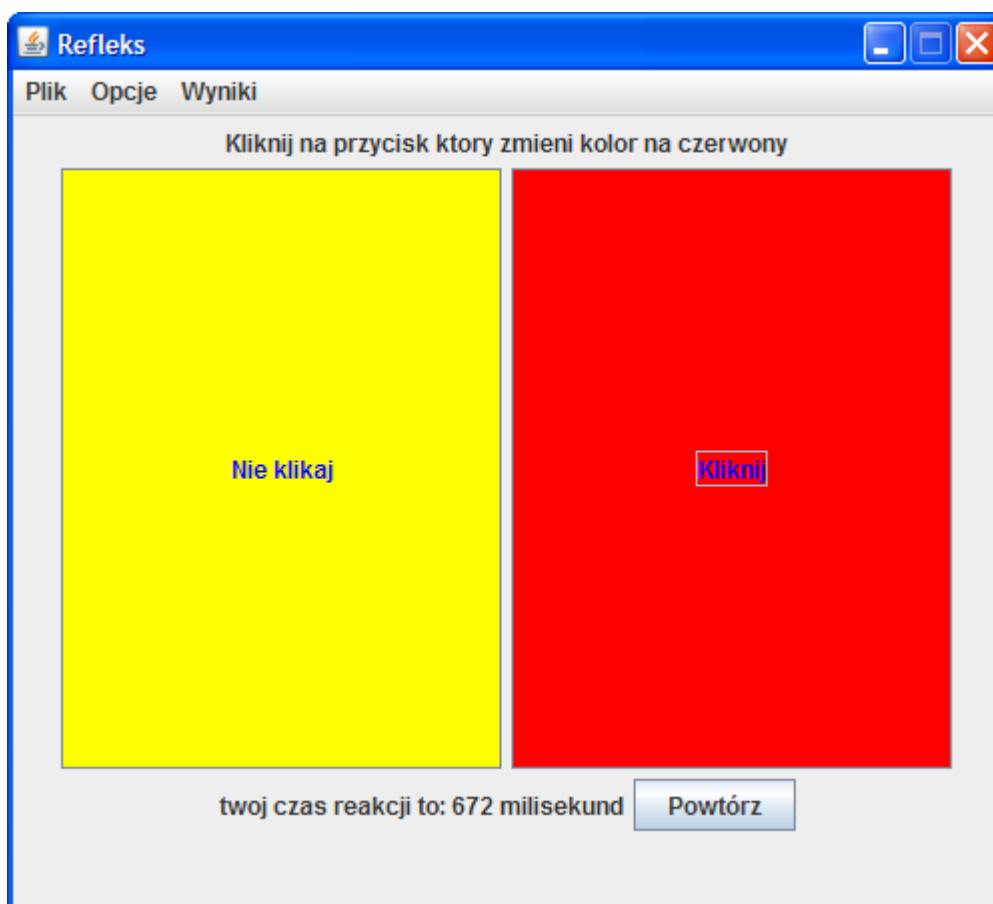


Laboratorium Inteligentnych Maszyn i Systemów

Politechnika Warszawska

Wydział Elektryczny

Test Refleksu



Dorian Gutowski dorian.gutowski@gmail.com

Semestr V, Informatyka, specjalność Inżynieria Oprogramowania

29.01.09

Spis treści:

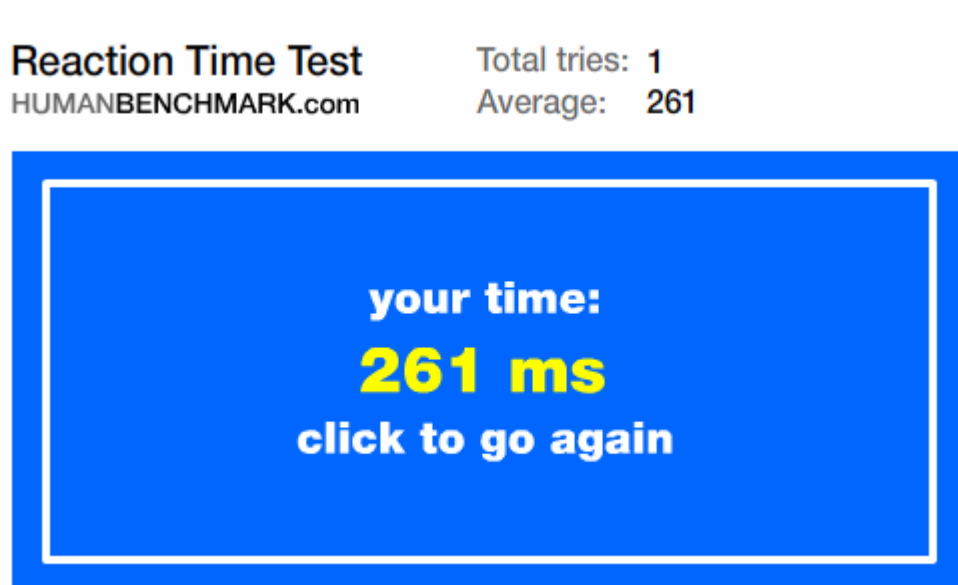
SPIS TREŚCI:	2
1 WPROWADZENIE	3
2 MOŻLIWE ROZWIĄZANIA	3
2.1 WYBÓR ROZWIĄZANIA	4
3 ZAŁOŻENIA PROJEKTU	4
4 REALIZACJA PROJEKTU	4
4.1 SZCZEGÓŁOWY OPIS REALIZACJI	4
4.2 OPIS IMPLEMENTACYJNY	9
4.3 OPIS URUCHOMIENIOWY	10
5 PODSUMOWANIE I WNIOSKI	10
6 MOŻLIWOŚCI ROZBUDOWY	11
BIBLIOGRAFIA	11
DODATEK	11

1 Wprowadzenie

Celem projektu jest stworzenie niewielkiego programiku testującego refleks. Użytkownik miał mieć możliwość sprawdzenia swojego czasu reakcji na nagły bodziec wzrokowy lub słuchowy.

