

که آن را توصیف می‌کند، به ارث ببریم. ابتدا باید به‌طور غیر رسمی عملیاتی را توصیف کنیم که لازمند و سپس وضعیت را.

این وضعیت شامل مجموعه‌ای از نوع  $T$  است. فرض می‌کنیم که بیشتر از ۱۱۰۰ قلم در این جدول وجود ندارد. همچنین باید یک‌سری عملیاتی را تعریف کنیم که در این وضعیت عمل می‌کنند:

- $INIT$  که شروع به حالی کردن جدول نشانه‌ها می‌کند.
- $add$  که قلمی را به جدول می‌افزاید.
- $remove$  که قلمی را از جدول حذف می‌کند.
- $number$  که با تعدادی از قلم‌ها در جدول نشانه برمی‌گردد. تعریف آن در زیر آمده است.

### SymbolTable [T]

$maxItems : \mathbb{N}$   
 $maxItems \leq 100$

$table : T$   
 $\#table \leq maxItems$

#### $INIT$

$table = \{\}$

#### $add$

$\Delta (table)$

$item? : T$

$\#table < maxItems$

$table' = table \cup \{item?\}$

#### $remove$

$\Delta (table)$

$item? : T$

$item? \in table$

$table' = table / \{item?\}$

#### $number$

$\equiv (table)$

$noInTable! : \mathbb{N}$

$noInTable! = \#table$

این‌ها عملیات بسیار ساده‌ای هستند که مستلزم عملیات استاندارد مجموعه‌اند هم‌چون  $U$  و از قالب نشان داده شده در مثال صف استفاده می‌کند. عملیات *add* تنها در صورتی تعریف می‌شود که جدول دارای فضایی برای قلم افزوده و عملیات *remove* تنها در صورتی تعریف می‌شود که قلمی حذفی در جدول باشد. عملیات *number* به وضعیت اثر نگذشته و بنابراین شامل عنصر (*table*)  $\equiv$  است. اگر می‌خواهیم جدولی را به‌عنوان جدول اسامی کارکنان از نوع *EMPLOYEE* به‌وجود آوریم پس از تمام چیزی که لازم داریم عبارت است از:

*EmployeeTab* = *SymbolTable*[*EMPLOYEE*]  
 [*emps/table*, *emp?/item?*, *emp!/item!*]

که در آن جدول به جدولی تبدیل شده که شامل قلم‌های *Employee* و پارامترهای ورودی تعریف شده به‌صورت کلی است که جایگزین پارامترهای خاص *emp?* و *emp!* هستند. طرح کلی *object Z* که این جدول جدید کارمندان را از طریق نشان دادن به‌صورت تجربی توصیف می‌کند در زیر آمده است. این جدول در برگرنده تعریف مجدد اپراتورهایی است که هم‌اکنون تعریف شده‌اند

<i>Emptable</i>
<i>e: FreqEmployeeTab</i>
<i>FreqEmpAdd</i> $\triangleq$ <i>e.add</i>
<i>FreqEmpRemove</i> $\triangleq$ <i>e.remove</i>
<i>FreqEmpInit</i> $\triangleq$ <i>e.INIT</i>
<i>FreqEmpNumber</i> $\triangleq$ <i>e.number</i>

حالا فرض کنیم که ما به استفاده از وراثت نیاز داریم. به‌منظور تشریح آن، باید فرض کنیم که ما به عملیات *Find* نیاز داریم که ارزش *Found* (یافته شده) را در صورت یافته شدن قلم‌ها در جدول نشانه برمی‌گرداند و مقداری *not found* می‌باشد که در جدول یافت نشده است. هم‌چنین برای این کاربرد فرض دیگری داریم که در آن وقتی عملیات *find* اجرا شده، اغلب از همین قلم استفاده می‌کنیم که در جدول است. وقتی این عملیات به‌کار گرفته می‌شود چک می‌کنیم که این قلم است قبل از شروع آن‌چه که می‌تواند یک جستجوی درازمدت در جدول باشد و به این ترتیب اطمینان حاصل می‌کنیم. قبل از تعریف این طرح لازم از طرح جدول اصلی نشانه استفاده کنیم تا این اطلاعات را در آن بگنجانیم. ابتدا عضو متمایزی از جدول را نیاز داریم که به‌عنوان *Freq Item* شناخته می‌شود. این قلمی است که اغلب مورد استفاده قرار می‌گیرد.

تعریف این جدول جدید نشانه خواهد بود: