال بعدی دیده می‌شود، می‌توان یک لیست را به آرایه فرستاد و با استفاده از حلقه‌ی for می‌توان کلمه‌های موجود در یک آرایه را بررسی کرد.

```
word sum_all(list A)
{
    word sum;
    sum = 0;
    for (a in A) {
        sum = sum + a;
    }
    return sum;
}
```

۱۲ هر برنامه‌ی تسلنگ می‌تواند شامل یک تابع با نام main باشد که اجرای برنامه با فراخوانی این تابع آغاز می‌گردد.

۱۳ تابع main بدون پارامتر است و یک word بر می‌گرداند که کد برگشتی برنامه را مشخص می‌نماید.

```
word main()
{
    list A;
    A[0] = readword();
    A[1] = readword();
    A[2] = readword();
    A[3] = readword();
    printword(sum_all(A));
    return 0;
}
```

۱۴ در زبان تسلنگ از عبارت شرطی if و حلقه‌ی while با ساختاری مشابه زبان C می‌توان استفاده کرد.

۱۵ مثال زیر استفاده از if را نمایش می‌دهد.

```
// Inefficient calculation of Fibonacci sequence
word fib(word n)
{
    if (n < 2)
        return 1;
    return fib(n - 1) + fib(n - 2);
}
```

قواعد تجزیه‌ی زبان تسلنگ

در ادامه ساختار BNF زبان تسلنگ نمایش داده شده است. در این ساختار اولویت‌های عملگرها (که مشابه عملگرهای زبان C هستند) در نظر گرفته نشده است. همچنین در برنامه‌های زبان تسلنگ، دو علامت / و حروفی که بعد از آن آمده‌اند تا آخر خط Comment محسوب می‌شوند.

```
prog ::= func |
      func prog
func ::= type iden ( flist ) { body }
body ::= stmt |
       stmt body
stmt ::= expr ; |
       defvar ; |
       if ( expr ) stmt |
       if ( expr ) stmt else stmt |
       for ( iden in expr ) stmt |
       while ( expr ) stmt |
       return expr ; |
       { body }
defvar ::= type iden
expr ::= iden ( clist ) |
       expr [ expr ] |
       expr = expr |
       expr + expr |
       expr - expr |
       expr * expr |
       expr / expr |
       expr % expr |
       expr < expr |
       expr > expr |
       expr == expr |
       expr <= expr |
       expr >= expr |
       expr or expr |
       expr and expr |
       ! expr |
       - expr |
       + expr |
```

```
        ( expr ) |
        iden |
        num
flist ::= |
        type iden |
        type iden , flist
clist ::= |
        expr |
        expr , clist
type ::= word |
        list
num ::= [0-9]+
iden ::= [a-zA-Z_][a-zA-Z_0-9]*
```