2 Możliwe rozwiązania

W Internecie można napotkać wiele programików tego typu, zazwyczaj napisanych we flashu. Większość po prostu mierzy czas pomiędzy zmianą koloru tła a kliknięciem przez użytkownika np.:



Rysunek 1. Prosty program testujący refleks.

<http://www.humanbenchmark.com/tests/reactiontime/index.php>

Program po kliknięciu start czeka chwilę, po czym kolor tła zmienia się z czerwonego na zielony. Po kliknięciu wyświetla się czas oraz obliczona zostaje średnia ze wszystkich prób.

Czasem można natrafić na całkiem oryginalne i ładnie wyglądające programiki, sprawdzające czas reakcji w zupełnie inny sposób. Poniżej widać program niesłużący tylko do zabawy, lecz również do uświadamiania ludziom spadku refleksu pod wpływem alkoholu:



Rysunek 2. Okno programu sprawdzającego jak szybko zatrzymamy się na światłach

2.1 Wybór rozwiązania

Po obejrzeniu kilku programów tego typu zdecydowałem się na stworzenie aplikacji wyświetlającej przycisk, lub kilka przycisków, które w pewnym momencie zmieniają kolor. Należy wtedy jak najszybciej na niego kliknąć, bądź najpierw wybrać właściwy przycisk i na niego kliknąć. Program mierzy czas reakcji i zapisuje go do statystyk, aby użytkownik wiedział czy robi postępy czy nie.

3 Założenia projektu

Program jest napisany w języku Java działa pod systemem Windows w środowisku programistycznym NetBeans 6.0.1.

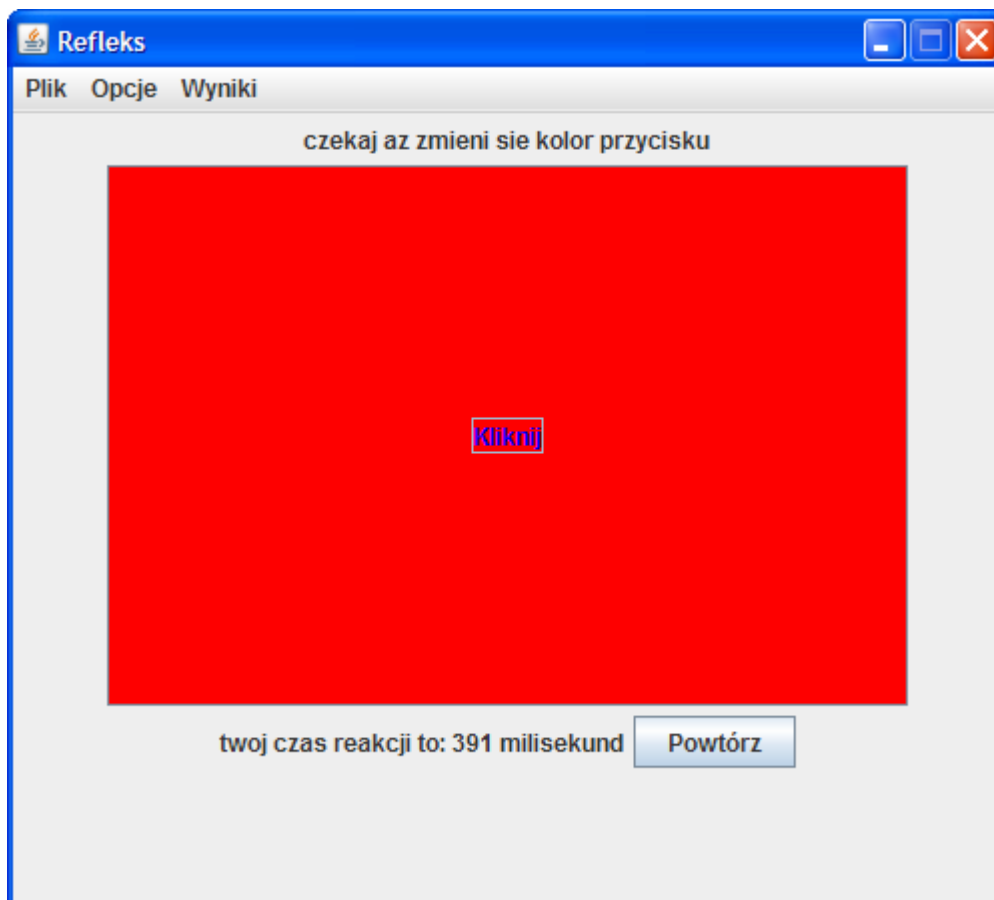
4 Realizacja projektu

4.1 Szczegółowy opis realizacji

Pierwsza wersja programu miała tylko jedną opcję, zmieniający kolor na czerwony prostokąt. Nie miał jeszcze ani zapisu uzyskanego czasu ani statystyk, miał na celu jedynie sprawdzić czy zastosowany przeze mnie sposób pomiaru czasu działa.

