|  |
| --- |
|  |
| МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ |
| Федеральное государственное бюджетное образовательное учреждение высшего образования  **"МИРЭА - Российский технологический университет"**  **РТУ МИРЭА** |
| Институт информационных технологий (ИТ) |
| Кафедра … |

|  |  |
| --- | --- |
| **ОТЧЕТ**  **ПО ПРАКТИЧЕСКОЙ РАБОТЕ №1** | |
| **по дисциплине** |  |
| **«Программирование на языке Java»**  **Тема: Многопоточные приложения, как основа распределённых клиент-серверных вычислений** | |
| Выполнил студент группы ИКБО-20-19 Анваржонов Ж Т |  |
| Принял ассистент кафедры ИиППО | Русляков А.А. |

|  |  |  |
| --- | --- | --- |
| Практические работы выполнены | « 28 » 09 2021г. |  |
| «Зачтено» | « » 2021 г. |  |

Москва 2020

**Теоретическое введение**

Цель: Ознакомится с моделью многопоточного программирования в Java.

**Постановка задачи**

Используя материалы данной практической работы необходимо написать многопоточную программу, в которой два потока записывают строку в стандартный вывод, по образцу PING PONG PING PONG PING PONG. Программа должна работать следующим образом:

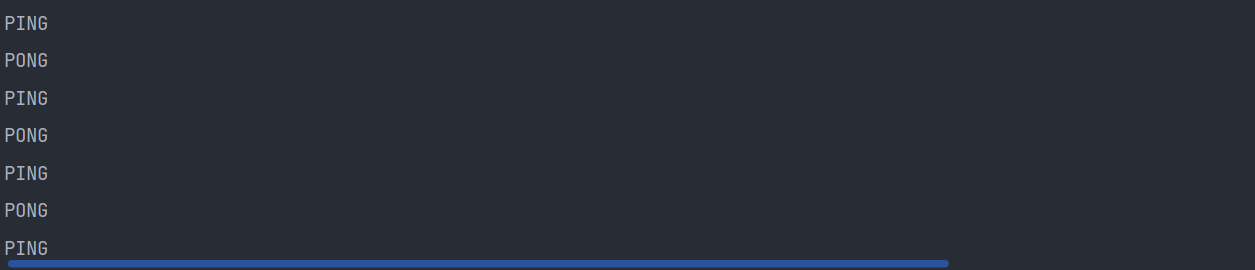
* 1-й поток печатает «Ping» и переходит в состояние ожидания.
* 2-й поток выходит из состояния ожидания, печатает «Pоng», уведомляет 1-й поток, возвращается в состояние ожидания.
* 1-й поток выходит из состояния ожидания, печатает «Pшng», уведомляет 2-й поток, возвращается в состояние ожидания.
* Шаги 2 и 3 повторяются и печатают «Ping Pong».

Программа должна быть реализована только с использованием Wait Notify, либо ReentrantLock.

**Программный код**

*public class* PingPong {  
  
 *private* String last = "PONG";  
  
 *private void* action (String message) {  
  
 *synchronized* (*this*) {  
 *while* (*true*) {  
 *if* (last.equals(message)) {  
 wait(*this*);  
  
 } *else* {  
 System.out.println(message);  
 last = message;  
 *sleep*(1000);  
 notifyAll();  
 }  
 }  
 }  
 }  
  
 *public static void* main(String[] args) {  
 PingPong pingPong = *new* PingPong();  
 *new* Thread( () -> pingPong.action("PING")).start();  
 *new* Thread( () -> pingPong.action("PONG")).start();  
 }  
  
 *private static void* sleep(*long* mills) {  
 *try* {  
 Thread.*sleep*(mills);  
 }*catch* (InterruptedException e){  
 e.printStackTrace();  
 }  
 }  
  
 *private void* wait(Object object) {  
 *try* {  
 object.wait();  
 }*catch* (InterruptedException e){  
 e.printStackTrace();  
 }  
 }  
}

**Вывод программы**



**Вывод**

В результате выполнения написали многопоточную программу, в которой два потока записывают строку в стандартный вывод, по образцу PING PONG PING PONG PING PONG.