Asst. Prof. Dr. Sasiporn Usanavasin, Dr. Jessada Karnjana, Dr. Kasorn Galajit and Dr. Akkarawoot Takhom {sasiporn.us, akkharawoot.aj, jessada.aj, kasorn.aj}@siit.tu.ac.th

School of Information, Computer, and Communication Technology, Sirindhorn International Institute of Technology, Thammasat University.

[DES103(Y23S2)-LAB10-REVIEW]

Timer

Learning Objectives

1. To learn the `Timer` class in `javax.swing`
2. To learn how to use a `Timer` object to create basic animation

****Remark**** A ***pointer finger*** **(** 👉 **)** refers to an explanation between students and their assigned TAs.

10.1 Timer

- A `Timer` is a non-GUI component that can fire an `ActionEvent` at a predefined rate.
- It can be used to control animations.

`javax.swing.Timer`	
`+Timer(delay: int, listener: ActionListener)`	Creates a `Timer` with a specified delay in milliseconds and an ActionListener.
`+addActionListener(listener: ActionListener): void`	Add an `ActionListener` to the `timer`.
`+start(): void`	Start this `timer`.
`+stop(): void`	Stop this `timer`.
`+setDelay(delay: int): void`	Sets a new delay value for this `timer`.

10.2 Using Timer to make an animation

- The `timer` object fires the `ActionEvent` with a constant rate
- Every time the `ActionEvent` is fired from a `timer`, variants of graphics/pictures are displayed
- When the timer keeps firing `ActionEvent` and the graphic keep changing, it creates animation

Asst. Prof. Dr. Sasiporn Usanavasin, Dr. Jessada Karnjana, Dr. Kasorn Galajit and Dr. Akkarawoot Takhom {sasiporn.us, akkharawoot.aj, jessada.aj, kasorn.aj}@siit.tu.ac.th

School of Information, Computer, and Communication Technology, Sirindhorn International Institute of Technology, Thammasat University.

[DES103(Y23S2)-LAB10-EXERCISES]

Students MUST adhere to the lab instructions and regulations provided below.

Please consult your TA to review your completed exercises and submit them on Google Classroom.

Be noticed that for all lab exercises, you need to define your *Java* project as the following name format:

<StudentID>_<Lab number>_<Exercise name>

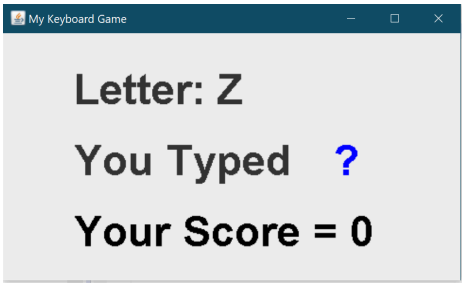
If your student's ID is 6722300208, the name format of your java project should be:

"6422300208_LAB10_KeyboardGame" for exercises 1, 2 and 3.

"6422300208_ LAB10_CatchingFlashBall" for exercises 4 and 5.

1. Students MUST explain your understanding to your TA before submitting your finished exercise,
2. Students can attach your finished exercise into Google Class, and click 'Turn-In' when your TA allows you to submit,

For today's exercises, students are going to use `Timer` for creating basic animation.

"<StudentID>_LAB10_KeyboardGame" for **exercise 1** and **2**	
KeyboardGame program displays a random character. The user types the letter he/she sees. If it matches the appearing letter, the score is updated. The final output should look like this:	

"<StudentID>_LAB10_KeyboardGame" for **exercise 3, 4** and **5**
CatchingFlashingBall program displays a black panel and shows an orange ball that appears at a random location. To catch the ball, the user needs to press the mouse inside of the ball. The ball stops moving when it is caught. The final output should look like this:


Asst. Prof. Dr. Sasiporn Usanavasin, Dr. Jessada Karnjana, Dr. Kasorn Galajit and Dr. Akkarawoot Takhom {sasiporn.us, akkharawoot.aj, jessada.aj, kasorn.aj}@siit.tu.ac.th
School of Information, Computer, and Communication Technology, Sirindhorn International Institute of Technology, Thammasat University.

