Практическая работа №2.1.

1. Обработка событий мыши.

1.1. **Задача:** Изучение теоретического материала. Закрепление полученных знаний путем разработки приложения на языке программирования Java в среде разработки eclipse.

1.2.**Тип занятий:** исследовательский.

1.3.**Содержание:** теоретический материал, задания для самостоятельного выполнения.

2.Теоретический материал к уроку

2.1. **Подключаемые библиотеки**

javax.swing - библиотека для создания графического интерфейса для программ на языке Java.

java.awt - содержит все классы для того, чтобы создать пользовательские интерфейсы и для рисования графики и изображений.

2.2. **extends** - ключевое слово для задания наследования.

пример:

public class ourclass extends JFrame {

}

2.3.  MouseAdapter это реализация интерфейса MouseListener

|  |  |  |
| --- | --- | --- |
| MouseEvent | MouseListener | mouseClicked(MouseEvent e) |
| mousePressed(MouseEvent e) |
| mouseReleased(MouseEvent e) |
| mouseEntered(MouseEvent e) |
| mouseExited(MouseEvent e) |
| MouseMotionListener | mouseDragged(MouseEvent e) |
| mouseMoved(MouseEvent e) |

3. Задания

3.1. уровень 1

1. Создайте новый проект и пакет в нем.
2. В вашем пакете создайте 3 класса: класс Main c методом main и два класса Window и Panel.
3. В класс Window импортируйте библиотеку javax.swing.\*; Сделайте класс Window наследуемым от класса JFrame и создайте его конструктор.
4. вызовите конструктор класса Window из класса Main.
5. Добавьте в конструктор класса Window следующие операторы:

//фокус на окно

setFocusable(**true**);

// Задание размеров и положения окна, первый параметр стартовая позиция //по горизонтали, вторая по вертикали, третий параметр размер в пикселях окна //по горизонтали, четвертый размер по вертикали

setBounds(0,0,800,600);

// Задание заголовка окна

setTitle("Обработка событий мыши");

// Операция завершения работы приложения при

// закрытии окна

setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

// Сделать окно видимым

setVisible(**true**);

1. Запустите программу на выполнение у вас должно выйти пустое окно.
2. Добавьте в класс Panel конструктор и импортируйте в него библиотеки javax.swing.\* и java.awt.\*.
3. Сделайте его наследуемым от класса JPanel.
4. вызовите конструктор класса Panel из класса Window до команды установки видимости окна: add(new Panel());
5. в классе Panel вне конструктора объявите 4 целочисленные переменные x1, y1, x2, y2 и одну переменную Graphics g.
6. Добавьте два метода для рисования в класс Panel:

**public** **void** paintComponent(Graphics g)

{

**super**.paintComponent(g);

}

**private** **void** setUpDrawingGraphics()

{

g = getGraphics();

}

1. добавьте встроенный класс:

**private** **class** MyMouseHandler **extends** MouseAdapter

{

**public** **void** mousePressed( MouseEvent e )

{

x1 = e.getX();

y1 = e.getY();

setUpDrawingGraphics();

x2=x1;

y2=y1;

}

**public** **void** mouseDragged( MouseEvent e )

{

x1 = e.getX();

y1 = e.getY();

g.drawLine(x1,y1,x2,y2);

x2=x1;

y2=y1;

}

}

1. Перед командой отрисовки линии установите любой цвет отличный от черного для рисования линии.
2. теперь в конструкоре класса Panel уставим цвет фона командой setBackground( здесь указать цвет ваш);

10) далее вызвать конструктор класса MyMouseHandler.

11) добавить прослушивателей:

**this**.addMouseListener( handler );

**this**.addMouseMotionListener( handler );

12) запустить, попробовать порисовать