ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«ПОВОЛЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ И ИНФОРМАТИКИ»

Факультет 1

Кафедра программной инженерии

**ОТЧЕТ**

**ПО ЛАБОРАТОРНОЙ РАБОТЕ №** 1

по дисциплине Операционные системы и оболочки

|  |  |  |
| --- | --- | --- |
| **ВЫПОЛНИЛИ** | | |
| студенты | РПИС-11  (группа) | Гареев Ф. Ф. и Яковлев П. Н.  (ФИО) |
|  | **ПРОВЕРИЛ** |  |
| (должность) | | Баженов А. Э. и Малахов С. В.  (ФИО) |

Самара

2023

**Цель работы:**

Научиться работать с потоками.

**Задание:**

Вариант 5.Заполнить три «сосуда» емкостью 1000мл. Каждый раз произвольно заполняются: 1 сосуд на 5мл, 2 сосуд на 7мл и 3 сосуд на 10мл. Каждый сосуд обрабатывает отдельный поток.

**Листинг программы:**

package com.lab.lab1guiv1;  
  
import javafx.application.Application;  
import javafx.scene.Scene;  
import javafx.scene.control.\*;  
import javafx.scene.layout.\*;  
import javafx.stage.Stage;  
import javafx.geometry.Pos;  
import javafx.event.ActionEvent;  
import javafx.event.EventHandler;  
import javafx.application.Platform;  
import java.util.concurrent.Exchanger;  
  
import java.io.IOException;  
import java.util.Random;  
  
public class HelloApplication extends Application {  
 VBox root,groupProgBar, grouplbl, grouplbl2;  
 HBox groubbarlbllbl2, groupfill, grouplbls;  
 MenuBar menuBar;  
 MenuItem menuPrior, menuPrior2, menuPrior3, apply;  
 Label barStatus, barStatus2, barStatus3;  
 Button btnFill, btnDrain;  
 ProgressBar progressBar, progressBar2, progressBar3;  
 Slider priority, priority2, priority3;  
  
 int[] cups = {5, 7, 10};  
 int[] prior = {2, 6, 8};  
  
 @Override  
 public void start(Stage stage) throws IOException {  
  
  
 //Group root = new Group();  
 priority = new Slider(1,9,0);  
 priority2 = new Slider(1,9,0);  
 priority3 = new Slider(1,9,0);  
  
 menuPrior = new MenuItem("Thread 1", priority);  
 menuPrior2 = new MenuItem("Thread 2", priority2);  
 menuPrior3 = new MenuItem("Thread 3", priority3);  
 apply = new MenuItem("Apply changes");  
 menuBar = new MenuBar(new Menu("Threads priority", root, menuPrior, menuPrior2, menuPrior3, apply));  
  
  
 Label barlbl = new Label("Bar 1:\t");  
 Label bar2lbl = new Label("Bar 2:\t");  
 Label bar3lbl = new Label("Bar 3:\t");  
  
 barStatus = new Label("0");  
 barStatus2 = new Label("0");  
 barStatus3 = new Label("0");  
  
  
 btnFill = new Button("Fill");  
 btnDrain = new Button("Drain");  
  
 Thread.*currentThread*().setPriority(10);  
  
  
 progressBar = new ProgressBar(1000);  
 progressBar.setMinSize(450,50);  
 progressBar2 = new ProgressBar(1000);  
 progressBar2.setMinSize(450,50);  
 progressBar3 = new ProgressBar(1000);  
 progressBar3.setMinSize(450,50);  
 progressBar.setProgress(0);  
 progressBar2.setProgress(0);  
 progressBar3.setProgress(0);  
  
 groupProgBar = new VBox(progressBar, progressBar2, progressBar3);  
 groupProgBar.setSpacing(10);  
 groupProgBar.setMinSize(470, 190);  
 groupProgBar.setAlignment(Pos.*BOTTOM\_CENTER*);  
  
 groupfill = new HBox(btnFill, btnDrain);  
 groupfill.setAlignment(Pos.*CENTER*);  
 groupfill.setMinHeight(50);  
 groupfill.setSpacing(50);  
  
 grouplbl = new VBox(barlbl, bar2lbl, bar3lbl);  
 grouplbl.setAlignment(Pos.*CENTER*);  
 grouplbl.setMinWidth(30);  
 grouplbl.setMinHeight(190);  
 grouplbl.setSpacing(45);  
  
 grouplbl2 = new VBox(barStatus, barStatus2, barStatus3);  
 grouplbl2.setAlignment(Pos.*CENTER*);  
 grouplbl2.setMinHeight(190);  
 grouplbl2.setSpacing(45);  
  
 grouplbls = new HBox(grouplbl, grouplbl2);  
 groubbarlbllbl2 = new HBox(groupProgBar, grouplbls);  
 groubbarlbllbl2.setAlignment(Pos.*BOTTOM\_CENTER*);  
  
