Projet tuteuré

-

Semestre 2

# C:\Users\Sébastien\git\Projetcalculatrice\ProjetS2\logo.png IUT Informatique Rodez - 1ère année -2014/2015-

# Sujet : Mini-Calculatrice / Mini-Tableur

# -En Java-

MIQUEL Jonathan, BAUBE Maxime, PERIES Mickaël, GRANIER Vincent, SANCHEZ Sébastien

Dossier-Projet

# A - Plan Projet

## I - Introduction

Nous allons réaliser un projet tuteuré dans le cadre de notre première année de DUT Informatique. Ce projet aura pour but de créer une application de Mini-Calculatrice / Mini-Tableur. Nous avons décidé de développer cette application en mode graphique, puisque, étant un groupe constitué de 5 personnes, nous pouvons répartir plus de tâches entre les membres du groupe, et donc développer une application plus complète.

Ce développement se fera à l'aide de plusieurs concepts et outils, qui permettront de réaliser la définition des besoins, la planification, la conception de l'application (COO), le développement en Java, la création d'IHM et la gestion de projet.

## II - Organisation du projet

### - Présentation du projet :

L'application que nous allons réaliser se doit d'être simple et efficace dans son utilisation, la calculatrice doit permettre des calculs simples, avec parenthèse et utilisant des cases mémoire. Ces dernières pourront être gérés via un gestionnaire de mémoire. Le tableur devra permettre les mêmes opérations (excepté avec les cases mémoire). Ces calculs seront stockés dans la cellule du tableur du choix de l'utilisateur. On pourra effectuer des modifications du tableur via des commandes qui permettront par exemple de copier les valeurs de cellules, ou de les remettre à zéro.

### - Cahier des charges initial :

### - Nos objectifs

Nous souhaitons développer ce projet en interface graphique, afin de mieux comprendre le fonctionnement des bibliothèques swing et awt en java. Cela permettra un accès plus rapide et simple à l'application, que n'importe qui pourra utiliser facilement. Cela permettra également au groupe de s'exercer à la mise en application du cours d'IHM dispensé en DUT Informatique. De plus, nous pensons qu'un tableur en mode console aurait été peu lisible. Nous comptons réaliser le projet dans sa totalité, toutes les commandes seront donc fonctionnelles dans le livrable final.

### - Finalités

Notre projet ne présentera aucune finalité financière, puisqu'il n'est pas destiné à la vente. La finalité stratégique de celui-ci étant simplement de nous faire découvrir le développement d'une "vrai" application informatique, et par conséquent de nous expérimenté sur la gestion de projet informatique, la conception orientée objet, la programmation orientée objet et la conception d'IHM.

### - Chef de projet : choix et rôle

Le chef de projet qui a été désigné pour diriger le développement de cette application est Sébastien SANCHEZ. Etant à l'aise en programmation, mais également en communication, il semble capable de diriger ce projet correctement, en pouvant aider les autres à avancer mais également en définissant les spécifications qui aideront les programmeurs dans leurs tâches.

Au sein du groupe, le chef de projet devra permettre à tous d'exprimer correctement leurs capacités (programmation, conception, IHM, communication... ) en leur fournissant des spécifications claires, associées à des tâches adaptées à chacun. En effet, nous sommes tous étudiants, et nous devons donc veiller à ce que le projet permette à tout le monde d'apprendre et de s'exercer, peu importe ses capacités. De plus, il devra également s'atteler aux tâches de programmation et de communication. En effet, pour le cas spécifique de notre projet, le chef de projet n'aura que peu de travail à fournir. Le principal travail étant la définition des tâches et la spécification (cahier des charges) et l'organisation assez simple grâce à l'utilisation de Git (voir plus bas : gestionnaire de configuration), le chef de projet pourra aider au développement de l'application.

### - Autres rôles importants

- Le gestionnaire de configuration : Mickaël PERIES sera chargé de cette mission, il devra s'assurer que chacun puisse avoir accès au projet, chez lui ou à l'IUT. Pour cela, il devra apprendre à utiliser Git, GitHub et EGit, puis aider les autres membres du groupe à l'utiliser.

- Le secrétaire : Le secrétaire sera chargé de rédiger les documents nécessaires au cours du projet. Ce rôle a dans un premier temps été attribué à Maxime BAUBE, mais nous ferons tourner ce rôle au sein du groupe.

### - Livrables

A la fin du projet, nous devrons livrer ce dossier (plan projet, spécifications, conception, codage, tests, conclusion), ainsi qu'un CD-ROM contenant également le dossier, un installateur de l'application, le code, et le PowerPoint qui sera utilisé pour la soutenance orale.

### - Cadre du projet

Qualité : Le cahier des charges devra être respecté dans sa totalité, l'application testée et fonctionnelle, et le dossier correctement rédigé.

Coûts : Le projet ne représentera aucun coût puisque nous ne sommes pas rémunérés. L'application ne sera pas vendue donc le retour sur investissement sera nul également.

Délais : Le projet dans sa totalité devra être livré le 1er juin au plus tard. La planification qui suit montre comment nous nous organiserons pour atteindre ce but.

### - Exigences / Risques identifiés

Le principal risque lié aux exigences est le délai, qui devra absolument être respecté. Ainsi, la planification devra être respectée au mieux pour ne pas engendrer de retard, et pour éviter de ne pas pouvoir fournir une application incomplète.

### - Planification

Nous avons suivi la planification suivante, mise au point par le Chef de projet, grâce à Microsoft Projects. De plus, les jalons ont été les réunions avec la MOA/MOE, dont les comptes-rendus suivent. Nous avons choisi de suivre un cycle itératif, permettant de mieux géré chaque méthode lorsqu'elle est implémentée dans l'application, puisqu'elles seront testées directement. Ainsi,

### - Bilan de la gestion de projet

La planification initiale ne s'est pas éloignée de la réalité. Nous avons réussi à suivre les délais sans prendre de retard. Cependant, les durées des tâches n'ont pas été exactement celles prévues, par exemple, la mise en place des calculs simples a été plus courte que prévue, tout comme la gestion de la mémoire, à l'inverse, l'implémentation des parenthèses et des calculs à plusieurs opérandes a été beaucoup plus longue que prévue, nécessitant 2 développeurs, pendant plus de temps. Ces changements se sont finalement équilibrés en fin de projet. La gestion de projet s'est, elle déroulée sans encombre, l'entente entre les différents membres du groupe a été très bonne même dans les moments de stress, et de prise de décision, qui se sont finalement faites assez facilement, en écoutant les avis de chacun.

# B - Spécifications

### Cahier des charges

Nous avons adapté le cahier des charges fourni par la MOA afin qu'il corresponde mieux à nos attentes, notamment au niveau de l'IHM.

Nous allons développer une Mini-calculatrice et un Mini-tableur en Java. Le programme sera divisé en deux parties indépendantes : une partie calculatrice qui comportera la mini-calculatrice de base et le gestionnaire de mémoire, et une seconde partie qui sera le mini-tableur.

Le résultat sera une application graphique complète avec laquelle nous pourrons lancer au choix la calculatrice ou le tableur.

### I - Mini-calculatrice simple

#### a) Interface graphique - fonctionnement général

L'interface graphique de la partie calculatrice sera simple :

- L'utilisateur pourra entrer les instructions de calcul qu'il souhaitera exécuter dans un champ modifiable.

- L'utilisateur appuie sur la touche "Entrée" du clavier, ou bien sur un bouton "Calculer" situé à droite du champ où il entrera les calculs à effectuer pour exécuter les commandes qu'il entre.

- Lorsque le calcul est effectué, le champ modifiable est vidé, et la commande entrée est affichée dans une aire de texte non modifiable suivie du résultat, affiché à la ligne, précédé du symbole "=", ou d'une erreur si la commande est incorrecte.

#### b) Fonctionnement du champ d'exécution de commandes : mode calculatrice simple

##### 1) Calculs simples

- Une instruction de calcul simple sera constituée de 3 parties :

*nombre opérateur nombre*

- Le nombre pourra être un entier ou un réel, l'utilisateur pourra entrer un nombre négatif à l'aide du signe "-" (ex : -15) et un réel grâce au signe "." (ex : 5.1) ; les deux signes pourront être combinés.

- Les opérateurs seront :"+" pour l'addition ; "-" pour la soustraction ; "\*" pour la multiplication ; "/" pour la division.

- L'utilisateur pourra insérer autant d'espaces qu'il le souhaite entre les différents composants de l'instruction, mais l'instruction ne pourra pas dépasser 75 caractères.

- Il pourra également entrer autant de nombre qu'il souhaite, séparés par des opérateurs (ex : 1+2\*1+5/2)

##### 2) Calculs avec parenthèses

L'utilisateur pourra utiliser des parenthèses pour effectuer des calculs plus complexes.

- L'utilisateur devra obligatoirement fermer une parenthèse ouverte

- Les calculs dans les parenthèses seront effectués en premier

- Les lignes de commandes pourront avoir les formes suivantes avec les parenthèses :

(5 + 4) \* (3 + 1)

(5 \* (4 + 2))

(5 + 1)

- On autorisera également les calculs avec plusieurs opérateurs sans parenthèses (ex : 1+5\*4/2 )

- On autorisera autant de niveaux de parenthèses que l'on souhaite (mais la commande ne doit pas excéder 75 caractères).

##### 3) Calculs utilisant la mémoire

La calculatrice disposera de 26 mémoires nommées par les lettres majuscules de l'alphabet, de A à Z. L'utilisateur pourra utiliser ces mémoires dans ses calculs.

- Initialisation d'une mémoire à une valeur : syntaxe : 3 + 5 = A

- Le calcul sera tout de même effectué et le résultat affiché

- La mémoire pourra être utilisé dans les calculs comme un nombre "normal".

- Les zones mémoires pourront être consultées et éditées en mode gestionnaire de mémoire (voir plus bas).

Ex : 9 + 4 = A

= 13

A+2\*(1+3)

= 21

##### 4) Erreurs dans les calculs

- Si l'utilisateur n'entre pas un réel ou une lettre majuscule en tant qu'opérandes, on renverra une erreur. (ex : 1 + sd

- Si l'opérande est erronée ( pas +-\*/) ou que l'utilisateur en entre plusieurs à la suite (sans opérandes : ex : 1+\*1), on renverra une erreur.

- Si l'utilisateur entre un réel avec une virgule au lieu d'un point, on renverra une erreur.

- S'il utilise un point sans chiffres avant ou après (ex : 1. / .1), on renverra une erreur.

- Si l'utilisateur commence son calcul par un opérateur (ou le termine) ex : \* 2 ; 2 /

- Si l'utilisateur utilise dans un calcul une variable non initialisée

- Si l'utilisateur entre une variable à affecter qui n'est pas une lettre majuscule

- S'il n'utilise pas le signe "=" pour affecter un résultat de calcul à une variable

- S'il ne referme pas une parenthèse

- S'il entoure une affectation de parenthèse : ex : (1+2=A)

- S'il fait toute autre faute de syntaxe de calcul

- S'il entre une fonction du gestionnaire de mémoire sans être en mode MEM

#### c) Fonctionnement du champ d'exécution de commandes : mode mémoire

Lorsque l'utilisateur entrera l'expression "MEM" dans le champ d'exécution de commandes, la calculatrice passera en mode gestion de la mémoire.

Ce mode permettra de faciliter les opérations faisant intervenir la mémoire de la calculatrice.

- La commande "QUIT" permettra de revenir au mode "normal" de la calculatrice.

- L'utilisateur pourra utiliser la commande "AIDE" précédemment définie, ou le bouton "AIDE" pour consulter les différentes commandes disponibles.

- Les commandes permettant de gérer la mémoire sont composées de 3 parties :

--> Le nom de la commande

--> La zone de la mémoire concernée (cette partie peut être vide)

--> Une valeur numérique (pour certaines commandes)

Ex : MUL A 5

- La zone de mémoire concernée peut être un intervalle (plusieurs mémoires éditées en même temps. Pour cela, l'utilisateur devra préciser la première et la dernière zone mémoire concernée, et entrer ".." entre celles-ci (ex : A..F)

- Comme pour le mode normal, l'utilisateur pourra entrer autant d'espace qu'il le souhaite entre les composants de la commande, mais celle-ci ne pourra pas excéder 75 caractères.

Commandes disponibles :

- RAZ : remet à zéro les cases mémoires spécifiées ou toutes les cases mémoire et renvoie le message "OK" (ex : RAZ A..H)

- INCR : Ajoute 1 aux cases mémoires spécifiées, et renvoie le message "OK" (ex : INCR Y)

- SOM : effectue la somme des cases mémoires spécifiées, et renvoie le résultat (ex : SOM A..G)

- PROD : effectue le produit des cases mémoires spécifiées, et renvoie le résultat (ex : PROD A..B)

- MOY : effectue la moyenne des cases mémoires spécifiées, et renvoie le résultat (ex : MOY A..Z)

- SQRT : modifie la valeur des cases mémoires spécifiées pour leur affecter la racine carrée de leur valeur initiale, et renvoie "OK" (ex : SQRT A)

- CAR : modifie la valeur des cases mémoires spécifiées pour leur affecter le carré de leur valeur initiale, et renvoie "OK" (ex : SQRT G..K)

- INIT : Initialise la zone mémoire avec la valeur spécifiée, et renvoie "OK" (ex : INIT A 5)

- ADD : Ajoute la valeur spécifiée à la valeur des cases mémoires concernées et renvoie "OK" (ex : ADD G..L 42)

- MUL : Multiplie la valeur des cases mémoires concernées par la valeur spécifiée et renvoie "OK" (ex : MUL S 10)

- EXP : Elève la valeur de chacune des cases de la zone mémoire avec la puissance spécifiée par la valeur et renvoie "OK" (ex : EXP V 4)

#### d) Erreurs

- En mode MEM, s'il entre un calcul

- En mode MEM, s'il entre une commande non disponible

- S'il entre les plages de variables dans le désordre (ex : D..A au lieu de A..D)

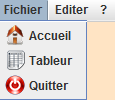
- Si, pour certains cas (ex : CAR) il utilise une ou plusieurs variables non initialisées

- Si les cases mémoires spécifiées ne sont pas des majuscules

- Si les plages ne sont pas séparées par ".." (ex : A,D ; A;D)

#### d) Menu

**Menu Fichier :**

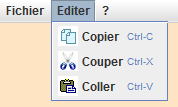


Accueil : permet de retourner au menu après avoir demandé de sauvegarder le tableur dans un fichier.

Tableur : permet de se déplacer directement vers le tableur.

Quitter : permet après confirmation par l’utilisateur de quitter complètement l’application.

**Menu Editer :**

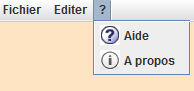


Copier : Permet de copier le contenu d’une cellule dans le presse papier.

Couper : Permet de couper le contenu d’une cellule dans le presse papier.

Coller : Permet de coller le contenu du presse papier dans une cellule.

**Menu Autre :**



A propos : Donne des informations sur les créateurs de l’application.

Aide : Affiche une nouvelle fenêtre contenant une aide approfondie pour aider l’utilisateur.

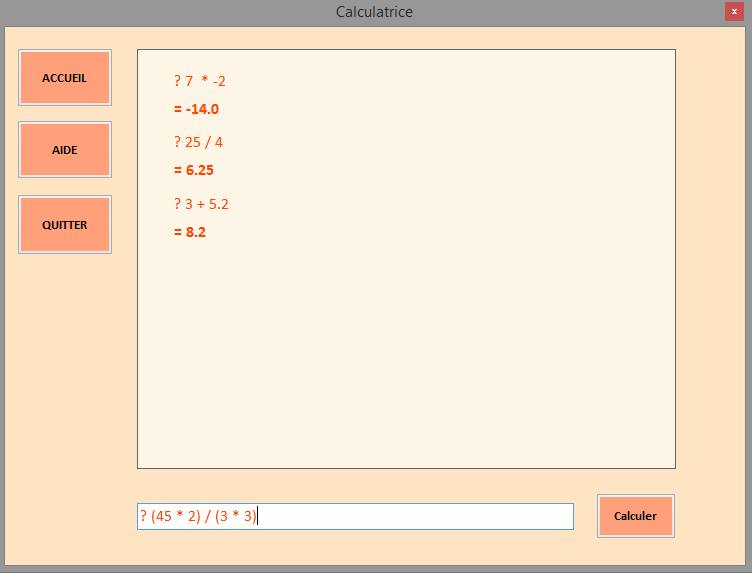
Menu pop-up : (s’affiche lors du clic droit dans le tableur)

Copier : Permet de copier le contenu d’une cellule dans le presse papier.

Couper : Permet de couper le contenu d’une cellule dans le presse papier.

Coller : Permet de coller le contenu du presse papier dans une cellule.

#### e) Première maquette de l'interface de la calculatrice



### II – Mini-Tableur

#### a) Interface Graphique – fonctionnement général

L’utilisateur peut entrer des commandes dans la ligne de commande, ou directement sur une cellule, il devra donc entrer le nom de la case cible uniquement dans ligne de commande.

Il existe 3 principaux types d’instructions : Le simple affichage ou le texte entré sera simplement stocké dans la cellule destination, le calcul disponible en ajoutant le symbole ‘=’ et le dernier type d’instruction est l’entrée d’une commande.

La touche ‘Entrée’ permet de réaliser la même action que le bouton ‘Valider’ disponible.

Les deux options précédentes demanderont une confirmation, pour éviter les "fausses manipulations" des utilisateurs.

Un menu contiendra l’option "AIDE" permettra à l'utilisateur de mieux comprendre le fonctionnement de l'application. Il recensera toutes les commandes possibles et ce qu'elles font. On pourra également accéder à cette fonction en tapant la commande "AIDE".

#### b) Les types d’entrées possibles

##### 1) Via ligne de commande

L’utilisateur doit rajouter la (les) case(s) destination dans les commandes d’affichage et de calcul.

L’affichage simple permet d’afficher sans modifications le texte passé en paramètre.

Le calcul est réalisé en ajoutant un ‘=’ après la cellule cible, si une erreur se produit durant le calcul un simple affichage du calcul sera réalisé.

Les commandes ne nécessitent pas d’indiquer une cellule cible mais elles demandent des paramètres variables.

