Задание 1

import java.awt.EventQueue;

import javax.swing.JFrame;

import javax.swing.JPanel;

import javax.swing.border.EmptyBorder;

import javax.swing.JLabel;

import javax.swing.JTextField;

import java.awt.Font;

import javax.swing.JToggleButton;

import javax.swing.JProgressBar;

import javax.swing.JScrollPane;

import javax.swing.JToolBar;

import javax.swing.JSlider;

import javax.swing.JTextArea;

import java.awt.Color;

import java.awt.SystemColor;

import javax.swing.JButton;

import java.awt.event.ActionListener;

import java.io.FileWriter;

import java.io.IOException;

import java.awt.event.ActionEvent;

import java.lang.String;

public class form extends JFrame {

private JPanel contentPane;

static JTextField textField;

static JTextField textField\_1;

static JSlider slider;

static JSlider slider\_1;

static JTextArea textArea;

static boolean[] flag = {false, false};

static int turn = 0;

static int N = 100;

static int j = 0;

static int i = 1;

static int num = 0;

static int stopPoint = 0;

static int numAllDigit=0;

static int[] mas = new int[100];

static int sleepProd = 2000;

static int sleepCons = 2000;

/\*\*

\* Launch the application.

\*/

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

/\*Thread forForm = createFrame();

forForm.start();

try {

forForm.join();

} catch (InterruptedException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}\*/

Thread t1 = new Thread(new Runnable() {

@Override

public void run() {

try {

form frame = new form();

frame.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

t1.start();

}

});

}

/\*\*

\* Create the frame.

\*/

public form() {

setBackground(SystemColor.inactiveCaption);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setBounds(100, 100, 457, 300);

contentPane = new JPanel();

contentPane.setBackground(SystemColor.activeCaption);

contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(null);

JLabel lblNewLabel = new JLabel("Записывает");

lblNewLabel.setFont(new Font("Tahoma", Font.PLAIN, 14));

lblNewLabel.setBounds(23, 13, 114, 14);

contentPane.add(lblNewLabel);

textField = new JTextField();

textField.setFont(new Font("Tahoma", Font.PLAIN, 14));

textField.setBounds(23, 38, 27, 20);

contentPane.add(textField);

textField.setColumns(10);

JLabel lblNewLabel\_1 = new JLabel("Читает");

lblNewLabel\_1.setFont(new Font("Tahoma", Font.PLAIN, 14));

lblNewLabel\_1.setBounds(222, 13, 105, 14);

contentPane.add(lblNewLabel\_1);

textField\_1 = new JTextField();

textField\_1.setFont(new Font("Tahoma", Font.PLAIN, 14));

textField\_1.setBounds(222, 38, 80, 20);

contentPane.add(textField\_1);

textField\_1.setColumns(10);

slider = new JSlider(100,4000);

slider.setBounds(12, 69, 200, 22);

contentPane.add(slider);

slider\_1 = new JSlider(100,4000);

slider\_1.setBounds(222, 69, 200, 22);

contentPane.add(slider\_1);

textArea = new JTextArea();

textArea.setFont(new Font("Monospaced", Font.PLAIN, 12));

textArea.setToolTipText("");

textArea.setBounds(15, 134, 400, 100);

contentPane.add(textArea);

//JScrollPane scrollPane = new JScrollPane(textArea);

//textArea.setEditable(false);

JButton btnNewButton = new JButton("GO");

btnNewButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

Thread t1 = new Thread(new Runnable() {

@Override

public void run() {

turn = 0;

N =500;

j = 0;

i = 1;

num = 0;

numAllDigit=0;

stopPoint = 0;

mas = new int[500];

try {

//log("Starting 2 processes (threads) ...");

Thread p0 = producer();

Thread p1 = consumer();

p0.start();

p1.start();

p0.join();

p1.join();

}

catch (InterruptedException e1) {}

}

});

t1.start();

/\*try {

//log("Starting 2 processes (threads) ...");

Thread p0 = producer();

Thread p1 = consumer();

p0.start();

p1.start();

p0.join();

p1.join();

}

catch (InterruptedException e1) {}\*/

}

});

btnNewButton.setFont(new Font("Tahoma", Font.PLAIN, 15));

btnNewButton.setBounds(162, 102, 87, 23);

contentPane.add(btnNewButton);

}

// 1. I want to enter CS.

// 2. Its the other process turn

// 3. I wait if you want too and your turn.

