

```
/// src/playground/PlaygroundGUI.java
```

```
package playground;

import being.insects.Cockroach;
import being.insects.Karankul;
import being.insects.Tarantul;
import beings.Being;
import beings.animals.Fox;
import beings.animals.GenericCat;
import beings.animals.Pets;
import beings.animals.my.Labrador;
import beings.animals.my.SphinxCat;
import beings.birds.Goose;
import beings.fish.Paltus;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.IOException;
import javax.swing.JOptionPane;
import javax.swing.Timer;

public class PlaygroundGUI extends javax.swing.JFrame {

    private Being[] b;
    private int cur_index = 0;
    private Timer st;

    /** Creates new form PlaygroundGUI */
    public PlaygroundGUI(Being[] bs) {
        initComponents();
        b = bs;
        st = createSleepTimer();
        st.start();
        displayBeing();
    }

    public Timer createSleepTimer(){
        Timer t = new Timer(1000, new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                long t = System.currentTimeMillis() % (72000);
                int h = (int) (t / 3000);
                int m = (int) (t / 50 % 60);
                timeField.setText(h + ":" + m);
                updateStatus((int)(t / 50));
            }
        });
        return t;
    }

    public void displayBeing()
    {
        Being bg = b[cur_index];
        beingNameField.setText(bg.getBeingName());
        nameField.setText(bg.getName());
        colorField.setText(bg.getColor());
        imageComponent.setIcon(bg.getImg());
        if (bg instanceof Pets)
        {
            sB.setEnabled(true);
        }
        else
        {
            sB.setEnabled(false);
        }
    }
}
```



```

}

public void updateStatus(int t){
    if (b[cur_index].isSleeping(t)){
        statusComponent.setText("Sleeping");
    }
    else {

        statusComponent.setText("Active");
    }
}

/** This method is called from within the constructor to
 * initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is
 * always regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code"> //GEN-BEGIN: initComponents
private void initComponents() {

    imageComponent = new javax.swing.JLabel();
    bL = new javax.swing.JLabel();
    beingNameField = new javax.swing.JTextField();
    nameL = new javax.swing.JLabel();
    nameField = new javax.swing.JTextField();
    colorL = new javax.swing.JLabel();
    colorField = new javax.swing.JTextField();
    sL = new javax.swing.JLabel();
    statusComponent = new javax.swing.JLabel();
    nextB = new javax.swing.JButton();
    prevB = new javax.swing.JButton();
    sB = new javax.swing.JButton();
    intrB = new javax.swing.JButton();
    jLabel1 = new javax.swing.JLabel();
    timeField = new javax.swing.JTextField();
    jLabel2 = new javax.swing.JLabel();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
    setTitle("Beings");

    bL.setText("Being Name:");

    beingNameField.setEditable(false);

    nameL.setText("Name:");

    nameField.setEditable(false);

    colorL.setText("Color:");

    colorField.setEditable(false);

    sL.setText("Status:");

    statusComponent.setText("Unknown");

    nextB.setText("Next >");
    nextB.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            nextBActionPerformed(evt);
        }
    });
}

```



```

prevB.setText("< Previos");
prevB.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        prevBActionPerformed(evt);
    }
});

sB.setText("Say");
sB.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        sBActionPerformed(evt);
    }
});

intrB.setText("Intr");
intrB.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        intrBActionPerformed(evt);
    }
});

jLabel1.setText("Time:");

timeField.setEditable(false);

jLabel2.setText("<html>Samarin Arthur 8<sup>2</sup> 2010");

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
            layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(sL)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addComponent(intrB, javax.swing.GroupLayout.PREFERRED_SIZE,
                    javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addComponent(sB, javax.swing.GroupLayout.PREFERRED_SIZE,
                    javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addComponent(prevB)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addComponent(nextB)
                .addContainerGap()
        )
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
            layout.createSequentialGroup()
                .addComponent(nameL)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addComponent(colorL)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addComponent(bL)
                .addContainerGap()
        )
        .addGroup(javax.swing.GroupLayout.Alignment.LEADING,
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                .addComponent(colorField, javax.swing.GroupLayout.Alignment.TRAILING,
                    javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addComponent(nameField, javax.swing.GroupLayout.Alignment.TRAILING,
                    javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addComponent(beingNameField,
                    javax.swing.GroupLayout.Alignment.TRAILING,

```



```

        javax.swing.GroupLayout.DEFAULT_SIZE, 416, Short.MAX_VALUE))
        .addContainerGap())
    .addGroup(layout.createSequentialGroup())
        .addComponent(jLabel1)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(timeField, javax.swing.GroupLayout.PREFERRED_SIZE, 198,
            javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jLabel2, javax.swing.GroupLayout.DEFAULT_SIZE, 254,
            Short.MAX_VALUE)
        .addContainerGap())
    .addComponent(imageComponent, javax.swing.GroupLayout.DEFAULT_SIZE, 523,
        Short.MAX_VALUE)
);
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addContainerGap()

.addGroup(layout.createSequentialGroup()
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(timeField, javax.swing.GroupLayout.PREFERRED_SIZE,
                    javax.swing.GroupLayout.DEFAULT_SIZE,
                    javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jLabel1)
                .addComponent(jLabel2))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(beingNameField, javax.swing.GroupLayout.PREFERRED_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE,
                javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(bL))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(nameField, javax.swing.GroupLayout.PREFERRED_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE,
                javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(nameL))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(colorField, javax.swing.GroupLayout.PREFERRED_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE,
                javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(colorL))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(sL)
            .addComponent(statusComponent,
                javax.swing.GroupLayout.PREFERRED_SIZE, 17, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(intrB)
            .addComponent(nextB)
            .addComponent(prevB)
            .addComponent(sB))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(imageComponent, javax.swing.GroupLayout.DEFAULT_SIZE, 193,
                Short.MAX_VALUE))
);
pack();
} // </editor-fold> // GEN-END: initComponents

```



```

private void intrBActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_intrBActionPerformed
    JOptionPane.showMessageDialog(this,
        b[cur_index].introduce(),
        "Introduce", JOptionPane.INFORMATION_MESSAGE);
} //GEN-LAST:event_intrBActionPerformed

private void sBActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_sBActionPerformed
    Being cb = b[cur_index];
    if (cb instanceof Pets)
    {
        JOptionPane.showMessageDialog(this, ((Pets) cb).say(), "Say",
            JOptionPane.INFORMATION_MESSAGE);
        if (cb.getAc() != null)
        {
            cb.getAc().play();
        }
    }
} //GEN-LAST:event_sBActionPerformed

private void nextBActionPerformed(java.awt.event.ActionEvent evt) {
    //GEN-FIRST:event_nextBActionPerformed
    cur_index = (cur_index + 1) % b.length;
    displayBeing();
} //GEN-LAST:event_nextBActionPerformed

private void prevBActionPerformed(java.awt.event.ActionEvent evt) {
    //GEN-FIRST:event_prevBActionPerformed
    cur_index--;
    if (cur_index < 0) cur_index = b.length - 1;
    displayBeing();
} //GEN-LAST:event_prevBActionPerformed

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {

        public void run() {
            try {
                Being[] beings = new Being[9];
                beings[0] = new Labrador("Mark");
                beings[0].loadImg("/img/dog.jpeg");
                beings[0].loadClip(PlaygroundGUI.class.
                    getResource("/sound/dog.wav"));
                beings[1] = new GenericCat("Ammi", "Black");
                beings[1].loadImg("/img/cat.jpg");
                beings[1].loadClip(PlaygroundGUI.class.
                    getResource("/sound/cat.wav"));
                beings[2] = new SphinxCat("Faraon");
                beings[2].loadImg("/img/sf.jpeg");
                beings[2].loadClip(PlaygroundGUI.class.
                    getResource("/sound/cat.wav"));
                beings[3] = new Goose("WhiteGoose", "White");
                beings[3].loadImg("/img/gus.jpeg");
                beings[3].loadClip(PlaygroundGUI.class.
                    getResource("/sound/gus.wav"));
                beings[4] = new Tarantul("Tarantul");
                beings[4].loadImg("/img/tarantul.png");
                beings[5] = new Karankul("Karankul");
                beings[5].loadImg("/img/dog.jpeg");
                beings[6] = new Fox("Firefox");
                beings[6].loadImg("/img/fox.jpeg");
            }
        }
    });
}

```



```

        beings[7] = new Cockroach("Tar", "Red");
        beings[7].loadImg("/img/tarak.jpeg");
        beings[8] = new Paltus("Paltuz");
        beings[8].loadImg("/img/paltus.jpeg");
        new PlaygroundGUI(beings).setVisible(true);
    } catch (IOException ex) {
        ex.printStackTrace();
    }
}
});
}
// Variables declaration - do not modify//GEN-BEGIN:variables
private javax.swing.JLabel bL;
private javax.swing.JTextField beingNameField;
private javax.swing.JTextField colorField;
private javax.swing.JLabel colorL;
private javax.swing.JLabel imageComponent;
private javax.swing.JButton intrB;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JTextField nameField;
private javax.swing.JLabel nameL;
private javax.swing.JButton nextB;
private javax.swing.JButton prevB;
private javax.swing.JButton sB;
private javax.swing.JLabel sL;
private javax.swing.JLabel statusComponent;
private javax.swing.JTextField timeField;
// End of variables declaration//GEN-END:variables
}

