

```

/*
 * Luokassa määriteltä kaikki alareunan napit.
 */

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

public class PainikeRivi extends JPanel implements ActionListener {

    private JButton käynnistäPysäytäNappi;
    private JButton tyhjennäNappi;
    private JButton arvoNappi;
    private JButton seuraavaNappi;
    private JLabel sukupolviKenttä;
    private JLabel solujaKenttä;
    private Font nappienKirjasin;
    private Insets nappienMarginaali;
    private RuudukkoAlusta ruudukkoAlusta;
    private boolean onkoKäynnissä;

    /*
     * Luokan konstruktori, jossa luodaan nappien ja
     * tekstikenttien sisältö ja ulkonäkö.
     */

    public PainikeRivi(RuudukkoAlusta ruudukkoAlusta) {

        // Alustetaan luokan muuttujat, mm. luodaan
        // napit ja tekstikentät.

        this.ruudukkoAlusta = ruudukkoAlusta;
        onkoKäynnissä = false;
        nappienKirjasin = new Font("SansSerif", Font.BOLD, 12);
        nappienMarginaali = new Insets(0, 3, 0, 3);

        käynnistäPysäytäNappi = new JButton("Käynnistä");
        tyhjennäNappi = new JButton("Tyhjennä");
        arvoNappi = new JButton("Arvo");
        seuraavaNappi = new JButton("Seuraava");

        sukupolviKenttä = new JLabel("Sukupolvi: 0");
        solujaKenttä = new JLabel("Soluja: 0");

        // Määritellään nappien fontit ja nappien
        // tekstien marginaalit.

        käynnistäPysäytäNappi.setFont(nappienKirjasin);
        tyhjennäNappi.setFont(nappienKirjasin);
        arvoNappi.setFont(nappienKirjasin);
        seuraavaNappi.setFont(nappienKirjasin);

        käynnistäPysäytäNappi.setMargin(nappienMarginaali);
        tyhjennäNappi.setMargin(nappienMarginaali);
        arvoNappi.setMargin(nappienMarginaali);
        seuraavaNappi.setMargin(nappienMarginaali);

        // Lisätään komponentit oman luokan säiliöön

        this.add(käynnistäPysäytäNappi);
    }
}

```

```

this.add(tyhjennäNappi);
this.add(arvoNappi);
this.add(seuraavaNappi);
this.add(Box.createGlue());
this.add(sukupolviKenttä);
this.add(solujaKenttä);

// Määritellään tapahtumakuuntelijat napeille

käynnistäPysäytäNappi.addActionListener(this);
tyhjennäNappi.addActionListener(this);
arvoNappi.addActionListener(this);
seuraavaNappi.addActionListener(this);

this.setBorder(BorderFactory.createEtchedBorder());
}

/*
 * Hoidellaan Käynnistä/Pysäytä-napin tapahtumat.
 */

private void käynnistäPysäytä() {
    if (onkoKäynnissä) {
        ruudukkoAlusta.pysäytäAjastin();
        käynnistäPysäytäNappi.setText("Käynnistä");
        onkoKäynnissä = false;
    }
    else {
        ruudukkoAlusta.käynnistäAjastin();
        käynnistäPysäytäNappi.setText("Pysäytä");
        onkoKäynnissä = true;
    }
}

/*
 * Pysäytetään ajastin ja asetetaan
 * Käynnistä/Pysäytä-napin teksti.
 */

private void pysäytä() {
    ruudukkoAlusta.pysäytäAjastin();
    käynnistäPysäytäNappi.setText("Käynnistä");
    onkoKäynnissä = false;
}

/*
 * Asetetaan Sukupolvi-tekstikentän teksti.
 */

public void asetaSukupolvi(int sukupolviNro) {
    sukupolviKenttä.setText("Sukupolvi: " + sukupolviNro);
}

/*
 * Asetetaan Soluja-tekstikentän teksti.
 */

```

```

public void asetaSoluja(int solujaNro) {
    solujaKenttä.setText("Soluja: " + solujaNro);
}

/*
 * Hoidellaan nappien aiheuttamat tapahtumat.
 */

public void actionPerformed(ActionEvent tapahtuma) {
    Object aiheuttaja = tapahtuma.getSource();

    // Käynnistä/Pysäytä-nappi

    if (aiheuttaja == käynnistäPysäytäNappi) {
        käynnistäPysäytä();
    }

    // Tyhjennä-nappi

    if (aiheuttaja == tyhjennäNappi) {
        pysäytä();
        ruudukkoAlusta.tyhjennä();
    }

    // Arvo-nappi

    if (aiheuttaja == arvoNappi) {
        pysäytä();
        ruudukkoAlusta.arvo();
    }

    // Seuraava-nappi

    if (aiheuttaja == seuraavaNappi) {
        pysäytä();
        ruudukkoAlusta.seuraavaSukupolvi();
    }
}
}

```