Read Chapter 11 from *Introduction to Java Programming, Brief Version*, (9th ed) by Liang, Y. D. (2013).

- Page references
  - Page 408 Section 11.2 (Superclasses and Subclasses)
  - Page 418 Section 11.5 (Overriding vs. Overloading)
  - Page 430 Section 11.11 (The Array List Class)

Eclipse and Java websites for additional reference

- http://www.cs.armstrong.edu/liang/intro9e/supplement.html
- http://help.eclipse.org/juno/index.jsp?topic=/org.eclipse.platform.doc.user/reference/ref-cheatsheets.htm

Keywords

- date
- GUI
- integer

Task 2: Create a Java Program to Display a Subclasses from Lab 6.1

Procedure

1. Open the Eclipse software that is installed on the Lab computer.
2. Create the following methods for both Checking classes:
   - public String accountType()
   - public int getId()
   - public double getBalance()
   - public double getAnnualInterestRate()
   - public void setId(int id)
   - public void setBalance(double balance)
   - public void setAnnualInterestRate(double annualInterestRate)
   - public double getMonthlyInterest()
   - public java.util.Date getDateCreated()
   - public void withdraw(double amount)
   - public void deposit(double amount)
   - public String getAlert(double amount)
3. Create a Java class called Savings that extends Account class.
4. Create the following methods for both Savings classes:
   - public String accountType()

- public int getId()
- public double getBalance()
- public double getAnnualInterestRate()
- public void setId(int id)
- public void setBalance(double balance)
- public void setAnnualInterestRate(double annualInterestRate)
- public double getMonthlyInterest()
- public java.util.Date getDateCreated()
- public void withdraw(double amount)
- public void deposit(double amount)
- public String getAlert(double amount)

5. Write a demo program to invoke the CheckingAccount and its methods as pictured in Figure 6-2-1.

```
public class CheckingDemo {
  public static void main (String[] args) {
    Checking check = new Checking(1122, 20000, 4.5);

    check.withdraw(10000);
    check.deposit(3000);
    System.out.println("Account type is " + check.accountType());
    System.out.println("Balance is " + check.getBalance());
    System.out.println("Monthly interest is " +
      check.getMonthlyInterest());
    System.out.println(check.getAlert(10000));
    System.out.println("This account was created at " +
      check.getDateCreated());
  }
}
```

Figure 6-2-1

6. Write a demo program to invoke the SavingsAccount and its methods pictured in Figure 6-2-2.

```
public class SavingsDemo {
  public static void main (String[] args) {
    Savings save = new Savings(1122, 20000, 4.5);

    save.withdraw(10000);
    save.deposit(3000);
    System.out.println("Account type is " + save.accountType());
    System.out.println("Balance is " + save.getBalance());
    System.out.println("Monthly interest is " +
```

```
            save.getMonthlyInterest());
          System.out.println(save.getAlert(10000));
          System.out.println("This account was created at " +
            save.getDateCreated());
        }
      }
```

Figure 6-2-2

7. Express the result to the console.
8. Check the Java file for errors and right-click the class in the project to display the context menu. Choose Run, Java Application in the context menu to run the class.
9. Save screenshots of the output similar to Figures 6-2-3 and 6-2-4 and submit them to your instructor.

```
Account type is Checking
Balance is 23000.0
Monthly interest is 86.25
** Withdrawn amount = 10000.0 is over the $10,000 limit **
This account was created at Sun Jul 26 01:43:30 PDT 2009
```

Figure 6-2-3: Sample Output of Checking Account

```
Account type is Savings
Balance is 13000.0
Monthly interest is 48.75
No alert message.
This account was created at Sun Jul 26 01:45:11 PDT 2009
```

Figure 6-2-4: Sample Output of Savings Account

Did it work?

Were you able to:

- Implement the CheckingDemo class?
- Display a checking account's balance amount by calling the getBalance() method?
- Display a checking account's monthly interest rate by calling the getMonthlyInterest() method?
- Display a checking account's over-withdrawal message by calling the getAlert() method?
- Display a checking account's created time by calling the getDateCreated() method?
- Implement the SavingsDemo class?
- Display a savings account's balance amount by calling the getBalance() method?
- Display a savings account's monthly interest rate by calling the getMonthlyInterest() method?