// 4. I enter CS (sleep).

// 5. I am done with CS.

static Thread producer() {

return new Thread(() -> {

//int j = 0;//так как это производитель

for (int n=0; n<N; n++) {

// log(j+": want CS"+n); // LOCK

flag[j] = true; // 1

turn = i; // 2

while (flag[i] && turn == i) Thread.yield(); // 3

// log(j+": in CS"+n);

////

sleep(50);

if(num % 3 != 0) {

mas[numAllDigit]=num;

textField.setText(Integer.toString(num));

numAllDigit = numAllDigit+1;

writeFile();

writeArea();

}

num++;

////

/\* sleepProd = 4100-slider.getValue();

sleep(sleepProd);\*/ // 4 тут сделать вывод символа в файл()сначала в носоль

//log(j+": done CS"+n); // UNLOCK

flag[j] = false; // 5

sleepProd = 4100-slider.getValue();

sleep(sleepProd);

}

});

}

static Thread consumer() {

return new Thread(() -> {

for (int n=0; n<N; n++) {

// log(i+": want CS"); // LOCK

flag[i] = true; // 1

turn = j; // 2

while (flag[j] && turn == j) Thread.yield(); // 3

String str="";

// log(i+": in CS"+n);

sleep(50);

if (numAllDigit>0) {

for (int i=stopPoint;i<numAllDigit;i++) {

if (mas[i] % 2 ==0) {

//textField\_1.setText(Integer.toString(mas[i]));

str = str+Integer.toString(mas[i])+" ";

for (int j=i; j<numAllDigit;j++) {

mas[j]=mas[j+1];

}

i--;

numAllDigit--;

stopPoint=i;

break;

}

}

/////////

textField\_1.setText(str);

writeFile();

writeArea();

}

//

/\*sleepCons = 4100-slider\_1.getValue();

sleep(sleepCons); \*/// 4

// log(i+": done CS"); // UNLOCK

flag[i] = false; // 5

sleepCons = 4100-slider\_1.getValue();

sleep(sleepCons);

}

});

}

public static void writeFile() {

try (final FileWriter writer = new FileWriter("fileDigit.txt", false))

{

for (int i = 0; i < numAllDigit; ++i)

{

final String s = Integer.toString(mas[i]);

writer.write(s);

writer.write(" ");

// writer.write(System.lineSeparator());

System.out.println(s);

}

}

catch(IOException e) {

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

}

}

public static void writeArea() {

String str = "";

for (int i = 0; i < numAllDigit; ++i)

{

str = str+Integer.toString(mas[i])+" ";

}

textArea.setText(str);

}

static void sleep(double t) {//time for timer

try { Thread.sleep((long)t); }

catch (InterruptedException e) {}

}

static void log(String x) {//HAVE to change

System.out.println(x);

}

}

Задание 2

package smokers;

import java.util.concurrent.Semaphore;

import java.util.concurrent.locks.Condition;

import java.util.concurrent.locks.Lock;

import java.util.concurrent.locks.ReentrantLock;

import java.awt.EventQueue;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.ImageIcon;

import javax.swing.JButton;

import java.awt.event.ActionListener;

import java.awt.event.ActionEvent;

import javax.swing.JSlider;

public class formSmoke {

JFrame frame;

static boolean[] table = new boolean[3];

static boolean tobacco = table[0];

static boolean paper = table[1];

static boolean matches = table[2];

static boolean push = true;

static int f=1;

static final Lock lock = new ReentrantLock();

static final Condition pusher = lock.newCondition();

static final Condition smokerA = lock.newCondition();

static final Condition smokerB = lock.newCondition();

static final Condition smokerC = lock.newCondition();

static JLabel lgreen;

static JLabel lblue;

static JLabel lred;

static JLabel Ltabak;

static JLabel Lpaper;

static JLabel Lfire;

static JSlider slider;

static JSlider slider\_1;

static JSlider slider\_2;

static int sleep =1000;

static JLabel greenMes;

static JLabel blueMes;

static JLabel redMes;

static final Semaphore sem = new Semaphore(2);

static final Object mutex = new Object();

private final JButton btnNewButton = new JButton("Go");

/\*\*

\* Launch the application.

\*/

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

formSmoke window = new formSmoke();

window.frame.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the application.

\*/

public formSmoke() {

initialize();

}

/\*\*

\* Initialize the contents of the frame.

