



Université Abdelmalek Essaadi

Faculté des sciences et techniques de Tanger

Département d'Informatique

Projet

CYCLE D'INGÉNIEUR

LOGICIELS ET SYSTÈMES INTELLIGENTS

**Sujet: Application mobile: Calculatrice
Scientifique.**

Réalisé par :

MOUDDENE LAMIAE.

SALMI ALI.

Encadré par:

ZILI HASSAN

Remerciements

Avant de commencer la présentation de ce travail, nous profitons de l'occasion pour remercier ALLAH pour la volonté, la force et la santé qu'il nous a donné afin de réaliser ce travail.

Au terme de ce projet, nous tenons à remercier toute personne ayant contribué, de près ou de loin, à l'aboutissement de ce travail, en particulier :

Nos respectueuses gratitudee à notre cher professeur Monsieur ZILI pour nous avoir donné le goût de l'apprentissage et d'avoir partagé avec nous sa passion pour l'enseignement. Nous avons grandement apprécié son implication et sa qualité d'enseignement dont nous avons bénéficié tout au long du semestre.

C'est grâce à lui que nous avons acquis de nouvelles compétences indispensables pour la réalisation de ce projet.

Merci à nos camarades de classe pour avoir partagé avec nous leurs connaissances.

Table de matière :

Remerciements	2
Table de matière :	3
Introduction :	3
I- Structure du projet :	4
I.1 Activity_main.xml :	4
I.2 MainActivity.java :	6
I.3 Interface graphique :	7
II- Application :	7
II.1 La méthode eval(String):	8
II.2 Opération d'addition :	9
II.3 Opération de division :	11
II.4 Opération de cosinus :	12
II.5 Opération de l'inverse :	13
II.6 La constante Pi:	15
II.7 Opération multiple:	15
III- Conclusion :	17

Introduction :

Développer une application Android sur mesure permet de répondre à vos besoins et à ceux de vos utilisateurs tant du point de vue fonctionnel, que qu'en matière de **mobilité** et de **praticité**. Aujourd'hui, la majorité des personnes ont leur téléphone toujours à proximité et consomment plusieurs heures de temps d'écran journaliers. C'est donc le périphérique idéal pour être sûr d'être avec eux :)

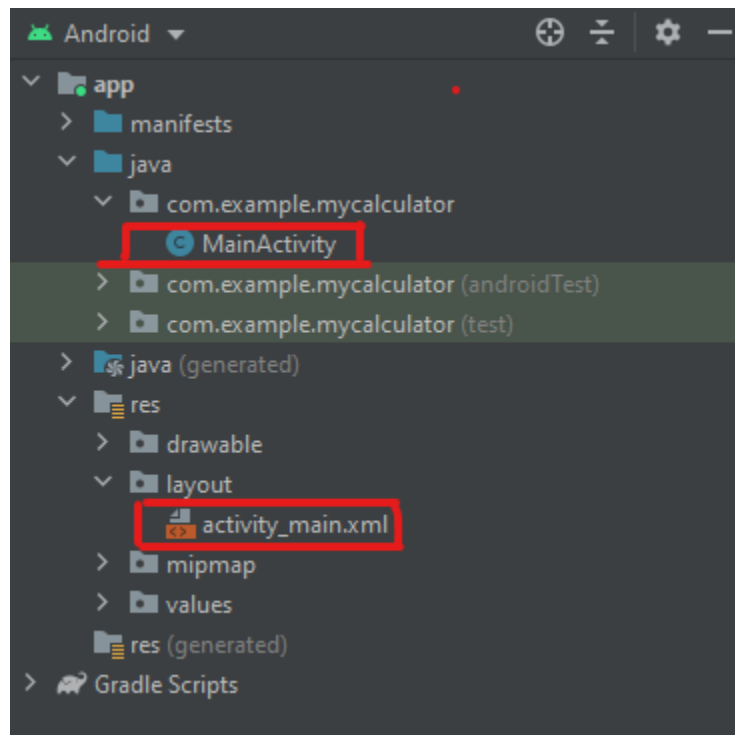
Rappelons-le, créer une application Android, c'est créer une application compatible avec le système d'exploitation Android. La flexibilité d'Android permet de **créer une unique application qui peut être rapidement adaptable aux différents environnements (smartphones, tablettes, montres connectées)**. La majeure partie des applications Android est distribuée via Google Play Store.

