

11.3 -

では、ジェネリック・パッケージの を ていきましょう。ジェネリック・パッケージのこの は、[Ada 95 RM section 12.8](#) からそのまま ってきたものです。

ここでは、プッシュとポップの をするスタックのためのジェネリック・パッケージを したい を えてみましょう。ここでは、このようなスタックを するための の の を します。 たちはいくつかの Item をストアし、 サイズを のスタック・パッケージを することになります。

```
generic   Size : Positive;   type Item is private;   package Generic_Stack is   procedure Push(E : in Item);   procedure Pop (E : out Item);   Overflow, Underflow : exception;   end Generic_Stack;
```

さて、 は されることを としますので、ここで の を せることにしましょう。

```
package body Generic_Stack is   type Table is array (Positive range <>) of Item;   Space : Table(1 .. Size);   Index : Natural := 0;   procedure Push(E : in Item) is   begin   if Index >= Size then   raise Overflow;   end if;   Index := Index + 1;   Space(Index) := E;   end Push;   procedure Pop(E : out Item) is   begin   if Index = 0 then   raise Underflow;   end if;   E := Space(Index);   Index := Index - 1;   end Pop;   end Generic_Stack;
```

プログラマーはどこでも Generic_Stack パッケージを することが となります。もしプログラマーが 200 の を の 「Stack_Int」という の しいパッケージを したいのなら、 の ようにします。

```
package Stack_Int is new Generic_Stack(Size => 200, Item => Integer);
```

「Size =>」と「Item =>」はオプションです。 ならばプログラマーはこれらを することが ますが、これらを めておいたほうがコードが になります。その の となった に、プログラマーは しい を Stack_Int にプッシュすることが できるようになります。

```
Stack_Int.Push(7);
```

: <http://www.adahome.com/Tutorials/Lovelace/s11s3.htm>