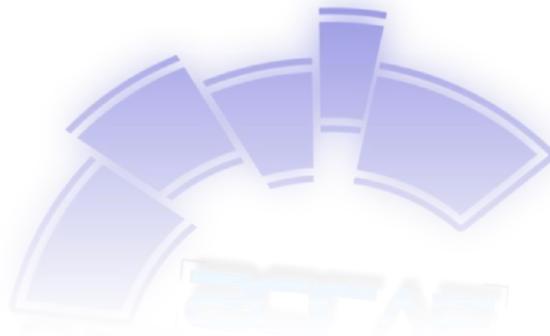


# BleuSolve

---

*Programme Java résolvant des équations en expliquant comment parvenir au résultat*



Travail de maturité sur le thème « Programmation en Java », proposé par Pierre Breguet  
Remis le 22 novembre 2011

Gymnase d'Yverdon, 2011

Tinh Di David Tang

# BleuSolve

---

*Programme Java résolvant des équations en expliquant comment parvenir au résultat*

## 1. Résumé

BleuSolve, un logiciel programmé en Java, permet de résoudre diverses sortes d'équations en expliquant à l'utilisateur comment parvenir au résultat souhaité et il contient aussi une application qui s'avère être très utile pour certain(e)s étudiant(e)s : le calculateur de moyenne.

BleuSolve permet de :

- Résoudre une équation du premier degré à une inconnue
- Résoudre une équation du deuxième degré à une inconnue
- Résoudre une équation du troisième degré à une inconnue (avec une solution entière comprise entre -10 et 10)
- Résoudre un système de deux équations du premier à deux inconnues
- Calculer les racines carrées d'un nombre complexe
- Résoudre une équation du premier degré à une inconnue dans l'ensemble des nombres complexes
- Effectuer une division polynomiale grâce à la méthode de Horner
- Calculer et prévoir des moyennes

Ce travail a un aspect très didactique, puisque l'objectif est d'aider des élèves qui auraient parfois de la peine à comprendre certains aspects des mathématiques. Ce programme, n'étant pas encore très utile au niveau de l'aide étant donné qu'il n'offre pas des résolutions très compliquées, permet néanmoins d'avoir déjà une petite idée de comment est-ce qu'on pourrait développer ce genre de logiciels d'aide pour les rendre plus performants et aussi si ce type d'aide est réellement utile.

Travail de maturité sur le thème « Programmation en Java », proposé par Pierre Breguet

Remis le 22 novembre 2011

Gymnase d'Yverdon, 2011

## 2. Table des matières

Page de garde.....	1
1. Résumé.....	2
2. Table des matières .....	3
3. Introduction.....	4
4. Manuel de l'utilisateur .....	4
4.1. Éléments importants à connaître .....	4
4.2. Démarrage .....	5
4.3. Écran d'attente .....	5
4.4. Les menus .....	6
4.5. La barre de menu.....	6
4.6. Les fenêtres de résolution .....	8
4.7. Les différents composants de BleuSolve .....	10
4.8. Quitter BleuSolve.....	22
5. Documentation de développement .....	23
5.1. Launcher.java.....	23
5.2. SplashScreen.java .....	23
5.3. Window.java .....	24
6. Conclusion .....	30
7. Annexes .....	31
7.1. Document de motivation et de présentation du sujet choisi .....	31
7.2. Fichiers PDF issus de la théorie .....	32
7.3. Code Java .....	40
8. Table des illustrations .....	108
9. Bibliographie .....	109
9.1. Crédit iconographique .....	109
9.2. Ouvrages consultés.....	109
9.3. Sites internet.....	109
10. Remerciements .....	110
11. Liens.....	110

### 3. Introduction

L'informatique occupe une place de plus en plus importante dans notre vie courante. Les appareils électroniques qui nous entourent nous permettent de vivre plus aisément notamment en effectuant les tâches répétitives dont on pourrait s'en passer. C'est là qu'intervient la programmation : elle permet d'effectuer infiniment les mêmes actions (programmation des robots-constructeurs) et/ou d'effectuer des calculs répétitifs beaucoup plus rapidement qu'un être humain.

C'est l'aspect répétitif des mathématiques qui m'a donné l'idée pour ce travail de maturité. Ne trouvez-vous pas que ce sont toujours les mêmes calculs, les mêmes opérations qu'on effectue pour résoudre un type particulier d'équation ? C'est grâce à cette répétition que des programmeurs ont réussi à réaliser des logiciels et des calculatrices qui permettent de résoudre des équations. Ces programmes fonctionnent à merveille, mais parfois on se demande comment ils sont arrivés au résultat. Ils nous permettent donc uniquement de vérifier nos résultats (à part si on a une confiance aveugle au logiciel), mais pourraient-ils nous expliquer par quelles étapes doit-on passer pour parvenir au résultat ? C'est à partir de cette optique-là que j'ai réalisé mon travail de maturité.

Ce travail a été programmé en Java, car c'est un langage de programmation qui est relativement intuitif et qui est compatible avec la plupart des machines.

## 4. Manuel de l'utilisateur

### 4.1. Éléments importants à connaître

Pour des raisons de programmation, à chaque fois que vous ouvrirez un PDF avec BleuSolve, une copie de ce dernier est automatiquement créée dans le même dossier que le logiciel. Mais vous pouvez supprimer tous les fichiers PDF en quelques clics dans le menu « Fichier » et lorsque vous fermez le programme, tous les fichiers seront automatiquement supprimés.

Lorsque vous inscrivez un caractère invalide (lettres, caractères spéciaux, etc...) ou rien dans les champs de texte, le programme fera comme si vous avez inscrit « 0 » dans ce champ de texte.

Il se peut fortement que vous voyez deux signes côte-à-côte (+ -) dans le résultat proposé. Veuillez-vous référer au PDF « Règle des signes<sup>1</sup> » pour interpréter correctement ces doubles signes.

Vous rencontrerez aussi peut-être « · - », cela signifie qu'on multiplie par un nombre négatif. Par exemple, si vous voyez «  $4 \cdot -2$  », interprétez-le comme ceci «  $4 \cdot (-2)$  ».

Dans le résultat, vous verrez peut-être des nombres avec un « E » à l'intérieur, par exemple «  $1.291E8$  ». Il faut l'interpréter comme un «  $\cdot 10^x$  ». Avec notre exemple, le nombre indiqué vaut «  $1.291 \cdot 10^8 = 129'100'000$  ».

---

<sup>1</sup> Pour accéder au PDF « Règle des signes » : allez dans la barre de menu, menu « Théorie », puis « Règle des signes ».

Il se peut que, à cause des arrondis de décimales, vous voyez une réponse très proche d'un nombre arrondi du type «  $1.81291E - 12$  », ce qui veut dire «  $1.81291 \cdot 10^{-12}$  ». La véritable réponse est très probablement « 0 » (dans ce cas-là).

#### 4.2. Démarrage

Pour pouvoir utiliser BleuSolve, il vous faudra, si ce n'est pas déjà fait, installer préalablement « Java ». Vous pouvez télécharger la dernière version de Java à l'adresse suivante : <http://www.java.com/fr/download/>.

Pour lancer BleuSolve, double-cliquez sur le fichier « BleuSolve » (ou « BleuSolve.jar »).



Image 1 : Fichier "BleuSolve"

#### 4.3. Écran d'attente

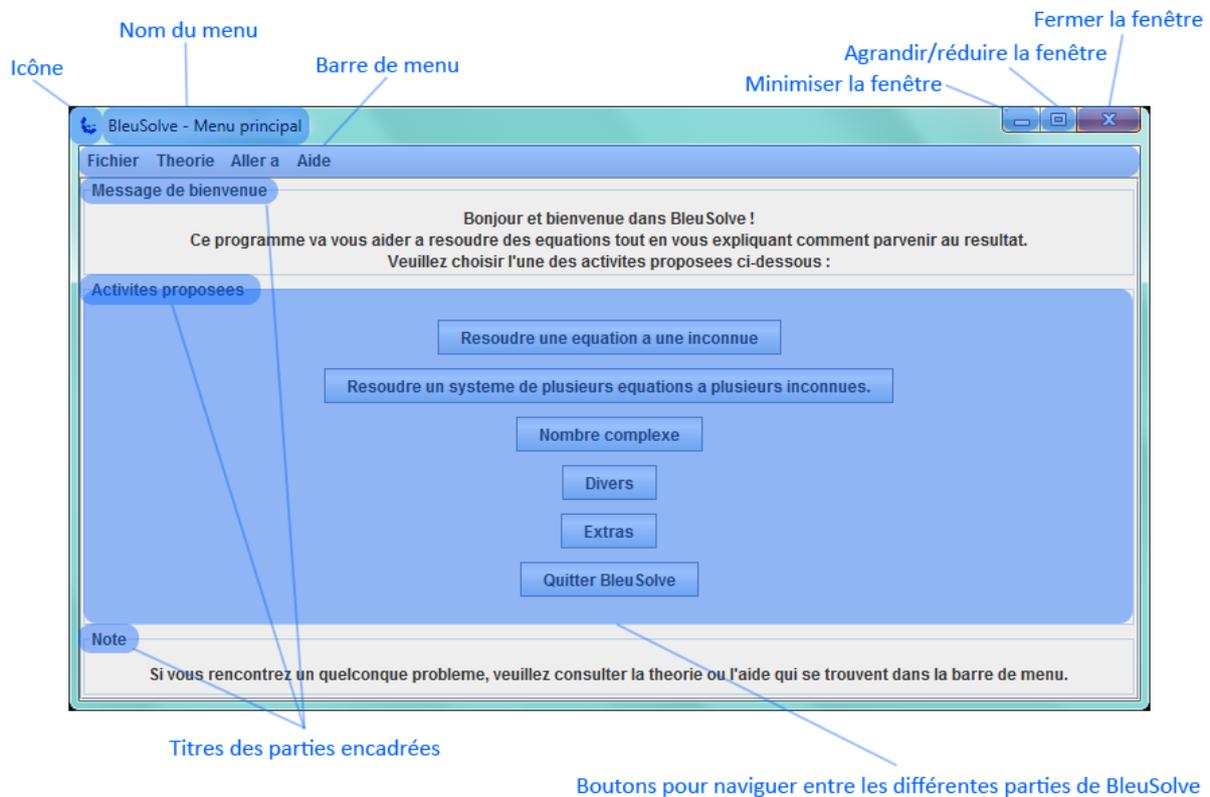
Quand vous lancez le programme, un écran d'attente s'affichera durant le chargement des différents composants du logiciel.



Image 2 : Image d'attente

#### 4.4. Les menus

Après l'écran d'attente, vous arriverez sur le menu principal<sup>2</sup> dont voici les différents composants (tous les menus sont calqués sur le même modèle) :



Exemple 1 : Explications des menus

#### 4.5. La barre de menu

Durant toute l'utilisation du logiciel, vous pouvez à tout moment avoir accès à la barre de menu qui se situe dans la partie supérieure de la fenêtre.

La barre de menu est composée de quatre menus :

- Le menu « Fichier »
- Le menu « Théorie »
- Le menu « Aller à »
- Le menu « Aide »



Image 3 : La barre de menu

##### 4.5.1. Le menu « Fichier »

Dans le menu « Fichier », vous pouvez :

- Effacer tous les fichiers PDF créés
- Retourner au menu principal

<sup>2</sup> À l'ouverture de BleuSolve, la fenêtre est en plein écran. Mais dans toutes les copies d'écran de ce document, les fenêtres ont été réduite (la taille de la fenêtre réduite par défaut est de 800 × 450 pixels).

- Quitter BleuSolve (*et effacer, par la même occasion, tous les fichiers PDF créés*)

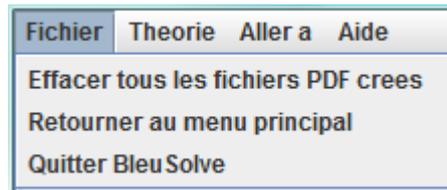


Image 4 : La barre de menu : Fichier

#### 4.5.2. Le menu « Théorie »

Dans le menu « Théorie », vous pouvez consulter de la théorie (Démonstrations, Règles, etc...) qui peut être utile pour comprendre certains résultats proposés par le logiciel.

La « Théorie » est actuellement composée de :

- La démonstration de la formule de résolution des équations du deuxième degré à une inconnue
- La formule de Cramer
- La méthode de Horner
- La règle des signes
- Quelques éléments théoriques sur les nombres complexes

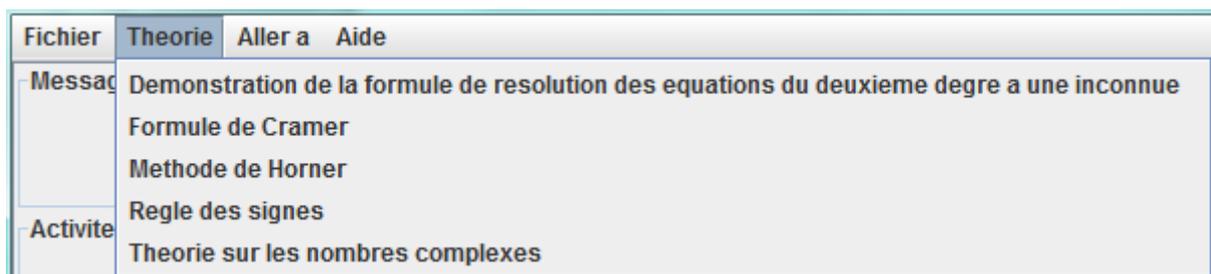


Image 5 : La barre de menu : Théorie

#### 4.5.3. Le menu « Aller à »

Vous pouvez accéder rapidement à toutes les fonctionnalités de BleuSolve dans le menu « Aller à ».

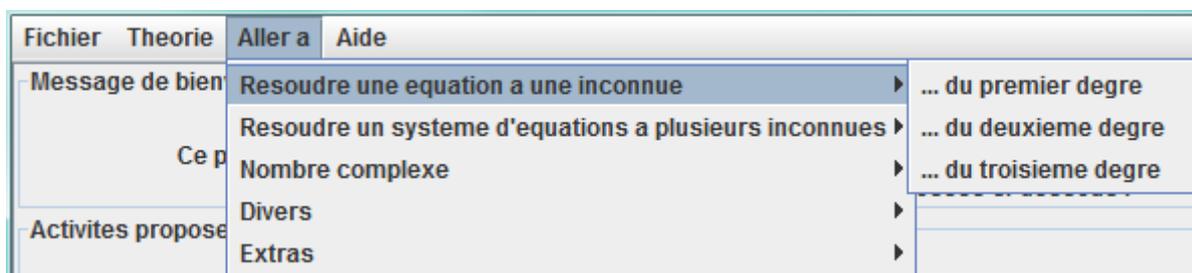


Image 6 : La barre de menu : Aller à

#### 4.5.4. Le menu « Aide »

Si vous rencontrez des problèmes lors de l'utilisation de BleuSolve, vous êtes invité(e) à consulter le menu « Aide » qui est composé de :

- Manuel de l'utilisateur (*que vous êtes en train de consulter*)

- Contacter le développeur (*pour envoyer un e-mail au développeur si vous avez des questions, des suggestions, des remarques ou autres...*)
- Blog de BleuSolve (*vous renvoie directement au blog de BleuSolve pour suivre l'actualité de ce dernier*)
- Page Facebook de BleuSolve (*vous renvoie directement à la page Facebook de BleuSolve pour suivre l'actualité de ce dernier*)
- À propos de BleuSolve (*si vous voulez en savoir plus sur le logiciel que vous utilisez*)

