Report concerning:

**Pipe Package**

By

**L. Dries**

dd. **16-08-2019**

Version: 1.00 Original version 16-08-2019 L. Dries

# Index

[1 Index 3](#_Toc16847580)

[2 Introduction 4](#_Toc16847581)

[3 General 5](#_Toc16847582)

[4 Internal structure 6](#_Toc16847583)

[5 Use of the package 7](#_Toc16847584)

[6 Listings 8](#_Toc16847585)

[6.1 Specification 8](#_Toc16847586)

[6.2 Body 10](#_Toc16847587)

# Introduction

A Pipe is a first-in first-out memory structure. Such a structure is for instance meaningful in a case of communication between asynchronous tasks. By pushing some value in the pipeanother task can read that valkuewhen needed without asking for the value. If the value is not yest present the task must ait until the value is present. Using such a device will use memory outside of the memory that is connected to the program by normal dimensions in your program. The standard garbage collection of ADA will clear that memory whenever not reachable anymore.

# General

The pipe package is a package that creates s pipe for one type of identifier. That can be an integer pipe, an unbounded string pipe, but also a pipe of a type that you have created yourself. But you can only put identifiers of that type in it. More pipes of the same type or a different one are possible. To set a value in the pipe is called pushing, getting one out is called popping. The last value pushed is always the first value popped.

# Internal structure

Internally every value pushed is automatically connected with a pointer to the one pushed before and the pointer to the value pushed first is saved generally. When a value is popped you get the value pointed to by that external pointer and the external pointer is shifted to the next one entered.

# Use of the package

Item is the type you use for the pipe

procedure Push ( Typ : Item );

The procedure Push pushes an item on the pipe

function Get return Item;

The function Get gets the top value of the pipe without removing it from the pipe

procedure Remove;

The procedure Remove removes the top value from the pipe without reading it

function Pop return Item;

The function Pop gets the top value from the pipe and removes it

procedure Clear;

The procedure Clear empties the complete pipe

function Get\_Length return integer;

The function Get\_Depth presents the number of levels within the pipe at that moment.

To use the Pipe package you have to declare it by:

package New\_Pipe is new Pipe(ID\_Line); to get a pipe consisting od element of the type ID\_Line.

You can then use statement as New\_Pipe.Pop(Line); where Line is of the type ID\_Line

# Listings

## Specification

-----------------------------------------------------------------------

-- Program\_Init A package to create INI files --

-- --

-- Copyright (C) 2019 L. Dries --

-- --

-- This library is free software; you can redistribute it and/or --

-- modify it under the terms of the GNU General Public --

-- License as published by the Free Software Foundation; either --

-- version 3 of the License, or (at your option) any later version. --

-- --

-- This library is distributed in the hope that it will be useful, --

-- but WITHOUT ANY WARRANTY; without even the implied warranty of --

-- MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU --

-- General Public License for more details. --

-- --

-- You should have received a copy of the GNU General Public --

-- License along with this library; if not, write to the --

-- Free Software Foundation, Inc., 59 Temple Place - Suite 330, --

-- Boston, MA 02111-1307, USA. --

-- --

-----------------------------------------------------------------------

package Pipe is

type Block\_Pipe is limited private;

type Pipe\_Pointer is limited private;

-----------------------------------------------------------------------------

-- The procedure Pushes a value Typ of the type Item in the pipe adding one

-- length to the pipe

-----------------------------------------------------------------------------

procedure Push ( Typ : Item );

-----------------------------------------------------------------------------

-- The function reads the first pushed value from the pipe but let the pipe

-- intact.

-----------------------------------------------------------------------------

function Get return Item;

-----------------------------------------------------------------------------

-- The procedure reduces the length of the pipe by one

-----------------------------------------------------------------------------

procedure Remove;

-----------------------------------------------------------------------------

-- The function pops a value of the type Item from the pipe reducing the

-- length of the pipe by one in the proces.

-----------------------------------------------------------------------------

function Pop return Item;

-----------------------------------------------------------------------------

-- The procedure Clears the pipe completely

-----------------------------------------------------------------------------

procedure Clear;

-----------------------------------------------------------------------------

-- The function gets the Pipe Length

-----------------------------------------------------------------------------

function Get\_Length return integer;

private

type Pipe\_Pointer is access Block\_Pipe;

type Block\_Pipe is record

typ : Item;

previous : Pipe\_Pointer := null;

next : Pipe\_Pointer := null;

end record;

FirstPP : Pipe\_Pointer := null;

LastPP : Pipe\_Pointer := null;

Length : integer := 0;

end Pipe;

## Body

-----------------------------------------------------------------------

-- Program\_Init A package to create INI files --

-- --

-- Copyright (C) 2019 L. Dries --

-- --

-- This library is free software; you can redistribute it and/or --

-- modify it under the terms of the GNU General Public --

-- License as published by the Free Software Foundation; either --

-- version 3 of the License, or (at your option) any later version. --

-- --

-- This library is distributed in the hope that it will be useful, --

-- but WITHOUT ANY WARRANTY; without even the implied warranty of --

-- MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU --

-- General Public License for more details. --

-- --

-- You should have received a copy of the GNU General Public --

-- License along with this library; if not, write to the --

-- Free Software Foundation, Inc., 59 Temple Place - Suite 330, --

-- Boston, MA 02111-1307, USA. --

-- --

-----------------------------------------------------------------------

package body Pipe is

procedure Push ( Typ : Item ) is

begin

if FirstPP = null then

FirstPP := new Block\_Pipe;

LastPP := FirstPP;

else

declare

NewPP : Pipe\_Pointer;

begin

NewPP := FirstPP;

while NewPP.next /= null loop

NewPP := NewPP.next;

end loop;

NewPP.next := new Block\_Pipe;

NewPP.next.previous := NewPP;

LastPP := NewPP.next;

end;

end if;

Length := Length + 1;

LastPP.typ := Typ;

end Push;

function Get return Item is

begin

if FirstPP = null then

return Null\_Item;

else

return FirstPP.typ;

end if;

end Get;

procedure Remove is

OldPP : Pipe\_Pointer;

begin

OldPP := FirstPP;

if FirstPP = LastPP then

FirstPP := null;

else

FirstPP := OldPP.next;

end if;

Length := Length - 1;

if Length < 0 then

Length := 0;

end if;

end Remove;

function Pop return Item is

value : Item;

begin

value := Get;

Remove;

return value;

end Pop;

procedure Clear is

begin

while FirstPP /= null loop

Remove;

end loop;

end Clear;

function Get\_Length return integer is

begin

return Length;

end Get\_Length;

end Pipe;