Государственное образовательное учреждение

высшего профессионального образования

«Уфимский Государственный Авиационный Технический Университет»

Кафедра «Вычислительной математики и кибернетики»

Отчет

По лабораторной работе

по дисциплине: Системы реального времени

Лабораторная работа № 1.

Producer-Consumer (Производитель-Потребитель)

Проверил

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Выполнил:

студент гр. ПРО-301в

Доронин С.Г.

Уфа 2013

# Задание

Разработать приложение, состоящее из двух функциональных частей. Первая время от времени отправляет сообщения (например, это может происходить при нажатии некоторой кнопки, или клавиши на клавиатуре) второй, вторая каким-то образом на них реагирует.

Размер буфера передачи между этими частями – 1 сообщение, а первая задача (передатчик) никогда не ждет очистки буфера. Т.е. если вторая часть (приемник) не успевает получить помещенное в буфер сообщение, а уже возникает новое сообщение, то это новое сообщение заменяет старое, а старое - теряется.

Здесь передатчик имитирует аппаратное обеспечение СРВ. Время обработки сообщения приемником должно быть 1-2с (можно воспользоваться функцией Sleep). Необходимо добиться того, чтобы при возникновении 2-3 сообщений подряд (до окончания обработки первого из них) ни одно из них не терялось, т.е. все обрабатывались приемником.

# Структура решения (этапы решения и их взаимосвязь)

Для решения этой задачи были созданы 7 классов:

1. Buffer - буфер, который хранит одно сообщение
2. BufferReader - поток, читающий буфер
3. BufferValue - значение для записи в буфер
4. BufferWriter - поток пишущий в буфер
5. Const - класс с константами
6. MainWindow - главное окно
7. Main - класс запускающий программу

Программа запускается с класса «**Main**», который внутри себя создает «**MainWindow**» (главное окно программы).

«**MainWindow**» внутри себя запускает класс-поток «**BufferReader**», который по таймеру каждые 100 миллисекунд, проверяет класс «**Buffer**», если в нем есть какое-либо значение, то он его перемещает в свой буфер, который реализован в виде списка с неограниченной длиной.

Так же «**MainWindow**» внутри себя запускает поток, который по таймеру каждые 2 секунды, проверяет буфер класса «**BufferReader**», если там появляются новые значения, то он их выводит на форму.

Так же на форме есть кнопка «**Push message to buffer**», которая посылает случайное сообщение с помощью класса «**BufferWriter**» в класс «**Buffer**», который может хранить только одно сообщение.

# Описание интерфейса с пользователем (руководство пользователя)

Программа запускается с файла **Producer-Consumer.jar** или с файла сценария **run.bat.** Для работы приложения требуется установленная JRE (Java Runtime Environment).

После запуска программы появится окно, имеющее вид (Рисунок 1).

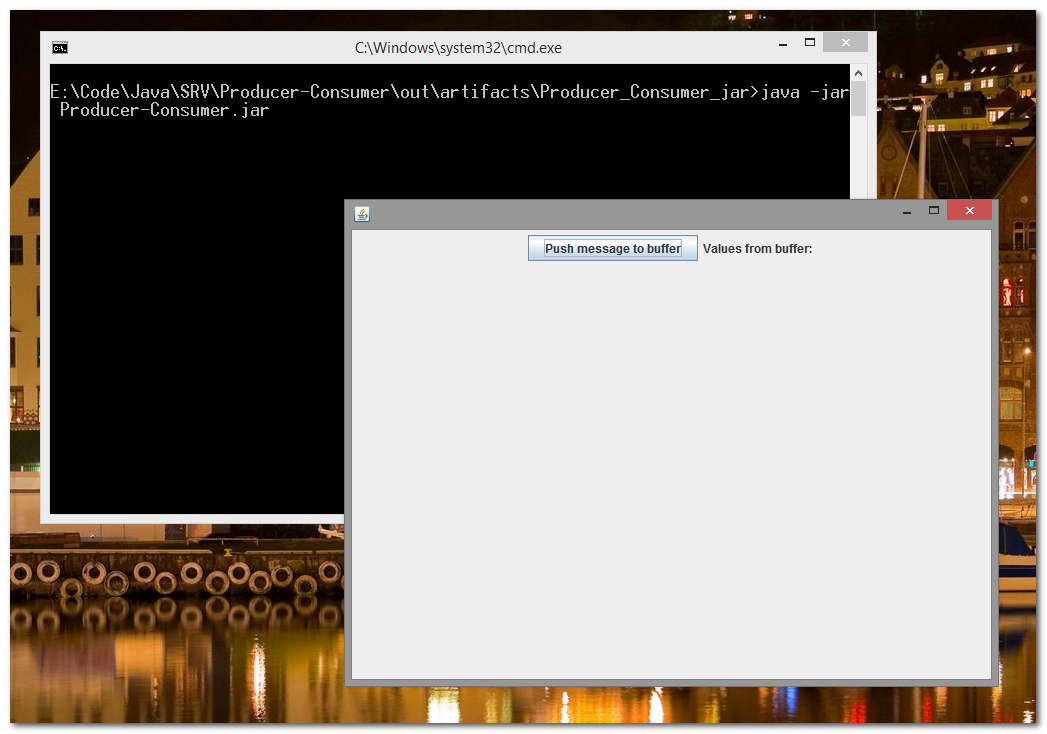


Рисунок . Главное окно программы

Сразу после запуска программы запускается поток «**BufferReader**» и поток читающий его внутренний буфер.

При нажатии на кнопку «**Push message to buffer**», будет послано случайное сообщение в класс буфер.

Поток посылающий сообщение в буфер, поток читающий буфер и сам буфер сообщают об изменении своих состояний в консоль (Рисунок 2).