```

## <!-- PlaygroundGUI.form -->

```
<?xml version="1.0" encoding="UTF-8" ?>
```

```

<Form version="1.3" maxVersion="1.7" type="org.netbeans.modules.form.forminfo.JFrameFormInfo">
  <Properties>
    <Property name="defaultCloseOperation" type="int" value="3"/>
    <Property name="title" type="java.lang.String" value="Beings"/>
  </Properties>
  <SyntheticProperties>
    <SyntheticProperty name="formSizePolicy" type="int" value="1"/>
  </SyntheticProperties>
  <AuxValues>
    <AuxValue name="FormSettings_autoResourcing" type="java.lang.Integer" value="0"/>
    <AuxValue name="FormSettings_autoSetComponentName" type="java.lang.Boolean" value="false"/>
    <AuxValue name="FormSettings_generateFQN" type="java.lang.Boolean" value="true"/>
    <AuxValue name="FormSettings_generateMnemonicsCode" type="java.lang.Boolean" value="false"/>
    <AuxValue name="FormSettings_i18nAutoMode" type="java.lang.Boolean" value="false"/>
    <AuxValue name="FormSettings_layoutCodeTarget" type="java.lang.Integer" value="1"/>
    <AuxValue name="FormSettings_listenerGenerationStyle" type="java.lang.Integer" value="0"/>
    <AuxValue name="FormSettings_variablesLocal" type="java.lang.Boolean" value="false"/>
    <AuxValue name="FormSettings_variablesModifier" type="java.lang.Integer" value="2"/>
  </AuxValues>

  <Layout>
    <DimensionLayout dim="0">
      <Group type="103" groupAlignment="0" attributes="0">
        <Group type="102" alignment="1" attributes="0">
          <EmptySpace pref="70" max="32767" attributes="0"/>
          <Component id="sL" min="-2" max="-2" attributes="0"/>
          <EmptySpace type="unrelated" max="-2" attributes="0"/>
          <Component id="statusComponent" min="-2" pref="66" max="-2" attributes="0"/>
          <EmptySpace max="-2" attributes="0"/>
          <Component id="intrB" min="-2" pref="84" max="-2" attributes="0"/>
          <EmptySpace max="-2" attributes="0"/>
          <Component id="sB" min="-2" pref="73" max="-2" attributes="0"/>
          <EmptySpace max="-2" attributes="0"/>
          <Component id="prevB" min="-2" max="-2" attributes="0"/>
        </Group>
      </Group>
    </DimensionLayout>
  </Layout>
</Form>

```



```

<EmptySpace max="-2" attributes="0"/>
<Component id="nextB" min="-2" max="-2" attributes="0"/>
<EmptySpace max="-2" attributes="0"/>
</Group>
<Group type="102" alignment="1" attributes="0">
  <Group type="103" groupAlignment="0" attributes="0">
    <Component id="nameL" alignment="0" min="-2" max="-2" attributes="0"/>
    <Component id="colorL" min="-2" max="-2" attributes="0"/>
    <Component id="bL" min="-2" max="-2" attributes="0"/>
  </Group>
  <EmptySpace max="-2" attributes="0"/>
  <Group type="103" groupAlignment="0" attributes="0">
    <Component id="colorField" alignment="1" pref="416" max="32767" attributes="0"/>
    <Component id="nameField" alignment="0" pref="416" max="32767" attributes="0"/>
    <Component id="beingNameField" alignment="1" pref="416" max="32767"
      attributes="0"/>
  </Group>
  <EmptySpace min="-2" max="-2" attributes="0"/>
</Group>
<Group type="102" alignment="0" attributes="0">
  <Component id="jLabel1" min="-2" max="-2" attributes="0"/>
  <EmptySpace max="-2" attributes="0"/>
  <Component id="timeField" min="-2" pref="198" max="-2" attributes="0"/>
  <EmptySpace max="-2" attributes="0"/>
  <Component id="jLabel2" pref="254" max="32767" attributes="0"/>
  <EmptySpace max="-2" attributes="0"/>
</Group>
  <Component id="imageComponent" alignment="0" pref="523" max="32767" attributes="0"/>
</Group>
</DimensionLayout>
<DimensionLayout dim="1">
  <Group type="103" groupAlignment="0" attributes="0">
    <Group type="102" alignment="0" attributes="0">
      <EmptySpace max="-2" attributes="0"/>
      <Group type="103" groupAlignment="3" attributes="0">
        <Component id="timeField" alignment="3" min="-2" max="-2" attributes="0"/>
        <Component id="jLabel1" alignment="3" min="-2" max="-2" attributes="0"/>
        <Component id="jLabel2" alignment="3" min="-2" max="-2" attributes="0"/>
      </Group>
      <EmptySpace max="-2" attributes="0"/>
      <Group type="103" groupAlignment="3" attributes="0">
        <Component id="beingNameField" alignment="3" min="-2" max="-2" attributes="0"/>
        <Component id="bL" alignment="3" min="-2" max="-2" attributes="0"/>
      </Group>
      <EmptySpace max="-2" attributes="0"/>
      <Group type="103" groupAlignment="3" attributes="0">
        <Component id="nameField" alignment="3" min="-2" max="-2" attributes="0"/>
        <Component id="nameL" alignment="3" min="-2" max="-2" attributes="0"/>
      </Group>
      <EmptySpace max="-2" attributes="0"/>
      <Group type="103" groupAlignment="3" attributes="0">
        <Component id="colorField" alignment="3" min="-2" max="-2" attributes="0"/>
        <Component id="colorL" alignment="3" min="-2" max="-2" attributes="0"/>
      </Group>
      <EmptySpace max="-2" attributes="0"/>
      <Group type="103" groupAlignment="3" attributes="0">
        <Component id="sL" alignment="3" min="-2" max="-2" attributes="0"/>
        <Component id="statusComponent" alignment="3" min="-2" pref="17" max="-2"
          attributes="0"/>
        <Component id="intrB" alignment="3" min="-2" max="-2" attributes="0"/>
        <Component id="nextB" alignment="3" min="-2" max="-2" attributes="0"/>
        <Component id="prevB" alignment="3" min="-2" max="-2" attributes="0"/>
        <Component id="sB" alignment="3" min="-2" max="-2" attributes="0"/>
      </Group>
      <EmptySpace max="-2" attributes="0"/>
      <Component id="imageComponent" pref="193" max="32767" attributes="0"/>
    </Group>
  </Group>
</DimensionLayout>
</Layout>
<SubComponents>
  <Component class="javax.swing.JLabel" name="imageComponent">

