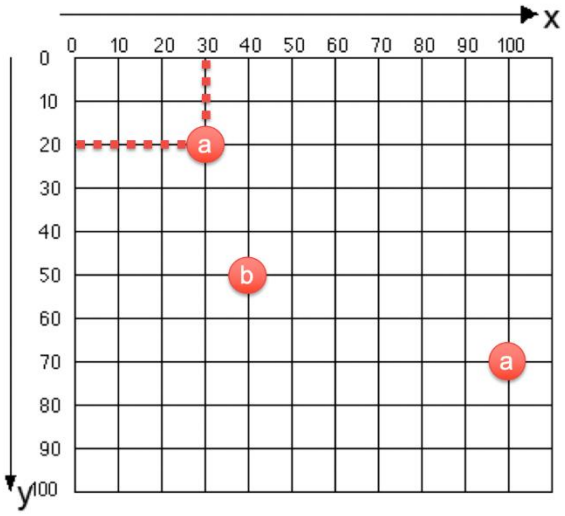
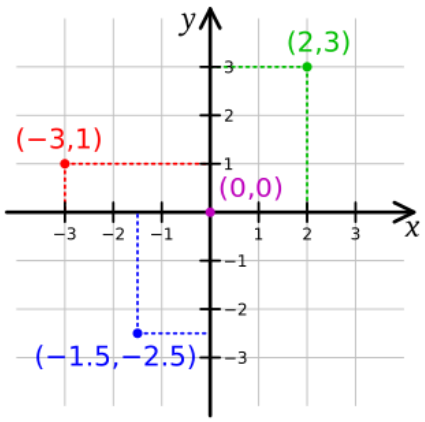
JAVA GRAFIKA UN LIETOTĀJA METODES

34.nodarbība

JAVA GRAFIKAS KOORDINĀTU SISTĒMA



Punkts	X	Y
a	30	20
b	40	50
c	100	70

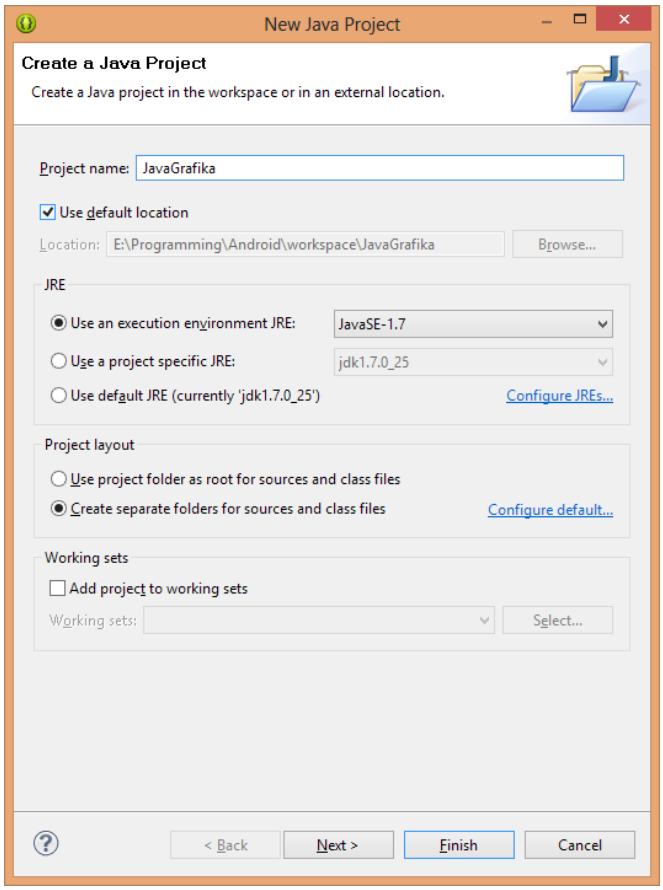


JAVA IEBŪVĒTĀS GRAFISKĀS METODES (GRAPHICS. …)