\*/

private void initialize() {

frame = new JFrame();

frame.setBounds(100, 100, 535, 371);

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.getContentPane().setLayout(null);

lgreen = new JLabel("");

lgreen.setIcon(new ImageIcon("C:\\\\Users\\\\User\\\\Desktop\\\\Лабы МОИ\\\\lab2\\\\Задание 2\\\\зеленый!!.png"));

lgreen.setBounds(26, 82, 90, 140);

frame.getContentPane().add(lgreen);

lblue = new JLabel("");

lblue.setIcon(new ImageIcon("C:\\\\Users\\\\User\\\\Desktop\\\\Лабы МОИ\\\\lab2\\\\Задание 2\\\\голубой.png"));

lblue.setBounds(279, 59, 63, 119);

frame.getContentPane().add(lblue);

lred = new JLabel("");

lred.setIcon(new ImageIcon("C:\\\\Users\\\\User\\\\Desktop\\\\Лабы МОИ\\\\lab2\\\\Задание 2\\\\красный.png"));

lred.setBounds(436, 76, 73, 164);

frame.getContentPane().add(lred);

Ltabak = new JLabel("");

Ltabak.setIcon(new ImageIcon("C:\\\\Users\\\\User\\\\Desktop\\\\Лабы МОИ\\\\lab2\\\\Задание 2\\\\табак.png"));

Ltabak.setBounds(55, 211, 100, 65);

frame.getContentPane().add(Ltabak);

Lpaper = new JLabel("");

Lpaper.setIcon(new ImageIcon("C:\\\\Users\\\\User\\\\Desktop\\\\Лабы МОИ\\\\lab2\\\\Задание 2\\\\бумага.png"));

Lpaper.setBounds(201, 59, 94, 68);

frame.getContentPane().add(Lpaper);

Lfire = new JLabel("");

Lfire.setIcon(new ImageIcon("C:\\\\Users\\\\User\\\\Desktop\\\\Лабы МОИ\\\\lab2\\\\Задание 2\\\\зажигалка.png"));

Lfire.setBounds(388, 172, 73, 48);

frame.getContentPane().add(Lfire);

greenMes = new JLabel("");

greenMes.setBounds(20, 50, 100, 14);

frame.getContentPane().add(greenMes);

blueMes = new JLabel("");

blueMes.setBounds(300, 10, 100, 14);

frame.getContentPane().add(blueMes);

redMes = new JLabel("");

redMes.setBounds(450, 25, 100, 14);

frame.getContentPane().add(redMes);

JLabel lblNewLabel = new JLabel("");

lblNewLabel.setIcon(new ImageIcon("C:\\\\Users\\\\User\\\\Desktop\\\\Лабы МОИ\\\\lab2\\\\Задание 2\\\\roomLittle.jpg"));

lblNewLabel.setBounds(0, 0, 525, 287);

frame.getContentPane().add(lblNewLabel);

btnNewButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

greenMes.setText(" ");

blueMes.setText(" ");

redMes.setText(" ");

new Thread(new SmokerA()).start();

new Thread(new SmokerB()).start();

new Thread(new SmokerC()).start();

new Thread(new Pusher()).start();

//int i = 0;

//while (i<5) {

/\* new Thread(new test()).start();

new Thread(new test2()).start();

new Thread(new test2()).start();

new Thread(new test()).start();

// i++;

//}\*/

}

});

btnNewButton.setBounds(409, 286, 100, 31);

frame.getContentPane().add(btnNewButton);

slider = new JSlider(1,20);

slider.setBounds(0, 286, 120, 26);

frame.getContentPane().add(slider);

slider\_1 = new JSlider(1,20);

slider\_1.setBounds(130, 287, 130, 26);

frame.getContentPane().add(slider\_1);

slider\_2 = new JSlider(1,20);

slider\_2.setBounds(270, 286, 130, 26);

frame.getContentPane().add(slider\_2);

JLabel lblNewLabel\_1 = new JLabel("\u0417\u0435\u043B\u0435\u043D\u044B\u0439");

lblNewLabel\_1.setBounds(35, 311, 46, 14);

frame.getContentPane().add(lblNewLabel\_1);

JLabel lblNewLabel\_2 = new JLabel("\u0421\u0438\u043D\u0438\u0439");

lblNewLabel\_2.setBounds(175, 311, 46, 14);

frame.getContentPane().add(lblNewLabel\_2);