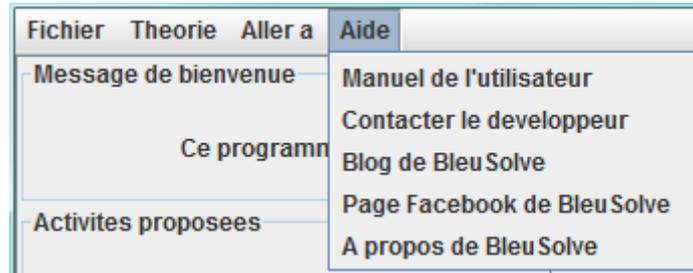
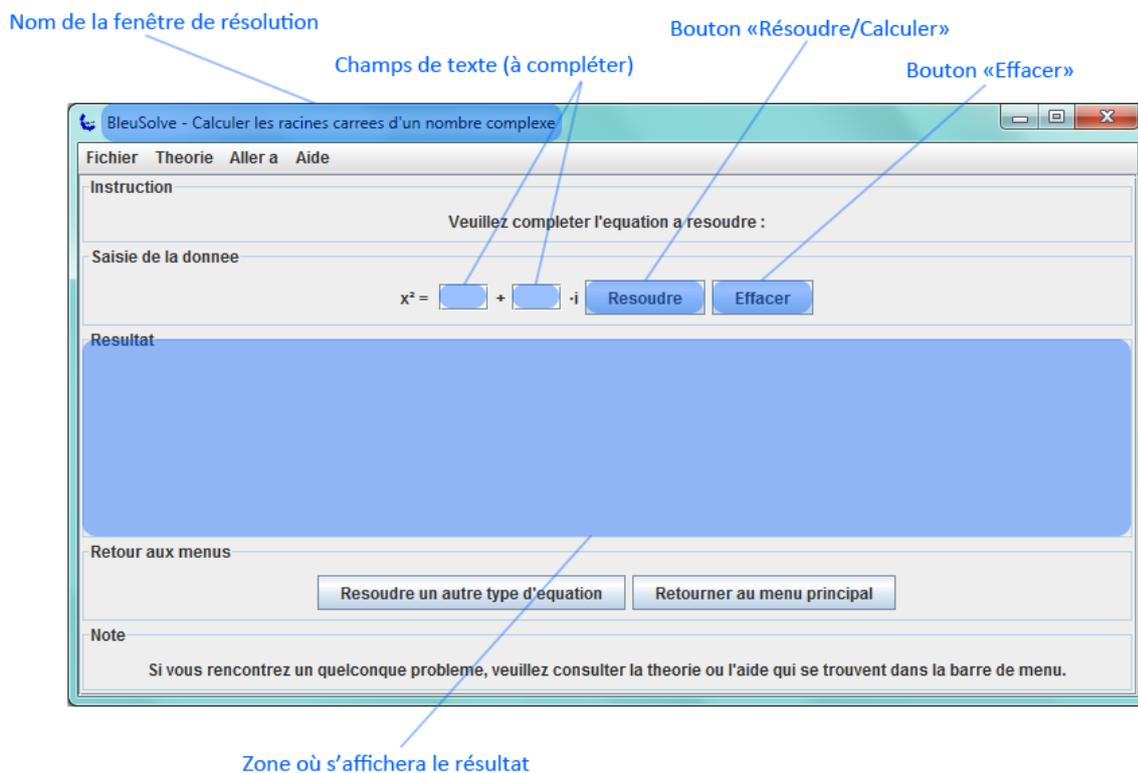


Image 7 : La barre de menu : Aide

#### 4.6. Les fenêtres de résolution

Toutes les fenêtres de résolution de ce logiciel sont calquées sur le même modèle, voici un exemple de l'une d'entre elles :



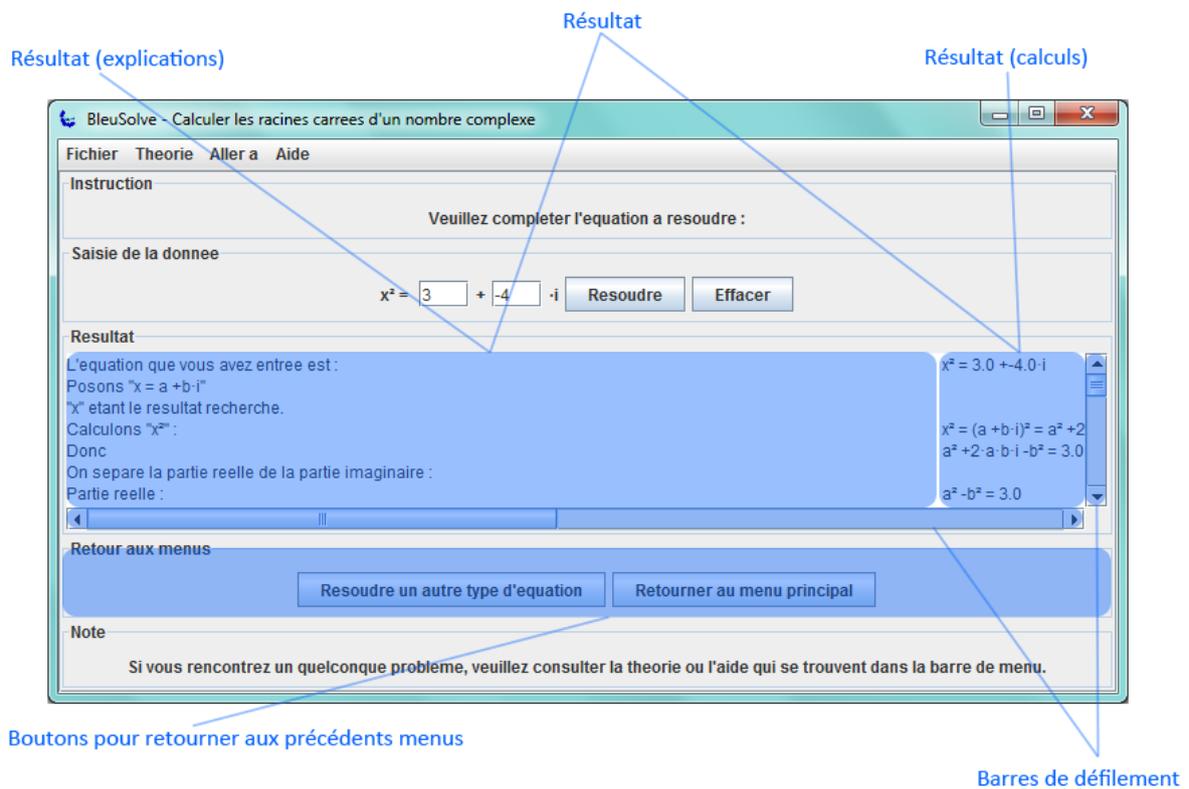
Exemple 2 : Explications des fenêtres de résolution

Inscrivez des nombres dans les champs de texte, puis cliquez sur le bouton « Résoudre » ou le bouton « Calculer ». Le résultat s'affichera dans la zone prévue à cet effet.

Si vous souhaitez réintroduire une nouvelle équation, vous pouvez cliquer sur le bouton « Effacer » qui remettra tout à zéro.

Remarquez que vous pouvez retourner soit au menu de résolution des équations à une inconnue, soit directement au menu principal grâce aux deux boutons qui se situent dans la zone encadrée « Retour aux menus ».

Après avoir cliqué sur le bouton « Résoudre », voici ce que vous obtiendrez :



### Exemple 3 : Explications des résultats

Le résultat est, dans la plupart du temps, séparé en deux parties (les explications et les calculs) pour faciliter la lecture des résultats et aussi pour éviter que vous vous ennuyiez après avoir lu plusieurs fois les mêmes explications (vous pourrez ainsi seulement lire les calculs).

Notez encore que si le résultat prend plus de place que la zone prédéfinie, des barres de défilement apparaîtront automatiquement pour que vous puissiez toujours avoir accès à la totalité du résultat. Pour naviguer dans le résultat, utilisez la molette de votre souris ou déplacez les barres de défilement avec votre souris.

#### 4.7. Les différents composants de BleuSolve

Dans le menu principal, vous pourrez choisir l'une des activités proposés :

- Résoudre une équation à une inconnue
- Résoudre un système d'équations à plusieurs inconnues
- Nombres complexes (*où vous pourrez résoudre des équations dans l'ensemble des nombres complexes*)
- Divers (*contient diverses activités en rapport avec les résolutions d'équations*)
- Extras (*contient diverses activités autres que des résolutions d'équations*)
- Quitter BleuSolve (*et effacer, par la même occasion, tous les fichiers PDF créés*)

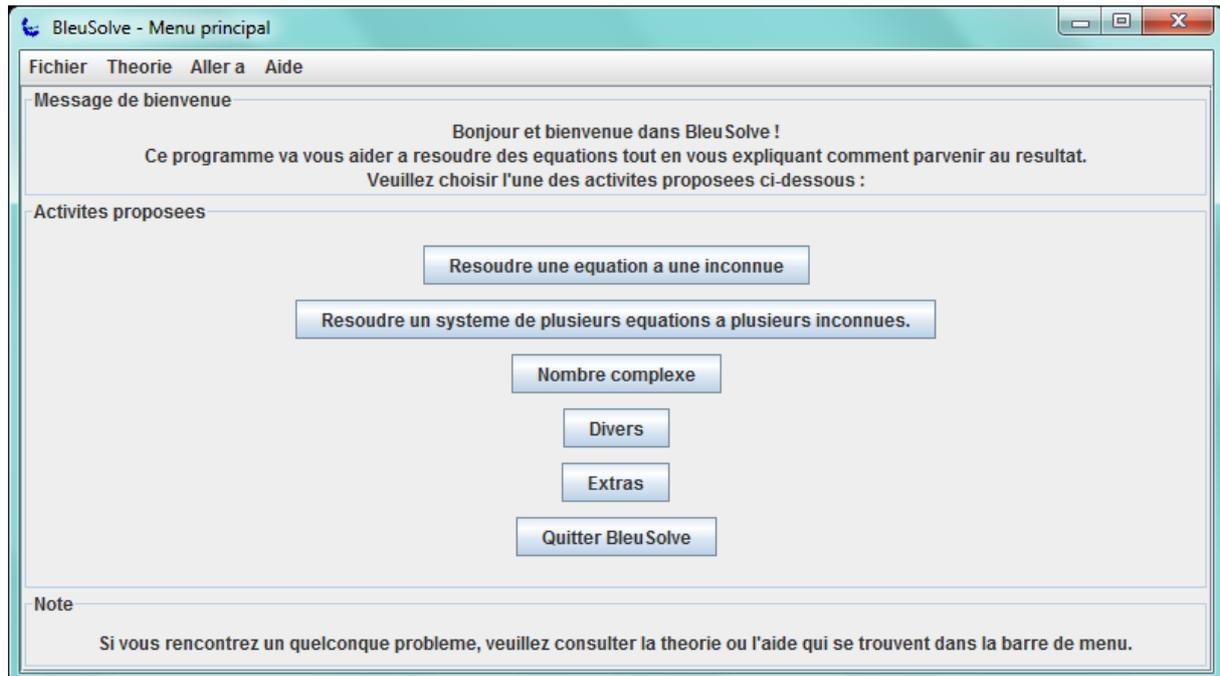


Image 8 : Menu principal

##### 4.7.1. Résoudre une équation à une inconnue

Dans ce menu, vous trouverez les activités suivantes :

- Résoudre une équation du premier degré à une inconnue
- Résoudre une équation du deuxième degré à une inconnue
- Résoudre une équation du troisième degré à une inconnue (avec une solution entière comprise entre -10 et 10)
- Retourner au menu principal
- Quitter BleuSolve (*et effacer, par la même occasion, tous les fichiers PDF créés*)

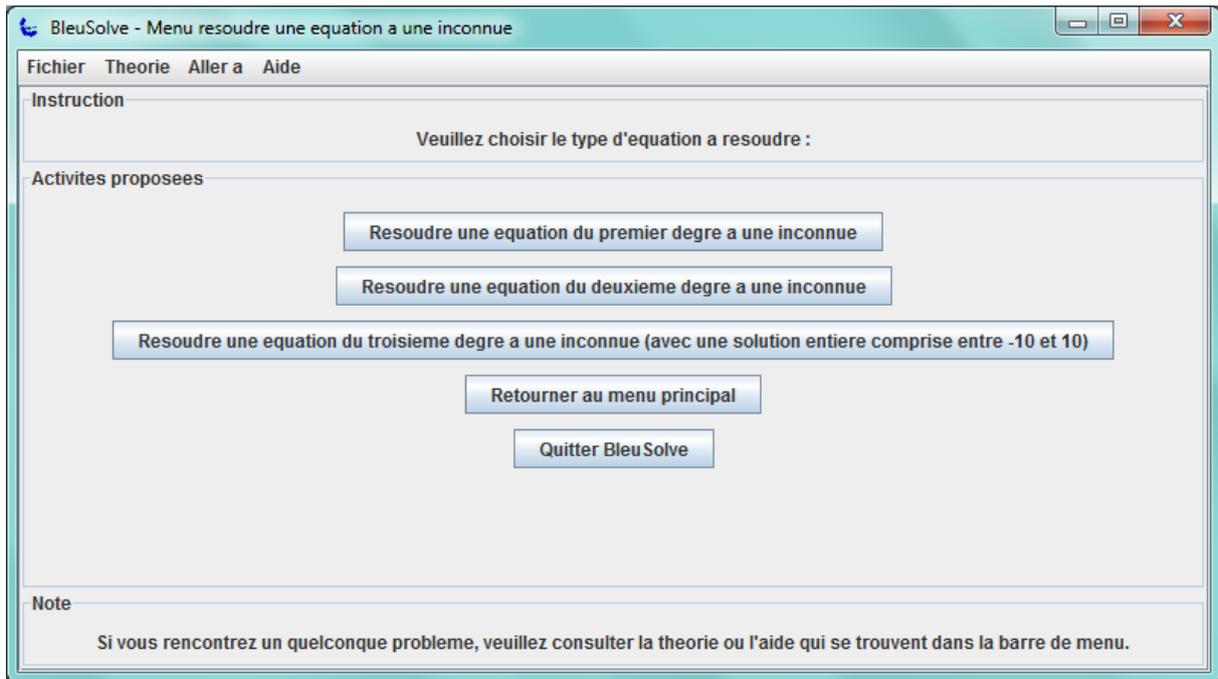


Image 9 : Menu résoudre une équation à une inconnue

#### 4.7.1.1. Résoudre une équation du premier degré à une inconnue

En arrivant dans la fenêtre de résolution, vous serez confronté à ceci :

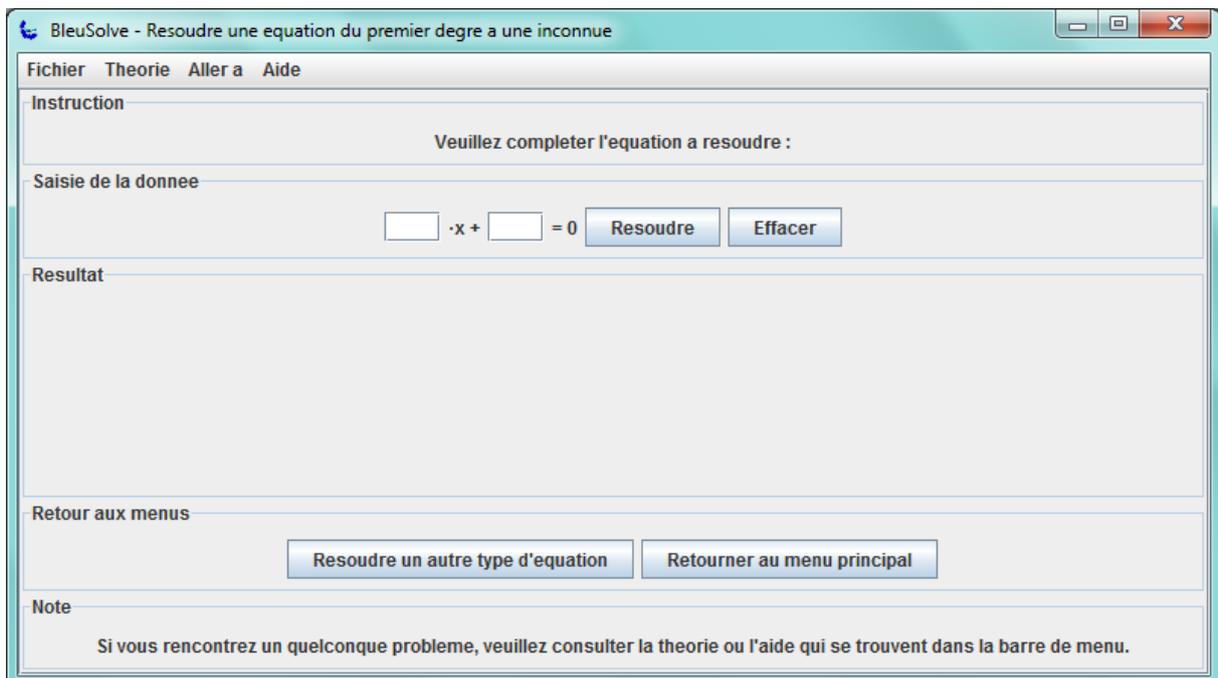


Image 10 : Résoudre une équation du premier degré à une inconnue

Si vous souhaitez, par exemple, résoudre «  $2 \cdot x - 4 = 0$  », inscrivez « 2 » dans le premier champ de texte et « -4 » dans le deuxième.

