|  |  |  |
| --- | --- | --- |
|  |  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |

Институт Информационных технологий

Кафедра Инструментального и прикладного программного обеспечения

**ПРАКТИЧЕСКАЯ РАБОТА №1**

**«Многопоточные приложения, как основа распределённых клиент-серверных вычислений»**

по дисциплине «Архитектура клиент-серверных приложений»

**Тема практической работы:**

**Студент группы** ИКБО-10-19 Полетаев Владимир Викторович

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

(подпись студента)

**Руководитель практической работы** преподаватель Степанов П. В.

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

(подпись руководителя)

Работа представлена «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_ 2021 г.

Допущен к работе «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2021 г.

Москва 2021

# **Цель работы**

Ознакомиться 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»

# **Решение**

Были созданы 2 потока, реализующие постановку задачи. Для качественного функционирования главные куски кода обоих потоков были обрамлены методом synchronized с общим монитором. Как только поток выводит на экран нужную надпись, он пытается «разбудить» другой поток, после чего сам засыпает посредством метода wait(). Исходный код:

public class Main {  
  
 public static void main(String[] args) {  
 Object object = new Object();  
 new Thread(() -> {  
 synchronized (object){  
 for(int i = 0;i <10 ; i++){  
 System.*out*.println("Ping");  
 object.notify();  
 try {  
 object.wait();  
 } catch (InterruptedException e) {  
 e.printStackTrace();  
 }  
 }  
 }  
 }).start();  
  
 new Thread(() -> {  
 synchronized (object) {  
 for(int i = 0; i<10; i++) {  
 System.*out*.println("Pong");  
 object.notify();  
 try {  
 object.wait();  
 } catch (InterruptedException e) {  
 e.printStackTrace();  
 }  
 }  
 }  
 }).start();  
 }  
  
}

# **Тестирование**

Запуск класса Main:

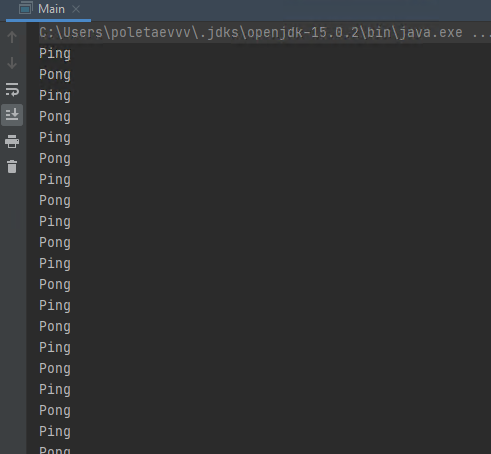


Рис. 9 Скриншот тестирования

1. **Вывод**

В результате выполнения поставленной задачи были получены навыки для работы с моделью многопоточного программирования в Java.