JLabel lblNewLabel\_3 = new JLabel("\u041A\u0440\u0430\u0441\u043D\u044B\u0439");

lblNewLabel\_3.setBounds(314, 311, 46, 14);

frame.getContentPane().add(lblNewLabel\_3);

}

}

Курильщик 1

package smokers;

import java.util.Random;

import static smokers.formSmoke.\*;

public class Pusher implements Runnable {

private Random rnd = new Random();

@Override

public void run() {

while (true) {

try {

lock.lock();

//if (push) {

while ((tobacco && paper) || (tobacco && matches) || (paper && matches)) {

try {

pusher.await();

//sem.acquire();

} catch (InterruptedException e) {

e.printStackTrace();

}

}

//

/\* try {

sem.acquire();

} catch (InterruptedException e) {

// TODO Auto-generated catch block

e.printStackTrace();\*/

//}

//

int temp = rnd.nextInt(slider.getValue()+slider\_1.getValue()+slider\_2.getValue());

if (temp <slider.getValue()) {

tobacco = false;

paper = true;

matches = true;

System.out.println("Pusher is pushing paper and matches");

Ltabak.setBounds(55, 211, 100, 65);

Lpaper.setBounds(248, 172, 94, 68);

Lfire.setBounds(315, 192, 73, 48);

pushing();

smokerA.signal();

}

if (temp >=slider.getValue() && temp<=slider.getValue()+slider\_1.getValue()) {

tobacco = true;

paper = false;

matches = true;

System.out.println("Pusher is pushing tobacco and matches");

Ltabak.setBounds(148, 189, 100, 65);

Lpaper.setBounds(201, 59, 94, 68);

Lfire.setBounds(315, 192, 73, 48);

pushing();

smokerB.signal();

}

if (temp >slider.getValue()+slider\_1.getValue()) {

tobacco = true;

paper = true;

matches = false;

System.out.println("Pusher is pushing tobacco and paper");

Ltabak.setBounds(148, 189, 100, 65);

Lpaper.setBounds(248, 172, 94, 68);

Lfire.setBounds(388, 172, 73, 48);

pushing();

smokerC.signal();

}

// push=false;

//sem.release();

//turn(temp);

} finally {

lock.unlock();

//sem.release();

}

}

}

private void pushing() {

try {

sleep = 6100-slider.getValue();

Thread.sleep(1000);

} catch (InterruptedException e) {

e.printStackTrace();

}

}

private void turn(int temp) {

if (temp == 0) {new Thread(new SmokerA()).start(); }

if (temp == 1) {new Thread(new SmokerB()).start(); }

if (temp == 2) {new Thread(new SmokerC()).start(); }

}

private void makingAndSmokingSigarette(int sleep) {

try {

//sleep = 6100-slider.getValue();

Thread.sleep(sleep);

} catch (InterruptedException e) {

e.printStackTrace();

}

}

}

Курильщик 1

package smokers;

import java.util.Random;

import static smokers.formSmoke.\*;

public class SmokerA implements Runnable {

private Random rnd = new Random();

@Override

public void run() {

while (true) {

try {

//makingAndSmokingSigarette(200);

//greenMes.setText("Хочет");

lock.lock();

greenMes.setText("Хочет курить");

while (!paper && !matches) {

try {

smokerA.await();

//sem.acquire();

} catch (InterruptedException e) {

e.printStackTrace();

}

}

/\*if (paper && matches) {

try {

sem.acquire();

} catch (InterruptedException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}\*/

int temp = rnd.nextInt(2);

if( temp == 0) {

greenMes.setText("Курит");

System.out.println("SmokerA are making and smoking cigarette");

paper = false;

matches = false;

tobacco = false;

Ltabak.setBounds(55, 211, 100, 65);

Lpaper.setBounds(102, 154, 94, 68);

Lfire.setBounds(102, 95, 73, 48);

//sleep = 6100-slider.getValue();

makingAndSmokingSigarette(1000);}

else {

paper = false;

matches = false;

tobacco = false;

greenMes.setText("Не хочет курить");

System.out.println("SmokerA don`want to smoke");

Ltabak.setBounds(55, 211, 100, 65);

Lpaper.setBounds(102, 154, 94, 68);

Lfire.setBounds(102, 95, 73, 48);

// sleep = 6100-slider.getValue();

makingAndSmokingSigarette(1000);

}

greenMes.setText("");

pusher.signal();

//sleep = 6100-slider.getValue();