Après avoir cliqué sur le bouton « Résoudre », vous obtiendrez ceci :

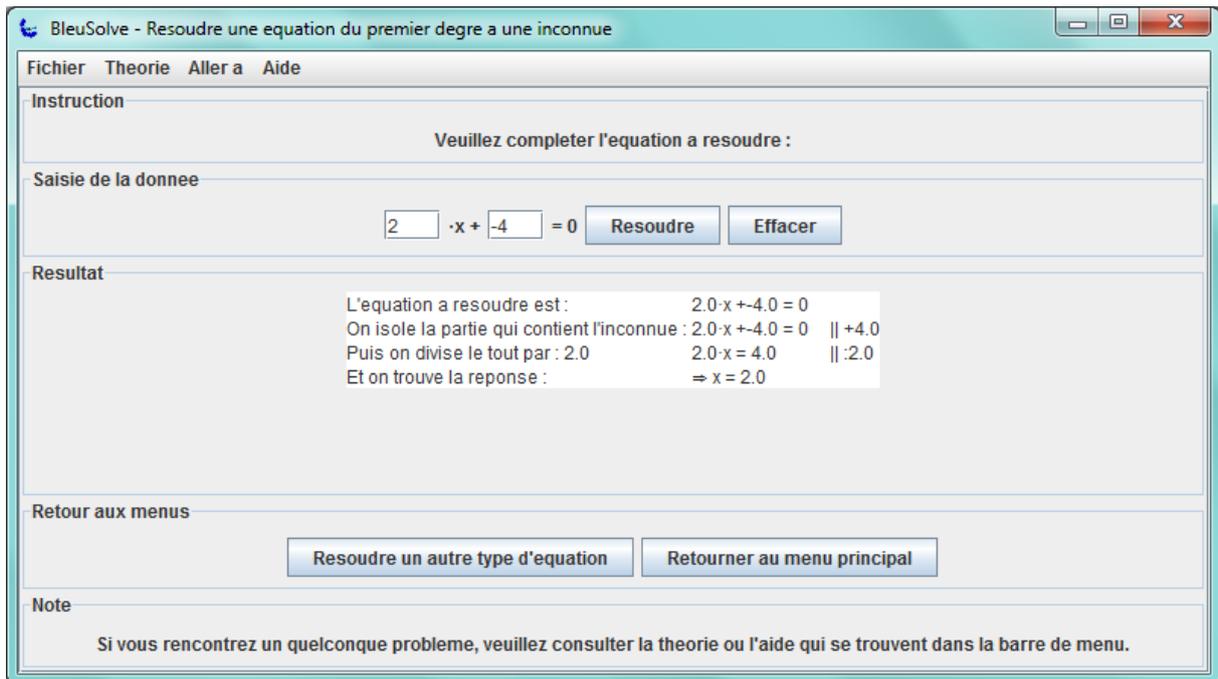


Image 11 : Résoudre une équation du premier degré à une inconnue - Résultat

#### 4.7.1.2. Résoudre une équation du deuxième degré à une inconnue

En arrivant dans la fenêtre de résolution, vous serez confronté à ceci :

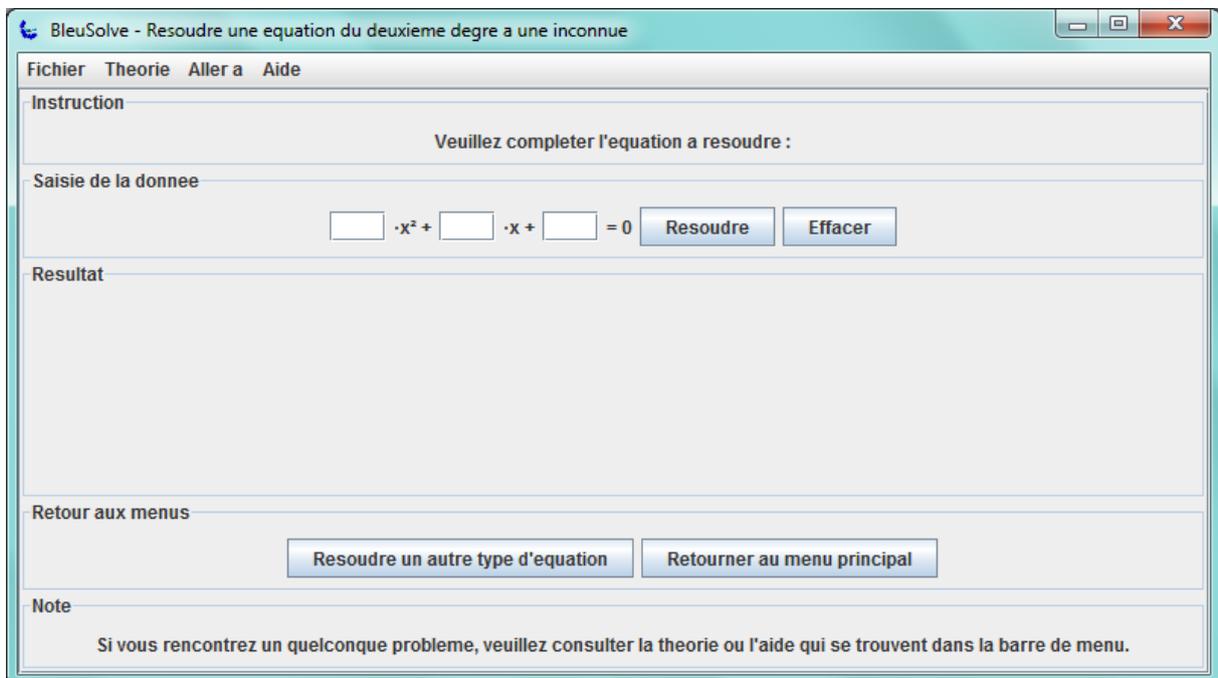


Image 12 : Résoudre une équation du deuxième degré à une inconnue

Si vous souhaitez, par exemple, résoudre «  $2 \cdot x^2 - 2 \cdot x - 12 = 0$  », inscrivez « 2 » dans le premier champ de texte, « -2 » dans le deuxième et « -12 » dans le troisième.

Après avoir cliqué sur le bouton « Résoudre », vous obtiendrez ceci :

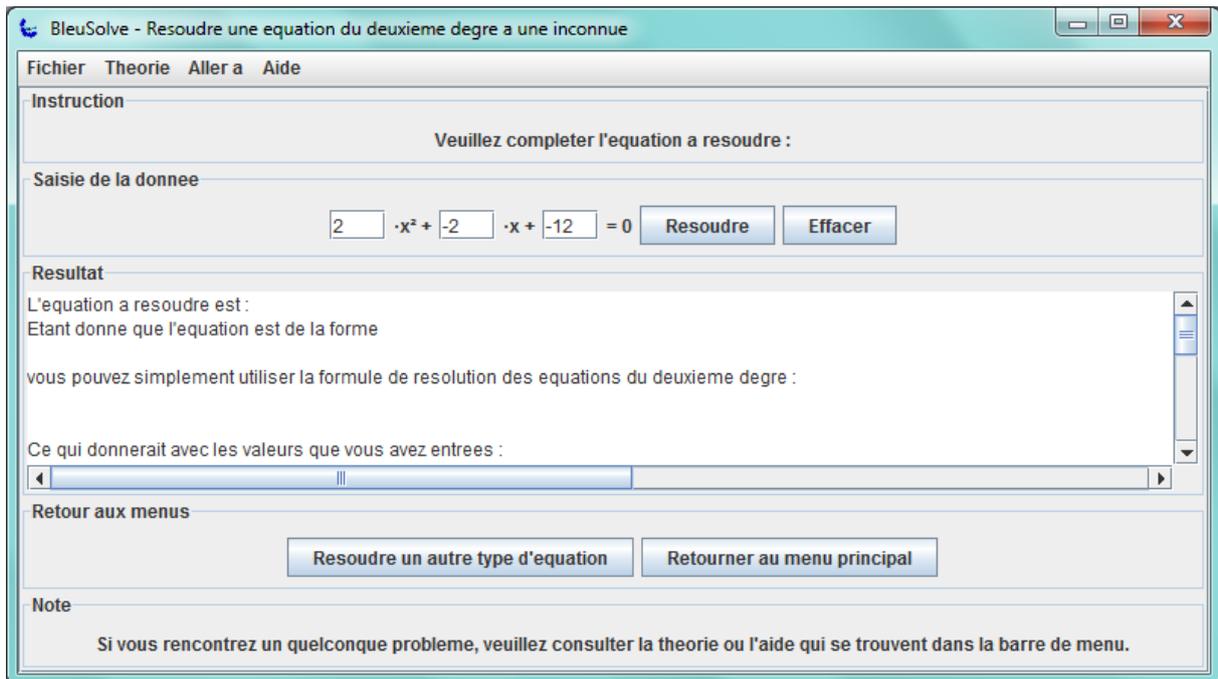


Image 13 : Résoudre une équation du deuxième degré à une inconnue – Résultat

Si le discriminant de votre équation est négatif (ce qui veut dire que votre équation n'a pas de solution dans l'ensemble des nombres réels), le logiciel résoudra l'équation dans l'ensemble des nombres complexes.

#### 4.7.1.3. Résoudre une équation du troisième degré à une inconnue (avec une solution entière comprise entre -10 et 10)

En arrivant dans la fenêtre de résolution, vous serez confronté à ceci :

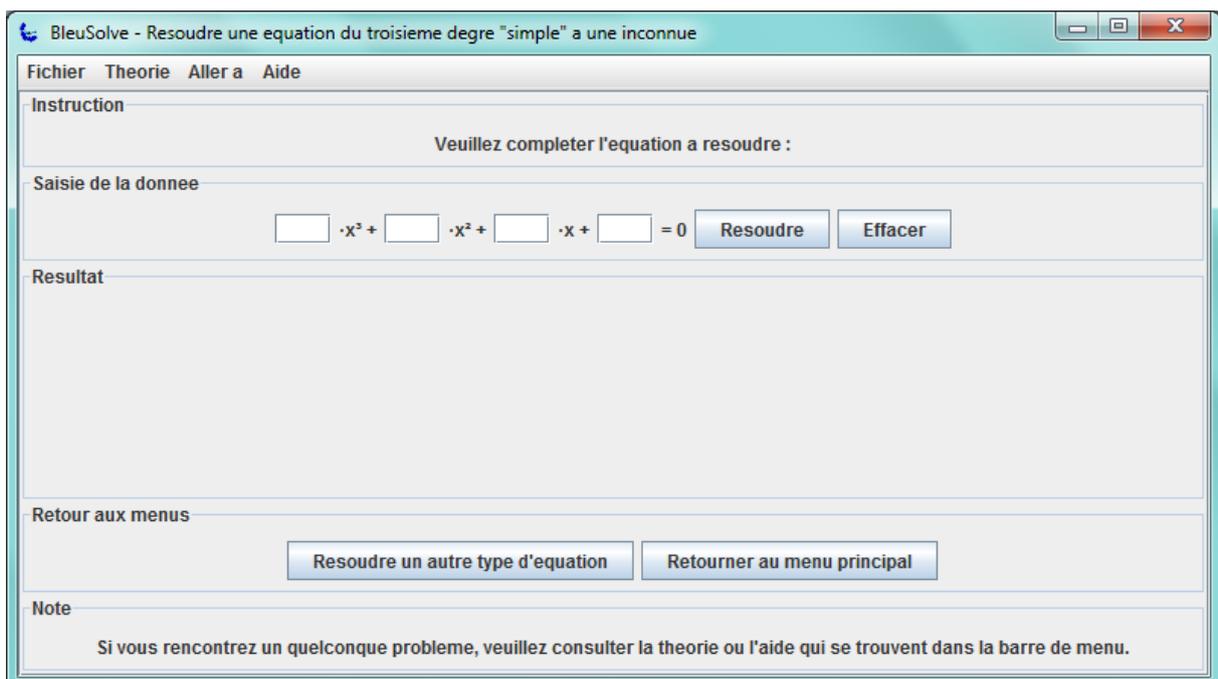


Image 14 : Résoudre une équation du troisième degré "simple" à une inconnue

Si vous souhaitez, par exemple, calculer «  $2 \cdot x^3 + 8 \cdot x^2 - 62 \cdot x - 140 = 0$  », inscrivez « 2 » dans le premier champ de texte, « 8 » dans le deuxième, « -62 » dans le troisième et « -140 » dans le quatrième. Après avoir cliqué sur le bouton « Résoudre », vous obtiendrez ceci :

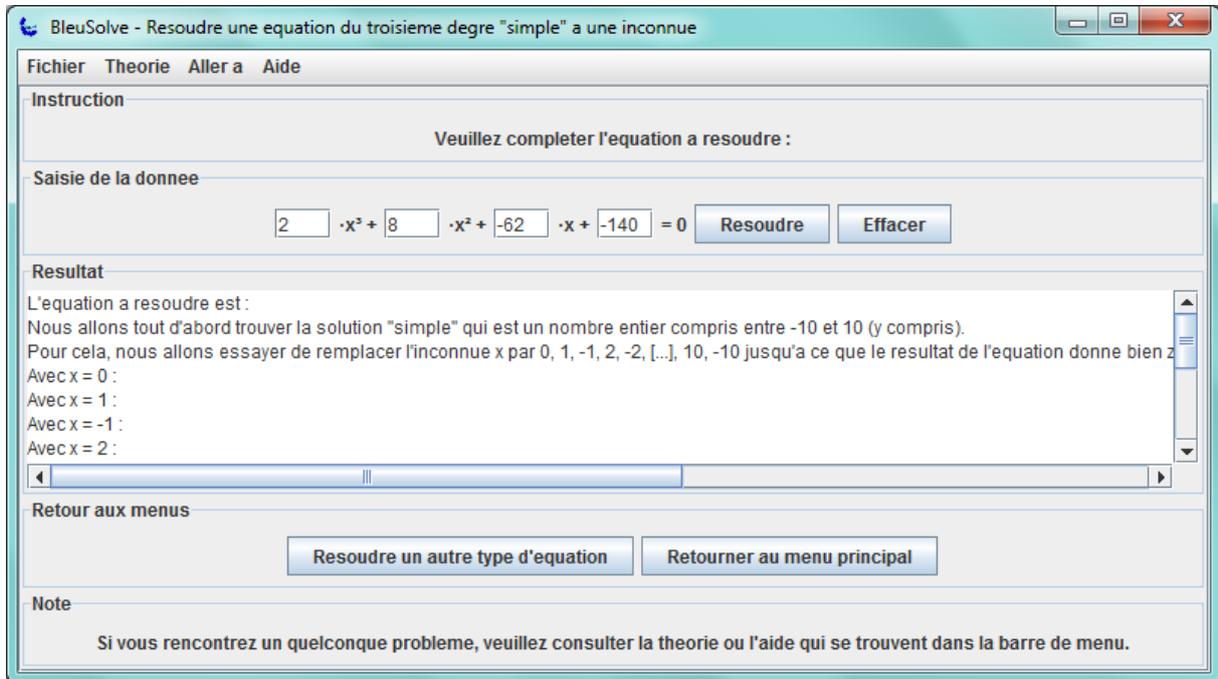


Image 15 : Résoudre une équation du troisième degré "simple" à une inconnue – Résultat

Le programme ne permet malheureusement que de résoudre des équations du troisième degré qui contiennent au moins une solution entière comprise entre -10 et 10 (y compris). Si une telle solution n'existe pas pour votre équation, le logiciel vous en informera.