Commandes disponibles :

COPIER de type cellule/cellule cellule/plage plage/plage

RAZ cellule : supprime le contenu d’une cellule

RAZ plage : supprime le contenu d’une plage

AIDE : affiche l’aide

##### 2) Directement depuis une case

L’affichage simple ne nécessitera pas d’indiquer la cellule source.

Le calcul est réalisé en ajoutant le signe ‘=’.

#### c) Erreurs

En cas de tentative d’affichage d’une cellule inexistante.

En cas de commande inconnue ou avec des erreurs dans les paramètres.

En cas d’erreur de syntaxe sur un calcul (voir erreurs de la calculatrice partie calcul, les variables étant ici des noms de cellules

Si on utilise le contenu d'une case qui n'est pas une opérande valide pour un calcul

Si le nombre d'arguments est incorrect (ou le type) ex : COPVAL A1 A2 A3

Si on entre une plage dans le mauvais ordre (ex : D4 A1)

#### d) Menu

**Menu Fichier :**

Nouveau : permet d’ouvrir un nouveau tableur après avoir demandé de sauvegarder l’ancien dans un fichier.

Charger : permet d’ouvrir un fichier contenant un tableur après avoir demandé de sauvegarder l’ancien dans un fichier à l’endroit voulu.

Sauver : permet de sauvegarder le tableur dans un fichier et un endroit choisi par l’utilisateur.

Accueil : permet de retourner au menu après avoir demandé de sauvegarder le tableur dans un fichier.

Calculatrice : permet de se déplacer directement vers la calculatrice simple après avoir demandé de sauvegarder le tableur dans un fichier.

**Menu Editer :**

Copier : Permet de copier le contenu d’une cellule dans le presse papier.

Couper : Permet de couper le contenu d’une cellule dans le presse papier.

Coller : Permet de coller le contenu du presse papier dans une cellule.

**Menu Autre :**

A propos : Donne des informations sur les créateurs de l’application.

Aide : Affiche une nouvelle fenêtre contenant une aide approfondie pour aider l’utilisateur.

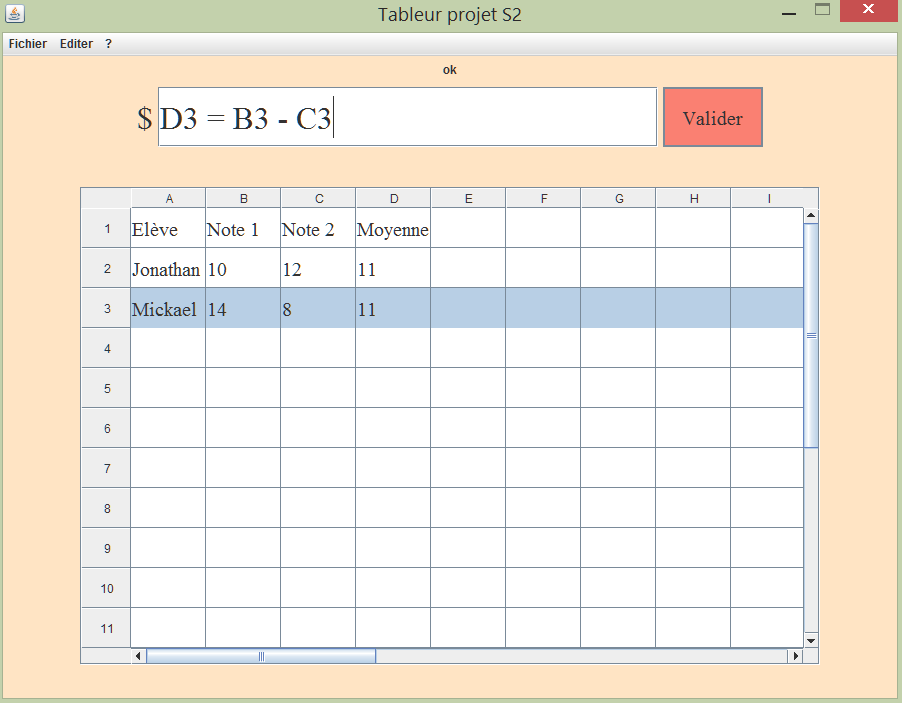
Menu pop-up : (s’affiche lors du clic droit dans le tableur)

Copier : Permet de copier le contenu d’une cellule dans le presse papier.

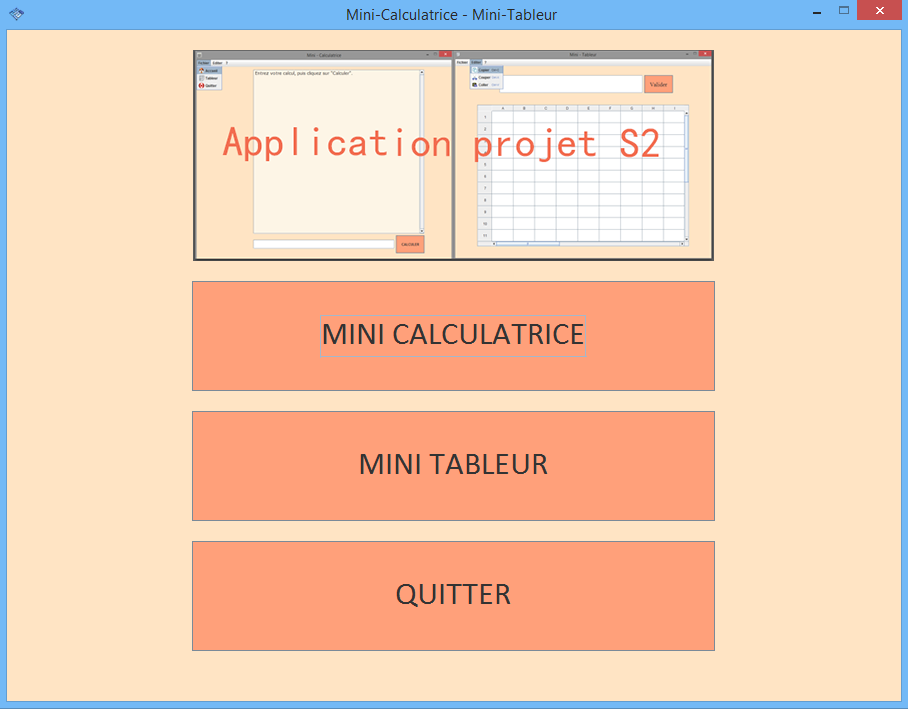
Couper : Permet de couper le contenu d’une cellule dans le presse papier.