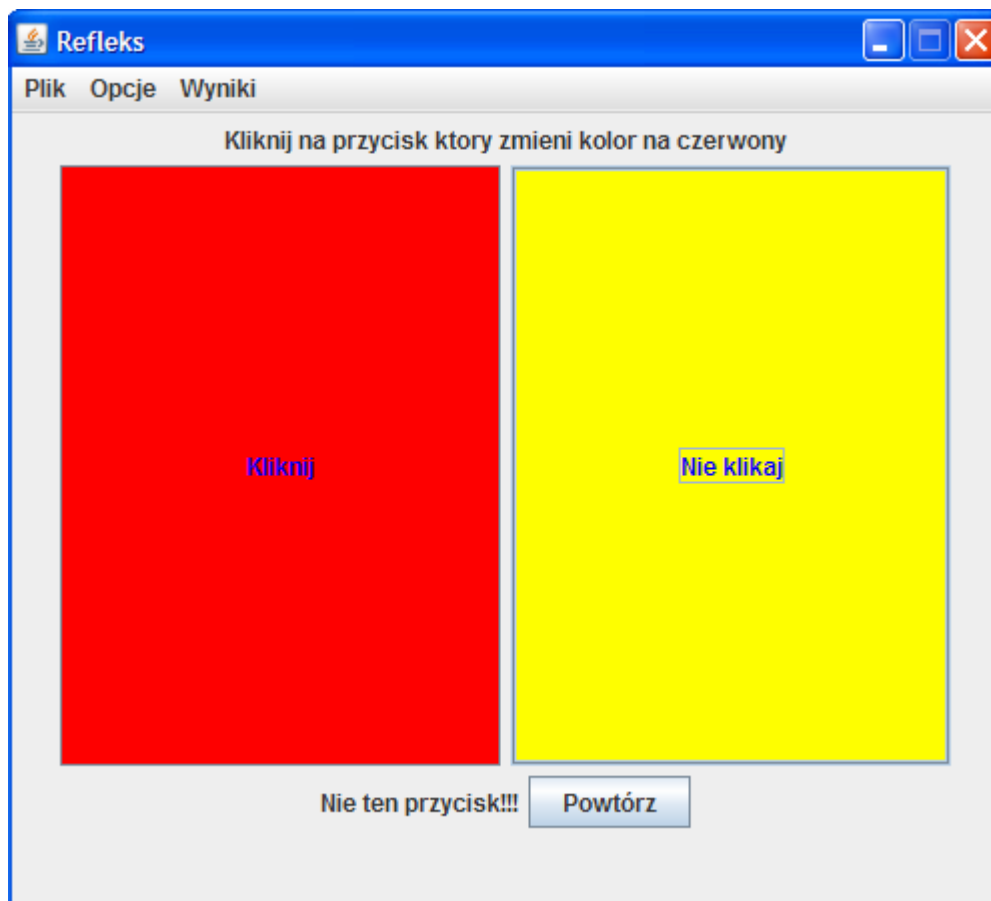
Klasa Stoper:

```
class Stoper {  
  
    private long sta = 0;  
    private long sto = 0;  
    private boolean run = false;  
  
    public void start() {  
        this.sta = System.currentTimeMillis();  
        this.run = true;  
    }  
  
    public void stop() {  
        if(this.run == true){  
            this.sto = System.currentTimeMillis();  
            this.run = false;  
        }  
    }  
  
    public long wynik() {  
        return ((sto - sta) - 2000);  
    }  
}
```



Rysunek 3. Pierwsza wersja programu

Następnie dodałem dwa kolejne tryby. W obu pojawiały się dwa prostokąty, należało kliknąć na jeden z nich. W pierwszym prostokąty zmieniały kolor na żółty i czerwony, w losowym ustawieniu. Należy kliknąć na czerwony, w razie kliknięcia żółtego program krzyczy ze przyciśnięto nie ten prostokąt.



Rysunek 4. Kliknięty żółty prostokąt

Trzeci tryb wyświetla w losowym ustawieniu 2 zdjęcia z 4 możliwych, przy czym jedno zawsze jest portretem Zdzicha. Wygląd obrazków jest bardzo rozpraszający, co powinno bardzo zwiększać trudność tego trybu a średni czas reakcji powinien bardzo wzrosnąć.

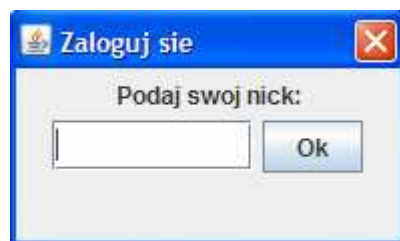


Rysunek 5. Kliknij na Zdzicha

Następnym krokiem było dodanie odporności na zbyt wczesne kliknięcie. Program wyświetla wtedy odpowiedni komunikat.

Ostatnim zagadnieniem były statystyki. Okazało się że jest to dużo bardziej pracochłonne niż przypuszczałem. Prawie połowa kodu programu zajmuje się stworzeniem statystyk.

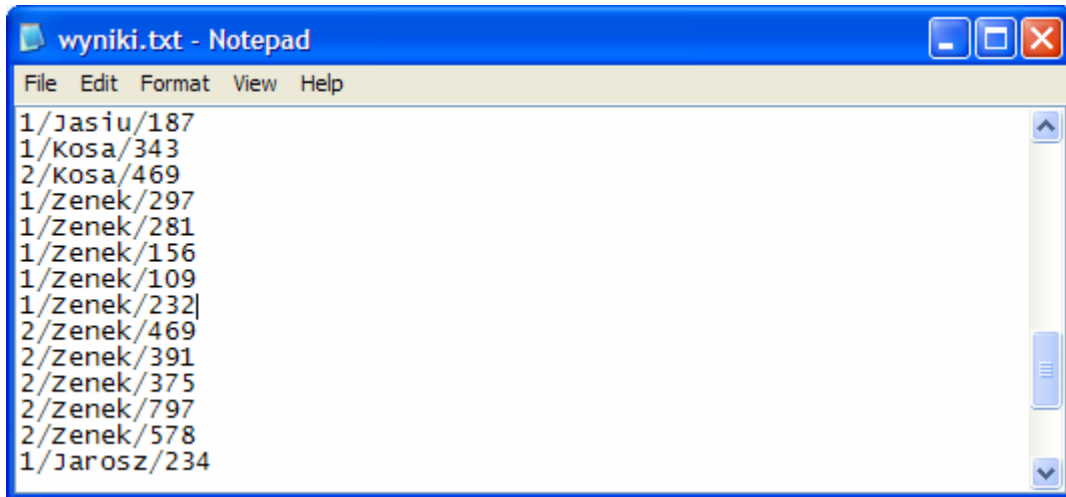
Tuz po uruchomieniu programu wyświetla się okienko proszące o wpisanie Nicku, widać je na rys 6.