- drawArc, fillArc – zīmē riņķa loku
- drawLine – zīmē līniju
- drawOval, fillOval – zīmē ovālu (apli)
- drawRect, fillRect – zīmē taisnstūri
  - drawRoundRect, fillRoundRect – ar noapaļotiem stūriem
- drawString – zīmē tekstu
- setColor – uzstāda zīmēšanas krāsu
- +klase Graphics2D. …
- ... http://docs.oracle.com/javase/7/docs/api/java/awt/Graphics.html

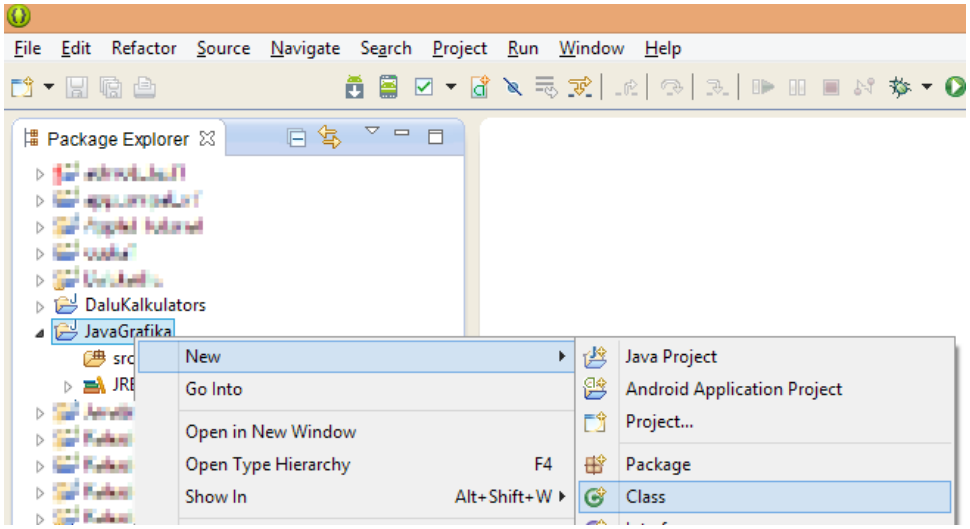
PROGRAMMA – JAVA GRAFIKA

- File/New/Java project
- Projekta nosaukums: **JavaGrafika**



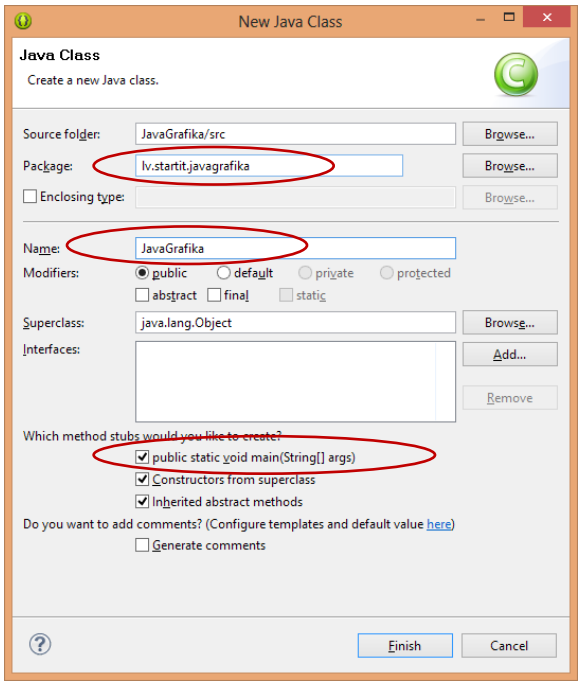
KLASE – JavaGrafika I

- Spiežot labo peles pogu uz projekta nosaukuma, izvēlas New/Class



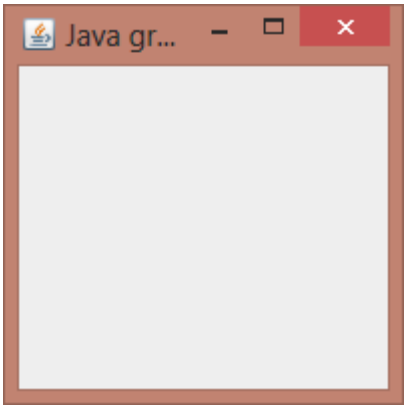
KLASE – JavaGrafika II

- Logā ievada:
  - Pakotne lv.startit.javagrafika
  - Nosaukums JavaGrafika
  - Ieslēdz izvēli public static void main