#### 4.7.2. Résoudre un système d'équations à plusieurs inconnues

Dans ce menu, vous trouverez les activités suivantes :

- Résoudre un système de deux équations à deux inconnues
- Retourner au menu principal
- Quitter BleuSolve (*et effacer, par la même occasion, tous les fichiers PDF créés*)

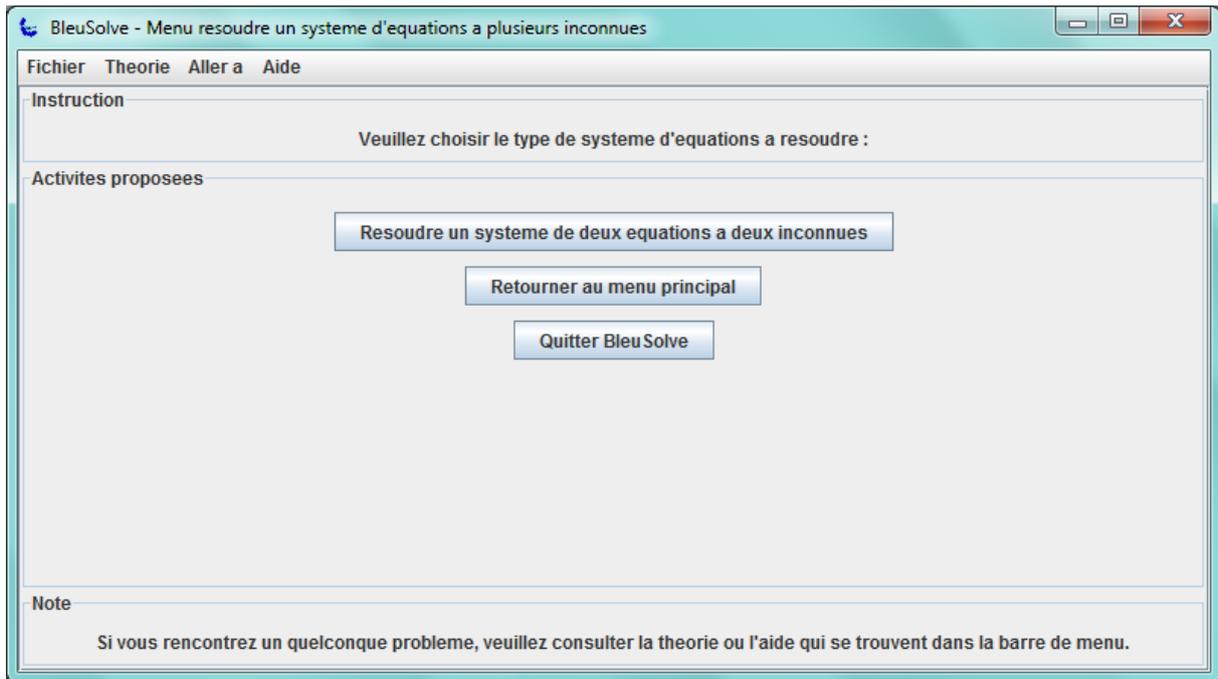


Image 16 : Menu résoudre un système d'équation à plusieurs inconnues

#### 4.7.2.1. Résoudre un système de deux équations à deux inconnues

En arrivant dans la fenêtre de résolution, vous serez confronté à ceci :

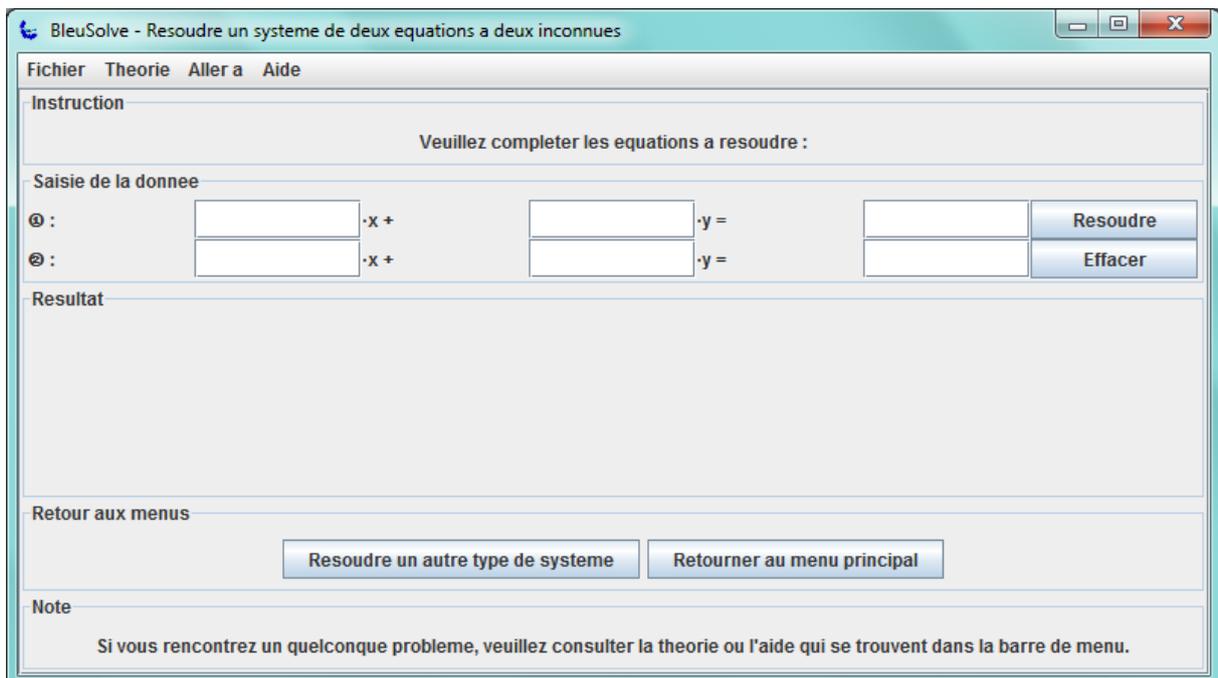


Image 17 : Résoudre un système de deux équations à deux inconnues

Si vous souhaitez, par exemple, résoudre le système qui est composé des deux équations suivantes : «  $7 \cdot x + 3 \cdot y = 5$  » et «  $x - 2y = 8$  ». Dans la première ligne, inscrivez « 7 » dans le premier champ de texte, « 3 » dans le deuxième et « 5 » dans le troisième. Dans la seconde ligne, inscrivez « 1 » dans le premier, « -2 » dans le deuxième et « 8 » dans le troisième.

Le résultat sera composé d'une à trois méthode(s) de résolution : La combinaison linéaire, la substitution et/ou la formule de Cramer.

Après avoir cliqué sur le bouton « Résoudre », vous obtiendrez ceci :

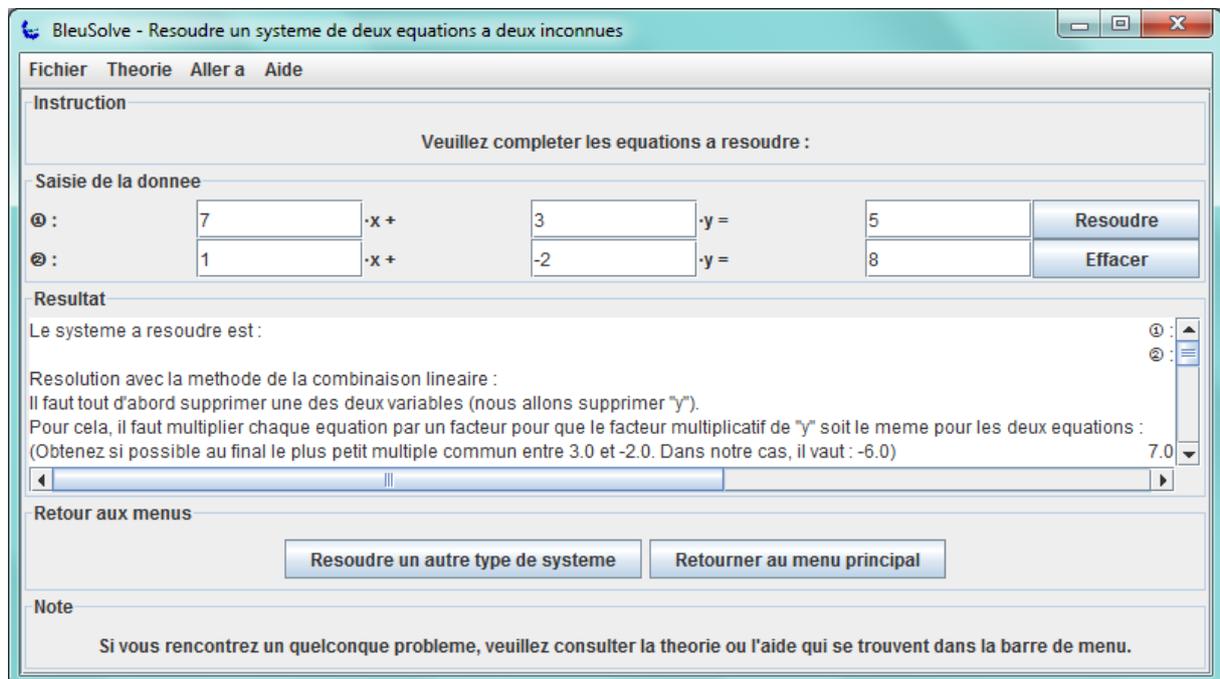


Image 18 : Résoudre un système de deux équations à deux inconnues – Résultat

Si c'est le cas, BleuSolve vous indiquera si votre système n'a pas de solution ou si la solution est l'équation d'une droite (infinité de solutions). Il vous montrera aussi parfois, la meilleure méthode à utiliser dans des systèmes particuliers.

#### 4.7.3. Nombres complexes

Dans ce menu, vous trouverez les activités suivantes :

- Calculer les racines carrées d'un nombre complexe
- Résoudre une équation du premier degré à une inconnue (*dans l'ensemble des nombres complexes*)
- Retourner au menu principal
- Quitter BleuSolve (*et effacer, par la même occasion, tous les fichiers PDF créés*)

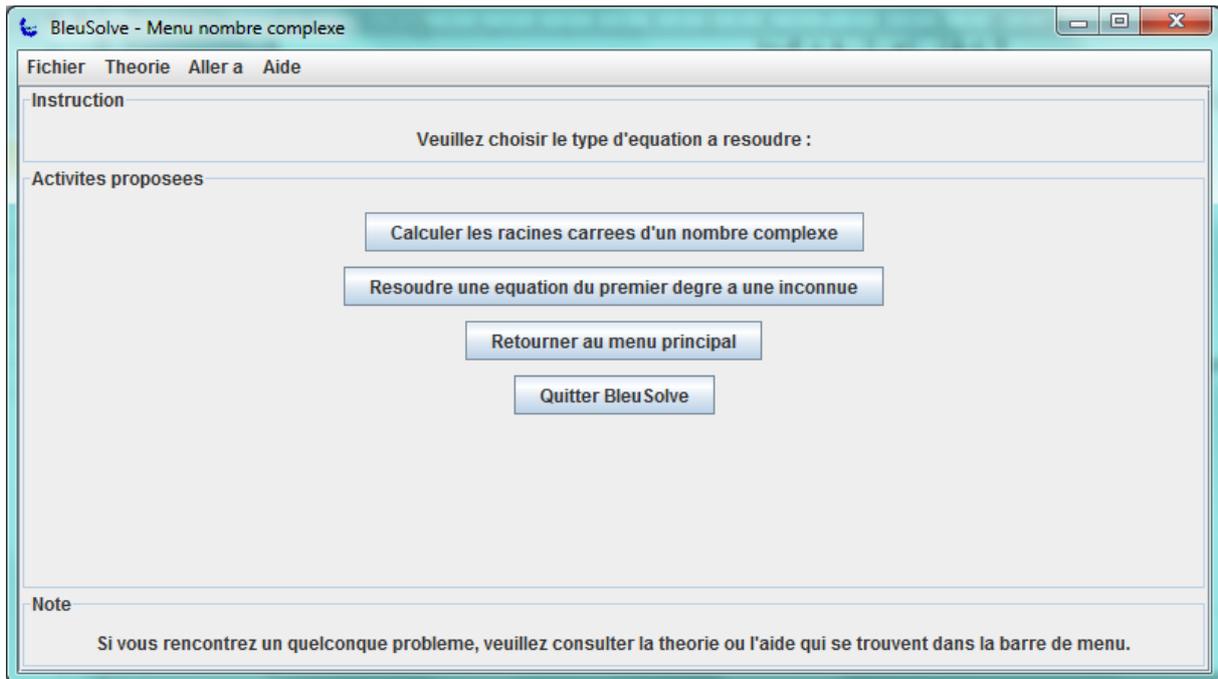


Image 19 : Menu nombre complexe

#### 4.7.3.1. Calculer les racines carrées d'un nombre complexe

En arrivant dans la fenêtre de résolution, vous serez confronté à ceci :

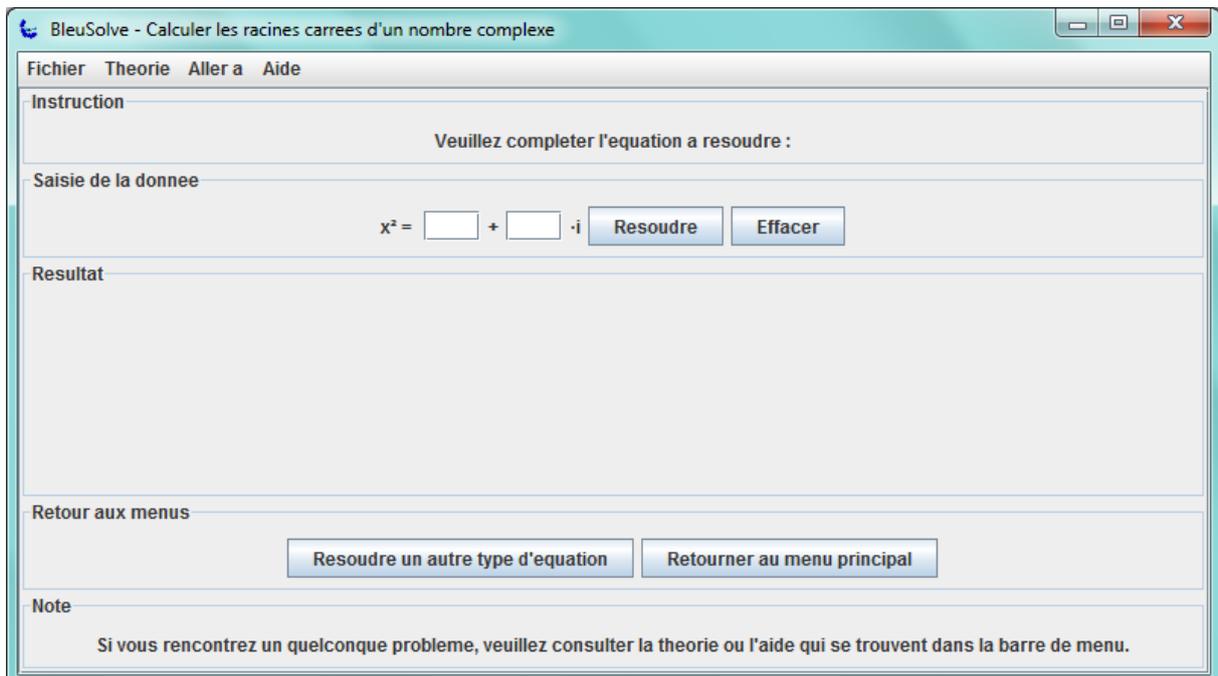


Image 20 : Calculer les racines carrées d'un nombre complexe

Si vous souhaitez, par exemple, calculer les racines carrées de «  $3 - 4 \cdot i$  », inscrivez « 3 » dans le premier champ de texte et « -4 » dans le deuxième.

Le résultat sera composé d'une résolution algébrique et d'une résolution en passant par la forme polaire.

