

## Certamen I ELO-329 Diseño y Programación Orientada a Objetos

En este certamen usted no podrá hacer preguntas. Si algo no está claro, indíquelo en su respuesta, haga una suposición razonable y resuelva conforme a ella.

Segunda parte, **con apuntes**:

### Tercera pregunta (35 pts.)

#### Detector de moscas

Se tiene un dispositivo detector de moscas que cuando éstas ingresan a su rango, éste alerta de su presencia.

- A. (17 puntos) Cree la clase FlyDetector y su programa principal TestDetector.java. Su programa tendrá sólo un detector de moscas y una única mosca será simulada moviendo un punto con las teclas w, a, s, z, las cuales cambian la posición (x, y) del punto a (x, y+5), (x-5, y), (x+5, y) y (x, y-5) respectivamente.

La ejecución del programa es del tipo:

```
$ java TestDetector <mx> <my> <rango>
```

Donde la posición inicial de la mosca es (mx, my). El detector se ubica en el origen y su radio de detección está definido por el parámetro "rango". Todos los parámetros son de tipo entero.

Si luego de mover la mosca, ésta se encuentra en la zona de detección, el detector de moscas debe enviar a consola el mensaje "Detectada".

El programa termina al ingresar x en lugar de un cambio de posición de la mosca (punto).

- B. (12 puntos) Cree JavaFxDetector.java, la versión gráfica del programa anterior en que el detector es representado por un círculo de radio 30 pixeles ubicado en el centro de la ventana. La mosca será representada por el mouse. Cada vez que el mouse ingresa al círculo, éste debe enviar a consola el mensaje "Detectada".

Sugerencia: Revise el ejemplo ShowCircleCentered.java

- C. (6 puntos) En un archivo readme.txt escriba el comando completo necesario para compilar TestDetector.java (pregunta b-a) y el comando necesario para compilar JavaFxDetector.java.

A)

```
import javafx.geometry.Point2D;
import javafx.scene.shape.Circle;

public class FlyDetector extends Circle {
    public FlyDetector(int r){
        super(r);
    }
    public void checkDetection (Point2D fly) {
        if (contains(fly))
            System.out.println("Detectado");
    }
}
```

```
import javafx.geometry.Point2D;
import java.util.Scanner;

public class TestDetector {
    public void executeUserInteraction (Scanner in, Point2D fly, FlyDetector flyDetector){
        char movement;
        boolean done =false;
        while (!done) {
            movement = in.next().charAt(0);
            switch (movement) {
                case 'w': fly = fly.add(0, 5);
                    break;
                case 'a': fly = fly.add(-5, 0);
                    break;
                case 's': fly = fly.add(5, 0);
                    break;
                case 'z': fly = fly.add(0, -5);
                    break;
                case 'x': done=true;
            }
            flyDetector.checkDetection(fly);
        }
    }
    public static void main(String [] args) {
        int mx = Integer.parseInt(args[0]);
        int my = Integer.parseInt(args[1]);
        Point2D fly = new Point2D(mx, my);
        int range = Integer.parseInt(args[2]);
        FlyDetector flyDetector = new FlyDetector(range);
        TestDetector td = new TestDetector();
        td.executeUserInteraction(new Scanner(System.in), fly, flyDetector);
    }
}
```

### Distribución de puntaje:

#### Aspectos funcionales (12 pts):

- 2 pts. Correcta lectura de las teclas de movimiento
- 1 pts. Condición de término del programa
- 2 pts. Desplazamiento adecuado del punto según se indica en enunciado
- 2 pts. Lógica de clase FlyDetector (notar que las implementaciones pueden variar)
- 2 pts. Lectura adecuada de los parámetros del programa
- 3 pts. Envío de mensaje bajo la condición correcta

#### Estructura del programa (5 pts)

- 2 pts Buen uso de lenguaje Java (métodos públicos etc)en clase FlyDetector
- 3 pts Buen uso del lenguaje Java en clase TestDetector y su método main

### B)

```
import javafx.application.Application;
import javafx.scene.layout.StackPane;
import javafx.stage.Stage;
```

```
import javafx.scene.Scene;
import javafx.scene.layout.Pane;
import javafx.scene.shape.Circle;
import javafx.scene.paint.Color; // los import no otorgan puntaje (lo hace el IDE)

public class JavaFxDetector extends Application {
    public void start(Stage primaryStage) {
        Pane pane = new StackPane();
        Circle circle = new Circle();
        circle.setRadius(30);
        circle.setStroke(Color.BLACK);
        circle.setFill(Color.WHITE);
        circle.setOnMouseEntered(e->System.out.println("Detectado"));
        pane.getChildren().add(circle);
        Scene scene = new Scene(pane, 200, 200);
        primaryStage.setTitle("P2b");
        primaryStage.setScene(scene);
        primaryStage.show();
    }
}
```

### Distribución de puntaje:

#### Aspectos funcionales (9 pts):

- 1 pts. La solución en efecto es gráfica.
- 2 pts. Ubica el detector en el centro de la ventana con la geometría pedida (30 pixles)
- 2 pts. Correcta configuración del evento “ingreso del mouse al detector” “setOnMouseEntered”
- 1 pts. Envío de mensaje a consola
- 3 pts. Correcta composición de los objetos (circule, pane, scene, stage) e invocación a show.

#### Estructura del programa (3 pts)

- 3 pts Buen uso de lenguaje Java y JavaFX

### C)

Comando para compilar TestDetector.java: // 3 pts.

```
$javac TestDetector.java
```

Comando para compilar JavaFxDetector.java: // 3 pts.

```
$ javac --module-path <path/to/javafx-sdk-17.0.2/lib> --add-modules javafx.controls JavaFxDetector.java
```

En < path/to/javafx-sdk-17.0.2/lib> reemplazar por la ruta completa donde está el directorio lib donde JavaFX fue ubicado.