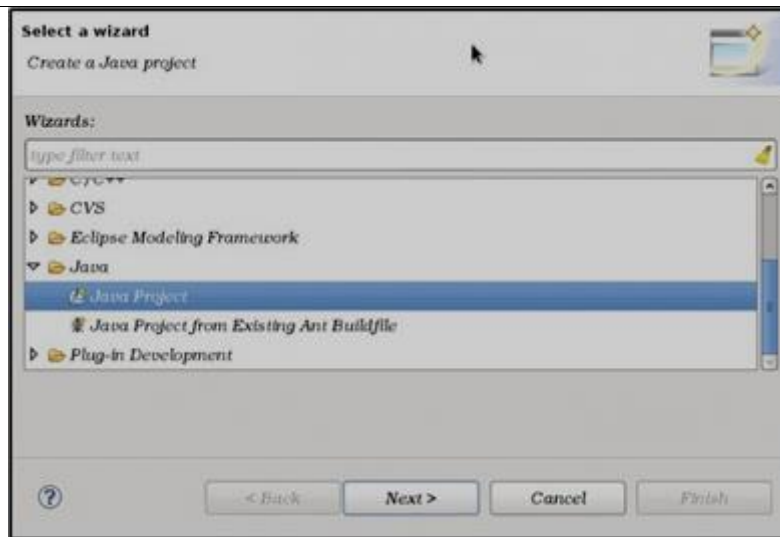
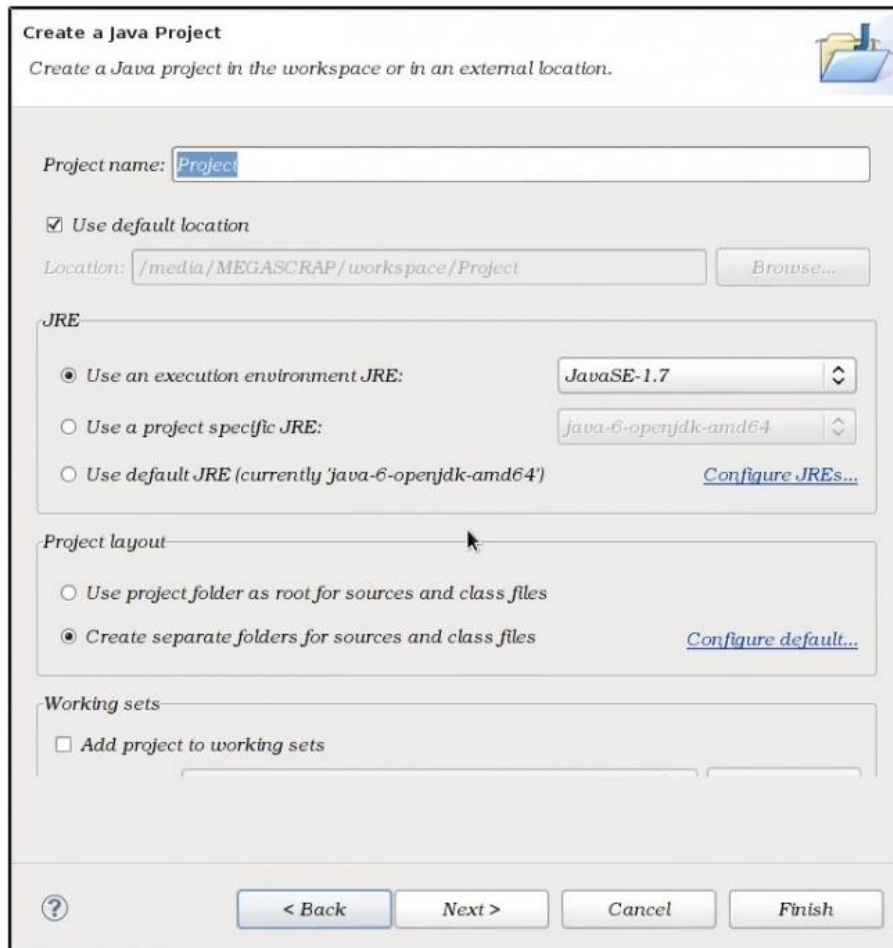