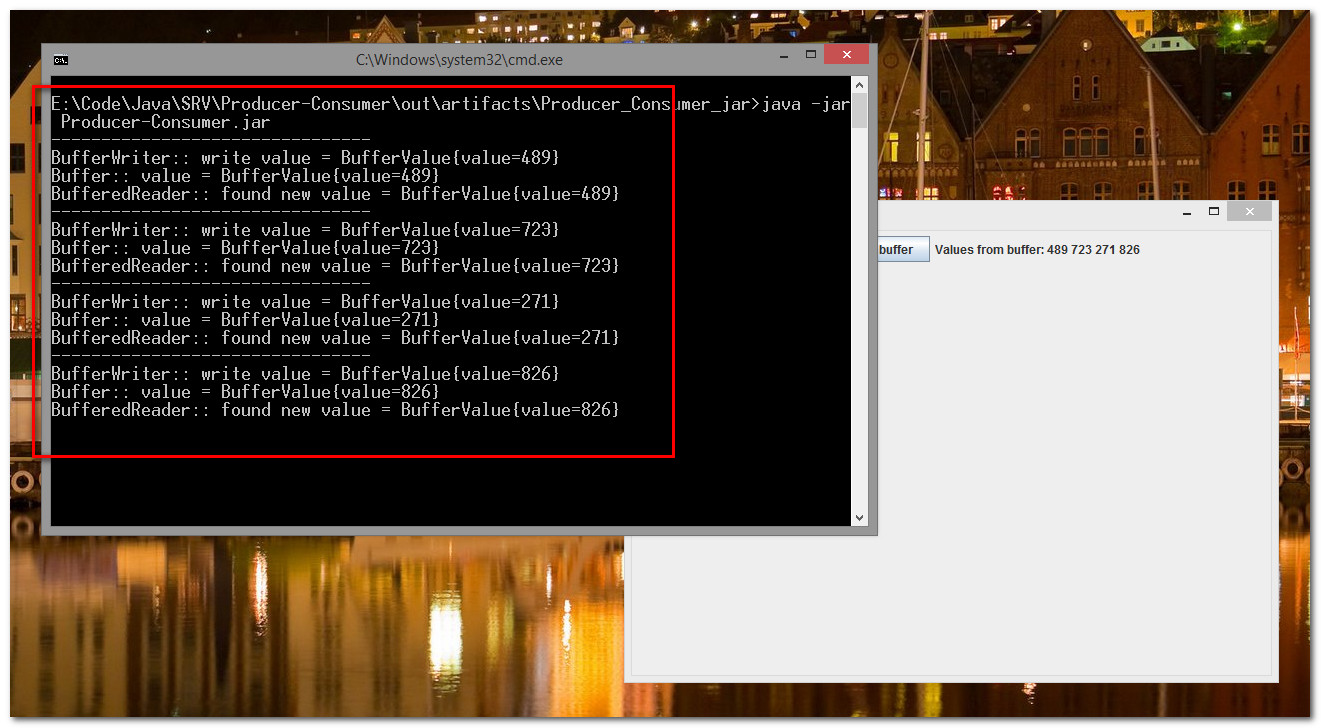


Рисунок . Консольное окно с журналом сообщений потоков

Сообщения обработанные главным потоком выводятся в лейбл на главном окне (Рисунок 3).

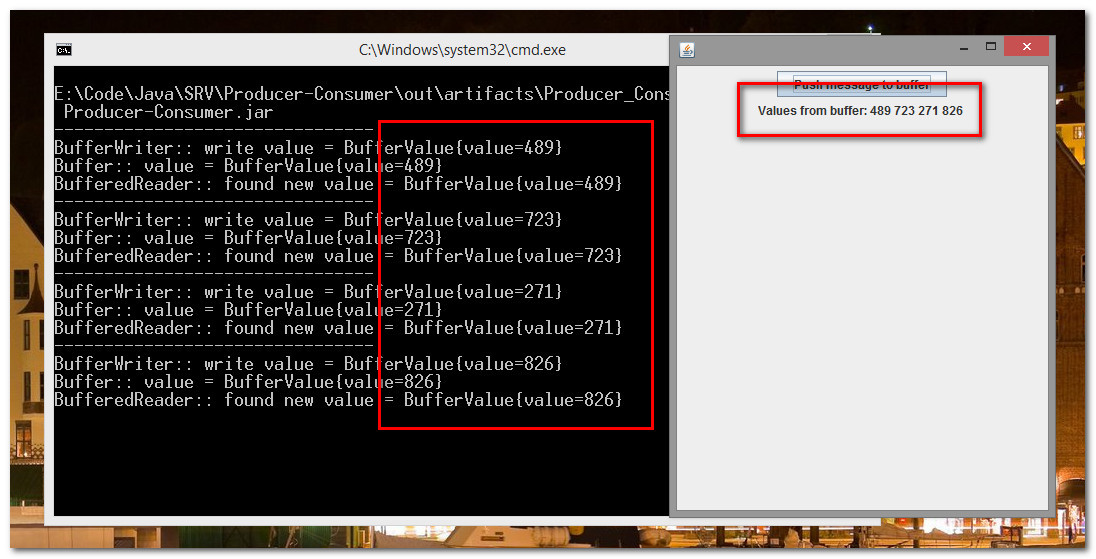


Рисунок . Вывод обработанных сообщений

Для завершения программы достаточно нажать на крестик в правом верхнем углу главного окна.

# Результаты

В результате выполнения работы была разработана программа, которая состоит из двух модулей: передатчика, имитирующего аппаратное обеспечение СРВ и приемника - потребителя.