Coller : Permet de coller le contenu du presse papier dans une cellule.

#### e) Première maquette de l’interface graphique



### III - Menu



Lors de l’exécution de l’application, une fenêtre s’ouvre. Cette fenêtre est le menu de l’application. Ce menu nous propose trois actions :

MINI-CALCULATRICE : Permet d’accéder à la partie calculatrice de l’application

MINI-TABLEUR : Permet d’accéder directement au tableur sans avoir à passer par la commande TAB que nous aborderons par la suite

QUITTER : Permet de quitter l’application. Pour quitter cette application, deux offres possibilités s’offre à l’utilisateur : soit il passe par la croix rouge et un message de confirmation s’affiche alors, soit il est sur la fenêtre calculatrice et il choisit de fermer l’application en appuyant sur la touche entrée après avoir saisi une chaîne vide. La calculatrice lui demandera alors de confirmer son choix.

# C - Conception

# D - Codage

# E - Tests

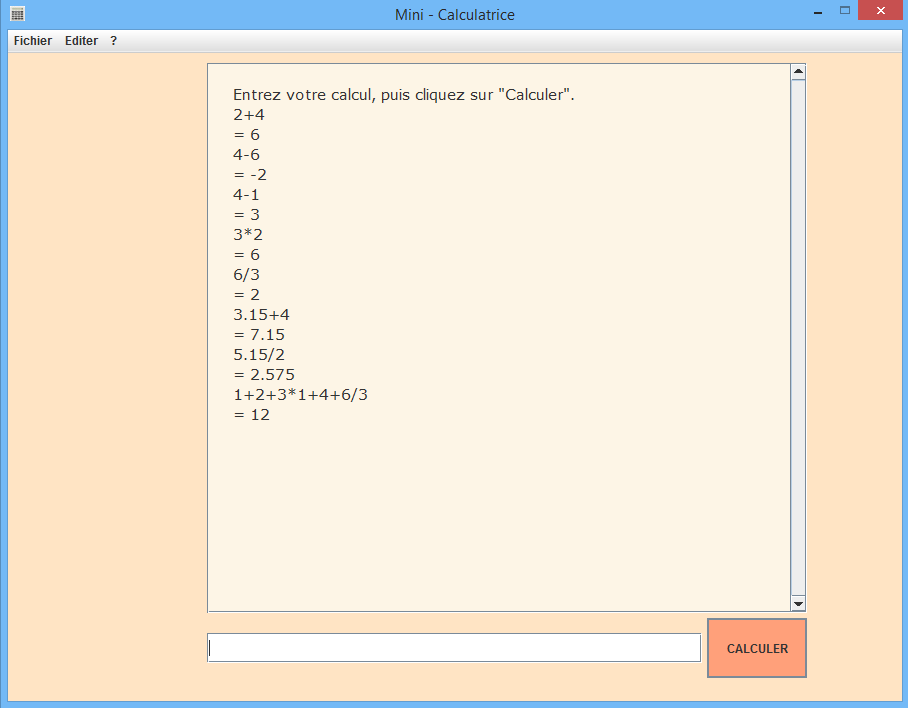
## 1) Programmes de test

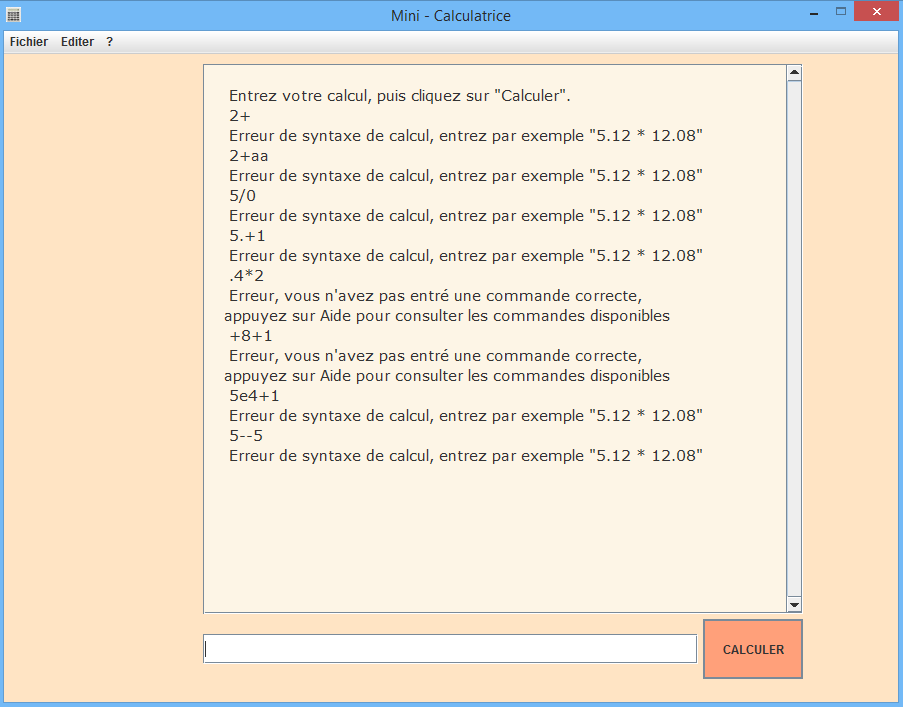
## 2) Résultats des programmes de test

## 3) Scénarios de test

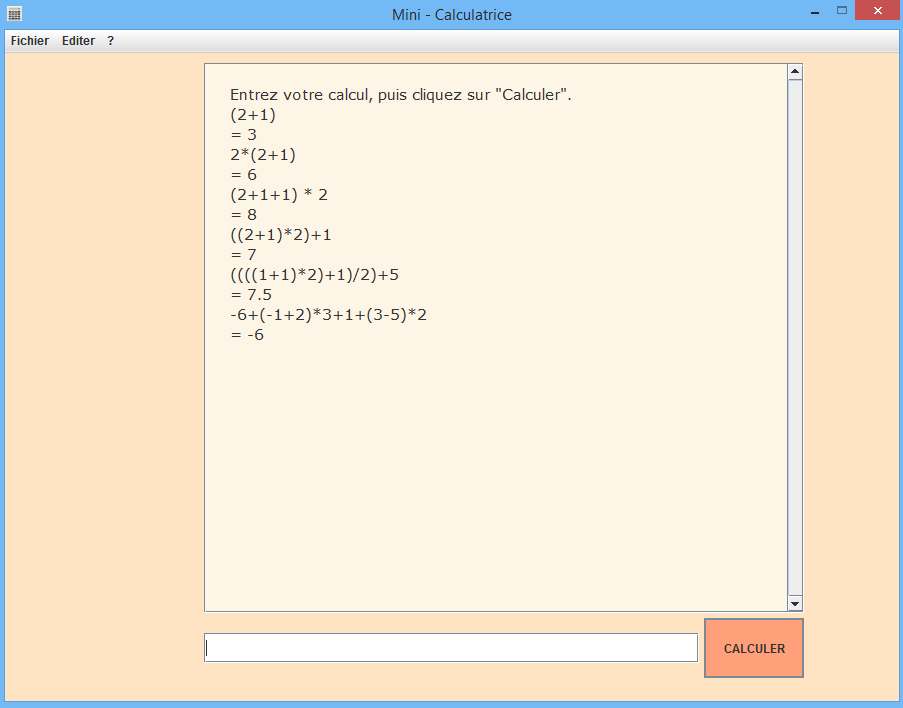
### I. Partie Calculatrice

**- Calculs simples, avec 1 ou plusieurs opérandes**

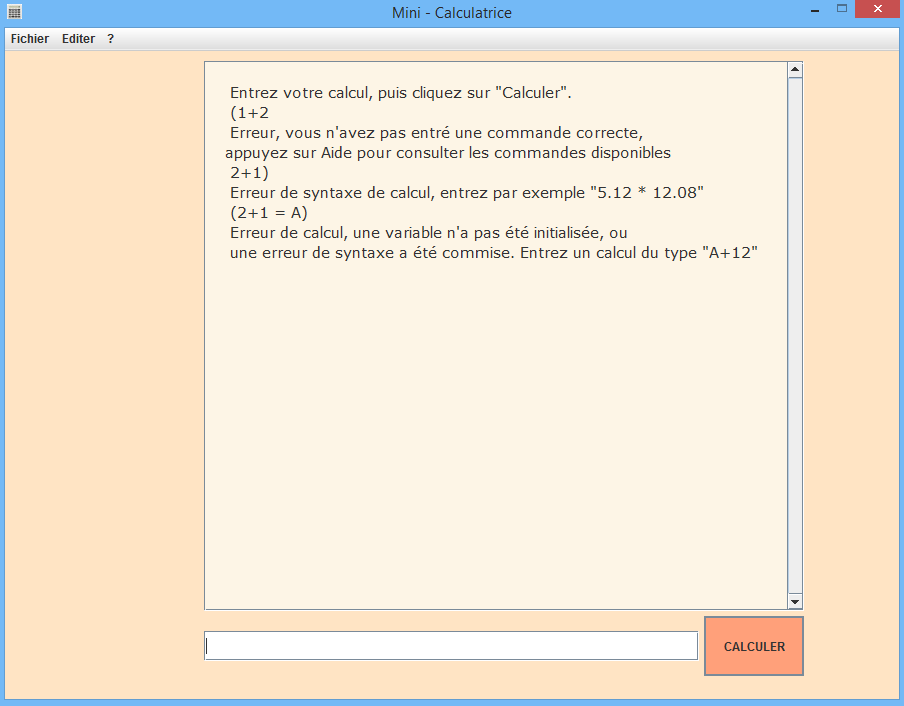
****

**Les Cas d’erreurs :  
  
**

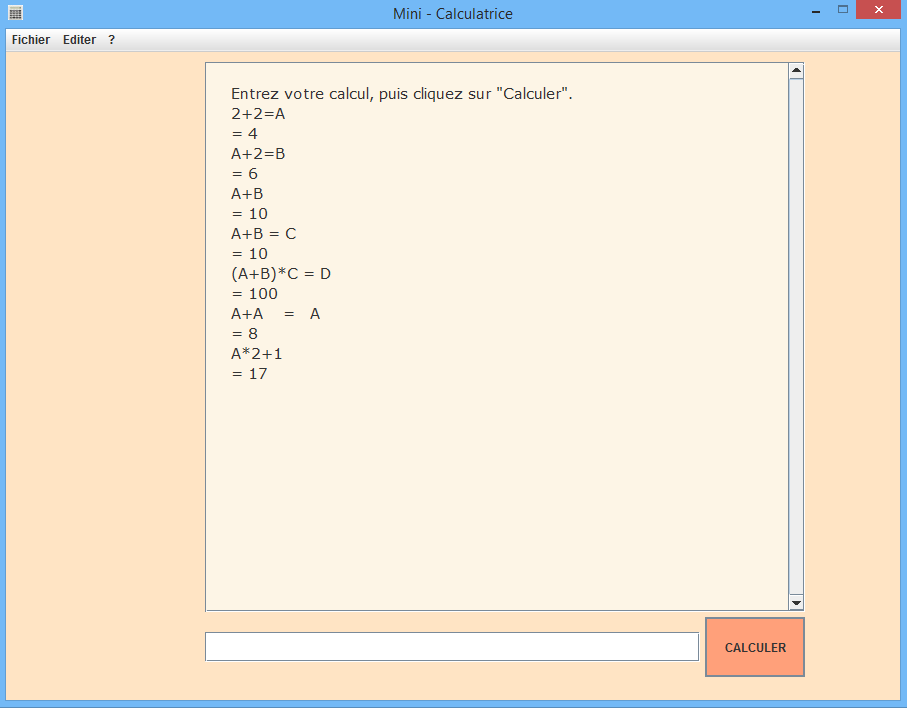
**- Calculs avec parenthèses**

****

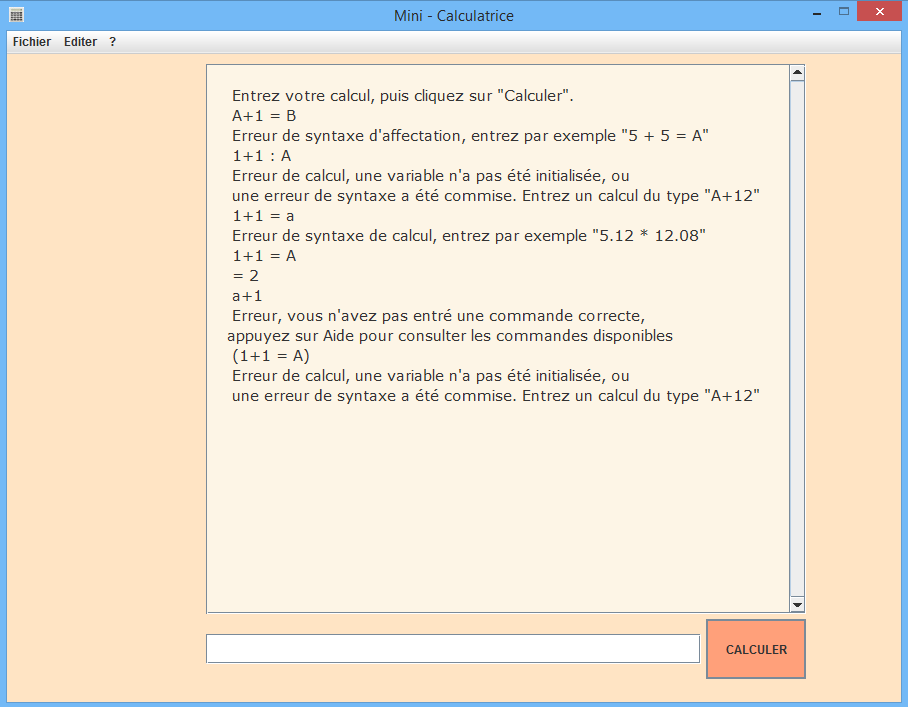
**- Erreurs :**

****

**Avec la mémoire :**

****

**- Erreurs :**

****

**- Mode MEM :**MEM

--> Mode mémoire actif.

QUIT

-->Mode mémoire inactif.

RAZ A

-->OK

RAZ A..D

-->OK

