

RAPPORT TP2

Kotlin
BOURDIER Loann
G4B



Table des matières

I.	EXERCICE 1	3
<hr/>		
A.	RAPPEL DE L'ENONCE	3
B.	TABLEAU	3
C.	EXPLICATION DE QUELQUES METHODES	4
A.	MAINACTIVITY1	4
B.	MAINACTIVTY2	4
II.	EXERCICE 2	5
<hr/>		
A.	RAPPEL DE L'ENONCE	5
B.	EXPLICATION DES METHODES	5
A.	METHODE TOSTRING (CLASSE LIVRE)	5
B.	METHODE ONCREATE (MAINACTIVITY1)	5
C.	METHODE ONSAVEINSTANCESTATE (MAINACTIVITY1)	5
D.	METHODE ONRESTOREINSTANCESTATE (MAINACTIVITY1)	6

I. Exercice 1

A. Rappel de l'énoncé

Définir 2 activités dont le but est de passer d'une activité à une autre par un bouton « Aller sur l'activité suivante » sur la première activité et « Revenir à l'activité précédente » sur la seconde activité.

Pour cela il faut surcharger les méthodes listées ci-dessous dans les >2 activités et afficher un Toast avec un message qui est le nom de la méthode :

- onCreate
- onStart
- onResume
- onSaveInstanceState
- onRestoreInstanceState
- onPause
- onStop
- onRestart
- onDestroy

B. Tableau

onCreate	Entrée : savedInstanceState: Bundle? Lancement de l'activité Changement d'orientation
onStart	Entrée : rien Lancement de l'activité Retour activité 1 et lancement activité 2 Changement d'orientation Maximisation de l'appli
onResume	Entrée : rien Lancement de l'activité 1 Lancement de l'activité 2 Retour activité 1 Changement d'orientation Maximisation de l'appli
onSaveInstanceState	Entrée : outState: Bundle Lancement de l'activité 2 Minimisation de l'appli Changement d'orientation Quit
onRestoreInstanceState	Entrée : savedInstanceState: Bundle Changement d'orientation
onPause	Entrée : rien Lancement de l'activité 2 Retour activité 1 Changement d'orientation Minimisation de l'appli

onStop	Entrée : rien Passage de l'activité 2 Retour activité 1 Changement d'orientation Minimisation de l'appli Quit
onRestart	Entrée : rien Retour activité 1 Maximisation de l'appli
onDestroy	Entrée : rien Retour activité 1 Changement d'orientation Quit

C. Explication de quelques méthodes

a. MainActivity1

Les explications de chaque ligne de la méthode **onCreate** sont en commentaire ci-dessous

```

override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_main)

    Toast.makeText( context: this, text: "onCreate", Toast.LENGTH_SHORT).show() //creation du Toast
    val btnSvt :Button! = findViewById<Button>(R.id.btnSvt) //initialiser le bouton
    btnSvt.setOnClickListener { //lors de l appuie sur le btn
        val intent = Intent( packageContext: this, MainActivity2::class.java) //créer un intent pour changer d'activité
        startActivity(intent) //lancer l'activité
    }
}

```

Nous faisons pareil pour la méthode **onStart** et **onResume** en changeant que la partie « text: » de la fonction « makeText »

Les explications de la méthode **onSaveInstanceState** sont en commentaire ci-dessous

```

override fun onSaveInstanceState(outState: Bundle) {
    super.onSaveInstanceState(outState)
    Toast.makeText( context: this, text: "onSaveInstanceState", Toast.LENGTH_SHORT).show() //création du Toast
}

```

Nous faisons pareil pour la méthode **onRestoreInstanceState**, **onPause**, **onStop**, **onRestart**, **onDestroy** en changeant que la partie « text: » de la fonction « makeText »

b. MainActivity2

Les explications de chaque ligne de la méthode **onCreate** sont en commentaire ci-dessous

```

override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_main2)

    Toast.makeText( context: this, text: "onCreate", Toast.LENGTH_SHORT).show() //creation du Toast
    val btnPcd :Button! = findViewById<Button>(R.id.btnPrCd) //initialiser le bouton
    btnPcd.setOnClickListener { //lors de l appuie sur le btn
        finish() //Fin de méthode
    }
}

```

Nous faisons pareil pour la méthode **onStart** et **onResume** en changeant que la partie « text: » de la fonction « makeText »

Les explications de la méthode **onSaveInstanceState** sont en commentaire ci-dessous

```

override fun onSaveInstanceState(outState: Bundle) {
    super.onSaveInstanceState(outState)
    Toast.makeText( context: this, text: "onSaveInstanceState", Toast.LENGTH_SHORT).show() //création du Toast
}

```

Nous faisons pareil pour la méthode **onRestoreInstanceState**, **onPause**, **onStop**, **onRestart**, **onDestroy** en changeant que la partie « text: » de la fonction « makeText »

II. Exercice 2

A. Rappel de l'énoncé

Créer une classe Livre comportant les éléments cités ci-dessous et ajouter la méthode toString() :

- Titre
- Auteur
- Nombre de pages

L'objectif de cet exercice est d'utiliser les 2 méthodes de l'activité 1 afin de sauvegarder un livre et afficher ce dernier au prochain lancement de l'activité

- onSaveInstanceState
 - o Créer un nouveau livre
 - o Afficher le livre
 - o Sauvegarder le livre
- onRestoreInstanceState
 - o Rechercher le livre sauvegardé
 - o Afficher le livre

Pour différencier les livres on ajoutera un suffixe (nombre random) au Titre de ce dernier

B. Explication des méthodes

a. Méthode toString (classe Livre)

Le but de cette méthode est de s'occuper de l'affichage d'un livre

```
class Livre (var Titre : String, var Auteur : String, var NombrePages : Int)
/*Création d'un classe Livre avec comme paramètres le titre, l'auteur et le nombre de pages */ {
    var random :Int = (1..12).shuffled().first()
    // variable donnant un nombre aléatoire entre 1 & 12

    override fun toString(): String { //création de la méthode
        return ("Titre : $Titre $random\nAuteur : $Auteur\nNombre de pages : $NombrePages")
        //retourne le titre, l'auteur et le nombre de page du livre
    }
}
```

b. Méthode onCreate (mainActivity1)

Les explications de chaque ligne de la méthode **onCreate** sont en commentaire ci-dessous

```
val livre :String = Livre( Titre:"livre", Auteur:"max", NombrePages: 32).toString()
//On créer un livre en appelant la méthode toString de la classe Livre

override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_main)

    Toast.makeText( context: this, text: "onCreate", Toast.LENGTH_SHORT).show() //creation du Toast
    val btnSvt :Button = findViewById<Button>(R.id.btnSvt) //initialiser le bouton
    btnSvt.setOnClickListener { //lors de l'appui sur le btn
        val intent = Intent( packageContext: this, MainActivity2::class.java) //créer un intent pour changer d'activité
        startActivity(intent) //lancer l'activité
    }

    var afflivre :TextView = findViewById<TextView>(R.id.off) //initialisation du textView
    afflivre.setText(livre) //donne la valeur livre créé précédemment au textView
}
```

c. Méthode onSaveInstanceState (mainActivity1)

Les explications de chaque ligne de la méthode **onSaveInstanceState** sont en commentaire ci-dessous

```
override fun onSaveInstanceState(outState: Bundle) {
    super.onSaveInstanceState(outState)
    Toast.makeText( context: this, text: "onSaveInstanceState", Toast.LENGTH_SHORT).show()
    //creation du Toast
}
```

d. Méthode `onRestoreInstanceState` (`mainActivity1`)

Les explications de chaque ligne de la méthode **`onRestoreInstanceState`** sont en commentaire ci-dessous

```
override fun onRestoreInstanceState(savedInstanceState: Bundle) {  
    super.onRestoreInstanceState(savedInstanceState)  
    Toast.makeText( context: this, text: "onRestoreInstanceState", Toast.LENGTH_SHORT).show()  
    //creation du Toast  
  
    //Recuperer les paramètres sauvegarde précédemment  
    val prefs : SharedPreferences! = getSharedPreferences( name: "MyPrefs", MODE_PRIVATE)  
    val livreb :String? = prefs.getString( key: "son", defValue: "actif")  
    var aff :TextView! = findViewById<TextView>(R.id.aff) //initialisation du textView  
    aff.setText(livreb + '\n' + '\n' + livre) //donne au setText les 2 derniers livres ajoutés  
}
```