

### \*\*\* linkモジュール・ライブラリの実装方法 \*\*\*

#### 1. linkモジュール・ライブラリ

linkモジュール・ライブラリは、デカルト言語において高速なリンクリストを実現するためのものです。

実装は異なるのですが、2重リンクの双方向リストと同様に扱えます。また、putメソッドにより任意の構造のデータをlinkの要素として追加可能です。

使用する場合には、以下のような手順です。

- 1) linkモジュールをincludeする。  
? <include link>;
- 2) linkモジュールを継承するオブジェクトを作る。  
::<lk  
inherit link;  
>;  
上の例では、新たにlkオブジェクトを定義してlinkモジュールを継承している。
- 3) 新規に作成したオブジェクトに対して、ノードの追加・削除やリストの前後両方向への走査を行います。

#### 2. 使用例

使用例を以下に示しましょう。

```
? <include link>;

::<lk
    inherit link;
>;

? ::lk <getlist #l>;
? ::lk <length #len>;

? ::lk <put a>;
? ::lk <put b>;
? ::lk <put c>;
? ::lk <put d>;
? ::lk <put e>;
? ::lk <put f>;
? ::lk <getlist #l>;
? ::lk <middle_node #n>;

? <list>;

? ::lk <next #p>;
```

```

? ::lk <prev #p>;
? ::lk <prev>;
? ::lk <prev #p>;
? ::lk <next>;
? ::lk <prev #p>;
? ::lk <prev #p>;
? ::lk <prev #p>;
? ::lk <prev #p>;

? ::lk <get #v>;

? ::lk <getlist #l>;

? <list>;

? ::lk <length #n>;

? ::lk <setlist (1 2 3 4 5 6 7 8 9 10)>
  ::lk <getlist _>
  ::lk <length #n>;

? ::lk <next _>
  ::lk <next _>
  ::lk <del>
  ::lk <getlist _>;

? ::lk <del>;
? ::lk <getlist _>;

? ::lk <top _>;
? ::lk <last _>;
? ::lk <top _>;
? ::lk <last _>;
? ::lk <getlist _>;

```

### 3. linkの操作

以下のメソッドが使えます。

<prev #prev1>  
一つ前のlinkに移ります。

<next #next1>  
一つ後ろのlinkに移ります。

<clear>  
linkの全要素を削除してクリアします。

<put #v>

引数に設定された値をlinkの一つのノード要素として追加します。

<get #v>

引数に、現在指しているlinkのノード要素を取り出します。

<del>

現在指しているlinkのノード要素を削除します。

<getlist #list>

linkに繋がっている要素をすべて接続したリストを返します。

<setlist #list>

引数に設定されたリストをlinkとして登録します。

<length #len>

linkに接続されているノード要素の長さを返します。

<top #n>

linkの先頭に移動します。

<last #n>

linkの最後尾に移動します。