```



```

</Component>
<Component class="javax.swing.JLabel" name="bL">
  <Properties>
    <Property name="text" type="java.lang.String" value="Being Name:"/>
  </Properties>
</Component>
<Component class="javax.swing.JTextField" name="beingNameField">
  <Properties>
    <Property name="editable" type="boolean" value="false"/>
  </Properties>
</Component>
<Component class="javax.swing.JLabel" name="nameL">
  <Properties>
    <Property name="text" type="java.lang.String" value="Name:"/>
  </Properties>
</Component>
<Component class="javax.swing.JTextField" name="nameField">
  <Properties>
    <Property name="editable" type="boolean" value="false"/>
  </Properties>
</Component>
<Component class="javax.swing.JLabel" name="colorL">
  <Properties>
    <Property name="text" type="java.lang.String" value="Color:"/>
  </Properties>
</Component>
<Component class="javax.swing.JTextField" name="colorField">
  <Properties>
    <Property name="editable" type="boolean" value="false"/>
  </Properties>
</Component>
<Component class="javax.swing.JLabel" name="sL">
  <Properties>
    <Property name="text" type="java.lang.String" value="Status:"/>
  </Properties>
</Component>
<Component class="javax.swing.JLabel" name="statusComponent">
  <Properties>
    <Property name="text" type="java.lang.String" value="Unknown"/>
  </Properties>
</Component>
<Component class="javax.swing.JButton" name="nextB">
  <Properties>
    <Property name="text" type="java.lang.String" value="Next &gt;"/>
  </Properties>
  <Events>
    <EventHandler event="actionPerformed" listener="java.awt.event.ActionListener"
      parameters="java.awt.event.ActionEvent" handler="nextBActionPerformed"/>
  </Events>
</Component>
<Component class="javax.swing.JButton" name="prevB">
  <Properties>
    <Property name="text" type="java.lang.String" value="&lt; Previos"/>
  </Properties>
  <Events>
    <EventHandler event="actionPerformed" listener="java.awt.event.ActionListener"
      parameters="java.awt.event.ActionEvent" handler="prevBActionPerformed"/>
  </Events>
</Component>
<Component class="javax.swing.JButton" name="sB">
  <Properties>
    <Property name="text" type="java.lang.String" value="Say"/>
  </Properties>
  <Events>
    <EventHandler event="actionPerformed" listener="java.awt.event.ActionListener"
      parameters="java.awt.event.ActionEvent" handler="sBActionPerformed"/>
  </Events>
</Component>
<Component class="javax.swing.JButton" name="intrB">
  <Properties>
    <Property name="text" type="java.lang.String" value="Intr"/>
  </Properties>

