

Examen

Interfaces graphiques

— Master d'informatique —

Janvier 2020, durée 2h.

L'examen se compose de trois exercices indépendants. Le seul document autorisé est une feuille de mémento (double A4). Tous les appareils électroniques y compris les montres sont interdits.

Lorsque la solution comporte du code **Java**, ces morceaux de code devront être écrits très lisiblement et très proprement pour être pris en compte. La présence de commentaires pertinents est bien sûr indispensable.

► **Exercice 1** L'interface du programme ci-dessous est constituée d'un unique bouton (**Button**). Le texte affiché par le bouton passe alternativement de **On** à **Off** à chaque clic sur celui-ci.

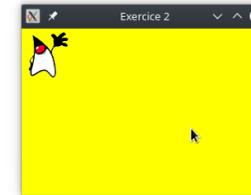


Le bouton est contrôlé par un objet d'une *classe interne anonyme*. Écrire le programme pour obtenir le résultat recherché, en complétant le trou dans le squelette de programme donné ci-dessous.

```
public class Exercice1 extends Application {
    boolean on = true;           // État du bouton
    Button bouton = new Button(on ? "On" : "Off");
    @Override
    public void start(Stage stage) {
        HBox.setMargin(bouton, new Insets(5, 5, 5, 5));
        // À COMPLÉTER ICI
        stage.setTitle("Exercice 1");
        stage.setScene(new Scene(new HBox(bouton)));
        stage.show();
    }
    public static void main(String[] args) {
        launch(args);
    }
}
```

Comment modifier le programme ci-dessus pour que le contrôleur du bouton soit la classe **Exercice1** ?

► **Exercice 2** L'interface du programme ci-dessous est constituée d'un panneau **Pane** et d'un objet **ImageView** contenant l'image d'un mobile. À chaque clic dans le panneau, le mobile vient se placer sous la souris. C'est le centre de l'image qui vient se placer sous la souris.



```
public class Exercice2 extends Application {
    @Override
    public void start(Stage stage) {
        Pane canvas = new Pane();
        canvas.setStyle("-fx-background-color: yellow;");
        ImageView iv = new ImageView(new Image("mobile.gif"));
        // À COMPLÉTER
        stage.setScene(new Scene(canvas, 300, 200));
        stage.setTitle("Exercice 2");
        stage.show();
    }
    public static void main(String[] args) {
        launch(args);
    }
}
```

► **Exercice 3** L'application ci-dessous affiche puis efface une cible circulaire rouge sur lequel l'utilisateur doit cliquer. Le rayon et la durée d'apparition de la cible sont contrôlés par deux curseurs (Slider). Écrire le programme pour obtenir le résultat recherché, en complétant les cinq trous dans le squelette de programme donné ci-dessous.



```
public class Exercice3 extends Application {
    boolean show = false;        // Cible présente
    boolean hit = false;        // Cible déjà touchée

    @Override
    public void start(Stage stage) {
        BorderPane root = new BorderPane();
        Label speedLabel = new Label(" Vitesse :");
        Slider speedSlider = new Slider(0.0, 100.0, 50.0);
        Label radiusLabel = new Label(" Rayon :");
        Slider radiusSlider = new Slider(2.0, 25.0, 10.0);

        // Mise en place des éléments de l'interface
        // À COMPLÉTER No 1

        Circle target = new Circle(100, 50, 20.0, Color.RED);
        // Liaison rayon de la cible avec le curseur
        // À COMPLÉTER No 2

        Pane canvas = new Pane();
        root.setCenter(canvas);
        canvas.setOnMousePressed(event -> {
            // Afficher "On target" si on clique sur la cible affichée
            // À COMPLÉTER No 3
        });

        Timeline forever = new Timeline(new KeyFrame(Duration.seconds(4), event -> {
            // Affichage de la cible aléatoirement positionnée
            // À COMPLÉTER No 4

            double time = 50/(1+speedSlider.getValue());
            // Suppression de la cible après time secondes
            // À COMPLÉTER No 5

        }));
        forever.setCycleCount(Animation.INDEFINITE);
        forever.play();

        stage.setScene(new Scene(root, 500, 250));
        stage.setTitle("Exercice 3");
        stage.show();
    }
    public static void main(String[] args) {
        launch(args);
    }
}
```