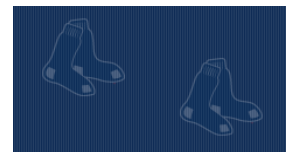


# NTOU Java Programming Homework 2

Spring 2025



# Homework 2-1<sub>1</sub>

2

- Please design a salary calculation application, which includes classes as follows:
  - ▣ *Employee* is abstract class, with an abstract method: *int getEarnings ()*.
  - ▣ *Employee* has subclasses:
    - (1) *SalariedEmployee*, which receives weekly salary: *weeklySalary\*weeks*.
    - (2) *HourlyEmployee*, which receives hourly salary.
      - If the working hours in a week are less than or equal to 40 hours, the salary is *hourlySalary\*hours*.
      - If the working hours are more than 40 hours, the salary for the part exceeding 40 hours is multiplied by 1.5 times.
    - (3) *CommissionEmployee*, which receives salary: *commissionRate\*grossSales*.
    - (4) *PieceWorker*, which receives salary: *wage\*piece* (number of completed works).

# Homework 2-1<sub>2</sub>

3

- Bonus is an interface used to define additional bonus. There are two implementation classes: *StaticBonus* and *DynamicBonus* (as attached):
  - *StaticBonus*: sets the bonus fixed at 10,000.
  - *DynamicBonus*: sets the bonus as 10% of salary.
- *Employee* must contain a field whose type is *Bonus*, which can be assigned to a *StaticBonus* or *DynamicBonus* object to calculate the bonus (員工可以有兩種Bonus形式).

# Homework 2-1<sub>3</sub>

4

- According to the above description, please design another class *EmployeeDataCollector* as the user interface:
  - ▣ Providing a text-based interactive UI to allow users to
    - choose the type of employees that should be added (by the Scanner API),
    - enter the required information for the selected type (for example, if you select *SalariedEmployee*, you need to enter the weekly salary and the number of working weeks), and
    - enter the bonus type (static or dynamic).
  - ▣ If the user inputs “-1”, the program should terminate, and display all entered employee details, total salary, and total bonus.

# Homework 2-1<sub>4</sub>

5

- All subtypes of *Employee* objects (Employee的子類別物件) need to be stored in an array of type *Employee*.
- All *Employee* objects in the array should be processed:
  - *toString()* is called to print the details.
  - *getEarnings()* is called to obtain individual salary (for summing up and output the summed salaries)
  - *getBonus()* of the Bonus field is called to get the bonus salary (for summing up and output the summed bonuses)
- Please note that the following six types of variables cannot appear in the program: *SalariedEmployee*, *HourlyEmployee*, *CommissionEmployee*, *PieceWorker*, *StaticBonus*, and *DynamicBonus*.

# Sample Output

6

- For expected results, please refer to hw-2-1-sampleOutput.txt.

# Hints

7

- Please refer to the textbook examples 10.4~10.9.
- An abstract *inputData* method can be added to the class *Employee*, and different input tasks can be implemented in the subclasses.
- The Bonus object should be set as a field of the Employee class.
- Please note that each *Employee* object may include either *StaticBonus* or *DynamicBonus* object for the field *bonus*.

## 2-2 Simple GUI<sub>1</sub>

8

- Write a temperature conversion (溫度轉換) application that converts among Fahrenheit (華氏), Celsius (攝氏), and Kelvin (克氏).
- The temperature should be entered from the keyboard (via a JTextField).
- A JLabel should be used to display the converted temperature. Use the following formula for the conversions.