Après avoir cliqué sur le bouton « Calculer », vous obtiendrez ceci :

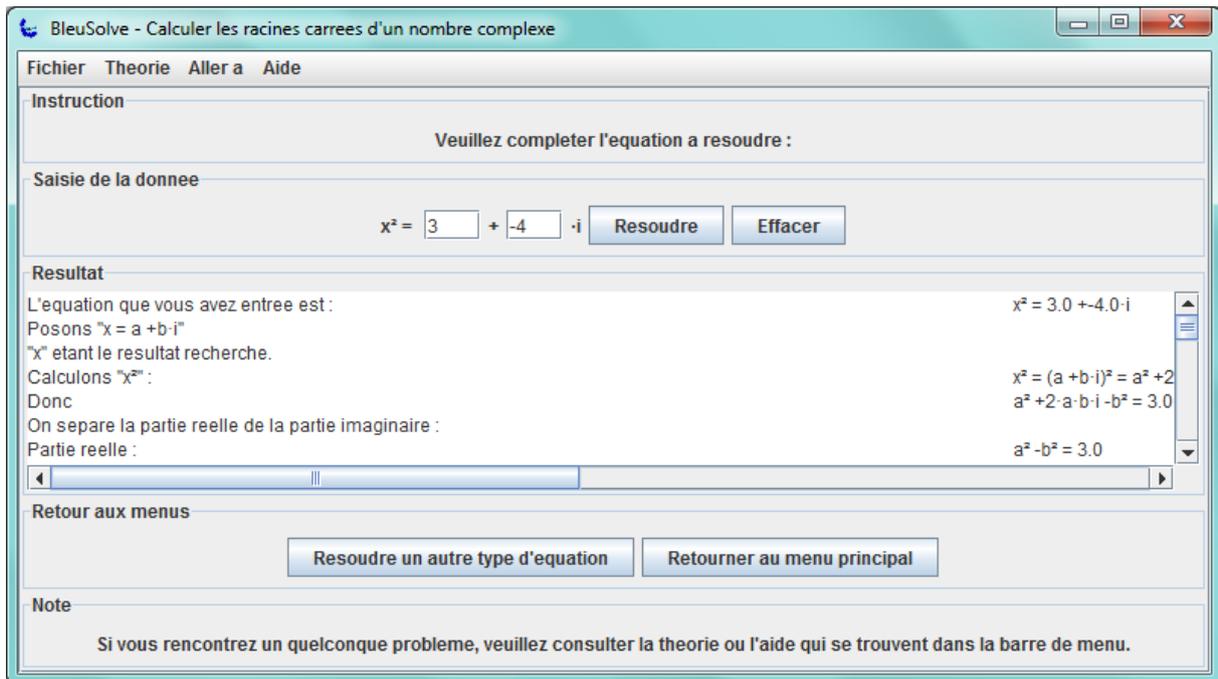


Image 21 : Calculer les racines carrées d'un nombre complexe - Résultat

#### 4.7.3.2. Résoudre une équation du premier degré à une inconnue (Nombre complexe)

En arrivant dans la fenêtre de résolution, vous serez confronté à ceci :

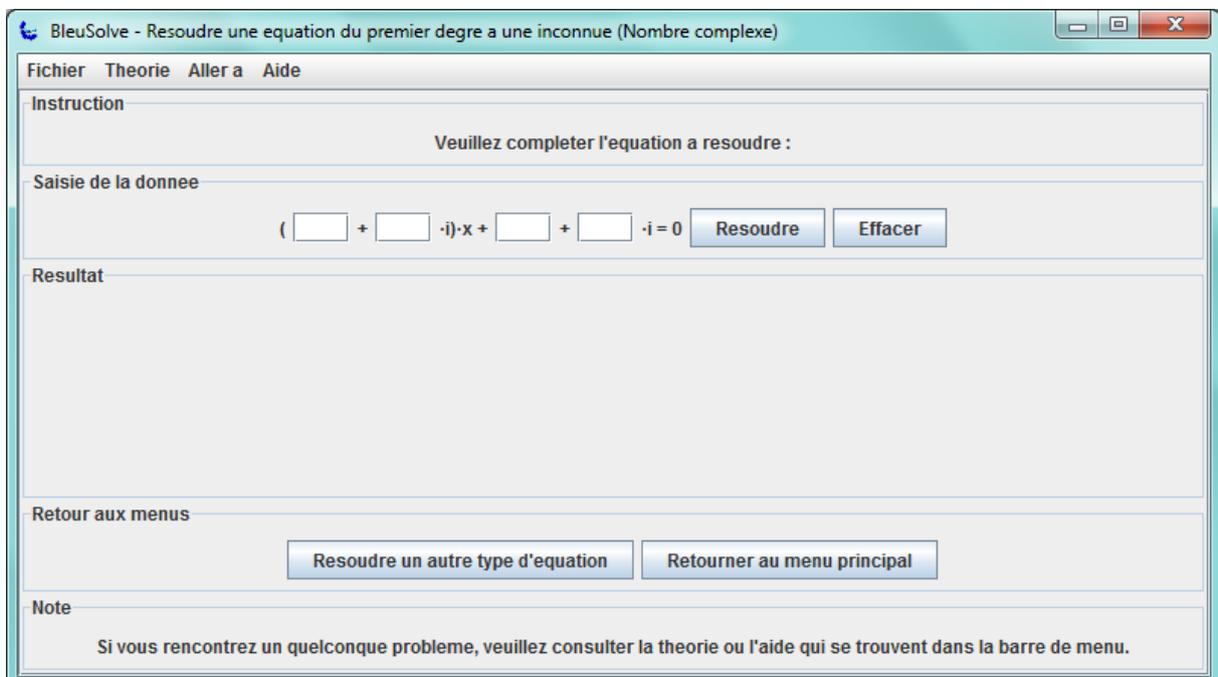


Image 22 : Résoudre une équation du premier degré à une inconnue (Nombre complexe)

Si vous souhaitez, par exemple, résoudre «  $(2 - i) \cdot x + (6 - 8 \cdot i) = 0$  », inscrivez « 2 » dans le premier champ de texte, « -1 » dans le deuxième, « 6 » dans le troisième et « -8 » dans le quatrième.

Après avoir cliqué sur le bouton « Résoudre », vous obtiendrez ceci :

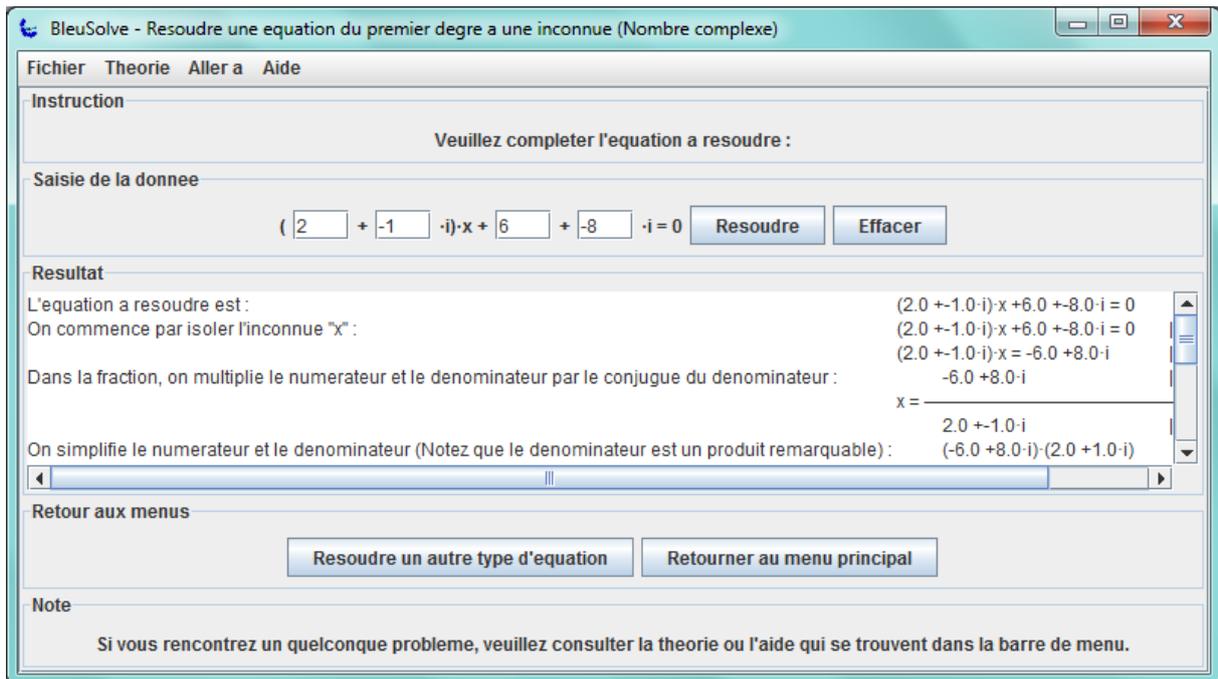


Image 23 : Résoudre une équation du premier degré à une inconnue (Nombre complexe) - Résultat

#### 4.7.4. Divers

Dans ce menu, vous trouverez les activités suivantes :

- Division polynomiale (avec la méthode de Horner)
- Retourner au menu principal
- Quitter BleuSolve (et effacer, par la même occasion, tous les fichiers PDF créés)

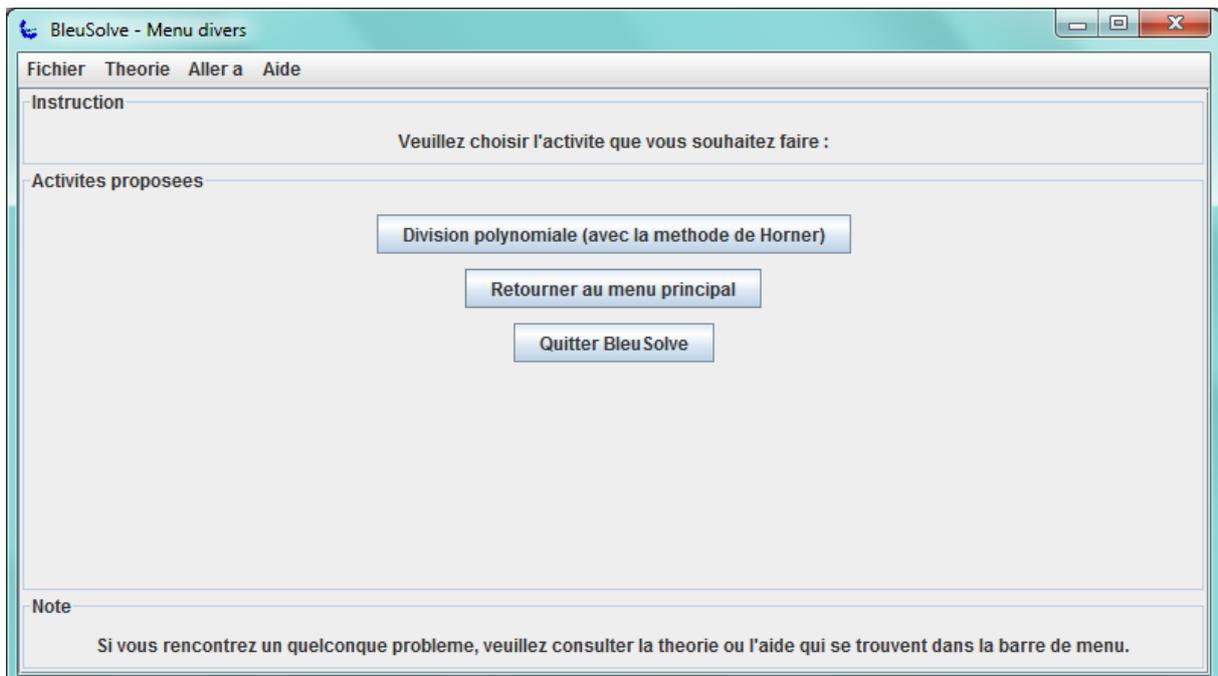


Image 24 : Menu divers

### 4.7.4.1. Division polynomiale (avec la méthode de Horner)

En arrivant dans la fenêtre de résolution, vous serez confronté à ceci :

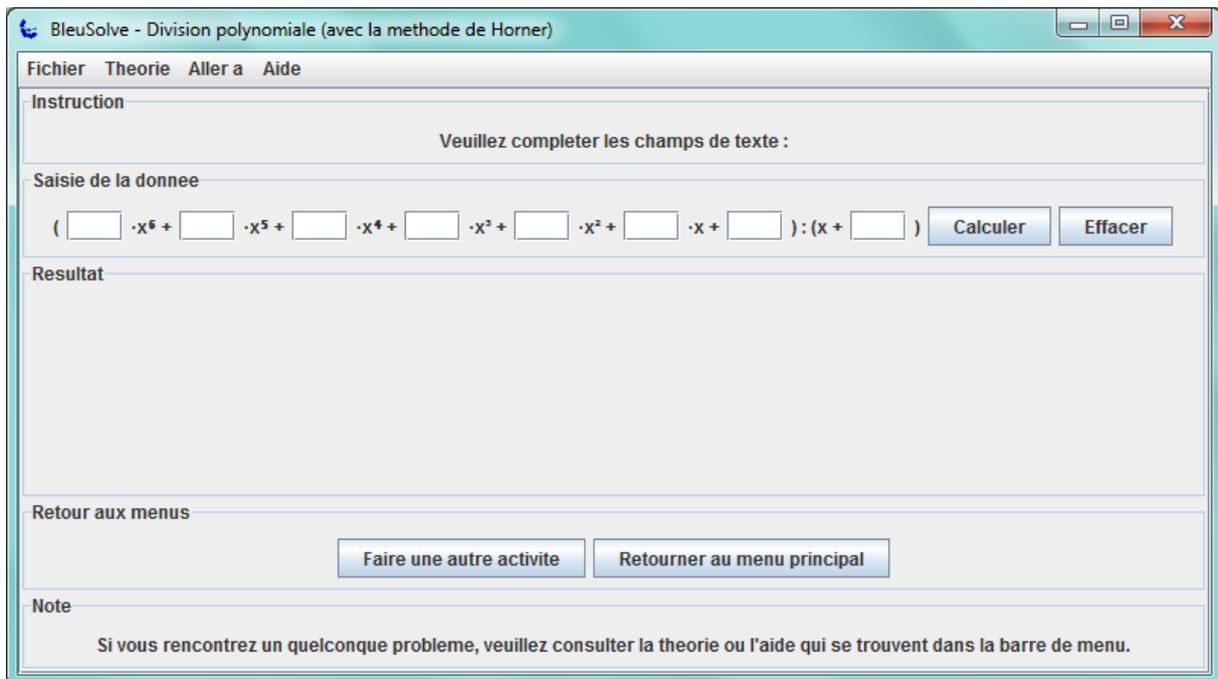


Image 25 : Division polynomiale (avec la méthode de Horner)

Si vous souhaitez, par exemple, diviser «  $-5 \cdot x^5 + 18 \cdot x^2 + 32$  » par «  $x - 2$  », inscrivez « 0 » ou rien dans le premier champ de texte, « -5 » dans le deuxième, « 0 » ou rien dans le troisième, « 0 » ou rien dans le quatrième, « 18 » dans le cinquième, « 0 » ou rien dans le sixième, « 32 » dans le septième et « -2 » dans le huitième. Après avoir cliqué sur le bouton « Calculer », vous obtiendrez ceci :

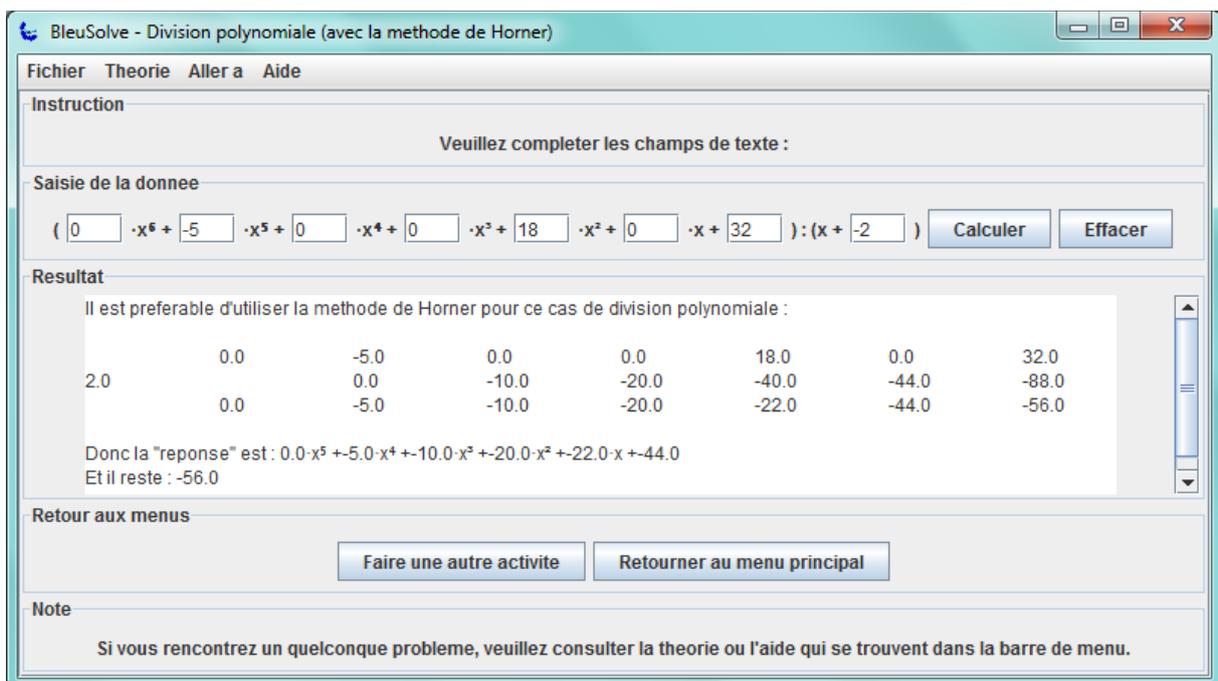


Image 26 : Division polynomiale (avec la méthode de Horner) - Résultat

#### 4.7.5. Extras

Dans ce menu, vous trouverez les activités suivantes :

- Calculateur de moyenne
- Retourner au menu principal
- Quitter BleuSolve *(et effacer, par la même occasion, tous les fichiers PDF créés)*

