Частное учреждение образования

«Колледж бизнеса и права»

Лабораторная работа № 9

Учащийся Е. А. Чужавко

группа: Т-992

2022

**Задание 1.**

Выведете состояние потока перед его запуском, после запуска и во время выполнения.

Результат с выполнением программы показан на рисунке 1.

package lr9;

public class Z1 {

public static void main(String[] args) {

SClass thread=new SClass();

Z1.CheckState(thread.getState());

thread.start();

try {

thread.join();

} catch (InterruptedException e) {

System.out.println(e.getMessage());

}

Z1.CheckState(thread.getState());

}

public static void CheckState(Thread.State st) {

if(st==Thread.State.NEW) {

System.out.println("Состояние New");

}

if(st==Thread.State.RUNNABLE) {

System.out.println("Состояние Runnable");

}

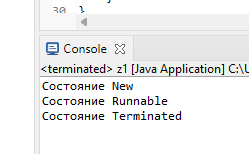
if(st==Thread.State.TERMINATED) {

System.out.println("Состояние Terminated");

}

}

}



*Рисунок 1.*

**Задание 2.**

Напишите программу, в которой создаются два потока, которые выводят на консоль своё имя по очереди.

Результат с выполнением программы показан на рисунке 2.

package lr9;

public class Z2 {

public static void main(String[] args) throws InterruptedException {

Object obj=new Object();

SomeClass sc1 = new SomeClass(obj);

SomeClass sc2 = new SomeClass(obj);

sc1.start();

sc2.start();

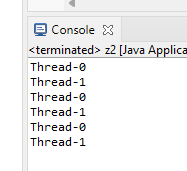
Thread.sleep(1000);

sc1.stop();

sc2.stop();

}

}



*Рисунок 2.*

**Задание 3.**

Дано два потока — производитель и потребитель. Производитель генерирует некоторые данные (в примере — числа). Производитель «потребляет» их. Два потока разделяют общий буфер данных, размер которого ограничен. Если буфер пуст, потребитель должен ждать, пока там появятся данные. Если буфер заполнен полностью, производитель должен ждать, пока потребитель заберёт данные и место освободится.

Результат с выполнением программы показан на рисунке 3.

package lr9;

import java.util.ArrayList;

public class Z3\_1 {

public static void main(String[] args) {

ArrayList<Integer> products=new ArrayList();

Thread Producer=new Thread(new Runnable() {

public void run() {

for(int i =0;i<3;i++) {

products.add(1);

System.out.println("Производитель добавил товар\nТоваров на складе: "+products.size());

}

while(true) {

try {

Thread.sleep(2500);

} catch (InterruptedException e) {

e.printStackTrace();

}

products.add(1);

System.out.println("Производитель добавил товар\nТоваров на складе: "+products.size());

}

}

});

Thread Customer=new Thread(new Runnable() {

public void run() {

while(true) {

try {

Thread.sleep(1300);

} catch (InterruptedException e) {

e.printStackTrace();

}

if(!products.isEmpty())

{

products.remove(0);

System.out.println("Покупатель купил товар\nТоваров на складе: "+products.size());

}

else {

System.out.println("Товаров на складе нет");

}

}

}

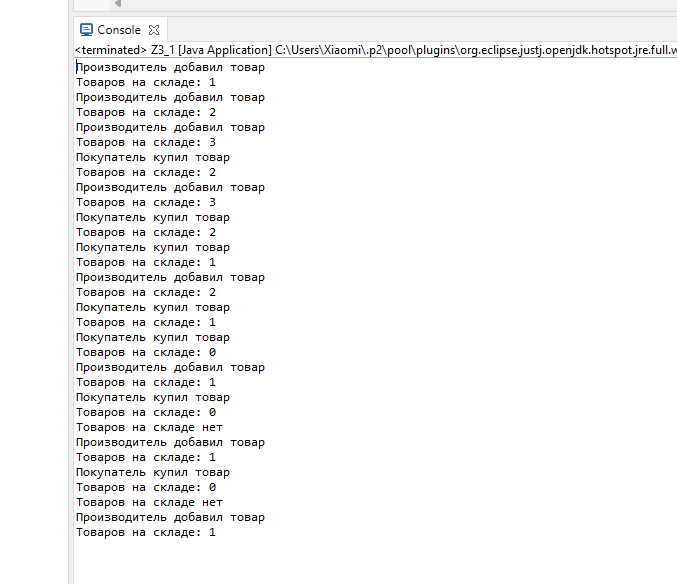
});

Producer.start();

Customer.start();

}

}



*Рисунок 3.*