$$\text{Celsius} = (\text{Fahrenheit} - 32) \times 5/9$$

$$\text{Kelvin} = \text{Celsius} + 273.15$$



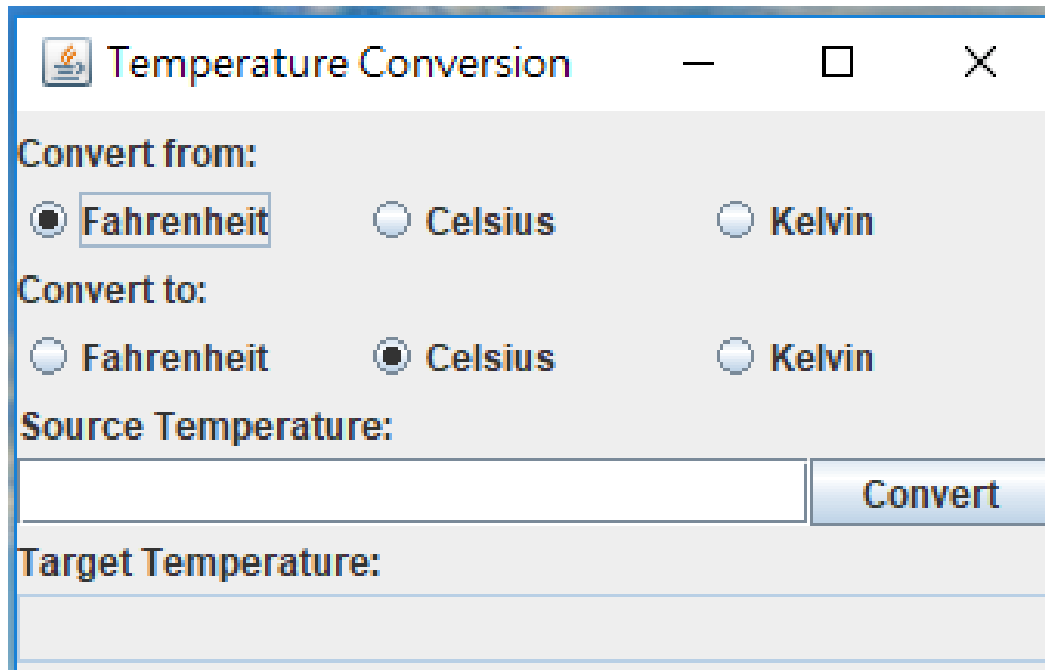
## 2-2 Simple GUI<sub>2</sub>

9

- The options for units of the source temperature and the destination temperature need to be JRadioButton.
  - ▣ Please pay attention to the use of ButtonGroup.
- The TextField displaying the conversion result needs to be set as uneditable ( *setEditable(false)* )
- All of *FlowLayout*, *BorderLayout*, and *GridLayout* will be used.
- The outer layer should be an 8 x 1 GridLayout.
  - ▣ Three options for units of the source temperature and target temperature should be placed in a JPanel.
  - ▣ The source temperature contains two components, the text field and the Button placed on the right (using BorderLayout).
  - ▣ After selecting the source and target units and inputting the source temperature, **press [Enter] or click the [Convert] button**, and the target temperature will be displayed.

## 2-2 Expected Results<sub>1</sub>

10



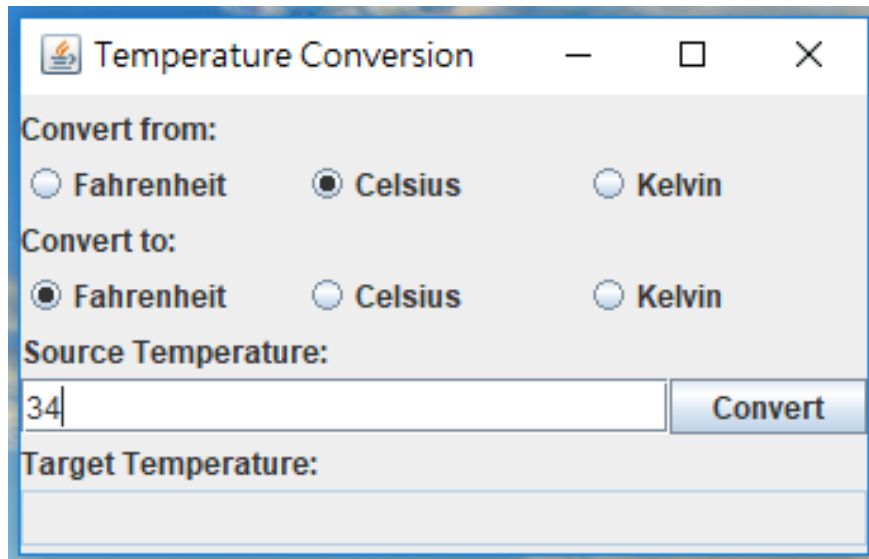
The screenshot shows a window titled "Temperature Conversion" with a standard Windows-style title bar (minimize, maximize, close buttons). The window contains the following elements:

- Convert from:** Three radio buttons are present: "Fahrenheit" (selected), "Celsius", and "Kelvin".
- Convert to:** Three radio buttons are present: "Fahrenheit", "Celsius" (selected), and "Kelvin".
- Source Temperature:** A text input field is present, currently empty.
- Convert:** A button labeled "Convert" is located to the right of the source temperature input field.
- Target Temperature:** A label is present, followed by an empty text input field.