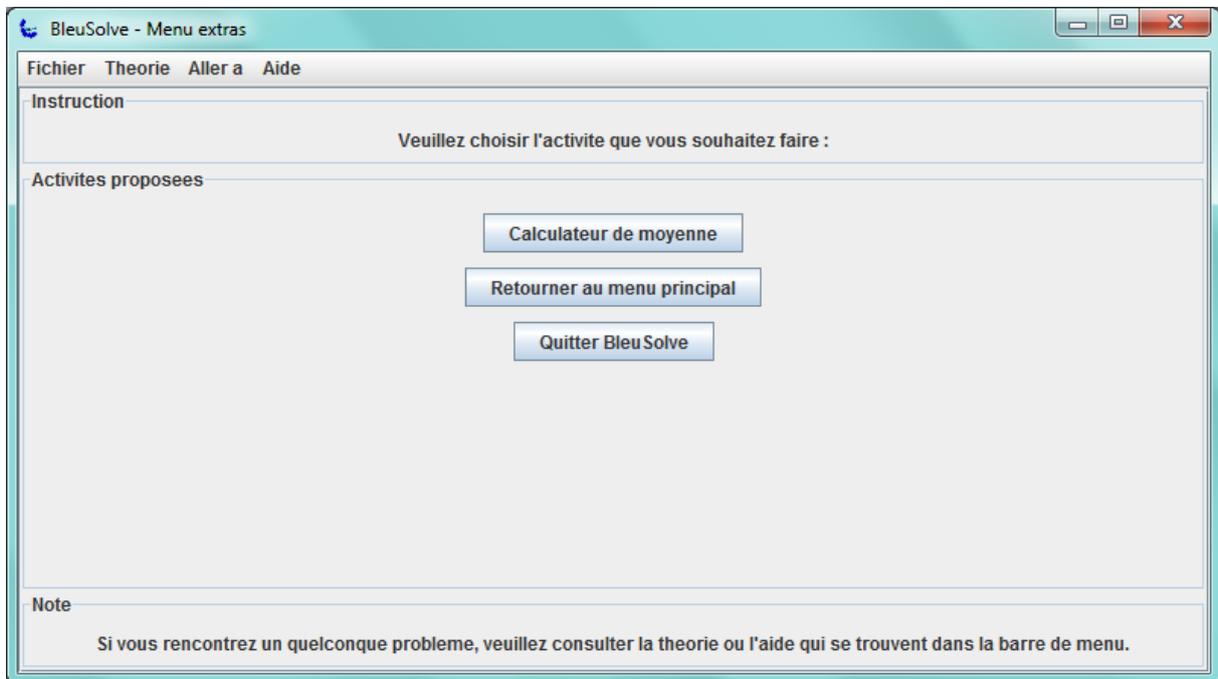


Image 27 : Menu extras

##### 4.7.5.1. Calculateur de moyenne

En arrivant dans la fenêtre de résolution, vous serez confronté à ceci :

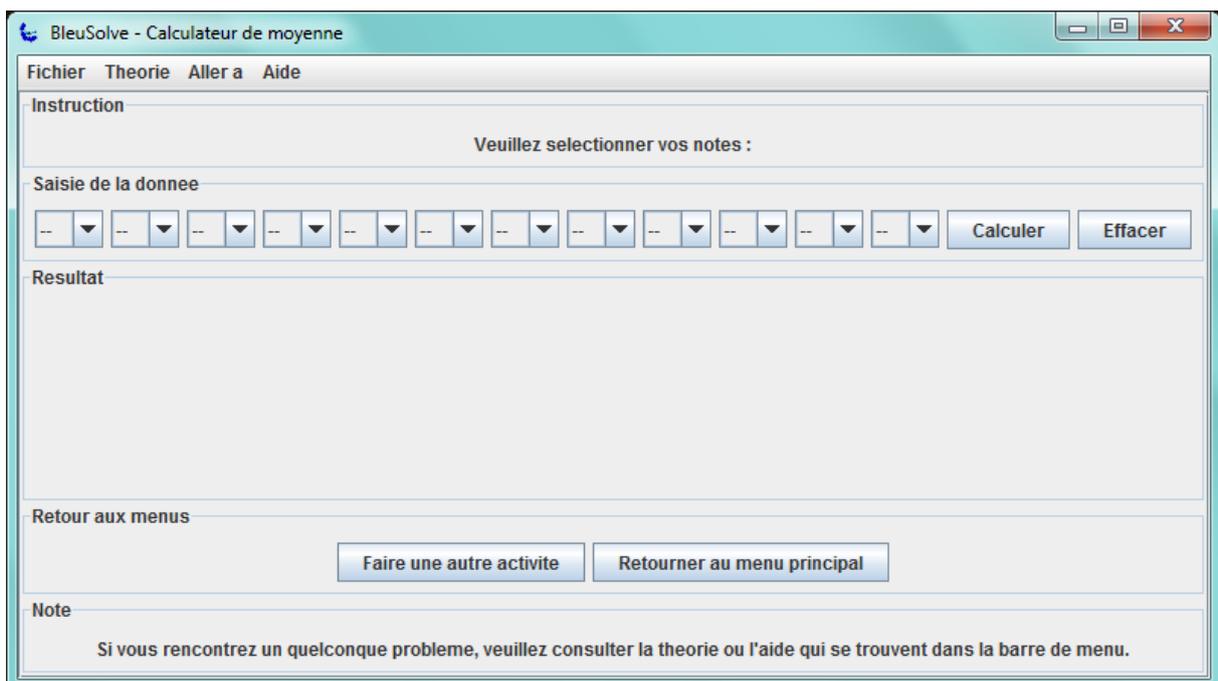


Image 28 : Calculateur de moyenne

Vous pouvez entrer jusqu'à douze notes grâce aux boîtes à liste déroulante. Cliquez sur l'une des boîtes et choisissez votre note. Quand « -- » est affiché, cela veut dire qu'il n'y a pas de notes.

Si, par exemple, vous avez les notes suivantes : 4 ; 6 ; 3.5 ; 4.5 ; 4 ; 5.5

Après avoir entré toutes vos notes et cliqué sur le bouton « Calculer », vous obtiendrez ceci :

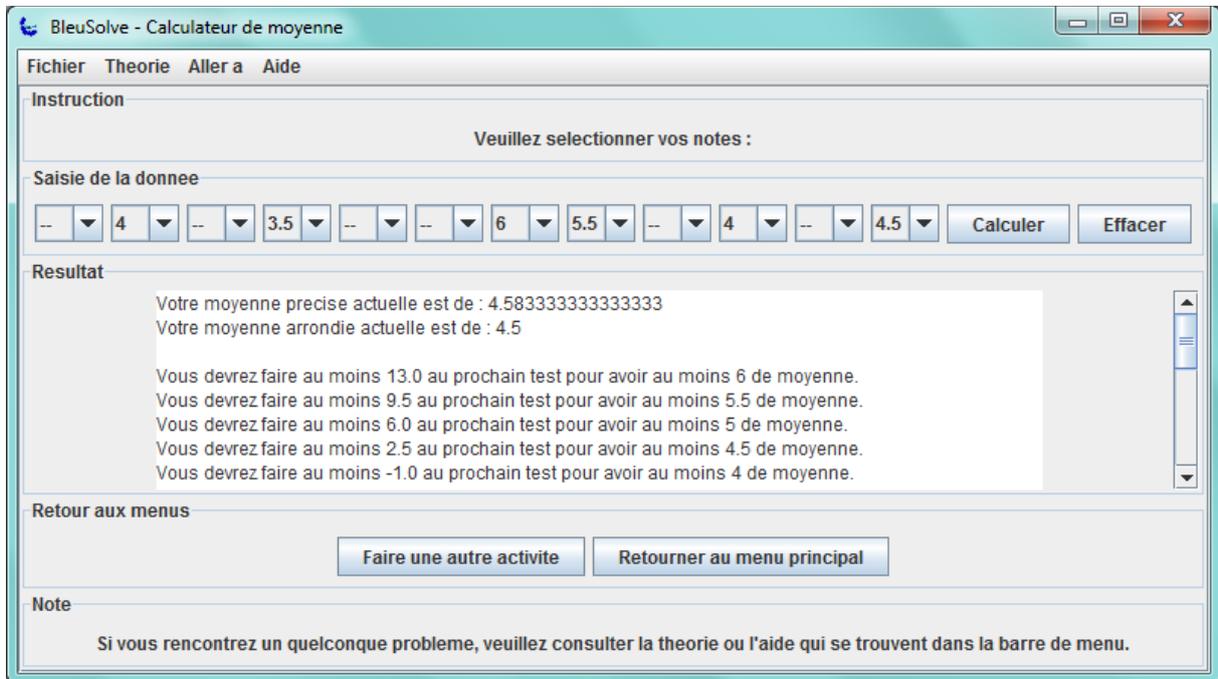


Image 29 : Calculateur de moyenne – Résultat

Le programme propose même des notes impossibles à obtenir (celles qui sont plus grandes que 6 ou plus petites que 1), car certain(e)s élèves calculent les moyennes avec des notes absurdes pour vérifier qu'il est très difficile d'avoir la moyenne désirée ou non-désirée.

#### 4.8. Quitter BleuSolve

Vous avez de multiples moyens pour quitter BleuSolve (cliquer sur la croix rouge, aller dans la barre de menu, dans les différents menus, etc...). À chaque fois que vous souhaitez quitter le logiciel, une confirmation s'affichera pour éviter d'éventuelle(s) fausse(s) manœuvre(s).

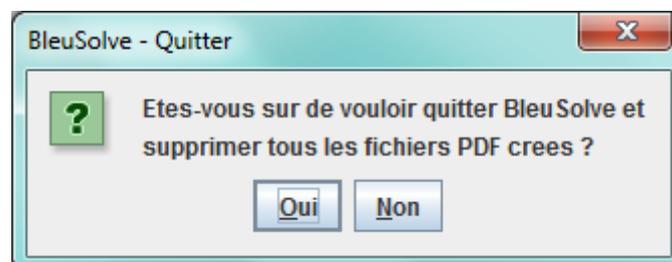


Image 30 : Boîte de confirmation avant de quitter

## 5. Documentation de développement

BleuSolve a été programmé avec l'aide du logiciel « Eclipse IDE for Java Developers<sup>3</sup> » (version 1.4.1). Il a été testé et fonctionne parfaitement avec « Java 6<sup>4</sup> » sous « Windows 7 ». Il est possible que certains problèmes de compatibilité surviennent quand une version antérieure de Java ou un autre système d'exploitation est utilisé.

Java est un langage de programmation orienté objet. Ce qui veut dire qu'on peut créer des « objets », définir ses caractéristiques et les faire interagir entre eux. Pour créer ces objets, on utilise des classes qui définissent des caractéristiques particulières qui nous serviront à créer des objets sur la base de ces classes. Dans le cas de BleuSolve, cette notion de « plusieurs objets semblables créés à partir d'une classe » n'est pas utilisée, puisqu'on n'a besoin que de créer un seul objet de chaque classe (un seul écran d'attente et une seule fenêtre).

BleuSolve est composé de trois classes qui ont chacune des fonctionnalités et un rôle bien distinct :

- Launcher.java
- SplashScreen.java
- Window.java

### 5.1. Launcher.java

La classe « Launcher.java » contient la méthode « main » du logiciel. C'est donc elle qui s'exécutera en premier et qui affichera la suite du programme.

Elle permet de créer un objet « splash » de la classe « SplashScreen.java » pour l'image d'accueil et de l'afficher.

Elle permet aussi de créer un objet « wind » de la classe « Window.java » pour la fenêtre principale et de l'afficher.

Puis, le programme fait une pause durant 1291 millisecondes (c'est le temps minimal d'affichage de l'écran d'accueil pour que l'utilisateur puisse voir le logo et lire les informations).

Finalement, l'objet « splash » se détruit (pour ne pas le garder en mémoire inutilement puisqu'on ne l'utilise qu'une seule fois) et « wind » est affiché tout à la fin pour que, quand la fenêtre s'affiche, tout est déjà mis en place (cela évite d'avoir une fenêtre vide).

### 5.2. SplashScreen.java

Cette classe permet d'afficher une image d'attente durant la création de l'objet « wind ».

Pour cela, elle définit la taille d'une fenêtre sans la barre de titre sur laquelle on affichera notre image et quelques informations à propos du logiciel, elle est composée de deux

---

<sup>3</sup> <http://www.eclipse.org/>

<sup>4</sup> <http://www.java.com/>

panneaux contenant chacun une étiquette qui est une image pour l'un et du texte pour l'autre.

### 5.3. Window.java

La classe « Window.java » affiche la fenêtre avec laquelle l'utilisateur va interagir.

Pour naviguer entre les différentes fonctionnalités de BleuSolve, la méthode utilisée est de créer tous les panneaux, puis d'afficher ou d'enlever des panneaux en suivant ce qu'on souhaite voir sur la fenêtre. À chaque fois que les panneaux changent, le panneau de fond se rafraîchit brièvement pour que l'utilisateur puisse voir le changement.

La classe « Window.java » est composée de plusieurs parties :

- Création des éléments
- Paramétrage de la fenêtre et de la barre de menu
- Paramétrage et fixation des éléments
- Affectation des actions aux boutons

#### 5.3.1. Création des éléments

C'est dans cette grande partie que tous les éléments qui constituent la fenêtre (panneaux, boutons, champs de texte, variables, etc...) sont créés.

Étant donné qu'il y a un très grand nombre d'éléments qui sont créés, il fallait qu'il y ait des règles pour définir leurs noms pour éviter d'éventuelles confusions et aussi pour faciliter les relectures :

- Le nom commence toujours par une minuscule et chaque mot est séparé par une majuscule.
- Le nom commence toujours par indiquer ce qu'il représente (panneau, bouton, ...) sauf pour les variables, certaines étiquettes qui s'appellent « message » et les zones de texte qui sont appelées « resultat ».
- Si l'élément n'est utilisé que pour une seule partie spécifique du logiciel, cette partie est indiquée (ReEq, ReCo, ...).
- Pour les panneaux, leur emplacement sur la fenêtre est indiqué (Nord, Centre, Sud).

#### 5.3.2. Paramétrage de la fenêtre et de la barre de menu

##### 5.3.2.1. Paramétrage de la fenêtre

La première chose que l'objet « wind » de la classe « Window.java » effectue, c'est de définir la taille, l'icône, le nom, la position de la fenêtre par rapport à l'écran, la maximisation et l'action qui suit la fermeture de la fenêtre. Elle ajoute aussi la barre de menu et le panneau de fond à la fenêtre. Le panneau de fond ajouté est en fait un panneau de défilement<sup>5</sup> sur lequel est fixé le vrai panneau de fond (où seront ensuite fixés tous les autres panneaux) pour que l'utilisateur puisse avoir accès au contenu même s'il réduit la fenêtre au-delà de la taille par défaut et qu'il n'y ait pas de problème d'affichage.

---

<sup>5</sup> Panneau qui affiche des barres de défilement quand le contenu est plus grand que le panneau en lui-même.

### 5.3.2.2. Paramétrage de la barre de menu

Puis, elle ajoute les options proposées dans la barre de menu. Pour que ces options servent à quelque chose, on ajoute des « ActionListener » qui définiront une succession d'évènement quand on clique dessus (on fera la même chose pour les boutons).

Si on clique sur « Retourner au menu principal » ou « Quitter BleuSolve », vous serez renvoyé à une classe qui se trouve à la fin du programme. Je vais aborder le sujet plus amplement dans « Affectation des actions aux boutons ».

Dans le menu « Théorie », l'utilisateur a la possibilité d'ouvrir des PDF qui se trouvent dans l'archive (pour pouvoir exporter BleuSolve avec tous ses composants dans le « .jar »). Pour que cela s'avère possible, le programme doit extraire le fichier dans le même dossier que le logiciel, puis il peut enfin l'ouvrir (il y a aussi le « Manuel de l'utilisateur » qui exécute le même code). Comme cette méthode (qui est la meilleure que j'ai trouvée) pollue un peu l'ordinateur de l'utilisateur en créant plein de fichiers, quand on ferme le programme, tous les fichiers créés seront automatiquement supprimés. Si vous deviez forcer BleuSolve à se fermer, les fichiers PDF ne seront malheureusement pas supprimés. Il faudra que vous les supprimiez manuellement (ils sont facilement reconnaissable grâce à leur préfixe « BleuSolve - ») ou que vous relanciez le programme pour les supprimer par la suite.

Pour que l'utilisateur puisse facilement supprimer tous les fichiers PDF créés sans pour autant fermer BleuSolve (cf. « Éléments importants à connaître »), il a la possibilité de le faire depuis la barre de menu sous le menu « Fichier ».

Grâce à la classe « Desktop », on peut facilement envoyer un mail avec le service de messagerie par défaut sur notre ordinateur ou encore ouvrir une page internet avec notre navigateur favori. Elle est compatible avec Windows et Mac OS notamment (on n'a plus besoin de vérifier le système d'exploitation comme avec d'autres méthodes).

« À propos » ouvre simplement une boîte de dialogue qui contient toutes les informations relatives à BleuSolve.

