

Лабораторная работа № 7

Тема: Работа с графическим интерфейсом пользователя

Цель: Разработка приложения с элементами графического интерфейса пользователя.

В работе рассматривается программа графопостроителя, демонстрирующая основы работы с графическим интерфейсом пользователя и обработкой событий компонентов. Программа графопостроителя основана на четырех классах, которые следует разместить в разных файлах. Создайте в Eclipse новый проект, а в нем новый пакет «MyGraph». Затем в данном пакете создайте четыре новых класса:

- «Graphic» — главный класс, в котором будет метод main();
- «Sinus» — класс графика синуса;
- «Cosinus» — класс графика косинуса;
- «X2» — класс графика параболы.

После создания указанных классов введите в них нижеследующий программный код.

Пример 1 . Построение графиков элементарных функций

Класс Graphic

```
package MyGraph;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class Graphic extends JFrame{
    Graphic(String s){
        super(s);
        setLayout(null);
        setSize(120,200);
        setVisible(true);
        this.setDefaultCloseOperation(EXIT_ON_CLOSE);
        this.setResizable(false);
        Button sin = new Button("Sin");
        sin.setBounds(5, 20, 100, 25);
        add(sin);
        Button cos = new Button("Cos");
        cos.setBounds(5, 70, 100, 25);
        add(cos);
        Button x2 = new Button("Парабола");
        x2.setBounds(5, 120, 100, 25);
        add(x2);
        sin.addActionListener(new ActionListener(){
            public void actionPerformed(ActionEvent event){
```

```

new Sinus("Синус");
}
});
cos.addActionListener(new ActionListener(){
public void actionPerformed(ActionEvent event){
new Cosinus("Косинус");
}
});
x2.addActionListener(new ActionListener(){
public void actionPerformed(ActionEvent event){
new X2("Парабола");
}
});
}
public static void main(String[] args) {
new Graphic("Графопостроитель");
}
}

```

Класс Sinus

```

package MyGraph;
import java.awt.Color;
import java.awt.Graphics;
import javax.swing.JFrame;
public class Sinus extends JFrame{

Sinus(String s){
super(s);
setLayout(null);
setSize(600,300);
setVisible(true);
this.setDefaultCloseOperation(DISPOSE_ON_CLOSE);
this.setResizable(false);
this.setLocation(100, 100);
}
public void paint(Graphics gr){
int y; int j=0; int k=0;
gr.setColor(Color.WHITE);
gr.fillRect(0, 0, 600, 300);
gr.setColor(Color.lightGray);
while(j<600){
gr.drawLine(j, 0, j, 300);
j+=30;
}
while(k<300){
gr.drawLine(0, k, 600, k);
k+=30;
}
gr.setColor(Color.BLACK);
gr.drawLine(300, 0, 300, 300);
gr.drawLine(0, 150, 600, 150);
gr.drawLine(120, 140, 120, 160);

```

```

gr.drawLine(480, 140, 480, 160);
gr.drawString("0", 305, 165);
gr.drawString("-"+"    ", 125, 140);
gr.drawString("    ", 485, 140);
gr.setColor(Color.RED);
for(double i=0;i<1000;i++){
y=(int)(80*Math.sin(Math.PI*i/180));
gr.drawLine((int)i-240, y+150, (int)i-240, y+150);
}
gr.dispose();
}
}

```

Класс Cosinus

```

package MyGraph;
import java.awt.Color;
import java.awt.Graphics;
import javax.swing.JFrame;
public class Cosinus extends JFrame{

Cosinus(String s){
super(s);
setLayout(null);
setSize(600,300);
setVisible(true);
this.setDefaultCloseOperation(DISPOSE_ON_CLOSE);
this.setResizable(false);
this.setLocation(200, 200);
}
public void paint(Graphics gr){
int y; int j=0; int k=0;
gr.setColor(Color.WHITE);
gr.fillRect(0, 0, 600, 300);
gr.setColor(Color.lightGray);
while(j<600){
gr.drawLine(j, 0, j, 300);
j+=30;
}
while(k<300){
gr.drawLine(0, k, 600, k);
k+=30;
}
gr.setColor(Color.BLACK);
gr.drawLine(300, 0, 300, 300);
gr.drawLine(0, 150, 600, 150);
gr.drawLine(120, 140, 120, 160);
gr.drawLine(480, 140, 480, 160);
gr.drawString("0", 305, 165);
gr.drawString("-"+"    ", 125, 140);
gr.drawString("    ", 485, 140);
gr.setColor(Color.RED);
for(double i=0;i<1000;i++){

```

```

y=(int)(80*Math.cos(Math.PI*i/180));
gr.drawLine((int)i-240, y+150, (int)i-240, y+150);
}
gr.dispose();
}
}