Initial UI

## 2-2 Expected Results<sub>2</sub>

11



Temperature Conversion

Convert from:

☐ Fahrenheit ☒ Celsius ☐ Kelvin

Convert to:

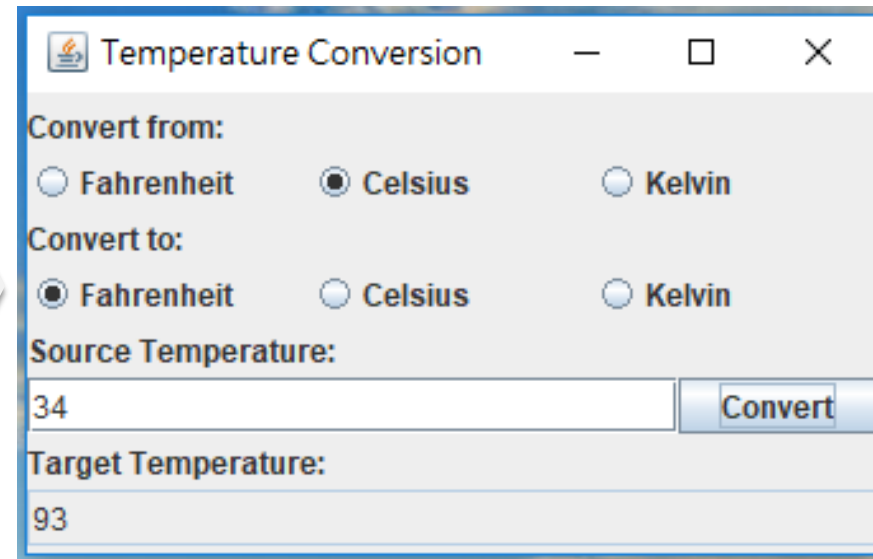
☒ Fahrenheit ☐ Celsius ☐ Kelvin

Source Temperature:

34

Convert

Target Temperature:



Temperature Conversion

Convert from:

☐ Fahrenheit ☒ Celsius ☐ Kelvin

Convert to:

☒ Fahrenheit ☐ Celsius ☐ Kelvin

Source Temperature:

34

Convert

Target Temperature:

93

Select the source and target temperature units, enter the source temperature, and **press [Enter] or click the [Convert] button**

# Hints

12

- To handle conversion from Fahrenheit to Kelvin, first convert Fahrenheit to Celsius, then Celsius to Kelvin.

# 2-3 GUI using Mouse Events<sub>1</sub>

13

- Please implement a GUI application that uses the MyShape hierarchy to create an interactive drawing application.
  - ▣ The three classes of the MyShape hierarchy require no additional changes.
- Class DrawPanel: represents the area on which the user draws the rectangles (MyRect objects).
  - ▣ An array shapes of type MyShape that will store all the rectangles the user draws.
  - ▣ An integer shapeCount that counts the number of shapes in the array.
  - ▣ A MyShape currentShape that represents the current rectangle the user is drawing.
  - ▣ A Color currentColor that represents the current drawing color.

## 2-3 GUI using Mouse Events<sub>2</sub>

14

- Class DrawPanel should also declare the following methods:
  - ▣ Overridden method paintComponent draws the shapes (rectangles) in the array.
    - Use instance variable shapeCount to determine how many shapes to draw.
    - Method paintComponent should also call currentShape's draw method. (呼叫目前正在繪製的長方形的draw函式)
  - ▣ Method clearDrawing removes all the shapes in the current drawing by setting shapeCount to zero.
    - It should call method repaint to refresh the drawing on the DrawPanel .