Школа:		
Дата:	ФИО учителя: Кашенова Ж.К.	
Класс: 7-8	Участвовали:	Не участвовали:
Тема урока: Работа с графическими изображениями		
Цели обучения, которые достигаются на данном уроке	Считывать, записывать и обработка изображений в среде Java	
Цели урока	<p>знать алгоритм разработки сценарии собственного проекта в игровой среде программирования, понятие анимации.</p> <p>реализация собственного проекта с добавлением звукового сопровождения и заставки в игровой среде программирования.</p> <p>улучшить собственную игру с добавлением анимации и новых объектов.</p>	
Критерии оценивания	<p>Различают вид графики по расширению файла</p> <p>Знают графические компоненты, элементы графических контекстов JFrame, JPanel, JButton, методы, свойства.</p> <p>Рисовать простейшие геометрические фигуры.</p>	
Воспитание ценностей	Уважение и сотрудничество отражается в парной работе. Самообучение, саморегуляция и творческий подход при работе над проектом.	
Предварительные знания	Работать с переменными, основные алгоритмические конструкции (ветвление, цикл), графические компоненты, операторы.	
Межпредметные связи	<p>Английский язык – работа с терминами.</p> <p>Информатика – расширения графических файлов</p>	
Запланированные этапы урока	Запланированная деятельность на уроке	Ресурсы
<p>Начало урока 2 мин.</p> <p>3 мин.</p>	<p>Организационный момент. Создание благоприятной среды на уроке, мотивация учащихся на успех.</p> <p>Проверка знаний. Метод: «Интерактивное тестирование».</p> <p>Цель: повторить графические компоненты, операторы</p> <p>Способы дифференциации: Темп</p> <p>Критерии оценивания: Знают графические компоненты, операторы</p> <p>Уровень мыслительных навыков: понимание, применение</p> <p>Задание: (И) Отвечают на вопросы теста.</p> <p>Дескриптор: - определяет растровые изображения, - определяет назначение методов setColor, drawLine, setBackground, drawRect, drawRect, drawOval, drawArc. - объясняет аргументы конструктора new Color(0, 0, 255)</p>	Слайды 1-5
Середина урока 33 мин.	<p>Практическая работа</p> <p>Среда программирования: Eclipse</p> <p>Eclipse - кроссплатформенная среда разработки, поддерживает множество языков программирования, среди которых есть и Java. Данный урок будет посвящён написанию простых графических приложений на языке Java в Eclipse.</p> <p>Для начала вам нужно загрузить и установить Eclipse. Распакуйте архив в удобное для вас место. После распаковки программа готова к работе, если на вашем компьютере установлена Java.</p>	



Запустите Eclipse и создайте новый проект: File>New>Project>Java Project.

Выберите версию Java, для которой будут создаваться проекты, если у вас установлено несколько версий. Введите имя проекта и нажмите Finish.



Создайте новый класс. Этот класс будет запускаться при запуске приложения, прорисовывать окно и элементы на нём.

Java Class

The use of the default package is discouraged.

Source folder:

Package:

Enclosing type:

Name:

Modifiers: public default private protected
 abstract final static

Superclass:

Interfaces:

Which method stubs would you like to create?

`public static void main(String[] args)`

Constructors from superclass

Inherited abstract methods

Do you want to add comments? (Configure templates and default value [here](#))

Generate comments

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

public class Window{
    public static void main(String[] args) {

        /* Задание заголовка окна*/
        JFrame w=new JFrame ("Окно с изображением");
        /*Задание размеров окна*/
        w.setSize(400, 400);

        /*
         * Если у окна не будет функции закрытия,
         * при нажатии крестика окно не закроется.*/
        w.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        /*
         * Менеджер определяет
         * каким образом в окне расположены объекты.*/
        w.setLayout(new BorderLayout(1,1));
    }
}
```

Таким образом будет описано окно, но для того, чтобы оно появилось на экране, нужно задать настройки видимости. Теперь добавьте то, что будете рисовать в окне. Для этого создайте новый класс в этом же файле, расширяющий класс `JComponent`. В `JComponent` описываются основные методы для объектов `java.swing`. Для `Graphics2D` описаны методы, позволяющие рисовать линии, заливать области. Так для рисования прямоугольника используется метод `drawRect()`.

```
/* Класс, который будет рисовать элементы*/
class Canvas extends JComponent{

    /*Метод, перерисовывающий элемент внутри окна
    *при обновлении*/
    public void paintComponent(Graphics g){
        super.paintComponents(g);
        Graphics2D g2d=(Graphics2D)g;

        /*
         * Устанавливает цвет рисования в зелёный*/
    }
}
```

```

g2d.setPaint(Color.GREEN);

/*   Рисует текущим цветом прямоугольник   */
g2d.drawRect(100, 100, 80, 20);

g2d.setPaint(Color.RED);
/*   Рисует текущим цветом в координатах (150,150)
строку "привет мир"*/
g2d.drawString("Привет мир", 150, 150);

g2d.setColor(Color.blue);
/*   Рисует текущим цветом овал в координатах (200,50)*/
g2d.fillOval(200, 50, 50, 20);

/*   Вызывает обновление себя после завершения рисования
super.repaint();
}
}

```

Теперь есть компонент, который будет прорисовываться при обращении к нему. Осталось прикрепить его к окну. Для этого в методе main добавьте элемент Canvas к окну и сделайте окно видимым.

```

Canvas canv=new Canvas();
w.add(canv);
w.setVisible(true);
}
}

```

После сохранения файла можно запустить программу, выбрав в меню Run>Run либо нажав комбинацию ctrl+F11.
Результат:



Создайте рисунок «Кораблик»

Конец урока
2 мин.

Рефлексия

Учащимся предлагается оценить свою работу с помощью приема «Лесница успеха», прикрепив стикер на лесницу успеха, отметив ступень на которой он находится.



Дифференциация – каким способом вы хотите больше оказывать поддержку? Какие задания вы даете ученикам более способным по сравнению с другими?	Оценивание – как Вы планируете проверять уровень освоения материала учащимися?	Охрана здоровья и соблюдение техники безопасности