- Display an savings account's over-withdrawal message by calling the getAlert() method?
- Display an savings account's created time by calling the getDateCreated() method?

Summary and Reminders

Summary

This unit discussed how to implement inheritance and polymorphism. The next unit will look at how to create GUI applications using abstract classes and interfaces.

Reminders

- Students should review Chapters 11 of Introduction to Java Programming before starting Lab 6.1 and 6.2.

(End of Unit 6)

Lab 7-1: Displaying Tic Tac Toe Board Using the Swing Class

Purpose

In this lab, you will display a frame that contains nine labels. A label may display a cross image icon, a not image icon, or nothing, as shown in the following figures. What to display is randomly chosen. Use the Math.random() method to generate integers 0, 1, or 2, corresponding to displaying a cross image icon, a not image icon, or nothing.

How it Fits In

Students will continue working with the basic building blocks of Java programs. They will work with a Java statement that will deal with constructor classes and the creation of a GUI.

Objectives

- Create a Java program
- Create a constructor
- Create a GUI

Estimated Completion Time: 40 minutes

Task 1: Study Material

Read Chapter 12 and 13 from *Introduction to Java Programming*, Brief Version, (9th ed) by Liang, Y. D. (2013).

- Page references
  - Page 446, Section 12.3 (The Java GUI API)
  - Page 447, Section 12.3.1 (Component Classes)
  - Page 451, Section 12.5 (Layout Manager)
  - Page 465, Section 12.10 (Image Icon)

Eclipse and Java websites for additional reference

- http://www.cs.armstrong.edu/liang/intro9e/supplement.html
- http://help.eclipse.org/juno/index.jsp?topic=/org.eclipse.platform.doc.user/reference/ref-cheatsheets.htm

Keywords

- Component class
- Helper class
- Layout Manager

Task 2: Create a Java Program to Display a Tic Tac Toe Board Using the Swing Class

Procedure

1. Open the Eclipse software that is installed on the Lab computer.
2. Create a Java class named TicTacToe.
3. Import javax.swing.* package to use all the Swing components.
4. Import java.awt.* package to use the framework of AWT.
5. Extend the TicTacToe class by creating a JFrame container class using inheritance.
6. Create two imageIcon variables for "cross" and "not" images.
7. Create a constructor to get the Content Pane to organize the layout structure. All the GUI controls will be sitting on the Content Pane.
8. Create a 3 x 3 labels layout on the frame by the code shown in Figure 8-1-1:

```
container.setLayout(new GridLayout(3, 3));
```

Figure 8-1-1

9. Declare a variable called mode to capture random numbers 0, 1, or 2 by the code shown in Figure 8-1-2.

```
int mode = (int)(Math.random() * 3);
```

Figure 8-1-2

10. Assign each label by a random imageIcon: 0 for cross, 1 for not, and 2 for nothing.
11. Express the result into a GUI frame.
12. Check the Java file for errors and right-click the class in the project to display the context menu. Choose Run, Java Application in the context menu to run the class.
13. Save screen shots similar to Figures 8-1-3 and 8-1-4 and submit them to your instructor.