Le menu « Aller à » est composé de plusieurs sous-menus qui contiennent eux-mêmes une redirection vers des classes qui se trouvent à la fin du programme. On arrive exactement sur les mêmes codes que quand on navigue avec les boutons sur la fenêtre. Nous en reparleront dans « Affectation des actions aux boutons ».

### 5.3.3. Paramétrage et fixation des éléments

C'est ici que la plupart des éléments sont fixés l'un à l'autre pour obtenir l'affichage et le positionnement souhaité des éléments de la fenêtre grâce à la méthode « .add() ».

#### 5.3.3.1. Mise en page de la fenêtre

Pour positionner les différents composants dans les panneaux, on utilise des gestionnaires de mise en forme. Il en existe plusieurs et on peut les combiner entre eux pour obtenir la mise en page désirée.

Trois de ces gestionnaires sont utilisés dans le programme :

- **FlowLayout** : c'est le gestionnaire par défaut. Les composants sont placés à la suite et se mettent automatiquement à la ligne suivante s'il n'y a pas suffisamment de place (comme quand on écrit un texte). On peut y régler l'espace vertical et horizontal entre les différents composants et aussi l'alignement de ces derniers dans le panneau.
- **BorderLayout** : il sépare le panneau en cinq parties (au maximum) : Nord, Centre, Sud, Est et Ouest. Quand on modifie la taille d'un panneau, seule la partie centrale change et les autres parties restent minimisées.
- **GridLayout** : il crée un grillage et on place des éléments dans les différentes cases de ce dernier.

Pour les « petits éléments » (champs de texte, étiquettes, etc...), la disposition est laissée par défaut (FlowLayout).

Pour les panneaux, le « BorderLayout » est utilisé pour que l'utilisateur puisse changer la taille de la fenêtre sans que la disposition ne soit trop modifiée. L'inconvénient avec ce gestionnaire, c'est que, lorsqu'on change la taille de la fenêtre, seul le panneau du centre change de taille et les panneaux nord et sud sont minimisés à une ligne (les emplacements est et ouest ne sont pas utilisés). Dans les panneaux de résolution où l'on a besoin d'avoir plus de zone (cinq zones), deux « BorderLayout » sont intégrés l'un dans l'autre pour garder le maximum de place pour le résultat.

Pour le message de bienvenue, le « GridLayout » est utilisé pour forcer le « BorderLayout.NORTH » à afficher trois lignes. J'ai aussi utilisé le « GridLayout » pour afficher les deux équations du système sur deux lignes.

Pour les menus, le « FlowLayout » est réutilisé, mais on augmente l'espace entre les boutons pour qu'ils se mettent à la ligne d'en dessous. Il est aussi réutilisé pour supprimer l'espace entre les résultats et ainsi avoir les explications et les calculs collés ensemble.

Certains panneaux (par exemple « panneauSud ») sont réutilisés à plusieurs reprises pour éviter de créer plusieurs panneaux parfaitement identiques.

### 5.3.3.2. Paramétrage des éléments

Après avoir fixé les différents éléments, certains doivent être réglés car les réglages proposés par défaut ne sont pas toujours très pratiques :

- Les résultats ne sont pas éditables
- Création des titres pour les panneaux
- Réglage de l'incrémentation des barres de défilement pour pouvoir se déplacer dans le résultat plus aisément (car l'incrémentation par défaut est trop petite)
- Réglage des boîtes à liste déroulante pour agrandir le nombre de notes proposées quand on clique sur la boîte et ainsi éviter de devoir utiliser la molette pour sélectionner sa note<sup>6</sup>

### 5.3.4. Affectation des actions aux boutons

Jusqu'à maintenant, le programme n'a fait que la mise en place des composants pour que l'utilisateur puisse résoudre des équations. Voici maintenant la plus grande et importante partie puisque tous les changements de menus et les résultats sont provoqués par l'action de cliquer sur des boutons. Autrement dit, à chaque clic sur un bouton, les panneaux affichés sur la fenêtre changent. Ces boutons sont affectés à des événements et il y en a deux sortes :

- Évènements à usage unique
- Évènements à usage multiple

#### 5.3.4.1. Évènements à usage unique

Les événements (ou actions) qui ne se déclenchent que par l'action d'un seul bouton sont programmés directement dans la classe (vous verrez que les autres seront légèrement à part).

Ces boutons sont les boutons « Résoudre » ou « Calculer » et les boutons « Effacer ». Les deux premiers permettent de créer et d'afficher les résultats, le bouton « Effacer », d'effacer les champs de texte et les résultats.

---

<sup>6</sup> Les notes ont été mises dans l'ordre décroissant, car je pars du principe que l'utilisateur fait de bonnes notes !

Nous allons voir maintenant ce que fait le programme quand on clique sur « Résoudre » :

- Tout d'abord, il affecte les nombre inscrits par l'utilisateur aux variables. C'est aussi à ce moment-là que les exceptions sont prises en compte : quand l'utilisateur entre un caractère invalide (lettre, caractère spéciaux, etc...) ou rien, le programme associe la valeur nulle à cette variable.
- Ensuite, il calcule le résultat en prenant bien en compte tous les cas particuliers (quand  $0 = 0$ ,  $2 = 0$ ,  $x/0$ , etc...)<sup>7</sup>. Le résultat est, dans la plupart du temps, séparé en deux parties : les explications « Expl » et les calculs « Calc ». Dans les résultats, le langage Unicode est utilisé pour les caractères spéciaux.
- Enfin, on monte la barre de défilement pour afficher le résultat depuis le début (car par défaut, l'utilisateur voit la fin du résultat quand il clique sur « Résoudre »).

Les boutons « Effacer » remettent simplement les valeurs de départ aux champs de texte et effacent le résultat.

Dans les résultats proposés, j'ai essayé d'expliquer le plus clairement possible chaque étape pour parvenir au résultat. Il se peut que vous trouviez que je détaille parfois un peu trop les calculs, mais il ne faut pas oublier qu'il y a des élèves qui ont vraiment de la peine à comprendre certains raisonnements et en détaillant j'espère que cela les aidera à comprendre pourquoi à tel moment, on effectue tel calcul et ainsi pouvoir suivre le raisonnement logique des différentes méthodes de résolution. Je suis tout à fait conscient que mes explications n'ont pas toujours eu la clarté voulue, mais parfois, il est beaucoup plus simple d'expliquer oralement que d'écrire une explication accessible à tous.

#### 5.3.4.1.1. Quelques résolutions particulières

La plupart des résultats sont construits de manière basique (voire la partie précédente), mais parfois, des résolutions nécessitent d'autres calculs :

- Pour factoriser une équation du deuxième degré à une inconnue, j'ai dû vérifier si un nombre est entier ou non. Pour cela, j'ai transformé ce nombre réel en entier, puis inversement, et j'ai égalisé le résultat au nombre de départ.
- Pour obtenir les résultats quand on résout une équation du troisième degré à une inconnue, j'ai utilisé une boucle pour trouver la solution simple. Le résultat est aussi particulier : une chaîne de caractères se crée, puis elle est ajoutée au résultat.
- Pour résoudre un système de deux équations à deux inconnues, on a parfois besoin de calculer le plus grand diviseur commun entre les deux facteurs multiplicatifs de « y » pour connaître leur plus petit multiple commun. Pour le calculer, j'ai utilisé l'algorithme d'Euclide qui se résume à ce schéma tiré de Wikipédia<sup>8</sup> :

---

<sup>7</sup> Je tiens quand même à préciser : « tous les cas particuliers que le développeur a trouvé ou que les testeurs lui ont signalés ». Car parfois, on ne peut malheureusement pas penser à toutes les possibilités.

<sup>8</sup> [http://fr.wikipedia.org/wiki/Algorithme\\_d'Euclide](http://fr.wikipedia.org/wiki/Algorithme_d'Euclide)

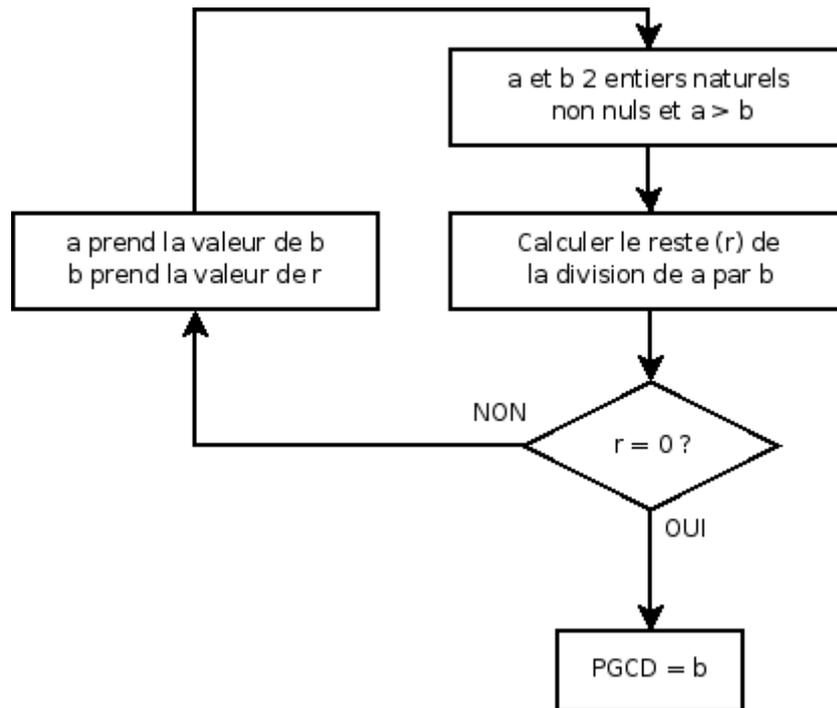


Image 31 : Algorithme d'Euclide

- Le calculateur de moyenne utilise des boîtes à liste déroulante au lieu des champs de texte (pour obliger l'utilisateur à entrer des notes qui sont possibles). J'ai aussi dû intégrer un compteur de nombre de note entrée qui se décrémente à chaque fois que l'utilisateur n'entre pas de note.

#### 5.3.4.2. Évènements à usage multiple

Pour les évènements qui peuvent être déclenchés par plusieurs boutons, des classes ont été créées pour éviter de recopier plusieurs fois les mêmes codes. Ces classes sont placés tout à la fin du programme en dehors de la classe « Window ». Cette méthode est utilisée pour tous les boutons « Quitter », pour les changements de menus et pour l'affichage des panneaux de résolution.

Le titre de la fenêtre change en même temps que les panneaux pour que l'utilisateur puisse savoir rapidement dans quelle partie du logiciel il se trouve actuellement.

À chaque changement de panneaux, on enlève tous les panneaux actuels, puis on ajoute les nouveaux et finalement, on rafraîchit le panneau de fond.

##### 5.3.4.2.1. Les menus

On reconnaît les panneaux qui sont créés spécifiquement pour les menus grâce au « [...]Me[...] » qui est contenu dans le nom des panneaux. Par exemple : « panneau**Me**PrNord ». La mise en page de ces panneaux est simple :

- Instruction pour que l'utilisateur sache ce qu'il doit faire (« Nord »)
- Redirection vers d'autres panneaux (« Centre »)
- Note (« Sud »)

#### 5.3.4.2.2. Les panneaux de résolutions

Ces panneaux sont reconnus grâce au « [...]Re[...] » qui est contenu dans leur nom. Par exemple : « panneauReEq1Centre ». Il y a un peu plus d'éléments à afficher dans les résolutions que dans les menus :

- Instruction pour que l'utilisateur sache ce qu'il doit faire (« Nord »)
- Donnée que l'utilisateur va dûment compléter (« CentreNord »)
- Résultat proposé par le logiciel (« CentreCentre »)
- Possibilité de retourner au précédent menu et au menu principal (« CentreSud »)
- Note (« Sud »)

## 6. Conclusion

À travers ce travail de maturité, j'ai découvert que le monde de la programmation est vraiment passionnant (même si parfois, on a l'impression de passer plus de temps sur des forums qu'à programmer). Comme on peut le constater, les prévisions que j'avais eues (cf. Présentation du sujet) étaient un peu trop ambitieuses par rapport à ce que j'ai réellement réussi à réaliser pour mon logiciel. Les simplifications ont été rapidement abandonnées car elles semblaient beaucoup plus compliquées à programmer que prévu. Je me suis donc concentré sur les résolutions (en ajoutant les nombres complexes). Certes, cela aurait été beaucoup plus intéressant si BleuSolve pouvait résoudre des équations du deuxième et troisième degré dans l'ensemble des nombres complexes ou encore de résoudre des systèmes de trois équations à trois inconnues, mais ces résolutions demanderaient un temps considérable à programmer. Car à part les résolutions, il y avait encore d'autres aspects dont je devais m'occuper : l'aspect didactique et la facilité d'utilisation notamment. Les explications devaient être claires et l'utilisateur doit avoir accès à toutes les options du logiciel intuitivement. Ce sont des choses qui passent dans la plupart du temps inaperçues quand on utilise un logiciel qui fonctionne bien, mais qui nécessite parfois beaucoup de recherche (tous les images et les PDF sont contenus dans l'archive, l'intégration des barres de défilement, les caractères spéciaux, etc...). C'est assez surprenant de réaliser qu'il y a une réflexion qui a été faite derrière chaque petit détail d'un programme.

Malgré le peu d'options que propose BleuSolve, je trouve qu'il donne déjà un avant-goût à ce qu'un programme peut faire en matière d'aide aux élèves en leur expliquant comment résoudre l'équation avec lequel ils ont des difficultés. Après ce travail de maturité, je pense qu'il serait peut-être intéressant de continuer à développer ce genre de logiciel et, s'il y a suffisamment de personnes qui trouvent BleuSolve utile, je me ferais un plaisir de l'améliorer durant mon temps-libre.

Et pour finir, je tiens à remercier toutes les personnes qui m'ont aidé à développer BleuSolve (leur nom figure dans les remerciements). Grâce à leur aide, j'ai pu corriger plusieurs problèmes que je ne m'en serais pas aperçu.

## 7. Annexes

### 7.1. Document de motivation et de présentation du sujet choisi

Voici le document de motivation et les prévisions de ce que contiendra le programme fini.

#### 7.1.1. Document de motivation

Après avoir aidé à de nombreuses reprises des élèves de première année du gymnase en mathématiques, j'ai remarqué que certains sujets qu'on traite en mathématiques standard sont réellement ennuyants car les tâches qu'on effectue sont très répétitives (par exemple : la factorisation, les résolutions d'équations, etc...). On pourrait donc créer un programme qui effectuerait ces tâches pour nous servir de contrôle. Mais ces programmes existent déjà dans les calculatrices programmables et dans les ordinateurs. Le problème, c'est que ces programmes nous donnent seulement la solution recherchée sans nous aider à comprendre comment on pourrait y arriver. Ils servent donc uniquement de moyen de contrôle des solutions qu'on aurait déjà trouvées auparavant.

Mais imaginons un (ou une) élève qui ne saurait pas comment résoudre une équation donnée, comment s'y prendrait-il (ou elle) ? Demanderait-il (ou elle) de l'aide à ses grands frères et/ou sœurs ? Ou à un autre élève qui aurait réussi ? Mais ne serait-il pas plus pratique s'il (ou elle) pouvait comprendre comment résoudre cette équation tout(e) seul(e) ? C'est pour aider cet(te) élève que j'aimerais créer un programme qui résoudrait des équations tout en indiquant comment on pourrait y arriver. Ce programme ne sera donc pas seulement un simple solveur d'équation, mais aussi une aide importante et utile à l'apprentissage des différentes méthodes de résolution d'équations. Il sera donc comme un professeur de mathématique qui expliquerait à l'élève comment résoudre une équation.

#### 7.1.2. Présentation du sujet

Après avoir entré une ou plusieurs équation(s) et une ou plusieurs variable(s), le programme pourra résoudre :

- les équations du premier degré à une inconnue
- les équations du deuxième degré à une inconnue
- les systèmes de  $n$  équations du premier degré à  $n$  inconnues
- les systèmes simples de  $n$  équations du deuxième degré à  $n$  inconnues

Il pourra aussi notamment :

- sortir une variable d'une équation
- factoriser une équation
- développer une équation
- simplifier une équation

Et si cela s'avère possible, le programme pourra faire tout cela dans l'ensemble des nombres complexe simple, sinon, il se contentera de l'ensemble des nombres réels.

Il résoudra toutes ces équations grâce notamment à :

- la formule de résolution d'équation du deuxième degré
- la méthode par substitution
- la méthode de la soustraction
- la mise en évidence
- la méthode des produits croisés
- etc...

Les résultats comprendront :

- Le(s) résultat(s) demandé(s) (pour autant qu'il y en ait au moins un)
- Chaque étape de calcul que le programme a effectuée
- une brève explication de la méthode utilisée pour chaque étape

Et les résultats seront affichés sous forme fractionnelle (tant que cela est nécessaire et/ou possible).