Rysunek 6. Podaj nick

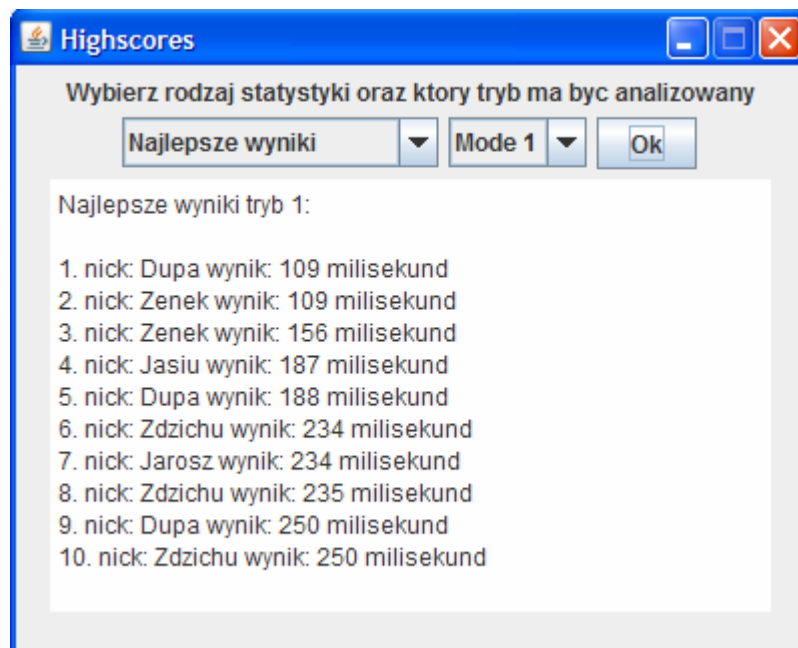
Wpisany Nick będzie używany później w statystykach. Okienko nie zamknie się, jeśli użytkownik nic nie wpisze lub wpisze spację.

Po każdej próbie testu refleksu czas reakcji wyświetlany na dole okna zostaje zapisany do pliku tekstowego wyniki.txt w formie tryb/nick/czas, co widać na rys7:



Rysunek 7. Zawartość pliku wyniki.txt

Po kliknięciu highscores w menu wyświetla się nowe okno (rys8) a wyniki zostają sczytane z pliku, każda linia, jako osobny łańcuch znaków. Każda linijka zostaje pocięta na intergera – tryb, Stringa – Nick i intergera – czas. Wyniki są przechowywane w kontenerze ArrayList.



Rysunek 8. Statystyki

Następnie elementy kontenera zostają posortowane rosnąco względem czasu reakcji. W zależności od wybranych opcji wyświetlają się: 10 najlepszych wyników wszystkich użytkowników, 10 najlepszych czasów danego użytkownika, średnia wszystkich lub średnia danego użytkownika, dla każdego z 3 trybów osobno.

Program posiada również zakładkę w menu nazwana 'info'. Po kliknięciu wyświetla się okienko z podstawowymi informacjami jak używać program.

4.2 Opis implementacyjny

Program składa się z 5 plików źródłowych.

Main.java:

Klasy:

Main: jest tam jedynie metoda main odpalająca konstruktor klasy SwingConsole. oraz

SwingConsole: klasa otwierająca okno w nowym wątku.

Okno.java:

Klasy:

Okno – wygląd okna głównego oraz odbiorniki zdarzeń do wszystkich obiektów w nim się znajdujących.

Dialog - małe okienko wyświetlające się po odpaleniu programu proszące o wpisanie swojego Nicku, który będzie potem zapisywany w statystykach.

Wait.java:

Klasy:

Stoper – służy do mierzenia czasu pomiędzy wykonaniem metody start a stop.

Wait – po wybraniu opcji i rozpoczęciu testu tworzy się nowy wątek, który oczekuje 2000 milisekund, po czym zmienia kolor przycisku/ przycisków.

Stat.java:

Klasa

Stat:

Posiada metody zapisz() – zapisuje czas każdej próby do pliku

Sczytaj() – czyta wyniki z pliku i umieszcza je w całości, jako Stringi w kontenerze ArrayList

Wytnij() – wycina z całych linijek pojedyncze zmienne i umieszcza je w nowym kontenerze.

Klasa **Linia** – typ danych kontenera przechowującego całe linie

Klasa **Podzielone** – typ danych kontenera przechowującego już pocięte linie.

Klasa **Sortuj** – sortuje elementy kontenera z elementami Podzielone po czasie reakcji, również wyciąga średnia czasów reakcji.

Scores.java:

Klasa Scores – nowe okno wyświetlające statystyki.

wyniki.txt:

– plik tekstowy, do którego są zapisywane wyniki kolejnych prób.

4.3 Opis uruchomieniowy

Program ma bardzo niewielkie wymagania sprzętowe i prawdopodobnie będzie działać na wszystkim, co jest w stanie uruchomić Windowsy.

4.4 Testowanie

Przy zwykłym pomiarze czasu reakcji na zmianę koloru tła średnie wyniki powinny być niewiele dłuższe niż 200 milisekund. W trybie drugim nie chodzi jedynie o szybka reakcje na dowolną zmianę, należy zauważyć który przycisk zmienił kolor na czerwony i wybrać go. Z tego powodu średnie czasy w tym trybie są znacznie dłuższe. Najdłuższe czasy reakcji zaobserwowałem w trybie 3. Nie tylko należy wybrać tam odpowiedni przycisk, ale najpierw trzeba rozpoznać konkretny kształt. W trakcie testowania programu przez większość czasu nie skupiałem się na czasie swojej reakcji lecz na tym czy wszystko działa jak powinno. Z dlatego a również z powodu mojego ciągłego zmęczenia i wyczerpania w trakcie testowania moje wyniki nie są zbyt wyśrubowane. W trybie pierwszym moja średnia wynosi 298ms, w trybie drugim 472ms a trzecim 513ms . Po przetestowaniu programu przez kilka innych osób średnie nieznacznie się poprawiły: tryb1 – 293ms tryb 2 – 467ms, tryb 3 – 490ms.

5 Podsumowanie i wnioski

Projekt jest prostym programikiem testującym czas reakcji na bodźce wzrokowe. Z powodu złego rozplanowania czasowego projektu nie udało mi się stworzyć kilku funkcji, które były planowane np. reakcji na dźwięk, prostokąta pojawiającego się w różnych miejscach okna itp.

6 Możliwości rozbudowy

Do programu przydałoby się dodać więcej opcji testowania refleksu, np. wybór z czterech przycisków, reakcja na dźwięk, reakcja na zmianę dźwięku lub muzyczka w tle rozpraszająca w trakcie robienia testu wzrokowego.

