Практическая работа №5.

1. Работа с графическими изображениями.

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. **Таймер**

Команда для задания таймера:

Timer имя\_таймера = **new** Timer(частота выполнения,**new** ActionListener() {

**public void** actionPerformed(ActionEvent e) {

блоккоманд;

}

});

Пример:Timer timer = **new** Timer(100,**new** ActionListener() {

**public void** actionPerformed(ActionEvent e) {

System.out.println("WOW!");

}

});

Запуск таймера осуществляется следующей командой: имя\_таймера.start();

Пример:timer.start();

2.2. **KeyListener – обработка событий клавиатуры**

Работая с приложением, которое имеет графический интерфейс, пользователь прибегает к помощи не только мыши, но и клавиатуры. JavaSwing даёт возможность разработчику приложения обработать различные события, которые поступают от клавиатуры в то время, когда пользователь нажимает клавиши. Давайте посмотрим, что необходимо сделать, чтобы иметь возможность слушать события клавиатуры

KeyListener имеет три метода: keyTyped, keyPressed и keyReleased.

Когда вызывается каждый из методов системой? Метод keyTyped вызывается системой каждый раз, когда пользователь нажимает на клавиатуре клавиши символы Unicode. Метод keyPressed вызывается системой в случае нажатия любой клавиши на клавиатуре. Метод keyReleased вызывается при отпускании любой клавиши на клавиатуре. Каждый раз, когда пользователь нажимает клавиши на клавиатуре и система вызывает методы keyTyped, keyPressed и keyReleased, в качестве параметра им передается объект KeyEvent, который содержит всю необходимую информацию о произошедшем событии. Отсюда можно узнать код клавиши, которая была нажата – метод getKeyCode. Стоит сказать, что события от клавиатуры будут генерироваться системой только тогда, когда компонент, который мы слушаем, находится в фокусе.

27 - код клавиши esc

37 - код клавиши стрелка влево

39 -код клавиши стрелка вправо.

2.3. **extends** это ключевое слово, предназначенное для расширения реализации какого-то существующего класса. Создается новый класс на основе существующего, и этот новый класс расширяет (extends) возможности старого.

**implements** это ключевое слово, предназначенное для реализации интерфейса (interface).

3. Задания

3.1. уровень 1

1. Откройте ваш проект Game.
2. Перед тем как добавлять таймер в ваш класс обязательно подключите следующие библиотеки:

java.awt.event.ActionEvent;

java.awt.event.ActionListener;

javax.swing.Timer;

1. В конструктор класса Pole добавьте таймер для перерисовки поля с частотой выполнения 50 (это означает игровое поле будет перерисовываться 20 раз в секунду). В блоке команд будет метод repaint(); // Запуск метода перерисовки поля.
2. Запустите таймер на выполнение.
3. Создайте класс MyKey без метода main с реализацией интерфейса KeyListener и подключите библиотеку java.awt.\*.
4. Сейчас имя класса у вас подчеркивается красным, все нормально, вы еще не добавили переопределенные методы, ничего не меняйте.
5. Объявите в классе MyKey переменную: **public** Pole gameP;
6. Добавьте метод, который срабатывает при нажатии клавиши на клавиатуре:

**public void** keyPressed(KeyEvent e)

{

// Получение кода нажатой клавиши

**int** key\_ = e.getKeyCode();

// Выход из программы, если нажат - Esc

**if** (key\_==27) System.*exit*(0);

**else if** (key\_==37) // Если нажата стрелка влево

{

// Контроль перемещения влево за пределы окна

**if** (gameP.x-30>-48) {

gameP.x-=30;

}

**else** {

gameP.x=752;

}

}

**else if** (key\_==39) // Если нажата стрелка вправо

{

// Контроль перемещения вправо за пределы окна

**if** (gameP.x+30<752) {

gameP.x+=30;

}

**else** {

gameP.x=-48;

}

}

}

**public void** keyReleased(KeyEvent e) {}

**public void** keyTyped(KeyEvent e) {}

1. В конструктор класса Window перед установкой фокуса setFocusable добавьте объявление переменной типа класса MyKey(то же самое, что вызов его конструктора) и подключите обработчик события для клавиатуры к окну следующей строчкой:addKeyListener(имя вашей переменной типа класса MyKey, которую объявили выше);

Найдите блок команд: gameP=**new** pole();

Container con = getContentPane();

con.add(gameP);.

Измените таким образом, чтобы использовалась переменная из класса MyKey, т.е. если вы завели переменную типа класса MyKey с названием mk, то ваш код примет вид:

mk.gameP=**new** pole();

Container con = getContentPane();

con.add(mk.gameP);

1. Запустите программу на выполнение и нажмите клавишу esc, осуществился ли выход из программы? если да, запустите еще раз и теперь проверьте работу клавиш стрелка влево и вправо, если ловец перемещается по экрану - все правильно.