Exercise 1: (2 points)

- **Java Project**: <StudentID>_LAB10_KeyboardGame
- **Objective**:

  To use the `Timer` class in `javax.swing`, and use a `Timer` object to create basic animation
- **Instruction:** Create a java project and write code in the following tasks.

  a. Add a new java class `KeyboardGame` without using any event and `timer`, complete the method `paintComponent` to generate the given output.

  Remark that the first line letter output is at `(75, 75)` and the line space is 75 pixels.

  b. Add a new java class `KeyboardGameTesting` and write a `main` method that has a frame to add an object `KeyboardGame`.

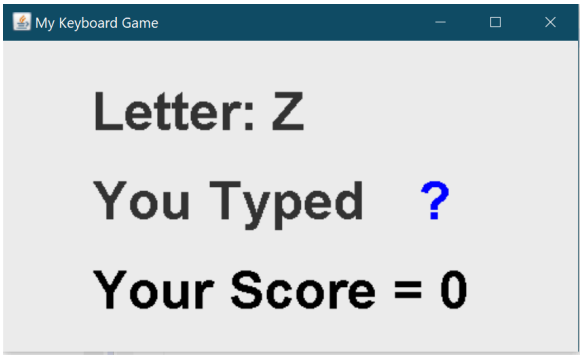
  Remark: a skeleton-code of the frame is as below:

```
JFrame frame = new JFrame(); //create an object of this frame

_________________________________ //create an object of BMICaculator
 frame.add(___________________________); //add JPanel into JFrame

//set a frame's resolution
frame.setTitle("My Keyboard Game");
 frame.setSize(500, 300); //set a frame's resolution
 frame.setLocationRelativeTo(null); //set a location at center the frame
 frame.setVisible(true); //set visible
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); //set default Close Operation
```

  c. Your running output of the `KeyboardGame` program, that makes the following GUI design as shown below:



Asst. Prof. Dr. Sasiporn Usanavasin, Dr. Jessada Karnjana, Dr. Kasorn Galajit and Dr. Akkarawoot Takhom {sasiporn.us, akkharawoot.aj, jessada.aj, kasorn.aj}@siit.tu.ac.th

School of Information, Computer, and Communication Technology, Sirindhorn International Institute of Technology, Thammasat University.

Exercise 2: (2 points)

- **Java Project**: <StudentID>_LAB10_KeyboardGame
- **Objective**: To learn how to register an appropriate listener to the source, and implement appropriate methods and their details for the specified listener to perform the assigned task.
- **Instruction:** Continue your work in Exercise 1, and write code in the following tasks.

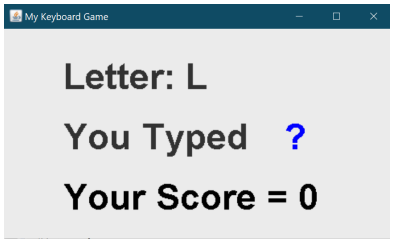
  This exercise, we are going to make a random character that appears at the first line after the `string Letter:` The letter changes every 1 second. To do this, you may follow the following instructions.

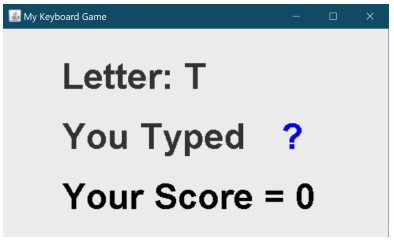
  a. Define the class `KeyBoardGame` to be an `ActionListener`.