```





```

<Events>
  <EventHandler event="actionPerformed" listener="java.awt.event.ActionListener"
    parameters="java.awt.event.ActionEvent" handler="intrBActionPerformed"/>
</Events>
</Component>
<Component class="javax.swing.JLabel" name="jLabel1">
  <Properties>
    <Property name="text" type="java.lang.String" value="Time:" />
  </Properties>
</Component>
<Component class="javax.swing.JTextField" name="timeField">
  <Properties>
    <Property name="editable" type="boolean" value="false"/>
  </Properties>
</Component>
<Component class="javax.swing.JLabel" name="jLabel2">
  <Properties>
    <Property name="text" type="java.lang.String"
      value="&lt;html&gt;Samarin Arthur 8&lt;sup&gt;2&lt;/sup&gt; 2010"/>
  </Properties>
</Component>
</SubComponents>
</Form>

```

```

/// src/beings/Being.java

```

```

package beings;

import java.applet.Applet;
import java.applet.AudioClip;
import java.io.IOException;
import java.net.URL;
import javax.swing.ImageIcon;

public class Being implements BeingDescription {

    public String beingName = "Being";
    public String color = "Natural";
    public String sound = "Mute";
    public String name = "Generic";
    public ImageIcon img = null;
    public AudioClip ac = null;

    public AudioClip getAc() {
        return ac;
    }

    public void setAc(AudioClip ac) {
        this.ac = ac;
    }

    public ImageIcon getImg() {
        return img;
    }

    public void setImg(ImageIcon img) {
        this.img = img;
    }

    public String getBeingName() {
        return beingName;
    }

    public String getColor() {
        return color;
    }

    public String getSound() {

```



```

    return sound;
}

public String getName() {
    return name;
}

public String introduce() {
    return ("I am a " + getBeingName() + " of " + getColor() +
        " color. My name is " + getName() + ". I can say \"" +
        getSound() + "\".");
}

public void loadImg(String path) {
    img = new ImageIcon(getClass().getResource(path));
}

public void loadClip(URL url) throws IOException {
    ac = Applet.newAudioClip(url);
}

public boolean isSleeping(int time) {
    return false;
}
}

```

```

/// src/beings/BeingDescription.java

```

```

package beings;

public interface BeingDescription {

    public String getBeingName();

    public String getColor();

    public String getSound();

    public String getName();
}

```

```

/// src/beings/animals/Dog.java

```

```

package beings.animals;

import beings.*;

public class Dog extends Being implements Pets {
    public Dog(String n) {
        name = n;
        beingName = "Dog";
        sound = "GAV-GAV";
        System.out.println(say());
    }
    public String say() {
        return (getName()+": "+sound+" "+sound+" "+sound+"!!!");
    }
    @Override
    public boolean isSleeping(int time)
    {
        return time > 1320 || time < 360;
    }
}

```



```
/// src/beings/animals/Fox.java
```

```
package beings.animals;

import beings.Being;

public class Fox extends Being implements Wild{

    public Fox(String n) {
        name = n;
        beingName = "Fox";
        sound = "Rrrr";
        color = "Red";
    }

    @Override
    public boolean isSleeping(int time)
    {
        return time > 1320 || time < 360;
    }
}
```

```
/// src/beings/animals/GenericCat.java
```

```
package beings.animals;

import beings.Being;

public class GenericCat extends Being implements Pets{

    public GenericCat(String n, String c) {
        name = n;
        beingName = "Generic Cat";
        sound = "Myau-myau";
        color = c;
        System.out.println(say());
    }

    public String say() {
        return (getName()+" : "+sound+" "+sound+"!!!");
    }

    @Override
    public boolean isSleeping(int time)
    {
        return time > 1320 || time < 360;
    }
}
```

```
/// src/beings/animals/Pets.java
```

```
package beings.animals;

public interface Pets {
    public String say();
}
```

```
/// src/beings/animals/Wild.java
```

```
package beings.animals;

public interface Wild {
}
```



### /// src/beings/animals/my/Labrador.java

```
package beings.animals.my;

import beings.animals.*;

public class Labrador extends Dog {

    public Labrador(String name) {
        super(name);
        color = "yellow";
    }
}
```

### /// src/beings/animals/my/SphinxCat.java

```
package beings.animals.my;

import beings.Being;
import beings.animals.Pets;

public class SphinxCat extends Being implements Pets{
    public SphinxCat(String n) {
        name = n;
        beingName = "Sphinx";
        color = "Yellow";
        sound= "Myau, Myau!";
    }

    @Override
    public boolean isSleeping(int time)
    {
        return time > 1320 || time < 360;
    }

    public String say() {
        return (getName()+" : "+sound+" "+sound+"!!!");
    }
}
```