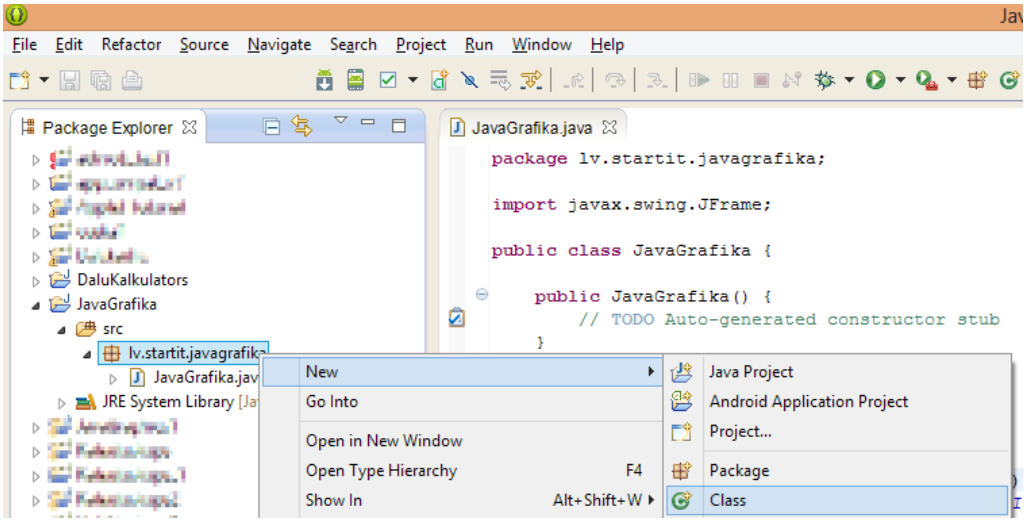
KLASES JavaGrafika KODS I

```
public class JavaGrafika {
  public static void main(String[] args) {
    // izveido ietvaru (logu)
    JFrame frame = new JFrame("Java grafika");
    // izvēlas, kas notiks, kad logu aizver
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    // uzstāda izmēru
    frame.setSize(200, 200);

    // parāda logu uz ekrāna
    frame.setVisible(true);
  }
}
```



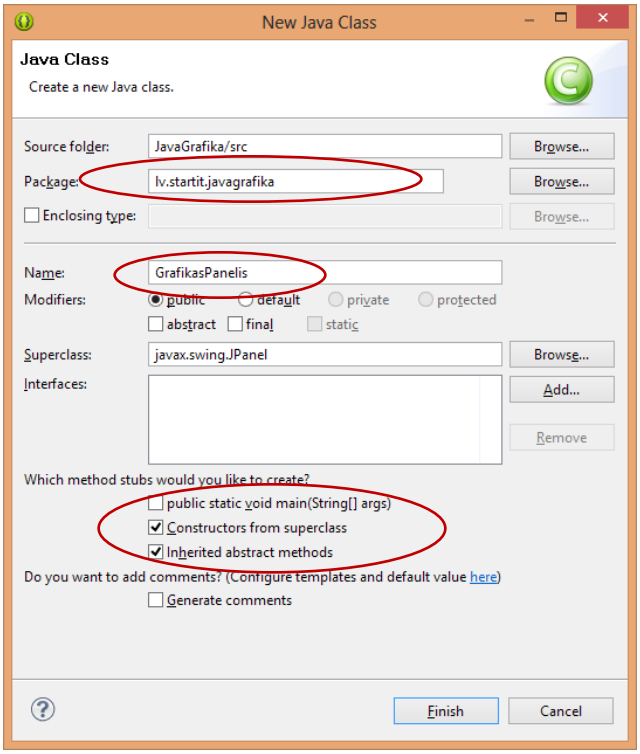
JAUNA KLASE – GrafikasPanelis

- Spiežot labo peles pogu uz pakotnes lv.startit.javagrafika, izvēlas New/Class



KLASE – GrafikasPanelis

- Logā ievada:
  - Pakotne lv.startit.javagrafika
  - Nosaukums GrafikasPanelis
  - **Superklase** javax.swing.JPanel
  - **Izslēdz** izvēli public static void main
  - Ieslēdz izvēli Constructors…