Bibliografia

- [1] Bruce Eckel, Thinking In Java, wydanie IV, Helion 2006.
- [2] <http://forum.4programmers.net>

Dodatek

Main.java

```
package swingg;

import java.awt.*;
import java.awt.event.*;
import java.util.*;
import javax.swing.*;
import java.util.concurrent.*;

class SwingConsole{
    public static void run(final JFrame f, final int width, final int
height){
        SwingUtilities.invokeLater(new Runnable(){
            public void run(){
```

```

        f.setTitle("Refleks");
        f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        f.setSize(width , height);
        f.setVisible(true);
        f.setResizable(false);
        f.setLocationRelativeTo(null);
    }
});
}
}

public class Main{

    public static void main(String[] args){
        SwingConsole.run(new Okno() , 500 , 450);

    }
}

```

Okno.java

```

package swingg;
import java.io.*;
import java.awt.*;
import java.awt.event.*;
import java.util.*;
import javax.swing.*;
import java.util.concurrent.*;
import java.awt.Color;

class Dialog extends JDialog{

    String nick = null;

    //okienko dialogowe czytające nick
    Dialog(){
        JLabel aa = new JLabel("Podaj swój nick:");
        final JTextField ee = new JTextField();
        ee.setPreferredSize(new Dimension(100, 25));
        JButton oo = new JButton("Ok");
    }
}

```

```

        setLayout(new FlowLayout());
        add(aa);
        add(ee);
        add(oo);
        setTitle("Zaloguj sie");
        setSize(200 , 120);
        setVisible(true);
        setLocationRelativeTo(null);
        this.setAlwaysOnTop(true);
        oo.addActionListener(new ActionListener(){
            public void actionPerformed(ActionEvent e){
                nick = ee.getText().trim();
                if(nick.length() != 0){
                    dispose();
                }
            }
        });
    }
}

//glowne okno
public class Okno extends JFrame{
    int eee = 0;
    int mod = 1;
    Stat za = new Stat();

    private JButton
        b1 = new JButton("Czekaj"),
        je = new JButton("Czekaj"),
        dw = new JButton("Czekaj"),
        tz = new JButton("Czekaj"),
        cz = new JButton("Czekaj"),
        start = new JButton("Start"),
        powt = new JButton("Powt6rzenie");

    private JLabel napg = new JLabel("Kliknij start aby rozpoczac");
    private JLabel czas = new JLabel();

    private JMenu[] menus = {new JMenu("Plik"), new JMenu("Opcje"), new

```

```

JMenu("Wyniki");

private JMenuItem[] items = {
    new JMenuItem("info"),
    new JMenuItem("wyjscie"),
    new JMenuItem("level 1"),
    new JMenuItem("level 2"),
    new JMenuItem("level 3"),
    new JMenuItem("Highscores"),
};

//zmiany wygladu okna przy zmianie trybu
public void model(){
    napg.setText("Kliknij start aby rozpoczac");
    start.setVisible(true);
    czas.setText("");
    powt.setVisible(false);
    je.setVisible(false);
    dw.setVisible(false);
    tz.setVisible(false);
    cz.setVisible(false);
    b1.setPreferredSize(new Dimension(400, 270));
    b1.setBackground(Color.WHITE);
    b1.setForeground(Color.BLUE);
    b1.setVisible(false);
}

void mode2(){
    napg.setText("Kliknij start aby rozpoczac");
    start.setVisible(true);
    czas.setText("");
    powt.setVisible(false);
    b1.setVisible(false);
    tz.setVisible(false);
    cz.setVisible(false);
    je.setPreferredSize(new Dimension(220, 300));
    dw.setPreferredSize(new Dimension(220, 300));
    je.setBackground(Color.WHITE);
    je.setForeground(Color.BLUE);
    dw.setBackground(Color.WHITE);
}

```

```

        dw.setForeground(Color.BLUE);
        je.setVisible(false);
        dw.setVisible(false);
    }

    void mode3(){
        napg.setText("Kliknij start aby rozpoczac");
        start.setVisible(true);
        czas.setText("");
        powt.setVisible(false);
        b1.setVisible(false);
        je.setVisible(false);
        dw.setVisible(false);
        tz.setPreferredSize(new Dimension(220, 300));
        cz.setPreferredSize(new Dimension(220, 300));
        tz.setBackground(Color.WHITE);
        tz.setForeground(Color.BLUE);
        cz.setBackground(Color.WHITE);
        cz.setForeground(Color.BLUE);
        tz.setVisible(false);
        cz.setVisible(false);
    }

    public Okno(){
        final Stoper s = new Stoper();
        final Dialog nic = new Dialog();

        //listenery do przyciskow
        b1.addActionListener(new ActionListener(){
            public void actionPerformed(ActionEvent e){
                s.stop();
                long w = s.wynik();
                if(w > 0){
                    czas.setText("twoj czas reakcji to: "+ w +
milisekund");
                    za.zapisz(1 ,nic.nick , w);
                }
                else{
                    Wait.czyczit(1);

```

```

        czas.setText("Nie oszukuj !!!!!!!!!!!");
    }

    powt.setVisible(true);
}
});

je.addActionListener(new ActionListener(){
    public void actionPerformed(ActionEvent e){
        s.stop();

        long w = s.wynik();
        if(w > 0){

            if(je.toString().contains("Kliknij") && eee==0){
                czas.setText("twoj czas reakcji to: "+ w + "
milisekund");

                za.zapisz(2 ,nic.nick , w);
            }
            else{
                eee = 1;
                czas.setText("Nie ten przycisk!!!");
            }
        }
        else{
            Wait.czyczit(1);
            czas.setText("Nie oszukuj !!!!!!!!!!!");

        }

        powt.setVisible(true);
    }
});

dw.addActionListener(new ActionListener(){
    public void actionPerformed(ActionEvent e){
        s.stop();
        long w = s.wynik();
        if(w > 0){

