Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Інститут комп’ютерних наук та інформаційних технологій

Кафедра програмного забезпечення



**ЗВІТ**

**Про виконання лабораторної роботи № 1**

**з дисципліни «Програмне забезпечення мережевих технологій»**

**на тему**

**«Java Applets and Threads»**

**Лектор:**

професор кафедри ПЗ

Мельник Р. А.

**Виконав:**

студент групи ПІ-31

Губ’як Р. І.

**Прийняв:**

старший викладач кафедри ПЗ

Гасько Р. Т.

«\_\_\_» \_\_\_\_\_\_\_\_ 2017 р.

∑ = \_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_

Львів – 2017

**ТЕМА РОБОТИ**

Java Applets and Threads

**МЕТА РОБОТИ**

Освоїти елементи програмування мовою JAVA при розробленні аплетів, а також засвоїти класи і методи бібліотек java.awt, java.applet. Навчитись використовувати потоки для розпаралелювання виконання різних задач

**ІНДИВІДУАЛЬНЕ ЗАВДАННЯ**

1. Створити аплет, який рисуватиме Progress Bar.
2. До кожного Progress Bar прив’язати потік, який змінюватиме його значення з часом.

**ТЕКСТ ПРОГРАМИ**

**//--------------------------------------------------------Applet that paints Progress Bar**

public class Main extends Applet implements ActionListener {

private int[] Array;

private ArrayList<Station> Stations;

private boolean IsBusy;

private SearchListener searchListener;

private JSpinner spinner;

private Button Button;

private boolean Repaint;

public void init() {

Label spinnerLabel = new Label("Count of numbers to sort");

spinner = new JSpinner(new SpinnerNumberModel(10000000, 100, 1000000000, 100000));

Button = new Button();

Button.setLabel("Generate and sort");

Button.addActionListener(this);

this.add(spinnerLabel);

this.add(spinner);

this.add(Button);

searchListener = new SearchListener(2727);

searchListener.onFoundListeners.add((address, port) -> EstablishConnection(address, port));

Stations = new ArrayList<Station>();

}

public void paint(Graphics g) {

spinner.repaint();

for (int i = 0; i < Stations.size(); ++i){

Station station = Stations.get(i);

g.setColor(Color.BLACK);

g.drawString(station.Address.toString(), 10, 90 + 40 \* i);

ProgressBarDrawer.Draw(g, Stations.get(i).ProgressBar);

}

}

public void actionPerformed(ActionEvent e) {

Button.setEnabled(false);

Stations.clear();

repaint();

new Thread(() -> {

Array = ArrayGenerator.Generate((int) spinner.getValue());

Stations = StationFinder.FindAll();

for (int i = 0; i < Stations.size(); ++i) {

ProgressBar progressBar = Stations.get(i).ProgressBar;

progressBar.Left = 10;

progressBar.Top = 100 + 40 \* i;

progressBar.Width = this.getWidth() - progressBar.Left - 10;

progressBar.Height = 10;

}

Controller controller = new Controller(Array, Stations);

controller.ControllerListeners.add(new ControllerListener() {

public void OnFinish(int[] array) {

Array = array;

Button.setEnabled(true);

IsBusy = false;

}

public void OnProgressBarChanged() {

repaint();

}

});

controller.StartControl();

}).start();

}

private void EstablishConnection(InetAddress address, int port){

if (IsBusy){

return;

}

ArrayList<InetAddress> local = StationFinder.GetLocalIpAddresses();

if (local.contains(address)){

return;

}

Socket socket = null;

try{

socket = new Socket(address, port);

}

catch(Exception e){

}

IsBusy = true;

Worker worker = new Worker(socket.getInetAddress());

worker.WorkerListeners.add(new WorkerListener() {

public void OnFinish() {

IsBusy = false;

}

});

try {

socket.close();

}

catch(Exception e){

}

worker.DoWork();

}

}

**//-------Usage of threads and changing the value of Progress Bar via OnSearchBarChanged()**

private void ReceiveResults(){

new Thread() {

public void run() {

WorkerSocket = null;

try {

WorkerSocket = new ServerSocket(2727);

} catch (Exception e) {

}

while (WaitForResponse) {

Socket socket = null;

ResultMessage result = null;

try {

socket = WorkerSocket.accept();

result = new Gson().fromJson(new BufferedReader(new InputStreamReader(socket.getInputStream())).readLine(), ResultMessage.class);

} catch (Exception e) {

continue;

}

Station station = null;

for (Station item : Stations) {

if (item.Address.equals(socket.getInetAddress())) {

station = item;

break;

}

}

try {

socket.close();

}

catch(Exception e){

}

if (result.Array.length == 0) {

if (station != null) {

if (result.Progress > station.ProgressBar.GetProgress()) {

station.ProgressBar.SetProgress(result.Progress);

}

if (result.Progress == 100) {

SendMergeClient(station);

}

OnProgressBarChanged();

}

} else {

Merge(result.Array);

}

}

}

}.start();

}

**ПРОТОКОЛ ВИКОНАННЯ ПРОГРАМИ**

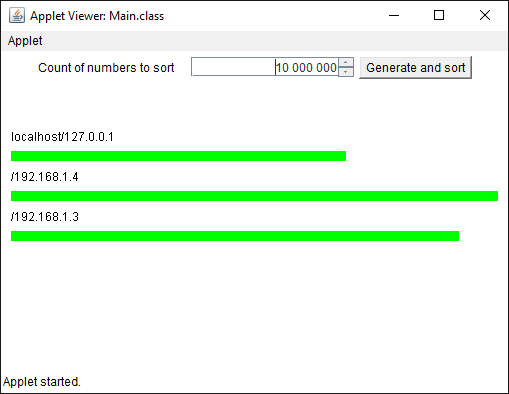


Рис. 1. Рисування Progress Bar у Applet з використанням Threads

**ВИСНОВОК**

Під час виконання цієї лабораторної роботи я освоїв елементи програмування мовою JAVA при розробленні аплетів, а також засвоїти класи і методи бібліотек java.awt, java.applet. Навчився використовувати потоки для розпаралелювання виконання різних задач. Створив аплет, який рисуватиме Progress Bar, а також до кожного Progress Bar прив’язав потік, який змінює його значення з часом.