KLASES GrafikasPanelis KODS I

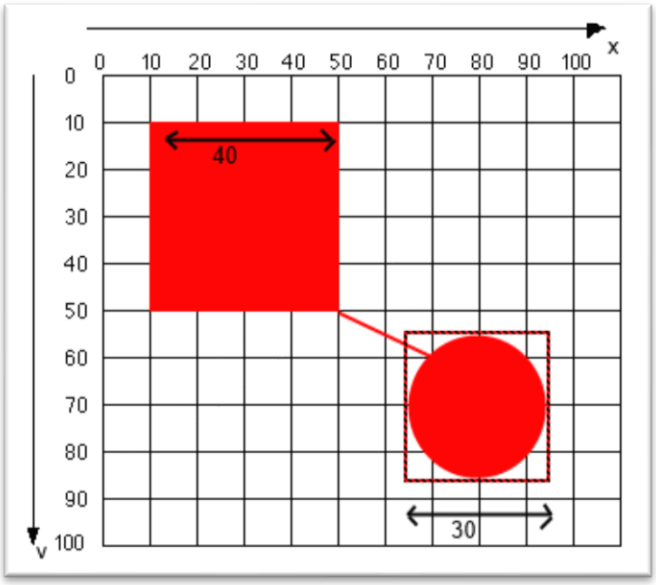
```
package lv.startit.javagrafika;
import java.awt.LayoutManager;
import javax.swing.JPanel;
public class GrafikasPanelis extends JPanel {
  public GrafikasPanelis() {
      // TODO Auto-generated constructor stub
  }
```

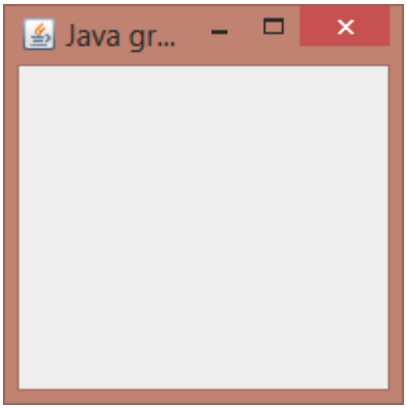
```
  public GrafikasPanelis(LayoutManager layout) {
      super(layout);
      // TODO Auto-generated constructor stub
  }
  public GrafikasPanelis(boolean isDoubleBuffered) {
      super(isDoubleBuffered);
      // TODO Auto-generated constructor stub
  }
  public GrafikasPanelis(LayoutManager layout, boolean
isDoubleBuffered) {
      super(layout, isDoubleBuffered);
      // TODO Auto-generated constructor stub
  }

}
```

KLASES GrafikasPanelis KODS II

```
package lv.startit.javagrafika;
import java.awt.LayoutManager;
import javax.swing.JPanel;
public class GrafikasPanelis extends JPanel {
  public GrafikasPanelis() {
      // TODO Auto-generated constructor stub
  }
  public GrafikasPanelis(LayoutManager layout) {
      super(layout);
      // TODO Auto-generated constructor stub
  }
  public GrafikasPanelis(boolean isDoubleBuffered) {
      super(isDoubleBuffered);
```

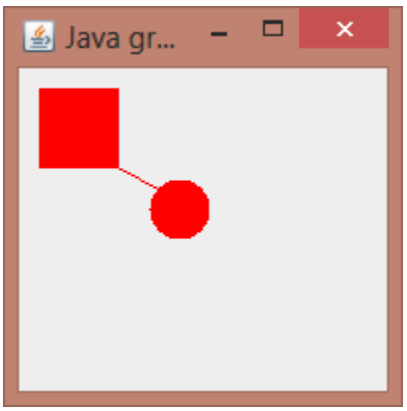
```
        // TODO Auto-generated constructor stub
    }
    public GrafikasPanelis(LayoutManager layout, boolean
isDoubleBuffered) {
        super(layout, isDoubleBuffered);
        // TODO Auto-generated constructor stub
    }
    public void paintComponent(Graphics g){
      super.paintComponent(g);
      // izvēlas krāsu
      g.setColor(Color.RED);
      // zīmē figūras
      g.fillRect(10, 10, 40, 40); // x, y, w, h
      g.drawLine(50, 50, 70, 60); // x1, y1, x2, y2
      g.fillOval(65, 55, 30, 30); // x, y, w, h
    }
}
```