```

Класс X2

```

package MyGraph;
import java.awt.Color;
import java.awt.Graphics;
import javax.swing.JFrame;
public class X2 extends JFrame{

X2(String s){
super(s);
setLayout(null);
setSize(600,300);
setVisible(true);
this.setDefaultCloseOperation(DISPOSE_ON_CLOSE);
this.setResizable(false);
this.setLocation(300, 300);
}
public void paint(Graphics gr){
int y; int j=0; int k=0;
gr.setColor(Color.WHITE);
gr.fillRect(0, 0, 600, 300);
gr.setColor(Color.lightGray);
while(j<600){
gr.drawLine(j, 0, j, 300);
j+=50;
}
while(k<300){
gr.drawLine(0, k, 600, k);
k+=50;
}
gr.setColor(Color.BLACK);
gr.drawLine(300, 0, 300, 300);
gr.drawLine(0, 150, 600, 150);
gr.drawString("0", 305, 165);
gr.setColor(Color.RED);
for(double i=0;i<1000;i++){
y=-(int)(i*i/300)+150;
gr.drawLine((int)i+300, y, (int)i+300, y);
gr.drawLine(-(int)i+300, y, -(int)i+300, y);
}
gr.dispose();
}
}

```

В классе Graphic описано окно размером 120x200, в котором расположены три кнопки. Нажатие кнопки обслуживается (обрабатывается)

специальной процедурой, которая называется процедурой обработки события `ActionEvent`. При нажатии кнопки происходит создание новой формы, в которой будет построен соответствующий график элементарной функции.

Классы обработчиков событий описаны внутри параметра метода добавления слушателя `addActionListener()`. Параметром этого метода является новый объект интерфейса `ActionListener` с одним методом `actionPerformed()`, который и отвечает за поведение программы, в случае если будет вызвано данное событие (нажатие на кнопки).

В классах, которые производят построение графиков, с использованием двух циклов происходит построение координатной сетки светло-серым цветом, затем черным цветом строятся оси абсцисс и ординат. Далее следуют строки, отвечающие за рисование графиков. Для синуса и косинуса вначале вычисляется значение y и методом `drawString()` рисуется «линия» длиной в один пиксел. Координата y вычисляется путем явного преобразования типа `double` в тип `int`, полученного в результате выполнения выражения

$$80 * \text{Math.cos}(\text{Math.PI} * i / 180).$$

Парабола строится аналогичным способом, в два этапа. На первом этапе строится положительная часть параболы и далее отрицательная.

Пример 2. Надписи на кнопках

```
public class ExtEventHandler extends JFrame {  
  
    ExtEventHandler(String s){  
        super(s);  
        setLayout(null);  
        setSize(100,200);  
        setVisible(true);  
        setResizable(false);  
        setDefaultCloseOperation(EXIT_ON_CLOSE);  
        Button b1 = new Button("Первая кнопка");  
        b1.setBounds(2, 5, 96, 22);  
        add(b1);  
        Button b2 = new Button("Вторая кнопка");  
        b2.setBounds(2, 100, 96, 22);  
        add(b2);  
        b1.addActionListener(new Handler(b1, b2));  
        b2.addActionListener(new Handler(b1, b2));  
    }  
  
    public static void main(String[] args) {  
        new ExtEventHandler("");  
    }  
}  
  
class Handler implements ActionListener{  
    private Button ba;  
    private Button bb;  
    String temp;
```

```
Handler(Button b1, Button b2){
    this.ba=b1;
    this.bb=b2;
}
public void actionPerformed(ActionEvent e) {
    temp = ba.getLabel();
    ba.setLabel(bb.getLabel());
    bb.setLabel(temp);
}
}
```

В классе ExtEventHandler приводится основное описание интерфейса пользователя: форма размером 100x200 и две кнопки с надписями: первая кнопка и вторая кнопка. В приложении для обработки событий, которые возникают при нажатии на кнопки, описан отдельный класс Handler, который использует интерфейс ActionListener. Поскольку в описании ActionListener нельзя напрямую использовать компоненты, описанные в конструкторе класса ExtEventHandler, то нужно создать ссылки на объекты, которые будут использованы. Ссылки создаются в конструкторе. Когда ссылки готовы, в процессе добавления слушателей, в параметрах указываем имена реальных компонентов. Таким образом, при нажатии на любую из двух кнопок меняются местами их надписи.

Задания к работе

1. Повторите примеры теоретической части лабораторной работы.
2. Создайте приложение с одной кнопкой, при нажатии которой на отдельной форме размером 500x500 рисуется «домик».
3. Сумматор. Создайте приложение, которое представляет собой форму 200x150. На форме разместите три текстовых поля и одну кнопку с надписью «Расчет». При нажатии на кнопку значения, введенные в первые два текстовых поля, складываются и результат записывается в третье поле.