

TD 2 : Gestion des événements

Objectifs

- se familiariser avec les différentes façons de gérer les événements.

Exercice 1. Différentes façons de gérer les événements

```
1 public class Ex1 extends Application {
2     Label label;
3
4     class myClickHandler implements EventHandler<MouseEvent> {
5         public void handle(MouseEvent event) {
6             label = new Label("toto");
7         }
8     }
9
10    public void start(Stage stage) {
11        VBox root = new VBox();
12        label = new Label("coucou");
13        Button button = new Button("change");
14        button.setOnMouseClicked(new myClickHandler());
15
16        root.getChildren().addAll(label, button);
17
18        Scene scene = new Scene(root);
19        stage.setTitle("Ex1");
20        stage.setScene(scene);
21        stage.show();
22    }
23
24    public static void main(String[] args) {
25        Application.launch(args);
26    }
27 }
```

Q 1. Quel est le type de classe utilisé pour gérer l'événement de clic sur le bouton ?

Q 2. Le code ci-dessus permet d'afficher une fenêtre avec un label et un bouton. Au lancement de l'application le label affiche le texte "coucou". On souhaite que le label affiche "toto" quand on clique sur le bouton mais ça ne fonctionne pas avec le code ci-dessus. Expliquez ce qu'il se passe et détaillez les modifications minimales à effectuer pour obtenir le résultat souhaité.

Q 3. La gestion des événements du bouton ne correspond pas à celle généralement attendue pour un bouton, corrigez là.

Q 4. Modifiez maintenant le code corrigé pour gérer l'événement de clic sur le bouton en utilisant une lambda expression. Note : le clic sur le bouton doit produire le même effet qu'avec le code ci-dessus.

Q 5. Faites la même chose en utilisant une classe externe.

Q 6. On souhaite maintenant garder le label dans la fenêtre mais avoir un nombre variable de boutons, définis au démarrage de l'application, et placés les uns en dessous des autres. Ces boutons seront intitulés "bouton 1", "bouton 2"... et on veut que le nom du bouton apparaisse dans le Label quand on clique dessus. En utilisant une seule boucle for, donnez le code permettant d'obtenir ce résultat.

Exercice 2. Carré perdu

```
1 public class Ex2 extends Application{
2
3     public void start(Stage stage) {
4         VBox root = new VBox();
5         Canvas canvas = new Canvas (500, 500);
6         GraphicsContext gc = canvas.getGraphicsContext2D();
7         Rectangle square = new Rectangle(50, 50, 20, 20);
8         gc.strokeRect(square.getX(), square.getY(), square.getWidth(), square.getHeight());
9
10        canvas.setOnMouseDragged(e -> {
11            gc.clearRect(0, 0, canvas.getWidth(), canvas.getHeight());
12            if (square.contains(new Point2D(e.getSceneX(),e.getSceneY())) {
13                square.setX(e.getSceneX() - 10);
14                square.setY(e.getSceneY() - 10);
15                gc.strokeRect(square.getX(), square.getY(), square.getWidth(), square.getHeight());
16            }
17        });
18
19        root.getChildren().add(canvas);
20
21        Scene scene = new Scene(root);
22        stage.setTitle("Hello square");
23        stage.setScene(scene);
24        stage.show();
25    }
26
27    public static void main(String[] args) {
28        Application.launch(args);
29    }
30 }
```

Q 1. Avec le code ci-dessus l'utilisateur perd le carré lorsqu'il bouge trop rapidement le pointeur de la souris. Expliquez l'origine du problème.

Q 2. Réalisez les modifications nécessaires pour résoudre ce problème.