KLASES JavaGrafika KODS II

```
public class JavaGrafika {
  public static void main(String[] args) {
    JFrame frame = new JFrame("Java grafika");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.setSize(200, 200);
    // izveido objektu GrafikasPanelis
    GrafikasPanelis grafika=new GrafikasPanelis();
    // pievieno paneli ievaram (galvenajam logam Jframe)
    frame.add(grafika);
    frame.setVisible(true);
  }
}
```



PIETRŪKST METODES

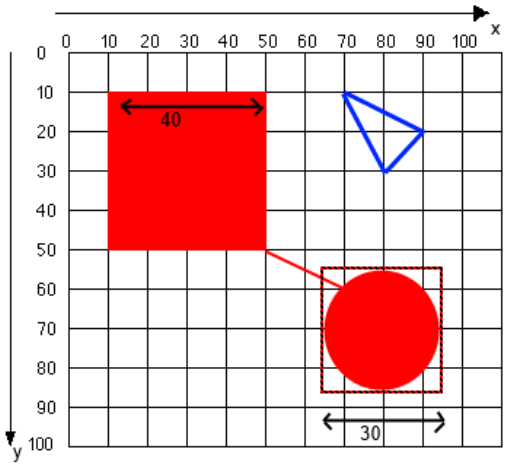
- Nav gatavu metožu:
  - trīsstūris,
  - paralelograms,
  - rombs,
  - trapece,
  - utt.
- Ko darīt?
- Veidosim paši savas metodes!

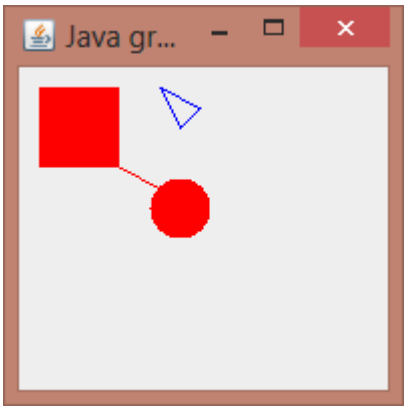
METODE drawTriangle

```
public class GrafikasPanelis extends JPanel {
. . .
  public void drawTriangle( Graphics g,
          int x1, int y1,
          int x2, int y2,
          int x3, int y3 ) {
      g.drawLine(x1, y1, x2, y2);
      g.drawLine(x2, y2, x3, y3);
      g.drawLine(x3, y3, x1, y1);
  }
}
```

TRĪSSTŪRA ZĪMĒŠANA

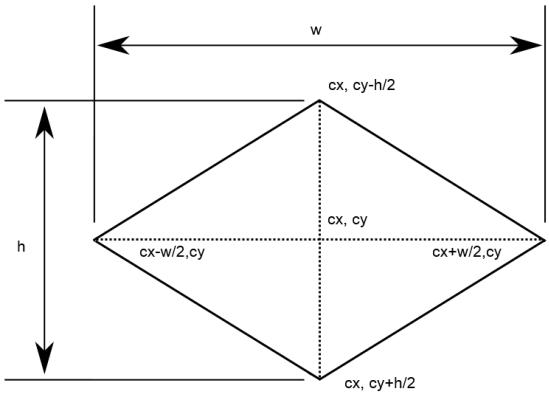
```
public void paintComponent(Graphics g) {
  super.paintComponent(g);
  g.setColor(Color.RED);
  g.fillRect(10, 10, 40, 40); // x, y, w, h
  g.drawLine(50, 50, 70, 60); // x1, y1, x2, y2
  g.fillOval(65, 55, 30, 30); // x, y, w, h
  g.setColor(Color.BLUE);
  drawTriangle(g, 70, 10, 90, 20, 80, 30);
}
```