### /// src/beings/birds/Goose.java

```
package beings.birds;

import beings.Being;
import beings.animals.Pets;

public class Goose extends Being implements Pets{

    public Goose(String n, String c) {
        name = n;
        beingName = "Goose";
        color = c;
        sound = "Ga-ga-ga";
        System.out.println(say());
    }
    public String say() {
        return (getName()+" : "+sound + "!!!");
    }
    @Override
    public boolean isSleeping(int time)
    {
        return time > 1380 || time < 300;
    }
}
```



```
/// src/beings/fish/Paltus.java
```

```
package beings.fish;

import beings.Being;
import beings.animals.Wild;

public class Paltus extends Being implements Wild{
    public Paltus(String n) {
        name = n;
        color = "Black";
        beingName = "Paltus";
    }

    @Override
    public boolean isSleeping(int time)
    {
        return time > 1380 || time < 360;
    }
}
```

```
/// src/being/insects/Cockroach.java
```

```
package being.insects;

import beings.Being;
import beings.animals.Wild;

public class Cockroach extends Being implements Wild {

    public Cockroach(String n, String c) {
        name = n;
        color = c;
        beingName = "Cockroach";
    }

    @Override
    public boolean isSleeping(int time)
    {
        return time < 350;
    }
}
```

```
/// src/being/insects/Karankul.java
```

```
package being.insects;

import beings.Being;
import beings.animals.Wild;

public class Karankul extends Being implements Wild{

    public Karankul(String n) {
        name = n;
        color = "Black";
        beingName = "Karankul";
    }

    @Override
    public boolean isSleeping(int time)
    {
        return time < 300;
    }
}
```



```
/// src/being/insects/Tarantul.java
```

```
package being.insects;
```

```
import beings.Being;
```

```
import beings.animals.Wild;
```

```
public class Tarantul extends Being implements Wild{
```

```
    public Tarantul(String n)
```

```
    {
```

```
        name = n;
```

```
        color = "Brown";
```

```
        beingName = "Tarantul";
```

```
    }
```

```
@Override
```

```
public boolean isSleeping(int time)
```

```
{
```

```
    return time < 200;
```

```
}
```

```
}
```



## Домашняя работа по первой лекции JavaME

### Задача 1

=====

Скачайте спецификации MIDP 2.0 и MMAPI 1.1

- **Спецификация MIDP 2.0:** <http://jcp.org/en/jsr/summary?id=118>
- **MMAPI 1.1:** <http://jcp.org/en/jsr/summary?id=135>

Найдите визуально, какие классы в пересечении этих API.

Какие это классы?

**Пакет javax.microedition.media:**

**интерфейсы** [Control](#) [Controllable](#) [Player](#) [PlayerListener](#)  
**классы** [Manager](#) и [MediaException](#)

**пакет** javax.microedition.media.control

**Интерфейсы** ToneControl, VolumeControl

### Задача 2

=====

Скачайте спецификации CHAPI 1.0 (JSR 211)

<http://jcp.org/en/jsr/summary?id=211>

Скачайте спецификации CLDC 1.1 (JSR 139)

<http://jcp.org/en/jsr/summary?id=139>

Скачайте спецификации CDC 1.1 (JSR 218)

<http://jcp.org/en/jsr/summary?id=218>

Ответьте на следующие вопросы:

1. Возможна ли реализация CHAPI 1.0 на устройстве с конфигурацией CLDC 1.1

**Да, возможна! В спецификации CHAPI говорится следующее:**

**«Configuration (CLDC/CDC):** A J2ME configuration is defined as a set of optimized Java APIs, class libraries, and a virtual machine which targets a family of devices with similar size and capability requirements. CHAPI is compatible with all versions of the Connected Limited Device Configuration (CLDC) and the Connected Device Configuration (CDC). »

2. Возможна ли реализация CHAPI 1.0 на устройстве с конфигурацией CDC 1.1

**Да, возможна! См выше**

### Задача 3

=====

- Присутствует ли интерфейс java.lang.Serializable в CLDC 1.1?

**нет, скачав спецификацию CLDC 1.1 вы сможете в этом убедиться.**

### Задача 4

=====

Попытайтесь найти опциональный пакет JavaME, который требует наличие конфигурации CDC (но не разрешает CLDC)

1. **Выберем список всех JavaME JSR-ов,**



<http://jcp.org/en/jsr/platform?listBy=1&listByType=platform>

2. Гуляя по этому списку, рано или поздно обнаружим JSR 169 JDBC Optional Package for CDC/Foundation Profile
3. В этой спецификации говорится:

«The JDBC for CDC / FP Optional Package defines a subset of the JDBC 3.0 API that can be used in conjunction with the Java 2 Micro Edition (J2ME) Connected Device Configuration / Foundation Profile (CDC / FP).»