Le terme « application Android » s'avère quelque peu inexact puisque Android ne possède pas son propre langage de programmation. Nos applications Android peuvent être développées en natif avec le langage Kotlin (recommandé par Google) et le langage JAVA, et également en crossplatform avec IONIC ou React Native.

L'outil principal pour le développement d'applications Android s'appelle **Android Studio**. Cet outil va permettre de créer les interfaces graphiques avec lesquels l'utilisateur interagira. Android Studio permet également de compiler, de déboguer et de signer une application.

Dans ce projet, on va réaliser une calculatrice scientifique avec le langage JAVA à l'aide d'**Android Studio**.

1- Structure du projet :



Le fichier **activity_main.xml** est la partie « externe » de la structure de notre projet (barre d'outils, bouton, TextView...). Alors ce fichier déterminera comment l'interface utilisateur globale de l'activité devrait être.

Le fichier **MainActivity.java** contient les actions et tous ce qui concerne les interactions d'utilisateurs avec l'interface graphique de l'application.

1.1 Activity_main.xml :

Exemple:

On va prendre comme exemple le bouton **C** :

```
<Button
    android:id="@+id/bc"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_weight="1"
    android:background="#000"
    android:text="C"
    android:textColor="#FFFFFF"
```

```

    android:textSize="30sp"
    app:backgroundTint="#E30202">

</Button>

```

Ligne 1 : cela indique ID de ce bouton.

Ligne 2 : **layout_width** spécifié la largeur de base de la vue, et l'attribut « **match_parent** » signifie que la vue doit être aussi grande que son parent.

Ligne 3 : **layout_height** spécifié la hauteur de base de la vue, et l'attribut « **match_parent** » signifie que la vue doit être aussi grande que son parent.

Ligne 3 : : **layout_weight** nous permettre de spécifier un rapport de taille entre plusieurs boutons.

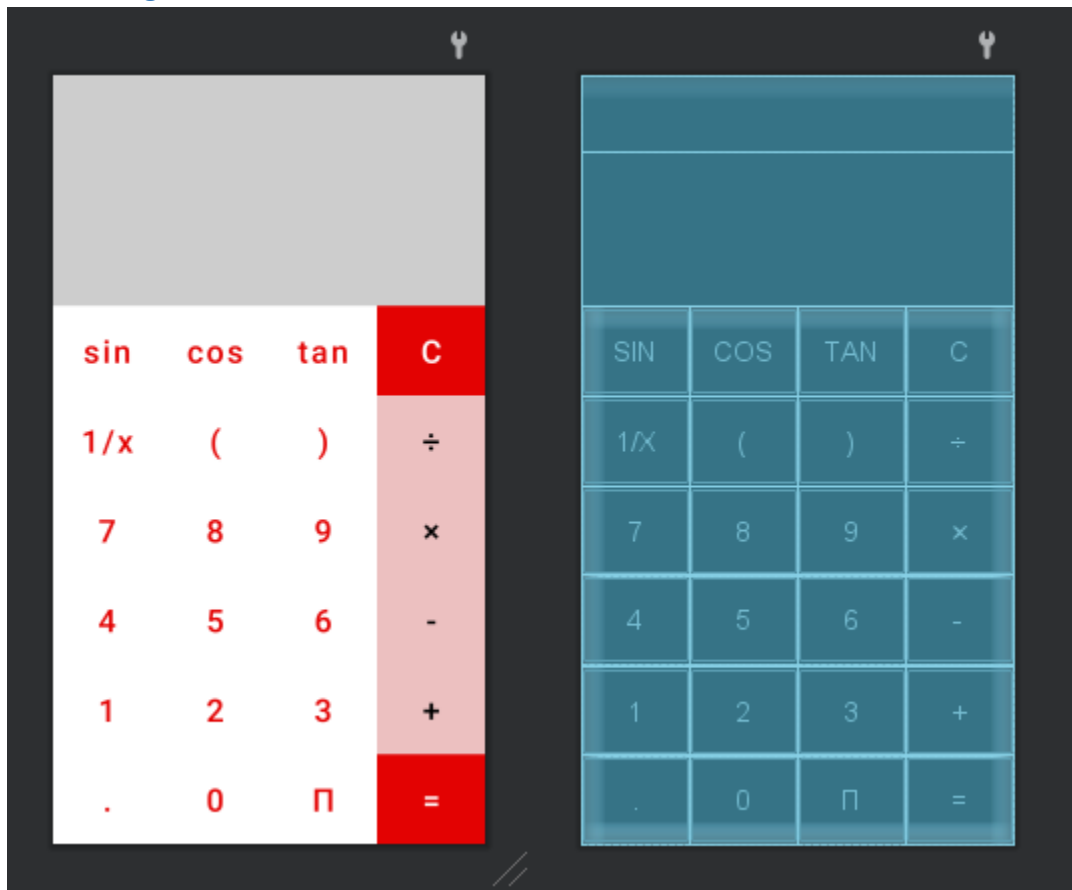
Ligne 4 : c'est la couleur du bouton.