  
 Exchanger ex = new Exchanger<Integer[]>();  
 AntiSortThread myAntiSort = new AntiSortThread(cups, ex);  
 Thread antiSort = new Thread(myAntiSort);  
  
 antiSort.setDaemon(true);  
  
 antiSort.start();  
  
  
 btnDrain.setDisable(true);  
  
  
 antiSort.setPriority(10);  
  
 btnFill.setOnAction(new EventHandler<ActionEvent>() {  
  
 @Override  
 public void handle(ActionEvent event) {  
  
   
 Thread thread = new MyThread(progressBar, barStatus, ex, 0, btnDrain);  
 thread.setPriority(prior[0]);  
 thread.setDaemon(true);  
 System.*out*.println(thread.getPriority());  
 thread.start();  
   
 Thread thread2 = new MyThread(progressBar2, barStatus2, ex, 1, btnDrain);  
 thread2.setPriority(prior[1]);  
 thread2.setDaemon(true);  
 System.*out*.println(thread2.getPriority());  
 thread2.start();  
   
 Thread thread3 = new MyThread(progressBar3, barStatus3, ex, 2, btnDrain);  
 thread3.setPriority(prior[2]);  
 thread3.setDaemon(true);  
 System.*out*.println(thread3.getPriority());  
 thread3.start();  
   
 btnFill.setDisable(true);  
  
  
 }  
 });  
 apply.setOnAction(new EventHandler<ActionEvent>() {  
 @Override  
 public void handle(ActionEvent actionEvent) {  
 prior[0] = (int) priority.getValue();  
 prior[1] = (int) priority2.getValue();  
 prior[2] = (int) priority3.getValue();  
 }  
 });  
  
 btnDrain.setOnAction(new EventHandler<ActionEvent>() {  
  
 @Override  
 public void handle(ActionEvent event) {  
  
 progressBar.setProgress(0);  
 progressBar2.setProgress(0);  
 progressBar3.setProgress(0);  
  
 barStatus.setText(String.*valueOf*(0));  
 barStatus2.setText(String.*valueOf*(0));  
 barStatus3.setText(String.*valueOf*(0));  
  
 btnFill.setDisable(false);  
 btnDrain.setDisable(true);  
 }  
 });  
  
  
  
 root = new VBox(menuBar, groubbarlbllbl2, groupfill);  
 Scene scene = new Scene(root, 600, 270);  
  
  
 stage.setResizable(false);  
 stage.setTitle("Hello!");  
 stage.setScene(scene);  
 stage.show();  
 }  
  
  
 public static void main(String[] args) {  
 *launch*();  
 }  
 void setPriority(Thread thread, Thread thread2, Thread thread3){  
 thread.setPriority((int) priority.getValue());  
 thread2.setPriority((int) priority2.getValue());  
 thread3.setPriority((int) priority3.getValue());  
 }  
}  
  
class MyThread extends Thread{  
 ProgressBar progressBar;  
 Label label;  
 double volume, speed;  
 Button button;  
 int ind;  
 int[] cups = {5, 7, 10};  
 Exchanger<int[]> exchanger;  
  
  
  
 public MyThread(ProgressBar progressBar, Label label, Exchanger<int[]> exchanger, int Ind, Button button){  
 this.ind = ind;  
 this.label = label;  
 this.progressBar = progressBar;  
 this.exchanger = exchanger;  
 this.button = button;  
  
  
 }  
  
 @Override  
 public void run() {  
 Runnable updater = new Runnable() {  
 @Override  
 public void run() {  
 progressBar.setProgress((volume += speed) / 1000);  
 label.setText(String.*valueOf*(volume));  
 }  
 };  
  
 while (volume < 1000) {  
  
 try {  
 cups = exchanger.exchange(cups);  
 speed = cups[ind];  
 Thread.*sleep*(50);  
  
 } catch (InterruptedException e) {  
  
  
 throw new RuntimeException(e);  
 }  
  
 Platform.*runLater*(updater);  
  
  
 }  
 button.setDisable(false);  
  
 }  
  
}  
class AntiSortThread extends Thread{  
  
  
 Exchanger<int[]> exchanger;  
 int[] cups = {5, 7, 10};  
  
 public AntiSortThread(int[] cups, Exchanger<int[]> ex){  
 exchanger = ex;  
 this.cups = cups;  
 }  
  
 @Override  
 public void run() {  
 Runnable updater = new Runnable() {  
 @Override  
 public void run() {  
 antisort();  
 }  
 };  
 while (true){  
 try {  
 Thread.*sleep*(100);  
 antisort();  
 exchanger.exchange(cups);  
 } catch (InterruptedException e) {  
 throw new RuntimeException(e);  
  
 }  
 Platform.*runLater*(updater);  
 }  
  
 }  
 public void antisort(){  
 Random random = new Random();  
 int buf;  
  
 for(int i = 0; i<cups.length; i++) {  
 int rand = random.nextInt(0, 2);  
 if (i != rand) {  
 buf = cups[i];  
 cups[i] = cups[rand];  
 cups[rand]=buf;  
 }  
 }  
 }  
}

Пример работы программы: 

