

Corrigé TD5

a. Soit la méthode suivante :

```
public void onCreateContextMenu(ContextMenu menu, View v,
    ContextMenuInfo menuInfo) {

    super.onCreateContextMenu(menu, v, menuInfo);
    menu.add(1, 1, 3, "Ajouter");
    menu.add(1, 2, 2, "Supprimer");
    menu.add(1, 3, 0, "Modifier");
    menu.add(1, 4, 1, "Quitter");
}
```

1. Définir l'élément graphique auquel elle est associée.

Cette méthode est associée à un menu contextuel.

Un menu contextuel est un menu flottant, associé à un événement sur un élément graphique de l'interface.

2. Que permet de faire cette méthode ? Dessiner son résultat.

Cette méthode permet d'ajouter quatre éléments au menu : Ajouter, supprimer, modifier et quitter.

Modifier
Quitter
Supprimer
Ajouter

b. Soit la méthode suivante :

```
public boolean onContextItemSelected(MenuItem item) {  
    ...  
    return super.onContextItemSelected(item);  
}
```

1. Que permet de faire cette méthode ?

Elle permet d'associer un comportement à chaque élément du menu.

2. On voudrait faire appel respectivement aux méthodes *ajouter()*, *supprimer()* et *modifier()* si on clique sur *Ajouter*, *Supprimer* ou *Modifier*, et fermer l'activité si on clique sur *Quitter*. Donner le code nécessaire pour faire cela.

```
switch (item.getItemId()){  
    case 1:  
        ajouter();  
        break;  
    case 2:  
        supprimer();  
        break;  
    case 3:  
        modifier();  
        break;  
    case 4:  
        finish();  
}
```

c. Quelle est la méthode qu'on doit surcharger si on veut définir le **contenu** du menu affiché quand on clique sur le bouton *Menu* du téléphone ? Donnez le code nécessaire pour faire cela.

onCreateOptionsMenu

```
public boolean onCreateOptionsMenu(Menu menu) {  
  
    menu.add(1, 1, 3, "Ajouter");  
    menu.add(1, 2, 2, "Supprimer");  
    menu.add(1, 3, 0, "Modifier");  
    menu.add(1, 4, 1, "Quitter");  
    return true;  
}
```

Corrigé TD6

Vous allez modifier le fichier *CalculActivity.java*

Q1 : Qu'est-ce qu'une activité en Android ? De quelle classe doit-elle hériter ?

Une activité est la composante principale pour une application Android. Elle représente l'implémentation métier dans une application Android, permettant de gérer l'ensemble des vues et ressources.

Elle hérite de la classe Activity.

Q2 : Quelle est la différence entre une activité et une application ? (0,5pt)

Une application peut contenir plusieurs activités.

Votre application va avoir le comportement suivant :

- L'utilisateur saisit deux entiers dans les champs d'édition *val1* et *val2*
- Il choisit l'opération à faire : *somme* ou *différence* en cliquant sur le bouton radio associé
- Il clique sur le bouton *calculer*. Le résultat est affiché dans le champs de texte *res*.

Q3 : Donner la ligne de code nécessaire pour associer l'interface définie avec l'activité *CalculActivity*.

setContentView(R.layout.choixOperation)

Vous avez créé une méthode dans le code pour définir le comportement de votre bouton.

Q4 : Donner la signature de cette méthode (son modificateur de visibilité, son type de retour, son nom et des paramètres).

public void calculer(View v){...}

public void onClick(View v) est acceptée aussi.

Vous avez créé deux attributs représentant les deux boutons radios :

```
private RadioButton b1;  
private RadioButton b2;
```

Vous voulez que *b1* soit associé au bouton radio *somme* et *b2* soit associé à *diff*.

Q5 : Dans quelle méthode proposez-vous de faire ces initialisations ? pourquoi ?

Dans la méthode onCreate

Q6 : Donner les lignes de code nécessaires pour faire cela.

```
b1 = (RadioButton) findViewById(R.id.somme) ;  
b2 = (RadioButton) findViewById(R.id.diff) ;
```

Passage d'une activité à une autre

Vous désirez créer une deuxième activité, appelée *ResultActivity*, pour y afficher le résultat.

Q7 : Donner les étapes nécessaires pour créer une autre activité.

- *Créer une nouvelle classe Java pour cette activité (elle doit hériter de Activity et contenir une méthode onCreate)*
- *Créer une nouvelle interface dans le répertoire layout*
- *Associer l'interface avec l'activité (grâce à setContentView)*
- *Ajouter l'activité dans le fichier Manifest.*

Le but est d'afficher une nouvelle interface en cliquant sur le bouton. Pour cela, nous avons besoin d'un *Intent*.

Q8 : Qu'est-ce qu'un Intent ?

Un Intent est un conteneur d'information. Il permet de passer des messages entre deux activités. L'activité appelante pourra ainsi transmettre des informations à l'application appelée, ainsi qu'au système Android.

Pour créer un nouvel Intent, nous avons besoin de la ligne de code suivante :

```
Intent i = new Intent(<contexte>, <cible>);
```

Q9: Quelle est la valeur de *contexte* et de *cible* dans notre cas ?

Context = CalculActivity.this

Cible = ResultActivity.class

Q10 : Donner la ligne de code nécessaire pour lancer la deuxième activité.

startActivity(i) ;

Dans notre application, nous avons calculé le résultat à afficher et l'avons stocké dans la variable *resultat*. Nous désirons l'envoyer à l'activité suivante.

Q11 : Donner la ligne de code nécessaire pour envoyer la variable *resultat* dans un message qu'on nommera *mess*.

i.putExtra("mess", resultat) ;