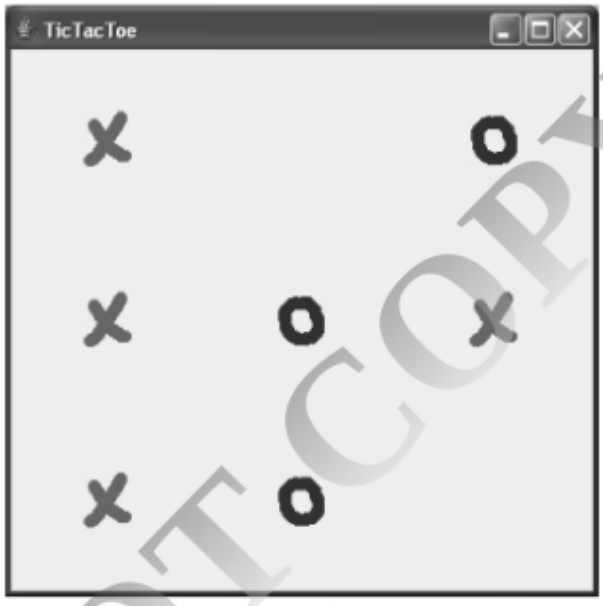

Figure 8-1-3

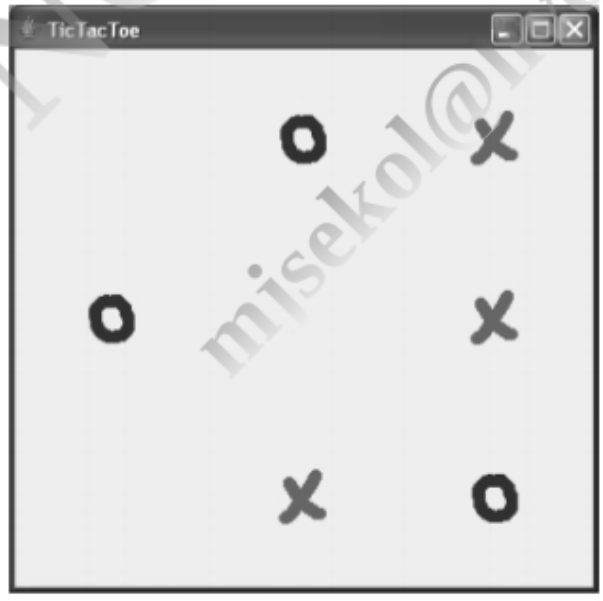

Figure 8-1-4

Did it work?

- Were you able to display a frame with nine random icon images (X, O, or blank) using the Swing class?

Unit 7 Lab 2: Displaying Tic Tac Toe Board Using the Graphics Class

Purpose

Create a custom panel that displays X, O, or nothing. What to display is randomly chosen whenever a panel is repainted. Use the Math.random() method to generate an integer 0, 1, or 2, corresponding to displaying X, O, or nothing. Create a frame that contains nine custom panels, as shown in Figures 8-2-3 and 8-2-4.

How it Fits In

Students will continue working with the basic building blocks of Java programs. They will work with a Java statement that will deal with cell classes and the creation of a GUI.

Objectives

- Create a Java program
- Create a Cell class
- Create a GUI

Estimated Completion Time: 45 minutes

Task 1: Study Material

Read Chapter 12 and 13 from *Introduction to Java Programming*, Brief Version, (9th ed) by Liang, Y. D. (2013).

- Page references
  - Page 446, Section 12.3 (The Java GUI API)
  - Page 462, Section 12.9 (Common Features of Swing GUI Components)
  - Page 470, Section 12.11.3 (Text Positions)

Eclipse and Java websites for additional reference

- http://www.cs.armstrong.edu/liang/intro9e/supplement.html
- http://help.eclipse.org/juno/index.jsp?topic=/org.eclipse.platform.doc.user/reference/ref-cheatsheets.htm

Keywords

- Component class
- Layout manager
- Swing GUI

Task 2: Create a Java Program to Display a Tic Tac Toe Board

Procedure

1. Open the Eclipse software that is installed on the Lab computer.
2. Create a Java class named TicTacToe2.
3. Import javax.swing.* package to use all the Swing components.
4. Import java.awt.* package to use the framework of AWT.
5. Extend the TicTacToe2 class by creating a JFrame container class using inheritance.
6. Create a constructor to get the Content Pane to organize the layout structure. All the GUI controls will be sitting on the Content Pane.
7. Create a 3 x 3-cell layout on the frame by the code shown in Figure 8-2-1.

```
container.setLayout(new GridLayout(3, 3));
```

Figure 8-2-1

8. Create a Cell class that extends JPanel by using inheritance. The Cell class will draw a graphics expression for each cell.
9. Assign each cell by a random number: 0 for cross, 1 for not, and 2 for nothing. See the code shown in Figure 8-2-2.

```
int mode = (int)(Math.random() * 3);

    if (mode == 0) {

    // see Listing 13.3, page 466 for drawing cross

    }
    else if (mode == 1) {

    // see Listing 13.5, page 435 for drawing circles

    }
```

Figure 8-2-2