|  |
| --- |
| ***Teme:***  *Procesi/niti, specializirane niti*  *UV (UI), oblikovanje, sestavljanje;*  *Nadzor nad elementi UI.* |

***Naloga ima 2 varianti, za minimalno izvedbo zadostuje ena sama***

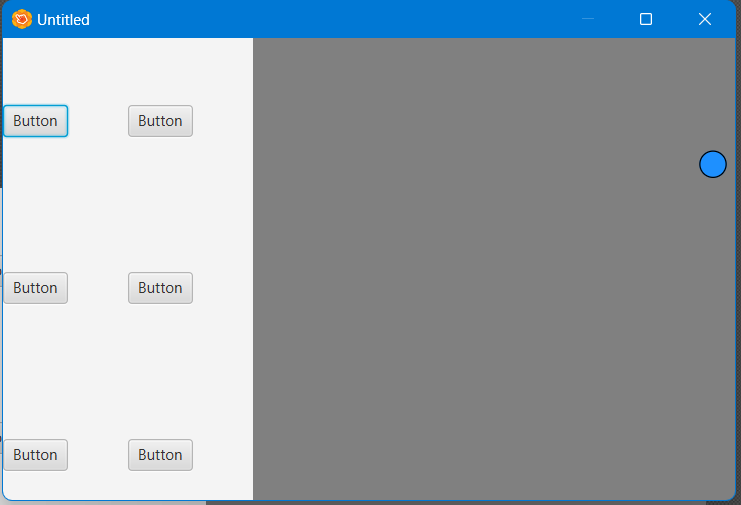
### Varianta A

Rezultat ene od predhodnih vaj je bila realizacija N-mestnega 7 segmentnega prikazovalnika, z vizualizacijo tehnološko izvedeno z JavaFX. Pričakovana realizacija vsebuje 'štetje' v obe smeri; s povečevanje in zmanjševanjem prikazane vrednosti.

1. Realizirajte odštevalnik (count-down), pri katerem od nastavljene vrednosti odštevate do 0 in ob tem dogodku sprožite alarm (zvok, dialog,…). Odštevanje naj bo avtomatično, vsako sekundo za eno enoto. Avtomatizem odštevanja realizirajte s procesom, ki bo vsako sekundo sprožil ustrezen signal prikazovalniku. (veliko možnosti: Task, Timer, TimerTask,…)
2. Realizirajte štoperico (enkratna meritev), pri kateri štejete navzgor; uporabnik jo lahko v določenem trenutno zaustavi in nato ponastavi. Interval med sosednjima korakoma štetja naj bo 1 sekunda.
3. Realizirajte štoperico (večkratni rezultati), s štetjem navzgor; uporabnik lahko prikazovanje štetja začasno zaustavi ali prikaz povrne do končne zaustavitve in ponastavitve. Da, 'pauzira' se zgolj prikazovanje, ne pa tudi štetje. Števni impulzi naj prihajajo na vsako sekundo

Primer rabe Timer/TimerTask je dan v prilogi tega dokumenta.

### Varianta B

Recimo, da scena sestavljena iz ozkega levega dela in širšega desnega. Levi del je zapolnjen z objekti (npr. gumbi kot na sliki). Ideja aplikacije je preprosta : ko kliknete poljuben objekt na levi, se vsi objekti sočasno začnejo gibati proti izbrani točki v desnem delu (modra krožnica) in jo dosežejo po petih sekundah.

Predstavljena scena je zgrajena kot: koren je BorderPane, v levem delu je GridPane 3x2, ki ima v vsaki celici po en gumb, v srednjem delu border-pane-ja je Pane, ki na naključno izbrani poziciji vsebuje še objekt tipa Circle .

Namig: za premik posameznega objekta uporabite tranzicijo (javafx.animation.Transition), vse objekte premaknite sočasno z uporabo vzporedne/sočasne tranzicije; tranzicijo za objekt naredite ob kreiranju objekta in jo sočasno dodajte v paralelno tranzicijo(javafx.animation.ParallelTransition). Event (dogodek) klika na poljubnem objektu naj to vzporedno tranzicijo sproži. (primer v uradni dokumentaciji o vzpordni tranziciji je dovolj ilustrativen)

Priloga A:

package si.mojaorganizacija.demofxml;

import java.util.Timer;

import java.util.TimerTask;

import javafx.application.Application;

import javafx.application.Platform;

import javafx.geometry.Insets;

import javafx.scene.Scene;

import javafx.scene.control.Alert;

import javafx.scene.control.Button;

import javafx.scene.layout.HBox;

import javafx.stage.Stage;

public class MojTimercek extends Application { // navadna nit v FX, proženje

int del = 5; // naj bo na 5 sekund

public void start(Stage st) {

Timer tm = new java.util.Timer();

//vse skupaj v enem hbox-u

HBox hb = new HBox(12);

//knof za začetek,

Button b = new Button("Timer na 5 sekund");

b.setOnAction(event -> {

//tm.schedule(new Ticker(), del\*1000); // čez DEL sekund

tm.schedule(new Ticker(), 5, del\*1000); // čez DEL sekund

});

hb.getChildren().addAll(b);

//scena

Scene sc = new Scene(hb);

st.setTitle("Moj Timer");

st.setScene(sc);

st.show();

}

// gnezden lokalni razred TimerTask - čisto navaden Thread (glej run/0) - interakcija s FX,

// Platform.runLater/0

private class Ticker extends TimerTask {

//run method

@Override

public void run() {

//method

Platform.runLater(() -> {

// alert je v FX preddefinirana kontrola - dialog ...

Alert a = new Alert(Alert.AlertType.INFORMATION);

a.setTitle("ALARM");

a.setHeaderText("5 sekund je poteklo");

a.setContentText("Naredi s tem karkoli- ker pridem še ... ");

a.showAndWait();

});

}

}

public static void main(String[] args) {

launch(args);

}

}