### Задача 5 (\*)

=====

- Скачайте JavaME Platform SDK по ссылке <http://download.oracle.com/javame/>
  - Напишите HelloWorldMIDlet , который бросит NullPointerException из метода startApp(). Запустите MIDlet на эмуляторе и пришлите stack trace (вывод программы)

Код нашего MIDlet-a:

```
----- HelloWorld.java -----
import javax.microedition.midlet.*;

public class HelloWorld extends MIDlet {
    public void startApp() {
        throw new NullPointerException();
    }
    public void pauseApp() {}
    public void destroyApp(boolean unconditional) {}
}
```

После некоторых «ковыряний» с JavaME Platform SDK 3.0, вы обнаружите, что System.out выводится в файл, название которого

"<ПАПКА ПРОФИЛЯ ПОЛЬЗОВАТЕЛЯ WINDOWS>javame-sdk\3.0\work\<НОМЕР УСТРОЙСТВА>"device.log

где номер устройства эмулятора (См. Строчку ID: ... в Window Status Bar для emulator.exe)

Весь Stack Trace будет выглядеть так:

```
[ 2547] INFO - VM - TRACE: <at java.lang.NullPointerException>, startApp threw an Exception
[ 2547] INFO - VM - java.lang.NullPointerException
[ 2547] INFO - VM - - HelloWorld.startApp(), bci=4
[ 2547] INFO - VM - - javax.microedition.midlet.MIDletTunnellImpl.callStartApp(), bci=1
[ 2547] INFO - VM - - com.sun.midp.midlet.MIDletPeer.startApp(), bci=7
[ 2547] INFO - VM - - com.sun.midp.midlet.MIDletStateHandler.startSuite(), bci=269
[ 2547] INFO - VM - - com.sun.midp.main.AbstractMIDletSuiteLoader.startSuite(), bci=52
[ 2547] INFO - VM - - com.sun.midp.main.CldcMIDletSuiteLoader.startSuite(), bci=8
[ 2547] INFO - VM - - com.sun.midp.main.AbstractMIDletSuiteLoader.runMIDletSuite(), bci=161
[ 2547] INFO - VM - - com.sun.midp.main.AppIsolateMIDletSuiteLoader.main(), bci=26
```





## Домашняя работа по второй лекции JavaME

### Задача 1

=====

Реализуйте один из алгоритмов сортировки используя лишь CLDC 1.1 API.

Входные параметры:

массив из N элементов типа int, заполненных случайными числами от 0 до N.

**Пример сортировки пузырьком:**

```
private static void bubblesort(int[] arr) {
    for (int i = arr.length; i != 1; i--) {
        for (int j = 0; j != i - 1; j++) {
            if (arr[j] > arr[j + 1]) {
                int tmp = arr[j];
                arr[j] = arr[j + 1];
                arr[j + 1] = tmp;
            }
        }
    }
}
```

### Задача 2 (\*)

=====

- Скачайте JavaME Platform SDK по ссылке <http://download.oracle.com/javame/>

3. Напишите MIDlet, с двумя командами:

«About» и «Exit»

4. По активации команды «About» - показывается имя и фамилия в Alert-е,

- По активации команды «Exit» - ваш MIDlet завершает исполнение

===== HelloWorld.java =====

```
import java.util.Random;
import javax.microedition.lcdui.Alert;
import javax.microedition.lcdui.AlertType;
import javax.microedition.lcdui.Command;
import javax.microedition.lcdui.CommandListener;
import javax.microedition.lcdui.Display;
import javax.microedition.lcdui.Displayable;
import javax.microedition.lcdui.Form;
import javax.microedition.midlet.*;

public class HelloWorld extends MIDlet implements CommandListener {
    protected void destroyApp(boolean unconditional) throws MIDletStateChangeException {
    }
    protected void pauseApp() {
    }
    protected void startApp() throws MIDletStateChangeException {
        Command okCommand = new Command("About", Command.OK, 1);
        Command exitCommand = new Command("Exit", Command.EXIT, 1);
        Form f = new Form("Family MIDlet");
        f.addCommand(okCommand); f.addCommand(exitCommand);
        f.setCommandListener(this); Display.getDisplay(this).setCurrent(f);
    }
    public void show() {
        Alert alert = new Alert("First and Second Names",
            "Boris Kvartskhava", null, AlertType.CONFIRMATION);
        alert.setTimeout(5000); // 5 secs
        Display.getDisplay(this).setCurrent(alert);
    }
    public void commandAction(Command c, Displayable d) {
        if (c.getCommandType() == Command.OK) {
            System.out.println("show!");
            show();
        } else {
            System.out.println("terminating!");
            notifyDestroyed();
        }
    }
}
```



## Домашняя работа по третьей лекции JavaME

### Задача 1

=====

- Укажите, каким образом из кода приложения можно понять, реализован ли на нём-- - JSR 211

#### Спецификация JSR 211 говорит:

«The system property `microedition.chapi.version` can be used to determine whether CHAPI is implemented, and if it is, the version number. If this specification is implemented in a CLDC Java runtime environment, the property `microedition.chapi.version` MUST have the value `"1.0"` or `"1.0.1"` as returned from the `java.lang.System.getProperty` method. For all other Java runtimes, the property value returned must be `"1.0.1"` »

#### - JSR 205

- Если на платформе реализована спецификация JSR 248 MSA 1.1, то через System property `wireless.messaging.version`

Иначе через проверку в вызове `Class.forName("wireless.messaging.message.Message")`

Если вызов бросает `ClassNotFoundException` – JSR не реализован, иначе — реализован.