RAZ D..A

-->Erreur, entrez une commande du type RAZ A..B

RAZ g

-->Erreur, entrez une commande du type RAZ A..B

INIT A 5

-->OK

INIT B..D 4

-->OK

INIT A

-->Erreur, veuillez entrer une commande du type "INIT A..D 2"

INIT a 5

-->Erreur, veuillez entrer une commande du type "INIT A..D 2"

INIT D..A 5

-->Erreur, veuillez entrer les variables dans l'ordre alphabétique : ex : "INIT A..G 10.17"

VOIR A

-->A = 4.0

VOIR B..D

-->B = 4.0

C = 4.0

D = 4.0

VOIR H

-->Variable H non déclarée

VOIR g

-->Erreur, entrez une commande du type "VOIR A..G"

VOIR D..A

-->Erreur, entrez une commande du type "VOIR A..G"

VOIR A-D

-->Erreur, entrez une commande du type "VOIR A..G"

INCR A

-->OK

INCR A..C

-->OK

INCR T

-->Erreur, cette variable ne peut pas être incrémentée car elle

n'est pas initialisée

INCR a

-->Erreur, entrez une commande du type "INCR A..G"

INCR C..A

-->Erreur, entrez les cases mémoires dans l'ordre alphabétique : ex : "INCR A..G"

INCR A..G

-->Attention, des variables n'ont pas pu être incrémentées car elles

n'étaient pas initialisées

AIDE

--> Ouvre la fenêtre d'aide

CAR A

--> OK

CAR A..C

--> OK

CAR T

--> Erreur, cette variable n'est pas initialisée

CAR A..R

--> Attention, des variables n'ont pas pu être traitées car elles

n'étaient pas initialisées

CAR D..A

--> Erreur, entrez les cases mémoires dans l'ordre alphabétique : ex : "CAR A..G"

CAR g

--> Erreur, entrez une commande du type : "CAR A..G"

SQRT A

--> OK

SQRT A..C

--> OK

SQRT T

--> Erreur, cette commande ne peut pas être traitée car la variable n'a pas été initialisée

SQRT A..R