## 2-3 GUI using Mouse Events<sub>3</sub>

15

- Class DrawPanel should also provide event handling to enable the user to draw with the mouse.
  - Create a single inner class that extends MouseAdapter to handle required mouse events in one class.
    - Override method mousePressed to assign currentShape a new MyRect object and initializes both points to the mouse position.
    - Override method mouseReleased to finish drawing the current shape and place it in the array.
    - Override method mouseDragged so that it sets the second point of the currentShape to the current mouse position and calls method repaint. (拖曳過程會即時繪圖)

## 2-3 GUI using Mouse Events<sub>4</sub>

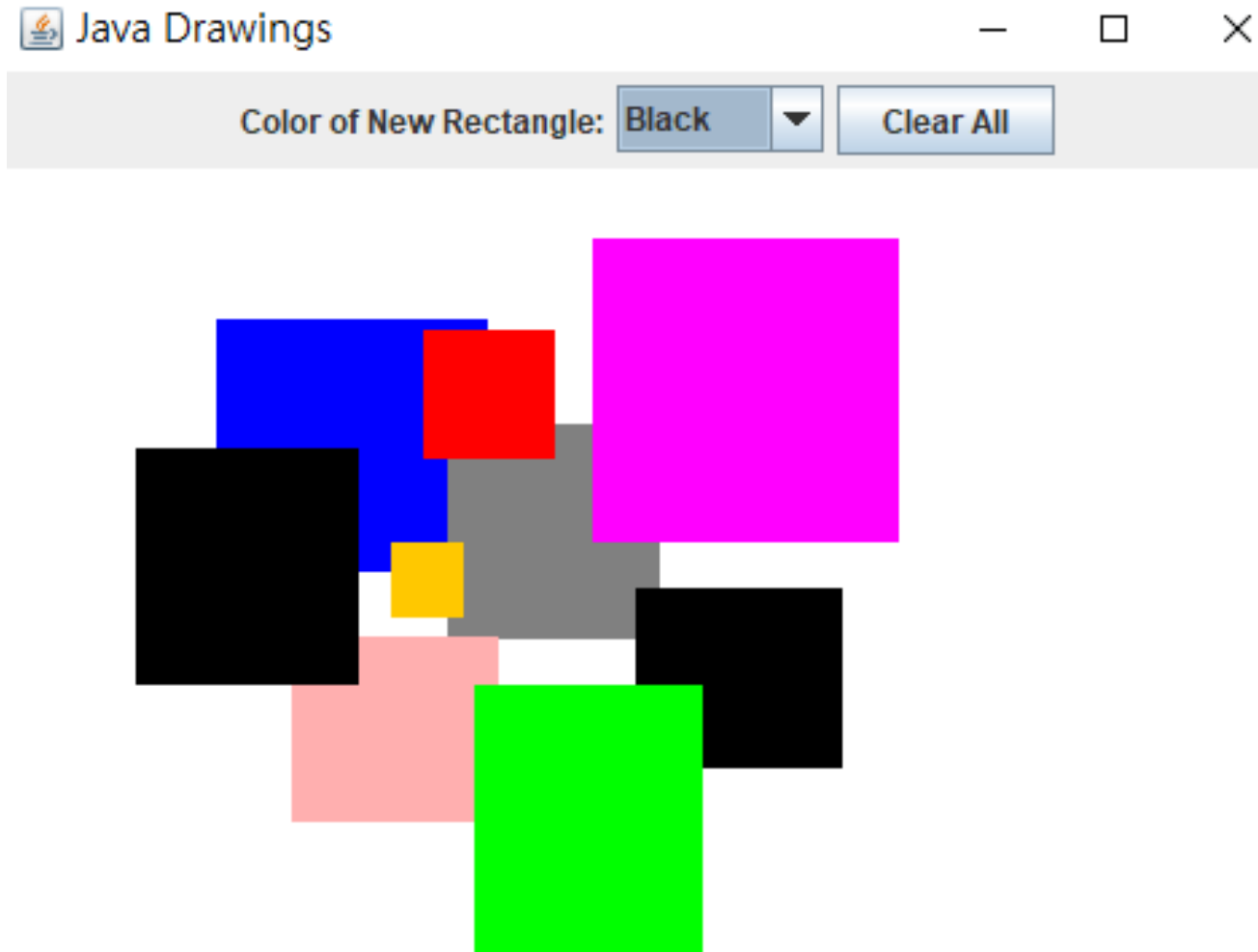
16

- Please also create a JFrame subclass called DrawFrame that provides a GUI that enables the user to control various aspects of drawing, including:
  - ▣ A text label and a combo box for selecting the color from the 10 predefined colors.
  - ▣ A button to clear all shapes from the drawing.
- In DrawFrame, each component's event handler should call the appropriate method in class DrawPanel.



# 2-3 Expected Results

17



# Requirements

18

- The naming should conform to the CamelCase style.
- “Package” is required: ntou.cs.java2025.
- Please submit files including .java files and .class files (upload them to TronClass).
- Code that fails to compile or execute is not accepted.