#### - JSR 185

- System property `microedition.sip.version`, так как спецификация JSR 185 говорит:

“Platforms where this API is implemented according to this specification shall include a system property with a key

`"microedition.sip.version"` . When `System.getProperty` is called with this key, implementations conforming

to this specification shall return the version string `"1.1.1"` .

”

### Задача 2

=====

Возьмите любой доступный вам телефон с JavaME “на борту”, и выпишите ответы на следующие вопросы:

5. Название модели и производителя телефона
6. Версия конфигурации
7. Версия профиля
8. Какие JSR опциональных пакетов реализованы
9. Какие реализованы зонтичные спецификации

#### Пример Nokia N95

JSR-75 ( File system )

JSR-82 ( Bluetooth )

JSR-120 ( Wireless Messaging API )

JSR-135 ( Mobile Media API )

JSR-179 ( Location API )

JSR-184 ( Mobile 3D Graphics API )

### Задача 3 (\*)

=====

Напишите MIDlet который выведет на экран ответы на все вопросы из задачи 2) за исключением названия модели и производителя телефона.



```

===== HelloWorld.java =====
import java.util.Random;
import javax.microedition.lcdui.Alert;
import javax.microedition.lcdui.AlertType;
import javax.microedition.lcdui.Command;
import javax.microedition.lcdui.CommandListener;
import javax.microedition.lcdui.Display;
import javax.microedition.lcdui.Displayable;
import javax.microedition.lcdui.Form;
import javax.microedition.midlet.*;

public class HelloWorld extends MIDlet implements CommandListener {

    protected void destroyApp(boolean unconditional) throws MIDletStateChangeException {
    }

    protected void pauseApp() {
    }

    protected void startApp() throws MIDletStateChangeException {
        Form f = new Form("Family MIDlet");
        for(int i=0; i < jsrsImplemented.length; i++) {
            Command c = new Command(propNames[i], jsrsImplemented[i],
                Command.OK, 1);
            f.addCommand(c);
        }

        Command exitCommand = new Command("Exit", Command.EXIT, 1);
        f.addCommand(exitCommand);
        f.setCommandListener(this); Display.getDisplay(this).setCurrent(f);
    }

    public void show(String message, boolean isImplemented) {
        String msg = message+" is"+((isImplemented)?" implemented":" not implemented");

        Alert alert = new Alert(msg);
        alert.setTimeout(5000); // 5 secs
        Display.getDisplay(this).setCurrent(alert);
    }

    public void commandAction(Command c, Displayable d) {
        if (c.getCommandType()== Command.OK) {
            System.out.println("c.getLabel() = " + c.getLabel());
            show(c.getLongLabel(), System.getProperty(c.getLabel()) != null);
        } else {
            System.out.println("terminating!");
            notifyDestroyed();
        }
    }

    String[] propNames = {
        "microedition.io.file.FileConnection.version",
        "bluetooth.api.version",
        "microedition.media.version",
        "microedition.location.version"
    };

    String[] jsrsImplemented = new String[]{
        "jsr 075",
        "jsr 082",
        "jsr 135",
        "jsr 179"
    };
}

```



Эталонное д/з для третьей лекции по JavaSE

**Задача:**

1. **Запустите среду NetBeans, откройте проект «PlayGround»**
2. **Создайте свой наследник класса `java.lang.Exception: IgnoreException`**
3. **Для каждого из животных создайте метод:**

***животное.meet(другое животное) throws IgnoreException, следующим образом:***  
- ***если животное того же вида, что и другое животное, то оно представляется.***  
- ***если другое животное - другого вида (например, кот встретил собаку), то наш метод бросает IgnoreException***

***Для сравнения видов животных вам понадобится конструкция `instanceof`***

4. ***В методе `Playground.main()` создайте 3 животных, по одному каждого вида, виды по вашему усмотрению,***
  - ***Создайте реализацию интерфейса `Runnable` (отдельный класс, давайте назовём его `Zoo`), в котором:***
    - ***есть конструктор с тремя параметрами, через который вы передадите созданных животных в `Zoo`.***
    - ***каждые 10 секунд 2-е из трёх животных (выбираются случайно) встречаются.***
    - ***В методе `Playground.main()` создайте и запустите нить с классом `Zoo`, что вы написали на предыдущем шаге.***
    - ***Создайте и запустите ещё одну нить с `Zoo`, работающую параллельно с первой. Измените ваш код таким образом, чтобы одно и то же животное не могло встречать кого-либо ещё одновременно в разных нитях.***