```



```

        if(dw.toString().contains("Kliknij") && eee==0){
            czas.setText("twoj czas reakcji to: "+ w +
milisekund");
            za.zapisz(2 ,nic.nick , w);
        }
        else{
            eee = 1;
            czas.setText("Nie ten przycisk!!!");
        }
    }
    else{
        Wait.czyczit(1);
        czas.setText("Nie oszukuj !!!!!!!!!!!");
    }

    powt.setVisible(true);
}
});

tz.addActionListener(new ActionListener(){
    public void actionPerformed(ActionEvent e){
        s.stop();
        long w = s.wynik();
        if(w > 0){

            if(tz.toString().contains("zdzichu") && eee==0){
                czas.setText("twoj czas reakcji to: "+ w +
milisekund");
                za.zapisz(3 ,nic.nick , w);
            }
            else{
                eee = 1;
                czas.setText("To nie jest Zdzichu!!!");
            }
        }
        else{
            Wait.czyczit(1);
            czas.setText("Nie oszukuj !!!!!!!!!!!");

        }

        powt.setVisible(true);
    }
});

```

```

    }
    });

    cz.addActionListener(new ActionListener(){
        public void actionPerformed(ActionEvent e){
            s.stop();
            long w = s.wynik();
            if(w > 0){

                if(cz.toString().contains("zdzichu") && eee==0){
                    czas.setText("twoj czas reakcji to: "+ w +"
milisekund");

                    za.zapisz(3 ,nic.nick , w);
                }
                else{
                    eee = 1;
                    czas.setText("To nie jest Zdzichu!!!");
                }
            }
            else{
                Wait.czyczit(1);
                czas.setText("Nie oszukuj !!!!!!!!!!!");

            }
            powt.setVisible(true);
        }
    });

    start.addActionListener(new ActionListener(){ //start
        public void actionPerformed(ActionEvent e){

            start.setVisible(false);
            if(mod == 1){
                napg.setText("czekaj az zmieni sie kolor przycisku");
                b1.setVisible(true);
                s.start();

                ExecutorService exec =
Executors.newCachedThreadPool();

                exec.execute(new Wait(b1));
                exec.shutdown();
            }
        }
    });

```

```

        else if(mod == 2){
            napg.setText("Kliknij na przycisk ktory zmieni kolor
na czerwony");

            je.setVisible(true);
            dw.setVisible(true);
            s.start();

            ExecutorService exec =
Executors.newCachedThreadPool();

            exec.execute(new Wait(je , dw));
            exec.shutdown();
        }
        else if(mod == 3){
            napg.setText("Jak najszybciej kliknij na przepiękny
portret Zdzicha");

            tz.setVisible(true);
            cz.setVisible(true);
            s.start();

            ExecutorService exec =
Executors.newCachedThreadPool();

            exec.execute(new Wait(tz , cz, 1));
            exec.shutdown();
        }
    }
});

powt.addActionListener(new ActionListener(){//powtorz
    public void actionPerformed(ActionEvent e){
        if(mod==1){
            napg.setText("czekaj az zmieni sie kolor przycisku");
            start.setVisible(false);
            b1.setBackground(Color.WHITE);
            czas.setText("");
            powt.setVisible(false);
            s.start();

            ExecutorService exec =
Executors.newSingleThreadExecutor();

            exec.execute(new Wait(b1));
            exec.shutdown();
        }else if(mod==2){
            eee = 0;
            napg.setText("Kliknij na przycisk ktory zmieni kolor
na czerwony");

```

```

        start.setVisible(false);
        je.setBackground(Color.WHITE);
        dw.setBackground(Color.WHITE);
        je.setText("Kliknij");
        dw.setText("Kliknij");
        czas.setText("");
        powt.setVisible(false);
        s.start();

        ExecutorService exec =
Executors.newCachedThreadPool();

        exec.execute(new Wait(je , dw));
        exec.shutdown();
    }else if(mod==3){
        eee = 0;
        napg.setText("Jak najszybciej kliknij na przepiękny
portret Zdzicha");

        start.setVisible(false);
        tz.setIcon(null);
        cz.setIcon(null);
        tz.setText("Kliknij");
        cz.setText("Kliknij");
        czas.setText("");
        powt.setVisible(false);
        s.start();

        ExecutorService exec =
Executors.newCachedThreadPool();

        exec.execute(new Wait(tz , cz , 1));
        exec.shutdown();
    }
}
});

items[0].addActionListener(new ActionListener(){//info
    public void actionPerformed(ActionEvent e){

        JFrame info = new JFrame();
        JTextArea ee = new JTextArea("Test Refleksu\n\n");
        ee.append("Programik mierzy czas reakcji na zmianę
przycisku\n");
        ee.append("mode 1: kliknij na przycisk jak najszybciej
po zmianie koloru\n");
        ee.append("mode 2: kliknij na przycisk który zmieni

```

```

kolor na czerwony\n");
        ee.append("mode 3: kliknij na portret zdzicha\n");
        ee.setEditable(false);
        info.add(ee);
        info.setTitle("Informacje");
        info.setSize(400 , 200);
        info.setVisible(true);
    }
});

items[1].addActionListener(new ActionListener(){//wyjscie
    public void actionPerformed(ActionEvent e){

        System.exit(0);
    }
});

items[2].addActionListener(new ActionListener(){//level1
    public void actionPerformed(ActionEvent e){
        mod = 1;
        model();
    }
});

items[3].addActionListener(new ActionListener(){//level2
    public void actionPerformed(ActionEvent e){
        mod = 2;
        mode2();
    }
});

items[4].addActionListener(new ActionListener(){//level3
    public void actionPerformed(ActionEvent e){
        mod = 3;
        mode3();
    }
});

items[5].addActionListener(new ActionListener(){//Highscores
    public void actionPerformed(ActionEvent e){

```

```

        Scores sss = new Scores(nic.nick);
    }
});

JMenuBar mb = new JMenuBar();
menus[0].add(items[0]);
menus[0].add(items[1]);
menus[1].add(items[2]);
menus[1].add(items[3]);
menus[1].add(items[4]);
menus[2].add(items[5]);

for(JMenu jm : menus)
    mb.add(jm);
setJMenuBar(mb);
setLayout(new FlowLayout());
add(napg);//kliknij aby rozpoczac
add(start);//start
add(b1);//przyc 1
add(je);
add(dw);
add(tz);
add(cz);
add(czas);
add(powt);
model();
}
}

