Министерство образования и науки РФ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Рыбинский государственный авиационный технический университет

имени П.А. Соловьева»

Факультет радиоэлектроники и информатики

Кафедра Математическое и программное обеспечение ЭВС

**ОТЧЕТ**

**по лабораторной работе №3**

**R**EADER 1

Студент группы ИПБ-15 Козин О.И .

*(Код) (Подпись, дата) (Фамилия И. О.)*

Рыбинск 2019

Оглавление

[Цель 2](#_Toc1572962)

[Простейшие взаимодействия 2](#_Toc1572963)

[Архитектура 2](#_Toc1572964)

[Анализ работы программ 3](#_Toc1572965)

[Приложение. Коды программ. 4](#_Toc1572966)

[Приложение. Интерпретация временных диаграмм 10](#_Toc1572967)

# Цель

1. Изучить механизм работы пакетов в языке программирования Ada.
2. Разработать архитектуру программы читатели-писатели с выделением её в пакет.
3. Запрограммировать пакет, его описание, и программу с его использованием.

# Простейшие взаимодействия

## Архитектура

Пакет

Controller

Read

Fin

Write

Function Read

Function Fin

Function Write

Reader

Writer

Программа

Рис. 1. Архитектура пакета с контроллером читателей-писателей, и его использования из внешней программы

## Анализ работы программ

Ниже представлен фрагмент временных диаграмм работы программ.

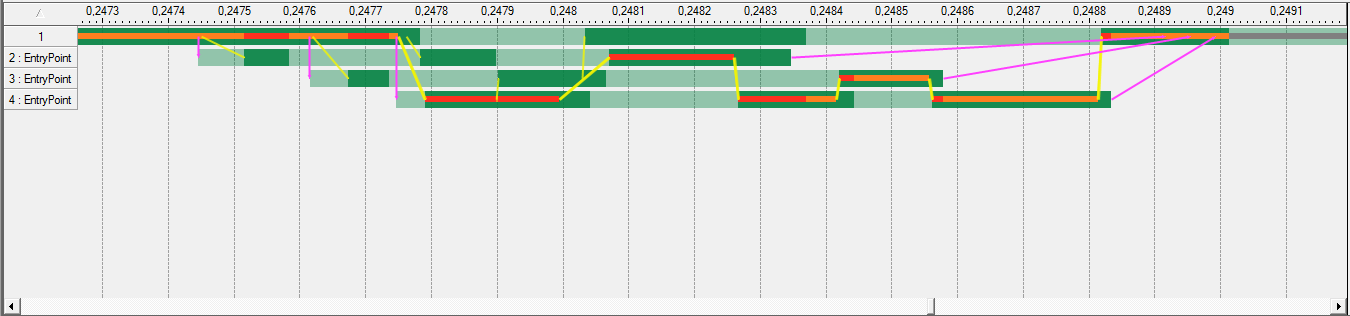


Рис. 2. Временные диаграммы работы программы с пакетом

# Приложение. Коды программ.

Описание пакета.

package RW\_Pack is

procedure Write;

procedure Fin;

procedure Read;

end RW\_Pack;

Код пакета.

with Text\_Io;

package body RW\_Pack is

use Text\_Io;

task Controller is

entry Read;

entry Write;

entry Fin;

end Controller ;

task body Controller is

State : Integer := 0;

Readers\_Count : Integer := 0;

G\_data : integer :=0;

begin

Put\_Line (

"Controller start");

loop

case State is

when 0 =>

select

accept Read do

State := 1;

Readers\_Count:=Readers\_Count+1;

Put\_Line ("read start");

Put\_Line(Integer'Image(Readers\_Count));

end Read;

or

accept Write do

State := 2;

Put\_Line ("write start");

end Write;

end select;

when 1 =>

select

accept Read do

State := 1;

Readers\_Count:=Readers\_Count+1;

Put\_Line (

"read start");

Put\_Line(Integer'Image(Readers\_Count));

end Read;

or

accept Fin do

Readers\_Count := Readers\_Count - 1;

Put\_Line ("Fin read");

Put\_Line(Integer'Image(Readers\_Count));

if (Readers\_Count = 0)

then

State := 0;

end if;

end Fin;

or

accept Write do

State := 3;

Put\_Line ("write req ");

end Write;

end select;

when 2 =>

select

accept Fin do State := 0;

G\_data := G\_data +1;

G\_data := G\_data +1;

G\_data := G\_data - 1;

Put\_Line (

"Fin write");

end Fin;

or

delay 0.1;

Put\_Line (

"Wait write Fin");

end select;

when 3 =>

select

accept Fin do Readers\_Count := Readers\_Count - 1;

Put\_Line (

"Stop read");

Put\_Line(Integer'Image(Readers\_Count));

if (Readers\_Count = 0)

then

State := 2;

Put\_Line (

"Start write");

end if;

end Fin;

end select;

when others => -- error

null;

end case;

end loop;

Put\_Line (

"Controller End");

end Controller;

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

procedure Write is

begin

controller.write;

end Write;

procedure Fin is

begin

controller.fin;

end Fin;

procedure Read is

begin

controller.read;

end Read;

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

Begin

Put\_Line ("Start RW");

End RW\_pack;

Использование пакета.

with Text\_Io;

with RW\_pack;

procedure P02 is

--use Ada.Text\_IO;

use Text\_Io;

use RW\_pack;

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

task type Reader is

entry Start;

end Reader;

task body Reader is

begin

accept Start;

loop

Read;

--delay 1.0;

Fin;

--delay 1.0;

end loop;

Put\_Line("Done");

end Reader;

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

task type Writer is

entry Start;

end Writer ;

task body Writer is

begin

accept Start;

loop

Write;

--delay 1.0;

Fin;

--delay 1.0;

end loop;

Put\_Line("Done");

end Writer ;

R\_1 : Reader;

R\_2 : Reader;

R\_3 : Reader;

W\_1 : Writer;

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

begin

R\_1.Start;

R\_2.Start;

R\_3.Start;

W\_1.Start;

end P02;

# Приложение. Интерпретация временных диаграмм