Ligne 5: cela indique le texte de ce bouton.

Ligne 6 : c'est la couleur du texte de ce bouton.

Ligne 7 : c'est la taille de notre bouton.

1.2 Design:



I.3 MainActivity.java :

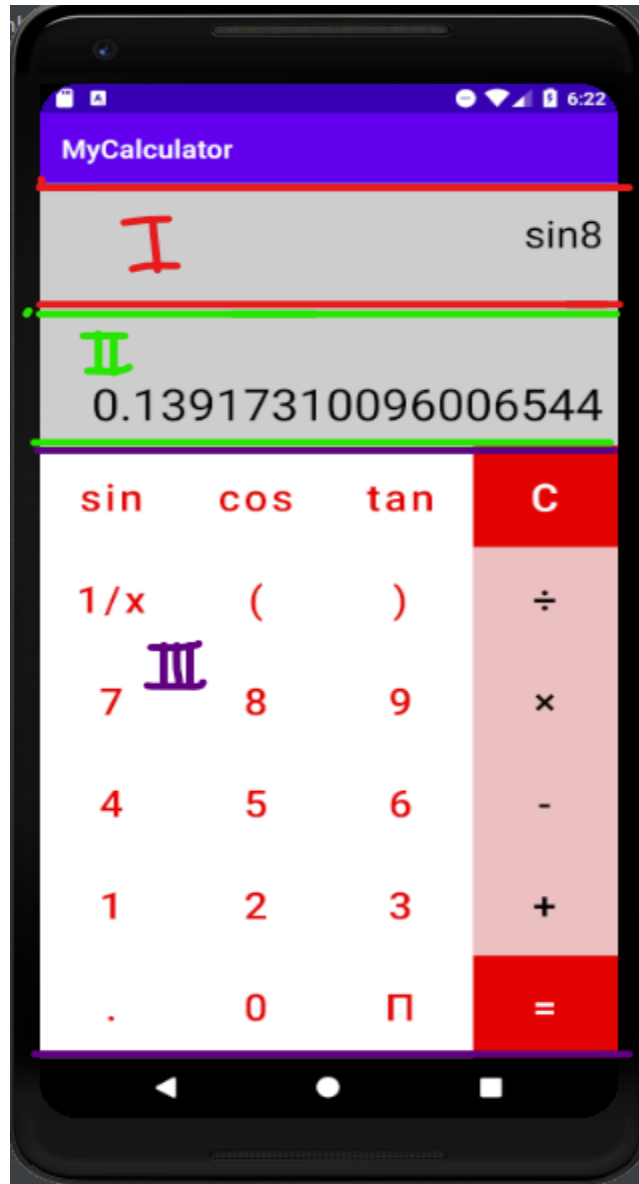
Exemple:

On va prendre comme exemple l'action sur le bouton **C** :

```
bc.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        tvmain.setText("0");  
    }  
});
```

La fonction **onClick** sera appelée lorsqu'on va cliquer sur le bouton **bc** à l'aide de la fonction **setOnClickListener**.

I.4 Interface graphique :



Pour ce projet on a décidé de deviser notre interface à 3 parties essentiels :

La 1ère partie a comme rôle d'afficher la dernière opération effectué.

