

### 3.4B. Graphics Supplemt

- 2.5절 Jframe , JOptionPane
- 3.4절 JOptionPane 추가

- 매 Chapter마다 Graphics Supplement
  - JFrame
  - JApplet (X)
  - Javafx → 3장 실습직전 (1,2,3장의 Javafx)





# 2.5 (optional) Graphics Supplement: Outline

- Style Rules Applied to a Graphics JFrame
- JOptionPane Class
  - provide you with a way to do windowing I/O for your Java Programs
- Inputting Numeric Types
- Multi-Line Output Windows



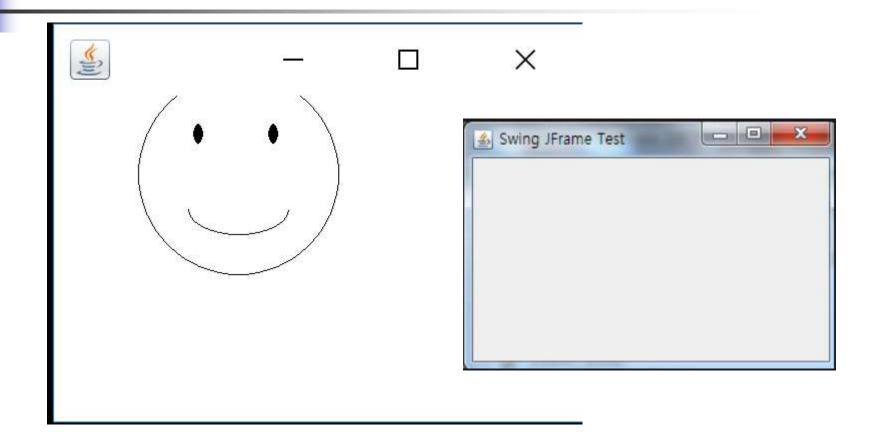
### Listing A java GUI Application Using the JFrame Class

```
import javax.swing.JFrame;
import java.awt.Graphics;
public class HappyFaceJFrame extends JFrame
  public static final int FACE_DIAMETER = 200;
  public static final int X_FACE = 100;
  public static final int Y_FACE = 50;
  public static final int EYE_WIDTH = 10;
  public static final int EYE_HEIGHT = 20;
  public static final int X RIGHT EYE = 155;
  public static final int Y RIGHT EYE = 100;
  public static final int X_LEFT_EYE = 230;
  public static final int Y_LEFT_EYE = Y_RIGHT_EYE;
  public static final int MOUTH_WIDTH = 100;
  public static final int MOUTH_HEIGHT = 50;
  public static final int X_MOUTH = 150;
  public static final int Y_MOUTH = 160;
  public static final int MOUTH START ANGLE = 180;
  public static final int MOUTH_DEGREES_SHOWN = 180;
```

```
public void paint(Graphics canvas)
//Draw face outline:
canvas.drawOval(X_FACE, Y_FACE, FACE_DIAMETER, FACE_DIAMETER);
//Draw eyes:
canvas.fillOval(X_RIGHT_EYE, Y_RIGHT_EYE, EYE_WIDTH, EYE_HEIGHT);
canvas.fillOval(X LEFT EYE, Y LEFT EYE, EYE WIDTH, EYE HEIGHT);
//Draw mouth:
canvas.drawArc(X_MOUTH, Y_MOUTH, MOUTH_WIDTH, MOUTH_HEIGHT,
       MOUTH START ANGLE, MOUTH DEGREES SHOWN);
   public HappyFaceJFrame()
          setSize(600,400);
          setDefaultCloseOperation(EXIT_ON_CLOSE);
   public static void main(String[] args)
          HappyFaceJFrame guiWindow = new HappyFaceJFrame();
          guiWindow.setVisible(true);
```









#### Listing 2.10 A Program with JOptionPane I/O

```
import javax.swing.*;

public class JOptionPaneDemo
{
   public static void main(String[] args)
   {
     String appleString;
     appleString =
        JOptionPane.showInputDialog("Enter number of apples:");
     int appleCount;
     appleCount = Integer.parseInt(appleString);
```



```
String orangeString;
  orangeString =
    JOptionPane.showInputDialog("Enter number of oranges:");
  int orangeCount;
  orangeCount = Integer.parseInt(orangeString);
  int totalFruitCount;
  totalFruitCount = appleCount + orangeCount;
  JOptionPane.showMessageDialog(
    null, "The total number of fruits = " + totalFruitCount);
  System.exit(0);
```



# Listing 2.10



When the unor effects CK, the retrictor spore street, each the next screeter (if any) is deployed.

#### Window 2



#### Window 3







#### JOptionPane

- class JOptionPaneDemo
  - simple java application program with a windowing interface
  - the three windows are produced
- - Swing library : for windowing interface
  - these libraries are called packages
  - JOptionPane Class is in this package.





- JOptionPane
  - can be used to construct windows that interact with the user.
  - imported by

```
import javax..*;
```

 produces windows for obtaining input or displaying output.





#### (JOptionPane)

 JOptionPane makes it easy to pop up a standard dialog box that prompts users for a value or informs them of something

Method Name	Description
showConfirmDialog	Asks a confirming question, like yes/no/cancel
showInputDialog	Prompt for some input.
showMessageDialog	Tell the user about something that has happened.
showOptionDialog	The Grand Unification of the above three.



- Use showInputDialog() for input.
- Only string values can be input.
- \_ ===
- showInputDialog
  - public static <u>String</u> showInputDialog(<u>Object</u> message) throws <u>HeadlessException</u>
  - Shows a question-message dialog requesting input from the user. The dialog uses the default frame, which usually means it is centered on the screen.



 Output is displayed using the showMessageDialog method.

```
JOptionPane.showMessageDialog(null, "The total
   number of fruits = " + totalFruitCount);
```

- showMessageDialog
  - public static void
     showMessageDialog(Component parentComponent,
     Object message) throws HeadlessException
  - Brings up an information-message dialog titled "Message".
  - Parameters:
    - : parentComponent determines the Frame in which the dialog is displayed; if null, or if the parentComponent has no Frame, a default Frame is used
    - message the Object to display



#### syntax

#### input

```
String_Variable =
JOptionPane.showInputDialogue(String_Express
ion);
```

#### output

```
JOptionPane.showMessageDialog(null,
String_Expression);
```





### Integer Class

- The Integer class <u>a value of the</u> <u>primitive type int</u> in an object. An object of type Integer contains a single field whose type is int.
- In addition, this class provides
  - 1) several for converting an int to a String and a String to an int
  - 2) other useful when dealing with an int.





#### the method

To convert an input value from a string to an integer use the parseInt() method from the Integer class, use appleCount = Integer.parseInt(appleString);

=====

#### parseInt

- public static int parseInt(<u>String</u> s) throws <u>NumberFormatException</u>
- Parses the string argument as a signed decimal integer. The characters in the string must all be decimal digits, except that the first character may be an ASCII minus sign '-' ('\u002D') to indicate a negative value.





### System Class

- System.exit(0) ends the program.
- \_ =====
- exit
  - public static void exit(int status)
  - Terminates the currently running Java Virtual Machine. The argument serves as a status code; by convention, a nonzero status code indicates abnormal termination. This method calls the exit method in class Runtime. This method never returns normally.
  - The call System.exit(n) is effectively equivalent to the call:
    - Runtime.getRuntime().exit(n)





### JOptionPane Cautions

If the input is not in the correct format, the

program will crash.



```
Exception in thread "main" java.lang.NumberFormatException: For input string: "k k"

at java.lang.NumberFormatException.forInputString(NumberFormatException.

java:48)

at java.lang.Integer.parseInt(Integer.java:447)

at java.lang.Integer.parseInt(Integer.java:497)

at JOptionPaneDemo.main(JOptionPaneDemo.java:13)

계속하려면 아무 키나 누르십시오...
```



### Fogetting System.exit(0)

- If you omit the last line (System.exit(0)), the program will not end, even when the oκ button in the output window is clicked.
  - the windows will all go away, but the "invisible" program is there.
  - it will just hang there, using up the computer resources and possibly keeping you from doing their things.
  - when you write a program with a windowing interface.....





# Why are some methods invoked with a Class name

- Instead of a calling objects
  - do not require a calling object and are invoked using the class name.



## Inputting Numeric Types

- JOptionPane.showInput Dialog can be used to input any of the numeric types.
  - Simply convert the input string to the appropriate numeric type.



