

GBIN6U03 - Examen de seconde session Année 2016-2017

Durée : 2h.

Tout document ou calculatrice interdit.

Pour chaque question, une partie des points peut tenir compte de la présentation.

Le barème est indicatif.

1 Itérateur (4 points)

Soit les deux interfaces suivantes :

```
public interface Liste {  
    // retourne un itérateur sur le début de la liste  
    public IterateurListe listIterator();  
}
```

```
public interface IterateurListe {  
    // vrai s'il y a un élément après la position courante de l'itérateur  
    public boolean hasNext();  
    // renvoie l'élément situé juste après la position courante de l'itérateur  
    // puis incrémente la position courante de l'itérateur  
    public int next();  
    // ajoute un élément dans la liste juste avant la position courante de  
    // l'itérateur  
    public void add(int element);  
    // supprime l'élément qui a été retourné par le dernier appel à next  
    public void remove();  
};
```

En supposant que vous disposiez d'une classe `ListeChaine` qui implémente l'interface `Liste` et dotée d'un constructeur sans argument permettant de créer une liste vide, implémentez une méthode qui prend une liste en paramètre et renvoie une copie de cette liste privée de ses doublons.

2 Chifoumi (8 points)

Le chifoumi est un jeu dans lequel deux joueurs doivent secrètement choisir entre "pierre", "feuille" et "ciseaux". Un fois leur choix fait, les deux joueurs se le montrent simultanément. Le gagnant est déterminé par les règles suivantes :

- la "pierre" bat les "ciseaux"
- les "ciseaux" battent la "feuille"
- la "feuille" bat la "pierre"

Si les joueurs ont fait le même choix, ils recommencent jusqu'à déterminer un gagnant.

Dans cette partie, vous devez écrire un programme dans lequel deux nouveaux threads seront créés pour simuler chacun un joueur de chifoumi. Les threads devront jouer en concurrence puis se synchroniser pour déterminer un éventuel gagnant jusqu'à ce que l'un des deux gagne. A l'issue de la partie, le programme principal devra afficher le gagnant (premier ou second thread).

Attention : le programmes multi-threadé qui vous est demandé doit impérativement exécuter les différents threads de manière concurrente (et non les uns après les autres). De plus, pour synchroniser les threads entre eux, une attente active sera pénalisée par rapport à une synchronisation correcte utilisant les primitives de synchronisation fournies par java.

3 Balle rebondissante (8 points)

Dans cette partie, vous devez écrire les parties manquantes du programme ci-dessous en vous aidant des fonctions données en annexes. Le but est d'obtenir un programme ouvrant une fenêtre au centre de laquelle se trouve une balle A d'un diamètre de 20 pixels se déplaçant dans une direction aléatoire. Une autre balle B devra être centrée sur le pointeur de la souris et suivre tous ses déplacements. La balle A devra rebondir sur les bords de la fenêtre et sur la balle B.

```
import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.canvas.Canvas;
import javafx.stage.Stage;
import javafx.scene.layout.BorderPane;

public class ProgrammePrincipal extends Application {
    @Override
    public void start(Stage primaryStage) {
        primaryStage.setTitle("Seconde session, rebondissons !");
        Canvas c = new Canvas();
        MonApplication app = new MonApplication(c);
        BorderPane b = new BorderPane(c);
        c.widthProperty().bind(b.widthProperty());
        c.heightProperty().bind(b.heightProperty());
        Scene s;
        s = new Scene(b, 800, 600);
        primaryStage.setScene(s);
        primaryStage.show();
        app.start();
    }

    public static void main(String [] args) {
        launch(args);
    }
}
```

Annexes

Classe Canvas (dans `javafx.scene.canvas`) :

```
public GraphicsContext getGraphicsContext2D();
public final double getWidth();
public final double getHeight();
```

méthodes héritées de `Node` :

```
public final void setOnMouseMoved(EventHandler<? super MouseEvent> value);
```

Classe `AnimationTimer` (dans `javafx.animation`) :

```
public void handle(long time);
public void start();
```

Classe `EventHandler<T>` (dans `javafx.event`) :

```
public void handle(T event);
```

Classe `MouseEvent` (dans `javafx.scene.input`) :

```
public final double getX();
public final double getY();
```

Classe `GraphicsContext` (dans `javafx.scene.canvas`) :

```
public void clearRect(double x, double y, double w, double h);
public void strokeOval(double x, double y, double w, double h);
```