```

Wait.java

```

package swingg;

import java.awt.*;
import java.awt.event.*;
import java.util.*;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.*;
import java.util.concurrent.*;
import java.awt.Color;

```

```

class Stoper {

    private long sta = 0;
    private long sto = 0;
    private boolean run = false;

    public void start() {
        this.sta = System.currentTimeMillis();
        this.run = true;
    }

    public void stop() {
        if(this.run == true){
            this.sto = System.currentTimeMillis();
            this.run = false;
        }
    }

    public long wynik() {

        return ((sto - sta) - 2000);

    }
}

public class Wait implements Runnable{
    int e, ran;
    JButton b1, b2;
    static int czy;
    Random rand = new Random();
    ImageIcon kwadrat = new ImageIcon("zdzichu.jpg");
    ImageIcon trojkat = new ImageIcon("rower.jpg");
    ImageIcon kolko = new ImageIcon("pies.jpg");
    ImageIcon gwiazdka = new ImageIcon("swieczka.gif");
}

```

```

public Wait(JButton jb){
    b1 = jb;
    czy = 0;
    e = 1;
}

public Wait(JButton jb, JButton jb2){
    b1 = jb;
    b2 = jb2;
    czy = 0;
    e = 2;
    ran = rand.nextInt(2);
}

public Wait(JButton jb, JButton jb2, int x){
    b1 = jb;
    b2 = jb2;
    czy = 0;
    e = 3;
    ran = rand.nextInt(6);
}

public static void czyczit(int i){
    czy = i;
}

void trzy(){
    try {
        TimeUnit.MILLISECONDS.sleep(2000);
        if(czy == 0) {
            b1.setText("");
            b2.setText("");
            if(ran==0){
                b1.setIcon(kwadrat);
                b2.setIcon(trojkat);
            }
            else if(ran ==1){
                b1.setIcon(trojkat);
            }
        }
    }
}

```



```

        b2.setIcon(kwadrat);
    }
    else if(ran ==2){
        b1.setIcon(kwadrat);
        b2.setIcon(kolko);
    }
    else if(ran ==3){
        b1.setIcon(kwadrat);
        b2.setIcon(gwiazdka);
    }
    else if(ran ==4){
        b1.setIcon(kolko);
        b2.setIcon(kwadrat);
    }
    else if(ran ==5){
        b1.setIcon(gwiazdka);
        b2.setIcon(kwadrat);
    }
}

} catch (InterruptedException ex) {
    Logger.getLogger(Wait.class.getName()).log(Level.SEVERE,
null, ex);
}
}

void dwa(){
    try {
        TimeUnit.MILLISECONDS.sleep(2000);
        if(czy == 0) {

            if(ran==0){
                b1.setText("Kliknij");
                b2.setText("Nie klikaj");
                b1.setBackground(Color.RED);
                b2.setBackground(Color.YELLOW);
            }

            else if(ran ==1){
                b1.setText("Nie klikaj");

```

```

        b2.setText("Kliknij");
        b1.setBackground(Color.YELLOW);
        b2.setBackground(Color.RED);
    }
}

} catch (InterruptedException ex) {
    Logger.getLogger(Wait.class.getName()).log(Level.SEVERE,
null, ex);
}
}

void jeden(){
    try {
        TimeUnit.MILLISECONDS.sleep(2000);
        if(czy == 0) {
            b1.setBackground(Color.RED);
            b1.setText("Kliknij");
        }

    } catch (InterruptedException ex) {
        Logger.getLogger(Wait.class.getName()).log(Level.SEVERE,
null, ex);
    }
}

public void run() {

    if(e==1){
        jeden();
    }
    else if(e==2){
        dwa();
    }else if(e==3){
        trzy();
    }
}
}
}

```

Stat.java

```

package swingg;
import java.io.*;
import java.util.*;
import java.util.logging.Level;
import java.util.logging.Logger;

class Sortuj{

    ArrayList<Podzielone>lajn = new ArrayList<Podzielone>();
    Sortuj(ArrayList<Podzielone> a){
        lajn = a;
    }

    void najlepsze(){
        int j;

        String nick;
        int czas;
        int mod;

        for(int i=0 ; i < lajn.size() ; i++){
            for(j= i+1 ; j < lajn.size() ; j++){//kepa-kurwa-sa?

                if(lajn.get(i).czas > lajn.get(j).czas){
                    mod = lajn.get(i).mod;
                    czas = lajn.get(i).czas;
                    nick = lajn.get(i).nick;

                    lajn.get(i).mod = lajn.get(j).mod;
                    lajn.get(i).czas = lajn.get(j).czas;
                    lajn.get(i).nick = lajn.get(j).nick;

                    lajn.get(j).mod = mod;
                    lajn.get(j).czas = czas;
                    lajn.get(j).nick = nick;
                }
            }
        }
    }
}

```

```

int srednia(int m){

    int x = 0 , sr = 0;

    for(Podzielone a : lajn){
        if(a.mod == m){
            sr+=a.czas;
            x++;
        }
    }

    sr = sr/x;

    return sr;
}

int tsrednia( String key , int m){

    int x = 0 , sr = 0;

    for(Podzielone a : lajn){
        if(a.mod == m && a.nick.equals(key)){
            sr+=a.czas;
            x++;
        }
    }

    sr = sr/x;

    return sr;

}

}

////////////////////////////////////
//obiekty do kontenerow

class Linia{
    static int counter;
    private final int id = counter++;
    String linijka;
}

```

```

    Linia(String a){
        linijka = a;
    }

    public int id(){return id;}
}

class Podzielone{

    static int counter;
    private final int id = counter++;
    String nick;
    int czas;
    int mod;
    //boolean czys = false;

    Podzielone(int c, String a , int b){
        nick = a;
        czas = b;
        mod = c;
    }

    public int id(){return id;}
}

////////////////////////////////////

public class Stat {

    //czytanie z pliku
    ArrayList<Linia> zaczytaj(){

        ArrayList<Linia> linie = new ArrayList<Linia>();
        BufferedReader bb =null , pw = null;
        try {

```

```

        bb = new BufferedReader(new InputStreamReader(System.in));
        String nazwaPliku = "wyniki.txt";
        pw = new BufferedReader(new FileReader(nazwaPliku));
        String linia;

        while((linia = pw.readLine())!= null){
            linie.add(new Linia(linia));
        }

        pw.close();
        bb.close();

    } catch (IOException ex) {
        Logger.getLogger(Stat.class.getName()).log(Level.SEVERE, null,
ex);

    }finally {
        try {
            bb.close();
            pw.close();
        } catch (IOException ex) {
            Logger.getLogger(Stat.class.getName()).log(Level.SEVERE,
null, ex);
        }
    }

    return linie;

}

//zapis do pliku
void zapisz(int mod, String nick , long wyn){
    PrintWriter plik = null;

    try {
        plik = new PrintWriter(new FileWriter("wyniki.txt", true));
    } catch (IOException e) {

```

```

        System.err.println("Nie udalo sie otworzyc pliku: " +
e.getLocalizedMessage());
        System.exit(1);
    }

    plik.println(mod + "/" + nick + "/" + wyn);

    plik.close();
}

//wycinanie zmiennych ze Stringa
ArrayList<Podzielone> wytnij(ArrayList<Linia> line){

    ArrayList<Podzielone> wyniki = new ArrayList<Podzielone>();

    int x;

    for(Linia w : line){

        char [] aa = w.linijka.toCharArray();

        String mo = "";
        String nicc = "";
        String ww = "";
        mo += aa[0];
        int wy = 0;
        int mod = Integer.parseInt(mo);

        for(int ii=2 ; ii< aa.length ; ii++){
            if(aa[ii] == 47){

                for(x = 2 ; x < ii ; x++){
                    nicc+= aa[x];
                }
                x++;
                for( ; x < aa.length ; x++){
                    ww += aa[x];
                }
                wy = Integer.parseInt(ww);
            }
        }
    }
}

```

```

                break;
            }
        }

        wyniki.add(new Podzielone(mod , nicc , wy));

    }
    return wyniki;
}
}

```

Scores.java

```

package swingg;

import java.io.*;
import java.awt.*;
import java.awt.event.*;
import java.util.*;
import javax.swing.*;
import java.util.concurrent.*;

//okno wyswietlajace statystyki
public class Scores extends JFrame{
    String nick;
    private JButton ok = new JButton("Ok");
    JLabel wyb = new JLabel("Wybierz rodzaj statystyki oraz ktory tryb ma
    byc analizowany");
    JComboBox opcje = new JComboBox();
    JComboBox mode = new JComboBox();
    JTextArea wyn = new JTextArea(13 , 32);

    Scores(String n){

        nick = n;

        setTitle("Highscores");
        setSize(400 , 325);
        setVisible(true);
    }
}

```



```

setResizable(false);
setLocationRelativeTo(null);

wyn.setVisible(false);
wyn.setEditable(false);

opcje.addItem("Najlepsze wyniki");
opcje.addItem("Twoje najlepsze wyniki");
opcje.addItem("Srednia");
opcje.addItem("Twoja srednia");
mode.addItem("Mode 1");
mode.addItem("Mode 2");
mode.addItem("Mode 3");
setLayout(new FlowLayout());
add(wyb);
add(opcje);
add(mode);
add(ok);

add(wyn);
wyn.setBorder(BorderFactory.createEmptyBorder(4, 4, 4, 4));

ok.addActionListener(new ActionListener(){
    public void actionPerformed(ActionEvent e){
        String op = (String) opcje.getSelectedItemAt();
        String mo = (String) mode.getSelectedItemAt();
        wyn.setVisible(true);
        Stat uu = new Stat();
        ArrayList<Linia> eoe = uu.sczytaj();
        ArrayList<Podzielone> qq = uu.wytnij(eoe);
        Sortuj ss = new Sortuj(qq);
        int lp = 1;

        if(op.equals("Najlepsze wyniki")){
            wyn.setText("Najlepsze wyniki ");
            ss.najlepsze();

            if(mo.equals("Mode 1")){
                wyn.append("tryb 1: \n\n");
                for(Podzielone a : ss.lajn){

```

```

        if(a.mod == 1 && lp<=10){
            wyn.append(lp + ". nick: "+a.nick+" wynik:
"+a.czas + " milisekund\n");
            lp++;
        }
    }
}
}else if(mo.equals("Mode 2")){
    wyn.append("tryb 2: \n\n");
    for(Podzielone a : ss.lajn){
        if(a.mod == 2 && lp<=10){
            wyn.append(lp + ". nick: "+a.nick+" wynik:
"+a.czas + " milisekund\n");
            lp++;
        }
    }
}
}else if(mo.equals("Mode 3")){
    wyn.append("tryb 3: \n\n");
    for(Podzielone a : ss.lajn){
        if(a.mod == 3 && lp<=10){
            wyn.append(lp + ". nick: "+a.nick+" wynik:
"+a.czas + " milisekund\n");
            lp++;
        }
    }
}

}
}else if(op.equals("Twoje najlepsze wyniki")){

    wyn.setText("Twoje najlepsze wyniki ");
    ss.najlepsze();

    if(mo.equals("Mode 1")){
        wyn.append("tryb 1:\n\n");
        for(Podzielone a : ss.lajn){
            if(a.mod == 1 && a.nick.equals(nick) &&
lp<=10){
                wyn.append(lp + ". wynik: "+a.czas+ "
milisekund\n");

                lp++;
            }
        }
    }
}
}else if(mo.equals("Mode 2")){

```



```
        }else if(mo.equals("Mode 2")){
            wyn.append("tryb 2: ");
            wyn.append(ss.tsrednia(nick , 2)+" milisekund\n");
        }else if(mo.equals("Mode 3")){
            wyn.append("tryb 3: ");
            wyn.append(ss.tsrednia(nick , 3)+" milisekund\n");
        }
    }
}
});
}
}
```