--> Attention, des variables n'ont pas pu être traitées car elles

n'étaient pas initialisées

SQRT D..A

--> Erreur, veuillez entrer les variables dans l'ordre alphabétique : ex : "SQRT A..G"

SQRT g

--> Erreur, veuillez entrer une commande du type : "SQRT A..G"

SOM A..C

= 15

SOM A

--> Erreur de syntaxe de calcul, entrez par exemple "SOM A..C"

(il se peut que des variables n'aient pas été initialisées)

SOM C..A

--> Erreur de syntaxe de calcul, entrez par exemple "SOM A..C"

(il se peut que des variables n'aient pas été initialisées)

SOM A..H

--> Erreur de syntaxe de calcul, entrez par exemple "SOM A..C"

(il se peut que des variables n'aient pas été initialisées)

SOM a..c

--> Erreur de syntaxe de calcul, entrez par exemple "SOM A..C"

(il se peut que des variables n'aient pas été initialisées)

MOY A..C

= 5

MOY A

--> Erreur de syntaxe de calcul, entrez par exemple "MOY A..C"

(il se peut que des variables n'aient pas été initialisées)

MOY C..A

--> Erreur de syntaxe de calcul, entrez par exemple "MOY A..C"

(il se peut que des variables n'aient pas été initialisées)

MOY A..T

--> Erreur de syntaxe de calcul, entrez par exemple "MOY A..C"

(il se peut que des variables n'aient pas été initialisées)

MOY A..c

--> Erreur de syntaxe de calcul, entrez par exemple "MOY A..C"

(il se peut que des variables n'aient pas été initialisées)

PROD A..C

= 125

PROD A

--> Erreur de syntaxe de calcul, entrez par exemple "PROD A..C"

(il se peut que des variables n'aient pas été initialisées)

PROD C..A

--> Erreur de syntaxe de calcul, entrez par exemple "PROD A..C"

(il se peut que des variables n'aient pas été initialisées)