//makingAndSmokingSigarette(sleep);

//sem.release();

//turn();

//}

} finally {

lock.unlock();

// sleep = 6100-slider.getValue();

// makingAndSmokingSigarette(sleep);

//sem.release();

}

}

}

private void makingAndSmokingSigarette(int sleep) {

try {

//sleep = 6100-slider.getValue();

Thread.sleep(sleep);

} catch (InterruptedException e) {

e.printStackTrace();

}

}

private void turn() {

new Thread(new Pusher()).start();

}

}

Курильщик 2

package smokers;

import static smokers.formSmoke.\*;

import java.util.Random;

public class SmokerB implements Runnable {

private Random rnd = new Random();

@Override

public void run() {

while (true) {

try {

//makingAndSmokingSigarette(200);

//blueMes.setText("Хочет");

lock.lock();

blueMes.setText("Хочет курить");

while (!tobacco && !matches) {

try {

smokerB.await();

//sem.acquire();

} catch (InterruptedException e) {

e.printStackTrace();

}

}

int temp = rnd.nextInt(2);

if( temp == 0) {

blueMes.setText("Курит");

System.out.println("SmokerB are making and smoking cigarette");

tobacco = false;

matches = false;

paper = false;

Ltabak.setBounds(185, 118, 100, 65);

Lpaper.setBounds(201, 59, 94, 68);

Lfire.setBounds(201, 11, 73, 48);

//sleep = 6100-slider\_1.getValue();

makingAndSmokingSigarette(1000);}

else {

blueMes.setText("Не хочет курить");

System.out.println("SmokerB dont want to smoke");

tobacco = false;

matches = false;

paper = false;

Ltabak.setBounds(185, 118, 100, 65);

Lpaper.setBounds(201, 59, 94, 68);

Lfire.setBounds(201, 11, 73, 48);

// sleep = 6100-slider\_1.getValue();

makingAndSmokingSigarette(1000);

}

blueMes.setText("");

pusher.signal();

//sleep = 6100-slider\_1.getValue();

// makingAndSmokingSigarette(1000);

//sem.release();

// turn();

} finally {

lock.unlock();

//sleep = 6100-slider\_1.getValue();

//makingAndSmokingSigarette(sleep);

//sem.release();

}

}

}

private void makingAndSmokingSigarette(int sleep) {

try {

//sleep = 6100-slider\_1.getValue();

Thread.sleep(sleep);

} catch (InterruptedException e) {

e.printStackTrace();

}

}

private void turn() {

new Thread(new Pusher()).start();

}

}

Курильщик 3

package smokers;

import static smokers.formSmoke.\*;

import java.util.Random;

public class SmokerC implements Runnable {

private Random rnd = new Random();

@Override

public void run() {

while (true) {

try {

//makingAndSmokingSigarette(200);

// redMes.setText("Хочет");

lock.lock();

redMes.setText("Хочет курить");

while (!tobacco && !paper) {

try {

smokerC.await();

//sem.acquire();

} catch (InterruptedException e) {

e.printStackTrace();

}

}

int temp = rnd.nextInt(2);

if( temp == 0) {

redMes.setText("Курит");

System.out.println("SmokerC are making and smoking cigarette");

Ltabak.setBounds(339, 113, 100, 65);

Lpaper.setBounds(367, 45, 94, 68);

Lfire.setBounds(388, 172, 73, 48);

//sleep = 6100-slider\_2.getValue();

makingAndSmokingSigarette(1000);

tobacco = false;

paper = false;

matches = false;}

else {

redMes.setText("Не хочет курить");

System.out.println("SmokerC don`t want to smoke");

Ltabak.setBounds(339, 113, 100, 65);

Lpaper.setBounds(367, 45, 94, 68);

Lfire.setBounds(388, 172, 73, 48);

// sleep = 6100-slider\_2.getValue();

makingAndSmokingSigarette(1000);

tobacco = false;

paper = false;

matches = false;

}

//sem.release();

//turn();

redMes.setText("");

pusher.signal();

} finally {

lock.unlock();

//sleep = 6100-slider\_2.getValue();

//makingAndSmokingSigarette(sleep);

//sem.release();

}

}

}

private void makingAndSmokingSigarette(int sleep) {

try {

Thread.sleep(sleep);

} catch (InterruptedException e) {

e.printStackTrace();

}

}

private void turn() {

new Thread(new Pusher()).start();

}

}