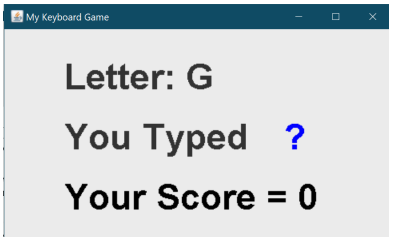
  b. Override the needed method for `ActionListener`.

  c. In the body of this method, use the method `getRandomChar()` to get the new character and then assign it to the property `displayedChar` then update the Graphics image.

  d. Run the `KeyBoardGame` program to see the changing letters in the output as shown below:







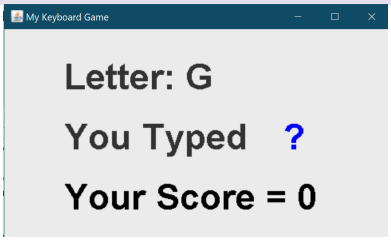
Asst. Prof. Dr. Sasiporn Usanavasin, Dr. Jessada Karnjana, Dr. Kasorn Galajit and Dr. Akkarawoot Takhom {sasiporn.us, akkharawoot.aj, jessada.aj, kasorn.aj}@siit.tu.ac.th

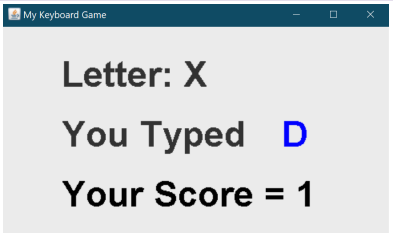
School of Information, Computer, and Communication Technology, Sirindhorn International Institute of Technology, Thammasat University.

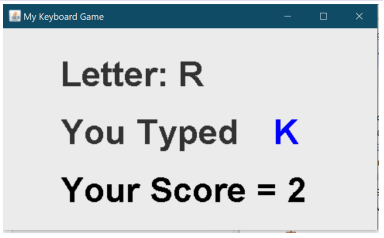
Exercise 3: (2 points)

- **Java Project**: <StudentID>_LAB10_KeyboardGame
- **Objective**:
  - To learn how to register an appropriate listener to the source, and implement appropriate methods and their details for the specified listener to perform the assigned task.
  - To use the `Timer` class in `javax.swing`, and use a `Timer` object to create basic animation
- **Instruction:** Continue your work in **Exercise 2**. In this exercise, we are going to get the response character from the keyboard and update the score if it matches with the `displayedChar`. To do this, you may follow the following instructions.
  a. Define the class `KeyBoardGame` to be a `KeyListener`.
  b. Override the needed method for `KeyListener`.
     - In the body of this method, get a character from the keyboard, assign it to the predefined response variable
     - If the variable response and displayed `Char` are the same, update the score.
     - Update the Graphics image from the `paintComponent` method
  c. Run the program to see the letter that you type and check if it runs correctly.

👉 To submit this exercise to TA, students **MUST** displays the `KeyBoardGame` program as below:







Asst. Prof. Dr. Sasiporn Usanavasin, Dr. Jessada Karnjana, Dr. Kasorn Galajit and Dr. Akkarawoot Takhom {sasiporn.us, akkharawoot.aj, jessada.aj, kasorn.aj}@siit.tu.ac.th

School of Information, Computer, and Communication Technology, Sirindhorn International Institute of Technology, Thammasat University.

Exercise 4: (2 points)

- **Java Project**: <StudentID>_LAB10_CatchingFlashBall
- **Objective**:
  - To learn how to register an appropriate listener to the source, and implement appropriate methods and their details for the specified listener to perform the assigned task.
  - To use the `Timer` class in `javax.swing`, and use a `Timer` object to create basic animation
- **Instruction:** Create a java project and write code in the following tasks.

a. Add a new java class `CatchingFlashingBall.`

b. Complete the program `CatchingFlashingBall` so that it shows the orange ball in the black panel and it changes the location of its appearance at random.

c. Define a class `TimerListener` which is a subclass of an interface `ActionListener` inside of the `CatchingFlashingBall` class.

   - **Remark:** The `TimerListener` class is <u>an inner class of the `CatchingFlashingBall` class</u>.

d. Override the abstract method of `ActionListener` for the `TimerListener` class.