METODE drawRhombus

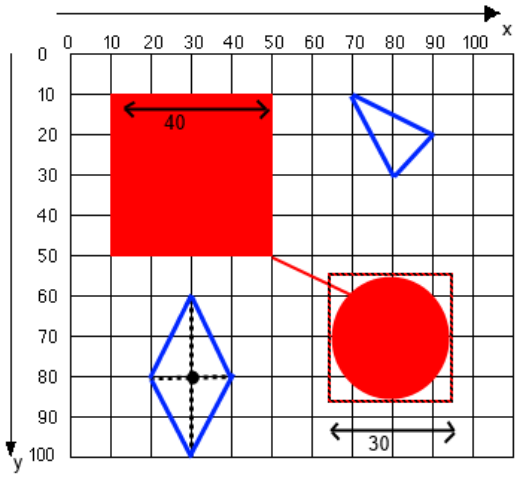
```
/* Tā, kā romba visas malas ir vienāda garuma, tad ievadīsim romba centru,
platumu un augstumu */
public class GrafikasPanelis extends JPanel {
  public void drawRhombus( Graphics g,
                           int cx, int cy,
                           int w, int h ) {
    g.drawLine( cx-w/2, cy, cx, cy-h/2 );
    g.drawLine( cx, cy-h/2, cx+w/2, cy );
    g.drawLine( cx+w/2, cy, cx, cy+h/2 );
    g.drawLine( cx, cy+h/2, cx-w/2, cy );
  }
}
```

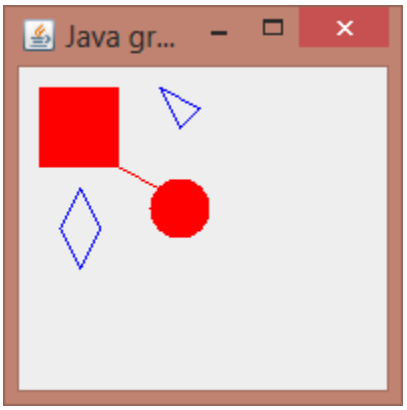


ROMBA ZĪMĒŠANA

```
public void paintComponent(Graphics g) {
  super.paintComponent(g);
  g.setColor(Color.RED);
  g.fillRect(10, 10, 40, 40); // x, y, w, h
  g.drawLine(50, 50, 70, 60); // x1, y1, x2, y2
  g.fillOval(65, 55, 30, 30); // x, y, w, h
  g.setColor(Color.BLUE);
```

```
  drawTriangle(g, 70, 10, 90, 20, 80, 30);
  drawRhombus(g, 30, 80, 20, 40 );
}
```





TRĪSSTŪRA LAUKUMS

```
public class GrafikasPanelis extends JPanel {
  public double getLineDistance( int x1, int y1,
                                 int x2, int y2 ) {
    return Math.sqrt( Math.pow(x2-x1, 2)+
              Math.pow(y2-y1, 2) );
  }

  public double getTriangleArea( int x1, int y1,
                                 int x2, int y2,
                                 int x3, int y3 ) {
    double a, b, c, p, S;
    a = getLineDistance(x1, y1, x2, y2);
    b = getLineDistance(x2, y2, x3, y3);
    c = getLineDistance(x3, y3, x1, y1);
    p = (a+b+c)/2; // pusperimetrs
    S = Math.sqrt(p*(p-a)*(p-b)*(p-c));
    return S;
  }
}
```

TRĪSSTŪRA LAUKUMA PARĀDĪŠANA

```
public void paintComponent(Graphics g) {
  super.paintComponent(g);
. . .
  g.setColor(Color.BLUE);
  drawTriangle(g, 70, 10, 90, 20, 80, 30);
  drawRhombus(g, 30, 80, 20, 40 );
  g.setColor(Color.BLACK);
  double d = getTriangleArea(70,10,90,20,80,30);
  g.drawString("S="+d, 90, 20);
}
```