PROD A..T

--> Erreur de syntaxe de calcul, entrez par exemple "PROD A..C"

(il se peut que des variables n'aient pas été initialisées)

PROD A..c

--> Erreur de syntaxe de calcul, entrez par exemple "PROD A..C"

(il se peut que des variables n'aient pas été initialisées)

ADD A 4

--> OK

ADD A..C 4

--> OK

ADD A

--> Erreur, veuillez entrer une commande du type "ADD A..D 4"

ADD a 4

--> Erreur, veuillez entrer une commande du type "ADD A..D 4"

ADD D..A 4

--> Erreur, veuillez entrer les variables dans l'ordre alphabétique : ex : "ADD A..G 5"

MUL A 2

--> OK

MUL A..C 2

--> OK

MUL A

--> Erreur, veuillez entrer une commande du type "MUL A..D 5"

MUL a 4

--> Erreur, veuillez entrer une commande du type "MUL A..D 5"

MUL D..A 2

--> Erreur, veuillez entrer les variables dans l'ordre alphabétique : ex : "MUL A..G 4"

EXP A 2

--> OK

EXP A..C 2

--> OK

EXP A

--> Erreur, veuillez entrer une commande du type "EXP A..D 5"

EXP a 2

--> Erreur, veuillez entrer une commande du type "EXP A..D 5"

EXP D..A 2

--> Erreur, veuillez entrer les variables dans l'ordre alphabétique : ex : "EXP A..G 5"

# F- Bilan

## 1) Bilan commun

🡪 Satisfaction des besoins

Au terme du projet, la totalité du cahier des charges, définit par la MOA, a été traitée et respectée. Malgré cela, quelques écarts ont été réalisés. En effet, lors des réunions avec la MOE, il a été convenu de modifier ce cahier des charges afin de privilégier le développement de l’application en interface graphique plutôt que sur la console. De plus, nous avons choisi de permettre à l’utilisateur de réaliser des opérations avec un nombre illimité de niveaux d’imbrication de parenthèses. Il en est de même pour le nombre d’opérateurs et d’opérandes. Par ailleurs, d’autres fonctionnalités ont été développées pour offrir à l’utilisateur plus de possibilités. Ainsi, le tableur propose de charger des fichiers ainsi que d’en sauvegarder. Le clic droit est également disponible (sur la calculatrice aussi) afin de permettre des actions d’édition telles que « copier », « couper », « coller ».

Au cours des différents comités de pilotage, le représentant de la MOE a semblé satisfait des avancées réalisées, des prototypes présentés ainsi que des choix entrepris par le groupe.

Le groupe de projet est donc satisfait de ce qu’il a pu fournir au cours de la réalisation de cette application.

🡪 Bilan de la gestion de projet

**La planification prévisionnelle** n’était finalement pas très éloignée de l’avancement effectif du projet. Les tâches les plus compliquées ont été l’apprentissage de l’utilisation des interfaces graphiques étant donné qu’aucun membre du groupe n’en avait auparavant réalisé dans ce langage, ainsi que les calculs comportant des parenthèses qui nous ont posé beaucoup de soucis.

Malgré cela, le respect des précédences n'a pas toujours été vérifié. En effet la partie concernant la Conception Orientée Objet de l’application, notamment les spécifications et les cas d’utilisation, n’a pas pu être réalisée au début, étant donné que nous n’avions pas encore abordé tous les éléments nécessaires en cours. De ce fait, la réalisation de cette partie a été retardée sans toutefois empêcher l’avancement des autres tâches.

Par ailleurs, **la planification du projet** ne montre pas toutes les retouches et détails apportés aux tâches déjà réalisées. Ces modifications ont souvent découlé de remarques évoquées lors des réunions avec la MOE.

Concernant **la gestion de communication**, nous n’avons pas rencontré de problème pénalisant l’avancement du projet. Le fait d’utiliser des outils de communication, tel que la messagerie instantanée Facebook pour le partage d’idées et d’informations, et des outils de partage, tel que GitHub, a permis de maitriser l’organisation du travail et de gérer la répartition des tâches.

Nous avons tout de même rencontré quelques problèmes. Tout d’abord, nous n’avons pu organiser que très peu de réunions. Enfin, l’utilisation de git nous a donné quelques sueurs froides puisque le changement de JRE au cours du projet a entraîné des conflits dans les classes, nous obligeant à récupérer les fichiers réalisés la veille. Ce problème a donc été source d’une importante perte de temps de travail.

Au niveau de la gestion humaine, la répartition des tâches s’est faîte presque naturellement, en fonction des capacités de chaque membre du groupe. En effet, les écarts de niveau de programmation imposaient que les tâches les plus complexes soient réalisées par les membres les plus aptes.

Tout au long du projet, la cohésion et l’entente au sein du groupe ont été bonnes. Cette cohésion a été facilitée par le fait que nous nous entendions déjà bien.

Le projet a vraiment été bénéfique pour chacun des membres du groupe puisque tout le monde a énormément progressé en programmation, conception, gestion de projet, IHM, travail en groupe et communication. De plus, nous avons découverts et appris à maitriser de nouveaux outils informatiques.

Nous avons donc pu avoir un premier aperçu de la réalisation d’un projet informatique en entreprise. Cela nous a permis de mieux appréhender certains concepts qui n’étaient pas maitrisés avant la réalisation du projet.

## 2) Bilans personnels

### - Sébastien SANCHEZ

Ce projet a été une très bonne expérience. D'une part d'un point de vues connaissances, puisque j'ai beaucoup appris en matière de programmation évènementielle et de conception d'interface graphiques en Java. J'ai également appris beaucoup sur la gestion d'un projet. Etant chef de projet, il était très important pour moi que l'entente au sein de l'équipe soit optimale, et ce fut le cas. Le travail en groupe a également été abordé avec l'utilisation de Git, qui a été un outil formidable pour permettre à tout le monde de travailler simultanément sur un code, mais qui est très difficile d'utilisation au premier abord.

L'étude de Git et des bibliothèques SWING et AWT sera de plus très bénéfique dans le futur, puisque utiles en entreprise.

L'ambiance au sein du groupe a vraiment été superbe, et de nombreux fous rires ont été au rendez-vous au cours de nos séances de programmation intenses et groupées.