Type Name	Method for Converting
byte	Byte.parseByte(String_To_Convert)
short	Short.parseShort(String_To_Convert)
int	<pre>Integer.parseInt(String_To_Convert)</pre>
long	Long.parseLong(String_To_Convert)
float	Float.parseFloat(String_To_Convert)
double	Double.parseDouble(String_To_Convert)

To convert a value of type <code>String</code> to a value of the type given in the first column, use the method given in the second column. Each of the methods in the second column returns a value of the type given in the first column. The <code>String\_To\_Convert</code> must be a correct string representation of a value of the type given in the first column. For example, to convert to an <code>int</code>, the <code>String\_To\_Convert</code> must be a whole number (in the range of the type <code>int</code>) that is written in the usual way without any decimal point.





### Multi-Line Output Windows

■ To output multiple lines using the method

JOptionPane.showMessage Dialog, insert the new
line character '\n' into the string used as the second argument.

```
OptionPane.showMessageDialog(null,
    "The number of apples\n" +
    "plus the number of oranges\n" +
    "is equal to " + totalFruit);
```



## Multi-Line Output Windows, cont.



Display 2.19
A Multiline Output Window



## Listing 2.11 Change Program with I/O Windows (part 1 of 2)

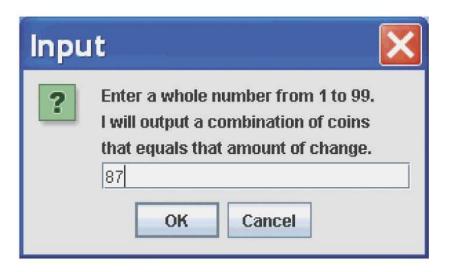
```
import javax.swing.*;
public class ChangeMakerWindow
  public static void main(String[] args)
    String amountString =
         JOptionPane.showInputDialog(
            "Enter a whole number from 1 to 99.\n"
          + "I will output a combination of coins\n"
          + "that equals that amount of change.");
    int amount, original Amount,
      quarters, dimes, nickels, pennies;
    amount = Integer.parseInt(amountString);
    originalAmount = amount;
```



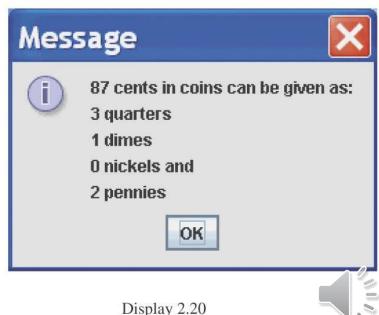
```
quarters = amount/25;
   amount = amount%25;
   dimes = amount/10;
   amount = amount%10;
   nickels = amount/5;
   amount = amount%5;
    pennies = amount;
   JOptionPane.showMessageDialog(null,
            originalAmount
           + " cents in coins can be given as:\n"
           + quarters + " quarters\n"
           + dimes + " dimes\n"
           + nickels + " nickels and\n"
           + pennies + " pennies");
   System.exit(0);
```







#### **Output Window**



Display 2.20 Change Program with I/O Windows



### 3.4 Graphics Supplement

- How to add color to your javaFX Drawings
- A JOptionPane Yes/No Window



### A JOptionPane Yes/No Window

- used to present the user with a yes/no question
- The window contains
  - the question text
  - two buttons labeled Yes and No.



#### example

```
int answer =
   JOptionPane.showConfirmDialog(null, "End
   program?", "End Check",
   JOptionPane.YES_NO_OPTION);
if (answer == JOptionPane.YES_OPTION)
   System.exit(0);
else
   System.out.println("once more");
```



## A JOptionPane Yes/No Window

```
import javax.swing.*;
public class JOptionPaneYesNoDemo
 public static void main(String[] args)
   int answer =
           JOptionPane.showConfirmDialog(null, "End program?",
                   "Want to end?", JOptionPane.YES NO OPTION);
       System.out.println("answer="+answer);
        if (answer == JOptionPane.YES_OPTION) //0
          System.exit(0);
        else if (answer == JOptionPane.NO_OPTION) //1
          System.out.println("One more time");
        else
     System.out.println("This is impossible");
   System.exit(0);
```



Display 3.21
An Applet that Uses Looping and Branching



- JOPtionPane.showConfirmDialog returns an int value named either YES\_OPTION or NO\_OPTION, but you do not need to think of them as ints.
- The second argument ("End program?" in our example) appears in the window.
- The third argument ("End Check" in our example) is displayed as the title of the window.



- The last argument (JOptionPane.YES\_NO\_OPTION in our example) requests a window with yes and no buttons.
- The first argument
  - (null in our example)
  - determines the Frame in which the dialog is displayed; if null, or if the parentComponent has no Frame, a default Frame is used
  - affects the placement of the window on the screen.—default frame
  - Simply use null for now.



