ParentProcess.java

```java
/*
 * Written by JJ Shepherd
 */
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.io.*;

public class ParentProcess{

    public static final int WIDTH = 600;
    public static final int HEIGHT = 600;

    public static final String BASE_CMD = "java";
    public static final String CLASS_PATH = "-classpath";
    public static final String BIN_PATH = "./bin";
    public static final String PROD_CLASS = "Producer";
    public static final String CONS_CLASS = "Consumer";
    public static final int SLEEP_TIME = 5000;

    public static final String FILE = "./file.txt";

    public static void main(String[] args) {
        //Make a frame
        JFrame frame = new JFrame("Parent Process");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);//To shut down main thread when the frame is closed.
        frame.setBounds(0,0,WIDTH,HEIGHT);
        //Create a button
        JButton button01 = new JButton("Start Processes");
        button01.setBounds(0, 0, WIDTH, HEIGHT);
        //Create the button's functionality
        button01.addActionListener(
                new ActionListener()
                {
                    //Overriding the ActionListener's method actionPerFormed
                    public void actionPerformed(ActionEvent e)
                    {
                        try
                        {
                            ProcessBuilder producerBuilder, consumerBuilder;
                            Process producer, consumer;

                            //Constructs a builder with all of the command line entries as strings
                            producerBuilder = new ProcessBuilder(BASE_CMD,CLASS_PATH,BIN_PATH,PROD_CLASS);
                            consumerBuilder = new ProcessBuilder(BASE_CMD,CLASS_PATH,BIN_PATH,CONS_CLASS);

                            //Start each process
                            producer = producerBuilder.start();
                            consumer = consumerBuilder.start();
                            //Capture the process's Standard Output (called InputStream)
                            BufferedReader prodRead = new BufferedReader(new InputStreamReader(producer.getInputStream()));
                            BufferedReader consRead = new BufferedReader(new InputStreamReader
(consumer.getInputStream()));

                            //Sleep for some number of milliseconds to let the other processes run
                            System.out.println("Sleeping for "+SLEEP_TIME+" milliseconds");
                            Thread.sleep(SLEEP_TIME);
                            System.out.println("Not Sleeping. Awake!");

                            //Close both processes
                            producer.destroy();
                            consumer.destroy();

                            //Count the lines produced
                            int prodCount = 0;
                            String line = "";
                            while((line = prodRead.readLine()) != null)
                            {
                                System.out.println(line);
                                prodCount++;
                            }
                            //Count the lines consumed
                            int consCount = 0;
                            while((line = consRead.readLine()) != null)
                            {
                                consCount++;
                                System.out.println(line);
                            }
                            //Every time this is run it will mostly result in different numbers for producers and consumers.
                            //This is the producer consumer problem and evokes a race condition
                            System.out.println("Producer produced "+prodCount+" Consumer Consumed "+consCount);

                        }
                        catch(Exception ex)
                        {
                            System.out.println(ex);
                        }
                    }
                });
        frame.add(button01);
        frame.setVisible(true);
    }
}

```