МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**Український державний університет науки і технологій**

Кафедра «Комп’ютерні інформаційні технології»

**Лабораторна робота №6**

**з дисципліни «Конструювання ПЗ»**

**на тему: «Програмування паралельних процесів у Erlang»**

Виконала: студент гр. ПЗ2011

Кулик С. В

Прийняв: Іванов О.П.

.

Дніпро, 2022

Лабораторна робота №6

**Тема.** Програмування паралельних процесів у Erlang.

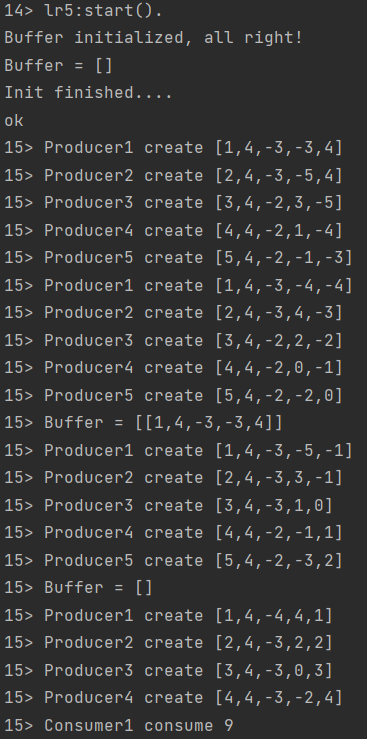
1. **Завдання**

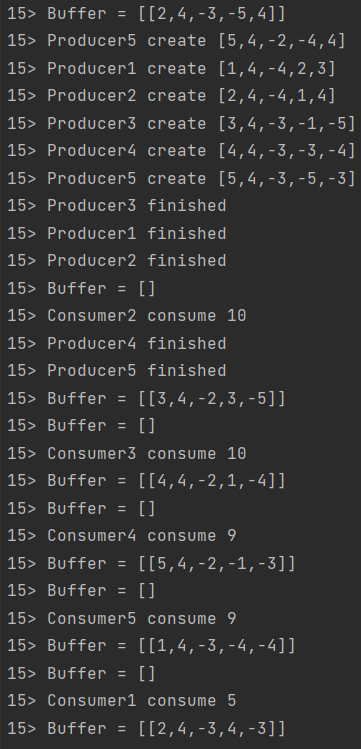
Реалізувати модель системи виробників споживачів. Взаємодія через загальний буфер з даними, в який виробники кладуть значення (числові списки), а споживачі забирають і виробляють операцію споживання (перетворення даних). Кожна дія над загальним буфером виводиться на екран консолі.

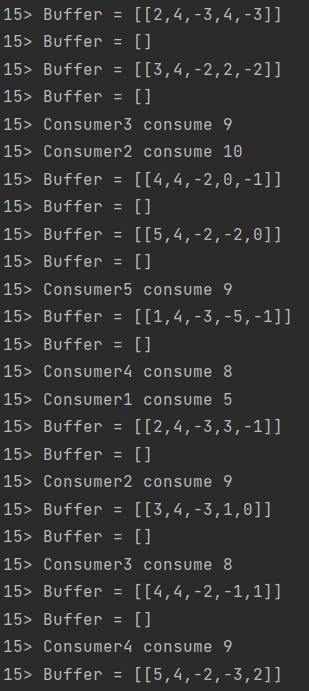
Варіант 15 - Знайти суму позитивних елементів.

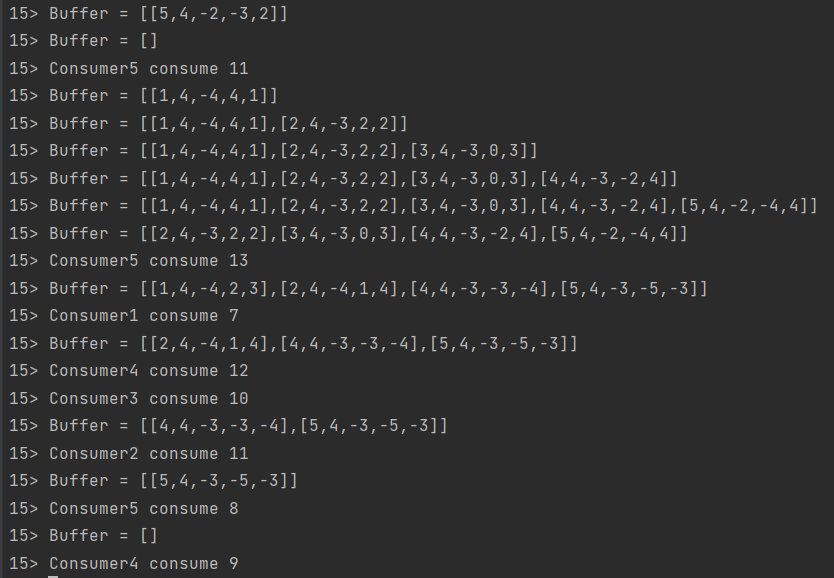
1. **Текст програми**
2. -module(lr5).  
   -compile(export\_all).  
   -compile(nowarn\_export\_all).  
     
   %%% процесс организующий общий буфер  
   bufer(Buf)->  
    io:format("Buffer = ~w~n",[Buf]),  
    receive  
    %% реализовать по заданию структуру буфера  
    {get,Pid} when (length(Buf)>0) -> [H|T]=Buf, Pid!{item,H}, bufer(T);  
    {put,Value} -> bufer(Buf ++ [Value]);  
    {clear} -> bufer([]);  
    {stop} -> ok  
    end.  
     
   %%% процесс производителя  
   producer(Number\_proc,0)-> io:format("Producer~w finished ~n",[Number\_proc]);  
   producer(Number\_proc,Count\_iter)->  
    random:seed(Number\_proc,Count\_iter,0),  
    Elem=[Number\_proc|produce(4)], % произведенный элемент  
    io:format("Producer~w create ~w~n",[Number\_proc,Elem]),  
    base!{put,Elem},  
    producer(Number\_proc,Count\_iter-1).  
     
   %%% генерировать числовой список  
   produce(0)->[];  
   produce(X)->[5-random:uniform(10)|produce(X-1)].  
     
   %%% процесс потребителя  
   consumer(Number\_proc,0)-> io:format("Consumer~w finished ~n",[Number\_proc]);  
   consumer(Number\_proc,Count\_iter)->  
    base!{get,self()},  
    receive  
    {item,Elem} -> io:format("Consumer~w consume ~w~n",[Number\_proc,consume(Elem)]),  
    consumer(Number\_proc,Count\_iter-1);  
    {stop}->ok  
    end.  
     
   %%% функция преобразования(потребления) элемента (реализовать по заданию)  
   sum\_([]) -> 0;  
   sum\_([H|T]) when H > 0 -> H + sum\_(T);  
   sum\_([\_|T]) -> sum\_(T).  
     
   consume(X)->sum\_(X).  
     
   start() ->  
    %% 1.создание процесса буфера  
    case whereis(base) of  
    undefined -> register(base,spawn(lr5, bufer,[[]]));  
    \_ -> base!{clear}  
    end,  
    io:format("Buffer initialized, all right!~n"),  
    %% 2. создание процессов производителей  
    [spawn(lr5,producer,[X,5]) || X <- [1,2,3,4,5]],  
    %% 3. создание процессов потребителей  
    [spawn(lr5,consumer,[X,5]) || X <- [1,2,3,4,5]],  
    io:format("Init finished....~n").
3. **Результати роботи**

****

****

****

****

****

**Висновки**

В даній лабораторній роботі було реалізовано систему виробників та споживачів з використанням мови Erlang.

Для реалізації системи було використано механізм процесів Erlang, які виконують ролі виробників та споживачів. Виробники випускають елементи у буфер, а споживачі забирають їх з буфера.