La 2ème partie est chargée d'afficher l'opération courante, et aussi d'afficher le résultat de cette dernière.

La 3ème partie est chargée d'afficher tous les boutons de notre calculatrice.

99- *Application :*

II.1 La méthode eval(String):

Cette méthode permet de diviser la chaîne de caractère passer en paramètre en des sous-chaînes, ensuite elle converti chaque sous chaîne à un variable **Double** pour effectuer l'opération demandé par l'utilisateur.

Lors du clique sur le bouton égale la méthode **eval()** sera appelé, et elle va prendre comme paramètre la chaîne du caractère stocké au niveau de l'écran .

```
public static double eval(final String str) {
    return new Object() {
        int pos = -1, ch;
        void nextChar() {
            ch = (++pos < str.length()) ? str.charAt(pos) : -1;
        }
        boolean eat(int charToEat) {
            while (ch == ' ') nextChar();
            if (ch == charToEat) {
                nextChar();
                return true;
            }
            return false;
        }

        double parse() {
            nextChar();
            double x = parseExpression();
            if (pos < str.length()) throw new RuntimeException("Unexpected: "
+ (char)ch);
            return x;
        }

        double parseExpression() {
            double x = parseTerm();
            for (;;) {
                if      (eat('+')) x += parseTerm(); // addition
                else if (eat('-')) x -= parseTerm(); // subtraction
                else return x;
            }
        }

        double parseTerm() {
            double x = parseFactor();
            for (;;) {
                if      (eat('*')) x *= parseFactor(); // multiplication
                else if (eat('/')) x /= parseFactor(); // division
                else return x;
            }
        }

        double parseFactor() {
            if (eat('+')) return parseFactor(); // unary plus
            if (eat('-')) return -parseFactor(); // unary minus

            double x;
            int startPos = this.pos;
```



```

        if (eat('(')) { // parentheses
            x = parseExpression();
            eat(')');
        } else if ((ch >= '0' && ch <= '9') || ch == '.') { // numbers
            while ((ch >= '0' && ch <= '9') || ch == '.') nextChar();
            x = Double.parseDouble(str.substring(startPos, this.pos));
        } else if (ch >= 'a' && ch <= 'z') { // functions
            while (ch >= 'a' && ch <= 'z') nextChar();
            String func = str.substring(startPos, this.pos);
            x = parseFactor();

            if (func.equals("sin")) x = Math.sin(Math.toRadians(x));
            else if (func.equals("cos")) x = Math.cos(Math.toRadians(x));
            else if (func.equals("tan")) x = Math.tan(Math.toRadians(x));

            else throw new RuntimeException("Unknown function: " + func);
        } else {
            throw new RuntimeException("Unexpected: " + (char)ch);
        }

        if (eat('^')) x = Math.pow(x, parseFactor()); // exponentiation

        return x;
    }
}.parse();
}

```

II.2 Opération d'addition :

Exemple :

Avant le clique sur le bouton égale:



Après le clique sur le bouton égale :



II.3 Opération de division :

Exemple :

Cas de la division sur 0 :



On remarque que dans le cas de la division sur 0 le résultat obtenu est **infinity** qui signifie l'infinie.

II.4 Opération de cosinus :

Exemple :



II.5 Opération de l'inverse :

Exemple :

L'inverse du nombre 0 :



On remarque que l'inverse du nombre 0 est égale à **infinity** qui signifie l'infinie.

II.6 La constante Pi:



Lorsqu'on clique sur le symbole « **pi** » le nombre « **3.1415..** » est affiché à l'écran.

II.7 Opération multiple:

Exemple 1 :



Exemple 2 :



999- Conclusion :

Grace à ce projet que nous avons eu l'opportunité de cumuler les connaissances théoriques avec celles de la pratique, ceci permet également de rentrer dans la vie active et découvrir plus précisément le milieu professionnel. Ce projet consiste à découvrir les outils de réalisation d'une Application mobile, à partir de notre vision et point de vue pour objectif de réaliser des différentes parties design et code JAVA , à l'aide de Android Studio. L'élaboration de ce rapport a pour principale source nos connaissances acquises tout le long de notre formation scolaire et nos recherches personnelles.