   - Your program should draw a ball in `paintComponent`. To set the location of the ball, use a Random object to assign the new `xcenter` and `ycenter` of the ball. The method `int nextInt(int max)` of the class Random can be used to generate a random integer from 0 to max.
   - Remark that the width and height of the container can be obtained using `getWidth()` and `getHeight()`, respectively.

e. Add a new java class `CatchingFlashingBallTesting` and write a `main` method that has a frame to add an object `CatchingFlashingBall`.

   Remark: a skeleton-code of the frame is as below:

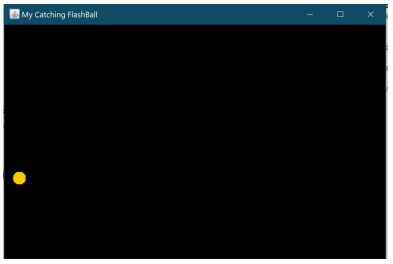
```
JFrame frame = new JFrame(); //create an object of this frame

_____________________________ //create an object of CatchingFlashingBall
 frame.add(__________________________); //add JPanel into JFrame

//set a frame's resolution
frame.setTitle("My Catching FlashBall"); //set Title of this frame
 frame.setSize(600, 400); //set a frame's resolution
 frame.setLocationRelativeTo(null); //set a location at center the frame
 frame.setVisible(true); //set visible
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); //set default Close Operation
```

Asst. Prof. Dr. Sasiporn Usanavasin, Dr. Jessada Karnjana, Dr. Kasorn Galajit and Dr. Akkarawoot Takhom {sasiporn.us, akkharawoot.aj, jessada.aj, kasorn.aj}@siit.tu.ac.th
School of Information, Computer, and Communication Technology, Sirindhorn International Institute of Technology, Thammasat University.

f. Run the `CatchingFlashingBall` program to see whether you get the correct output as shown below:



Exercise 5: (2 points)

- **Java Project**: <StudentID>_LAB10_CatchingFlashBall
- **Objective**:
  - To learn how to register an appropriate listener to the source, and implement appropriate methods and their details for the specified listener to perform the assigned task.
  - To use the `Timer` class in `javax.swing`, and use a `Timer` object to create basic animation
- **Instruction** Continue **Exercise 4**, and write code in the following tasks.

a. Define a class `Catcher` which is a subclass of an interface `MouseListener` inside of the `CatchingFlashingBall` class.
   - **Remark1:** The `Catcher` class is <u>an inner class of the `CatchingFlashingBall` class</u>. It can access all properties and methods of the `CatchingFlashingBall`.

b. Write the `MouseListener` and its required overridden methods to this program to get the location where the <u>mouse it pressed</u>.
   - Write more codes in <u>the overridden method</u> the `Catcher` to this program to get the location where the <u>mouse it pressed</u>.
   - **Remark2:** If the distance between the center of the ball and the mouse point location is less than the radius of the ball, the mouse point location is inside of the ball.

     The method `calculateDistance` between two points is provided as below:

```
double calculateDistance(int x1, int y1, int x2, int y2) {
        return Math.sqrt(Math.pow(x1 - x2, 2) + Math.pow(y1 - y2, 2));
```

Asst. Prof. Dr. Sasiporn Usanavasin, Dr. Jessada Karnjana, Dr. Kasorn Galajit and Dr. Akkarawoot Takhom {sasiporn.us, akkharawoot.aj, jessada.aj, kasorn.aj}@siit.tu.ac.th

School of Information, Computer, and Communication Technology, Sirindhorn International Institute of Technology, Thammasat University.

c. Also write more codes in the overridden method the `Catcher` to the program so that the ball stops when the user presses the mouse inside of it.

d. Run the `CatchingFlashingBall` program to see whether you get the correct output as shown below:

👉 To submit this exercise to TA, students **MUST** displays the Catching FlashingBall program as below: