

TD04

Matière : DEVELOPPEMENT D'APPLICATIONS MOBILES NATIVES

Classe : SEM21

## Questions de cours

- 1- Donner le nom de quatre technologies d'écran tactile.
- 2- Quels sont les trois processeurs standards vendus par ARM ? quel est le processeur le plus performant et quel est le processeur qui consomme moins d'énergie que les autres ?
- 3- Quels sont les trois types d'applications mobiles ?
- 4- Qu'est-ce qu'une application mobile native ?

## Problème

"MeteoVille" est une application android qui permet de saisir les données météorologiques d'une ville (exemple : "Sfax"). Les données météorologiques sont : la vitesse du vent (exemple : 15), l'heure (exemple : hL=6) et les minutes (exemple : mL=15) du lever du soleil et l'heure (exemple : hC=17) et les minutes (exemple : mC=45) de son coucher.

L'état de la mer est calculé en respectant le tableau suivant :

Indice dans spVent	0	1	2
Etat Mer	Calme	Agitée	Très agitée

La longueur de la journée est calculée en fonction de hL, mL, hC et mC.

**Exemple** : hL=6 mL=15 hC=17 mC=45

longueurEnMinute = (17\*60+45) - (6\*60+15) = 690

nbHeure = 690 div 60 = 11

nbMinute = 690 mod 60 = 30

Le longueur de la journée est : "11 H 30 Min"

Lorsque l'utilisateur clique sur le bouton **btnCalculer**, si edVille ou edHL ou edML ou edHC ou edMC est vide, un message d'erreur "Remplir toutes les données SVP!" s'affiche, sinon si toutes les données sont valides (hL<23 et mL<60 et hC<23 et mC<60 et hL<hC) alors l'état de la mer est affiché dans edEtatMer et la longueur de la journée est affichée dans edLongJour, sinon un message d'erreur "Vérifier les données SVP!" est affiché.

Lorsque l'utilisateur clique sur le bouton **btnAfficher**, si edVille et edEtatMer et edLongJour ne sont pas vides, un message qui contient la ville, l'état de la mer et la longueur de la journée est affiché.

Lorsque l'utilisateur clique sur le bouton **btnEffacer**, toutes les EditText sont effacées, spVent sélectionne la première valeur (15) et le curseur est placé dans edVille.

En utilisant les indications :

- 1- Donner le code de la méthode init().
- 2- Donner le code de la méthode ajouterEcouteurs().
- 3- Donner le code de la méthode calculer().
- 4- Donner le code de la méthode afficher().
- 5- Donner le code de la méthode effacer().

## Indications

L'interface de l'activité est la suivante :

The screenshot shows the 'MetoVille' application interface. It has a blue header with the title 'MetoVille'. Below the header, there are two main sections: 'Données' (Data) and 'Resultats' (Results).

**Données section:**

- 'Ville' (City) is set to 'Sfax'.
- 'Vitesse Vent' (Wind Speed) is set to '30' with a unit dropdown showing 'Km/H'.
- 'Lever de soleil' (Sunrise) is set to '6 H' and '15 Min'.
- 'Coucher de soleil' (Sunset) is set to '17 H' and '45 Min'.
- At the bottom of this section are three buttons: 'CALCULER', 'AFFICHER', and 'EFFACER'.

**Resultats section:**

- 'Etat de la mer' (Sea State) is set to 'Agitée'.
- 'Longueur de la journée' (Day Length) is set to '11 H 30 Min'.

**Contenu de spVent dropdown:**

A dropdown menu labeled 'Contenu de spVent' is shown to the right of the 'Vitesse Vent' field. It contains three options: '15', '30', and '60'. An arrow points from the '30' option in the dropdown to the 'Vitesse Vent' input field.

Les trois messages de l'activité sont les suivants :

Longueur de la journée

Remplir toutes les données SVP!

Longueur de la journée

11 H 30 Min

Vérifier les données SVP!

Agitée

Ville : Sfax

Etat de la mer : Agitée

Longueur de la journée : 11 H 30 Min

Le code de la classe MainActivity est le suivant :

```
package c.com.metoville;
//imports
public class MainActivity extends AppCompatActivity {
    private EditText edVille;
    private Spinner spVent;
    private EditText edHL;
    private EditText edML;
    private EditText edHC;
    private EditText edMC;
    private Button btnCalculer;
    private Button btnAfficher;
    private Button btnEffacer;
    private EditText edEtatMer;
    private EditText edLongJour;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        init();
    }
}
```

```
private void init() {
    ...
}

private void ajouterEcouleurs() {
    ...
}

private void calculer() {
    ...
}

private void afficher() {
    ...
}

private void effacer() {
    ...
}
}
```

```
}
```