Ceci n'est, bien entendu, qu'une base sur laquelle je vais développer mon programme. J'adapterai le programme en fonction de mes connaissances en mathématiques et en programmation, des limites de la programmation java et de mon temps à disposition.

## 7.2. Fichiers PDF issus de la théorie

Voici tous les fichiers PDF accessibles depuis le menu « Théorie » de BleuSolve.

### 7.2.1. Démonstration de la formule de résolution des équations du deuxième degré à une inconnue

<p style="text-align: center;"><i>Soit <math>a \cdot x^2 + b \cdot x + c = 0</math></i></p> <p style="text-align: center;"><i><math>a, b, c</math> étant connus et <math>a \neq 0</math></i></p> $x = \frac{-b \pm \sqrt{b^2 - 4 \cdot a \cdot c}}{2 \cdot a}$
--

#### 7.2.1.1. Démonstration :

$$a \cdot x^2 + b \cdot x + c = 0$$

On soustrait « c » des deux côtés de l'équation :

$$a \cdot x^2 + b \cdot x = -c$$

On divise tout par « a » (« a » étant différent de 0, car on ne peut pas diviser par 0) :

$$x^2 + \frac{b}{a} \cdot x = -\frac{c}{a}$$

#### 7.2.1.2. Intermède :

Notons que «  $x^2 + \frac{b}{a} \cdot x$  » ressemble au début d'une formule remarquable de type :

$$A^2 + 2 \cdot A \cdot B + B^2 = (A + B)^2$$

Il faut donc trouver la partie manquante :  $B^2$

Si on fait correspondre les deux équations :

$$x^2 = A^2$$

$$\frac{b}{a} \cdot x = 2 \cdot A \cdot B \rightarrow x = \frac{2 \cdot A \cdot B \cdot a}{b}$$

Il ne nous reste donc plus qu'à trouver «  $B^2$  » :

$$x^2 = A^2 \text{ sachant que } x^2 = \left( \frac{2 \cdot A \cdot B \cdot a}{b} \right)^2$$

$$\frac{2^2 \cdot A^2 \cdot B^2 \cdot a^2}{b^2} = A^2$$

$$B^2 = \frac{b^2}{4 \cdot a^2}$$

### 7.2.1.3. Suite de la démonstration :

On va donc additionner «  $\frac{b^2}{4 \cdot a^2}$  » des deux côtés de l'équation pour garder l'égalité :

$$x^2 + \frac{b}{a} \cdot x + \frac{b^2}{4 \cdot a^2} = -\frac{c}{a} + \frac{b^2}{4 \cdot a^2}$$

Il faut maintenant factoriser la partie de gauche de l'équation grâce à la formule remarquable vu précédemment :

$$\left( x + \frac{b}{2 \cdot a} \right)^2 = \frac{b^2}{4 \cdot a^2} - \frac{c}{a}$$

Et pour simplifier la partie de droite, il faut d'abord tout mettre au même dénominateur :

$$\left( x + \frac{b}{2 \cdot a} \right)^2 = \frac{b^2}{4 \cdot a^2} - \frac{4 \cdot a \cdot c}{4 \cdot a^2}$$

$$\left( x + \frac{b}{2 \cdot a} \right)^2 = \frac{b^2 - 4 \cdot a \cdot c}{4 \cdot a^2}$$

Il faut mettre le tout à la racine carrée :

$$x + \frac{b}{2 \cdot a} = \pm \sqrt{\frac{b^2 - 4 \cdot a \cdot c}{4 \cdot a^2}}$$

*Note : On met le «  $\pm$  », car un nombre et son inverse, tous deux mis au carré, donneront le même résultat.*

$$x + \frac{b}{2 \cdot a} = \pm \frac{\sqrt{b^2 - 4 \cdot a \cdot c}}{2 \cdot a}$$

Pour finir, on isole «  $x$  » :

$$x = -\frac{b}{2 \cdot a} \pm \frac{\sqrt{b^2 - 4 \cdot a \cdot c}}{2 \cdot a}$$

Et on trouve le résultat tant attendu :

$$x = \frac{-b \pm \sqrt{b^2 - 4 \cdot a \cdot c}}{2 \cdot a}$$

■

### 7.2.2. Formule de Cramer

La « Formule de Cramer (Gabriel) » (connu aussi sous le nom de « Règle de Cramer » ou « méthode de Cramer ») permet de résoudre des systèmes d'équations :

Nous allons voir ici comment utiliser la formule de Cramer dans un système de deux équations à deux inconnues :

Dans le système suivant :

$$\begin{cases} a \cdot x + b \cdot y = c \\ d \cdot x + e \cdot y = f \end{cases}$$

$$x = \frac{\begin{vmatrix} c & b \\ f & e \end{vmatrix}}{\begin{vmatrix} a & b \\ d & e \end{vmatrix}} = \frac{c \cdot e - f \cdot b}{a \cdot e - d \cdot b}$$

$$y = \frac{\begin{vmatrix} a & c \\ d & f \end{vmatrix}}{\begin{vmatrix} a & b \\ d & e \end{vmatrix}} = \frac{a \cdot f - d \cdot c}{a \cdot e - d \cdot b}$$

On peut donc aussi en conclure que, quand  $\begin{vmatrix} a & b \\ d & e \end{vmatrix}$  est égal à zéro, il n'y a pas de solution, puisqu'on ne peut pas diviser par zéro.

### 7.2.3. Méthode de Horner

La « méthode de Horner (William George) » (connu aussi sous le nom de « règle de Ruffini (Paolo) » ou d'« algorithme de Ruffini-Horner ») est une alternative à la division euclidienne qui est utilisée dans le cas où le diviseur<sup>9</sup> est un binôme unitaire<sup>10</sup>.

Voici l'explication de la méthode de Horner par le biais d'un exemple :

Prenons le polynôme :  $2 \cdot x^3 - 44 \cdot x + 5$

Et divisons le par :  $x + 5$

Écrivons les facteurs multiplicatifs de chaque monôme du dividende<sup>11</sup> à la suite, dans l'ordre et sans oublier les « trous » :

<sup>9</sup> Diviseur : dans «  $a/b = c + \text{reste}$  », «  $b$  » est le **diviseur**.

<sup>10</sup> Binôme unitaire : polynôme dont le degré maximal est de 1 et dont le monôme de degré 1 n'a pas de facteur multiplicatif. Exemple : «  $x + 5$  » est un binôme unitaire, mais pas «  $3 \cdot x + 5$  », ni «  $x^2 + x + 5$  ».

$$2 \qquad 0 \qquad -44 \qquad 5$$

Calculons le zéro du diviseur :  $x + 5 = 0 \rightarrow x = -5$

Et notons-le dans la marge :

$$2 \qquad 0 \qquad -44 \qquad 5$$

-5

Maintenant, descendons le premier chiffre « 2 » et multiplions-le par « -5 » :

$$\begin{array}{r}
 2 \qquad 0 \qquad -44 \qquad 5 \\
 -5 \\
 \hline
 2 \qquad -10
 \end{array}$$

Dans la colonne du « 0 », on additionne « 0 » à « -10 » et on écrit le résultat dans la dernière ligne :

$$\begin{array}{r}
 2 \qquad 0 \qquad -44 \qquad 5 \\
 -5 \\
 \hline
 2 \qquad -10
 \end{array}$$

Et on refait les mêmes opérations jusqu'à la fin :

$$\begin{array}{r}
 2 \qquad 0 \qquad -44 \qquad 5 \\
 -5 \\
 \hline
 2 \qquad -10 \qquad 6 \qquad -25
 \end{array}$$

Interprétons maintenant la dernière ligne comme suit :

Le premier nombre est le facteur multiplicatif du premier monôme de la réponse qui est de degré « n-1 », sachant que « n » est le degré maximal du dividende.

<sup>11</sup> Dividende : dans «  $a/b = c + \text{reste}$  », « a » est le **dividende**.

Le deuxième nombre est le facteur multiplicatif du deuxième monôme de la réponse qui a un degré de moins que le premier terme.

Et ainsi de suite jusqu'au dernier nombre qui est tout simplement le reste de la division.

Comme les explications sont peu claires, on vous laisse découvrir le résultat :

Le résultat de la division est :  $2 \cdot x^2 - 10 \cdot x + 6$

Et le reste est de :  $-25$

Vous pouvez vérifier que le résultat est le même qu'une division euclidienne, mais cette méthode de Horner est beaucoup plus rapide et vous avez moins de risque de faire des erreurs de calculs.

### 7.2.3.1. Exemple :

Dividende :  $3 \cdot x^4 - 50 \cdot x^2 + x$

Diviseur :  $x - 4$

	<b>3</b>	<b>0</b>	<b>-50</b>	<b>1</b>	<b>0</b>
<b>4</b>		<b>12</b>	<b>48</b>	<b>-8</b>	<b>-28</b>
	<b>3</b>	<b>12</b>	<b>-2</b>	<b>-7</b>	<b>-28</b>

Quotient<sup>12</sup> :  $3 \cdot x^3 + 12 \cdot x^2 - 2 \cdot x - 7$

Reste :  $-28$

### 7.2.4. Règle des signes

Quand vous résolvez des équations avec BleuSolve, il arrive que vous vous retrouviez avec deux signes à la suite. Rassurez-vous, ce n'est pas une faute de frappe, mais ceci a bel et bien été fait exprès, pour des raisons de programmation.

On va maintenant vous expliquer comment interpréter ces doubles-signes :

Ce que vous verrez dans les calculs	L'interprétation que vous devrez en faire	Moyen mnémotechnique
++	+	Les <b>amis</b> de mes <b>amis</b> sont mes <b>amis</b>
+-	-	Les <b>amis</b> de mes <b>ennemis</b> sont mes <b>ennemis</b>
-+	-	Les <b>ennemis</b> de mes <b>amis</b> sont mes <b>ennemis</b>
--	+	Les <b>ennemis</b> de mes <b>ennemis</b> sont mes <b>amis</b>

<sup>12</sup> Quotient : dans «  $a/b = c + \text{reste}$  », «  $c$  » est le **quotient**.

En règle générale :

- quand il y a un nombre **impair** de signe **négatif**, le résultat sera **négatif**
- et quand il y a un nombre **pair** de signe **négatif**, le résultat sera **positif**

Vous pourrez bien sûr utiliser cette méthode quand vous vous retrouvez avec des puissances de nombre négatif.

Par exemple :

$$(-1)^{18} = 1$$

$$(-1)^{1291} = -1$$

$$(-2)^3 = -8$$

$$(-4)^2 = 16$$

### 7.2.5. Les nombres complexes

Voici un tout petit peu de théorie concernant les nombres complexes qui vous serviront peut-être à comprendre certains résultats proposés par le programme « BleuSolve ».

#### 7.2.5.1. Forme algébrique

La forme algébrique d'un nombre complexe est :

$$a + b \cdot i$$

Elle est formée d'une partie réelle «  $a$  » et d'une partie imaginaire «  $b$  ».

#### 7.2.5.2. Forme polaire

On peut représenter tous les nombres complexes par un point sur un système d'axes en deux dimensions. L'abscisse correspond à la partie réelle et l'ordonnée à la partie imaginaire.

On définit que le module «  $r$  » d'un nombre vaut la distance qui sépare le point à l'origine (cette distance vaut aussi la norme du vecteur qui va de l'origine au point ou inversement).

$$r = \sqrt{a^2 + b^2} \geq 0$$

On définit aussi l'argument «  $\theta$  » qui vaut l'angle entre la droite qui relie l'origine au point et l'axe positif des abscisses.

La forme polaire d'un nombre complexe est définie par le couple suivant :

$$[r; \theta]$$

Mais il existe une infinité d'arguments pour un nombre, car on peut ajouter «  $k$  » tour(s) complet(s) à l'argument sans que la représentation graphique de l'angle change :

$$[r; \theta + k \cdot 2 \cdot \pi] \quad | \quad k \in \mathbb{Z}$$

Voici un exemple de représentation graphique issu de Wikipédia<sup>13</sup> :

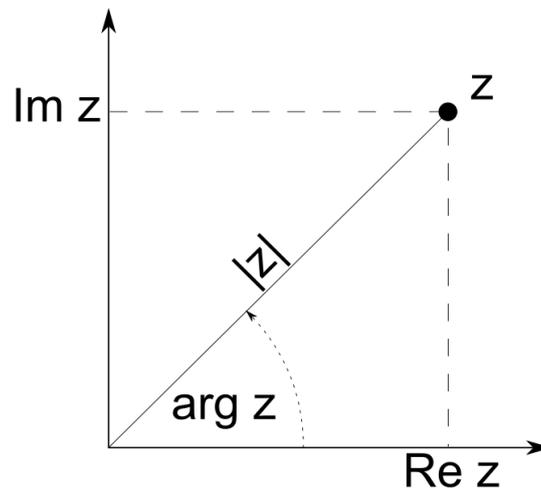


Image 32 : Représentation graphique d'un nombre complexe

$$z = (a; b)$$

$$\text{Re } z = \text{partie réelle} = a$$

$$\text{Im } z = \text{partie imaginaire} = b$$

$$|z| = \text{module} = r$$

$$\arg z = \text{argument} = \theta$$

#### 7.2.5.2.1. Quelques propriétés

$$[r; \theta] \cdot [r'; \theta'] = [r \cdot r'; \theta + \theta']$$

$$\frac{[r; \theta]}{[r'; \theta']} = \left[ \frac{r}{r'}; \theta - \theta' \right]$$

#### 7.2.5.3. Nombre complexe à la puissance $n$

Il existe une formule pour calculer un nombre complexe à la puissance  $n$  et nous allons la démontrer par récurrence :

##### 7.2.5.3.1. Théorème

$$[r; \theta]^n = [r^n; n \cdot \theta] \quad | \quad n \in \mathbb{N}$$

##### 7.2.5.3.2. Démonstration

La démonstration par récurrence consiste à vérifier qu'un théorème est vrai pour le premier cas ( $n = 1$ ), puis de montrer qu'il est aussi vrai pour le cas suivant ( $n + 1$ ). Si c'est le cas, la formule est démontrée, puisqu'elle est vérifiée pour tous les nombres. Cette méthode de démonstration s'apparente aux dominos : Si on fait tomber le premier domino et si on est sûr que chaque domino va faire tomber le suivant, tous les dominos tomberont.

##### 7.2.5.3.2.1. Amorce

$n = 1$  :

<sup>13</sup> [http://fr.wikipedia.org/wiki/Nombre\\_complex](http://fr.wikipedia.org/wiki/Nombre_complex)

$$[r; \theta]^1 = [r^1; 1 \cdot \theta]$$

$$[r; \theta] = [r; \theta]$$

Donc la formule est vérifiée pour  $n = 1$ .

#### 7.2.5.3.2.2. Hérité

On suppose que le théorème est vrai pour  $n$  :

$$[[r; \theta]^n = [r^n; n \cdot \theta]]$$

On veut montrer qu'il est aussi vrai pour  $n + 1$  :

$$[[r; \theta]^{n+1} = [r^{n+1}; (n + 1) \cdot \theta]]$$

$$[r; \theta]^{n+1} = [r; \theta] \cdot [r; \theta]^n = [r; \theta] \cdot [r^n; n \cdot \theta] = [r^{n+1}; (n + 1) \cdot \theta]$$

■

Maintenant qu'on a démontré le théorème pour  $n \in \mathbb{N}$ , on va essayer de l'appliquer dans  $\mathbb{Z}$  :

#### 7.2.5.3.3. Théorème

Si la formule  $[r; \theta]^n = [r^n; n \cdot \theta]$  est vraie quand  $n \in \mathbb{Z}$ , alors cette formule devrait aussi être vérifiée :

$$[r; \theta]^{-n} = [r^{-n}; -n \cdot \theta] \quad | \quad n \in \mathbb{N}$$

#### 7.2.5.3.4. Démonstration

$$[r; \theta]^{-n} = \frac{1}{[r; \theta]^n} = \frac{1}{[r^n; n \cdot \theta]} = \frac{[1; 0]}{[r^n; n \cdot \theta]} = \left[ \frac{1}{r^n}; 0 - n \cdot \theta \right] = [r^{-n}; -n \cdot \theta]$$

Donc

$$[r; \theta]^n = [r^n; n \cdot \theta] \quad | \quad n \in \mathbb{Z}$$

■

## 7.3. Code Java

### 7.3.1. Launcher.java

```
// Fichier : BleuSolve
// Date de commencement : 2 juin 2011
// Auteur : Tinh Di David Tang

public class Launcher {

    public static void main(String[] args) {

        // Creation d'un objet de la classe "SplashScreen"
        SplashScreen splash = new SplashScreen();

        // Affichage de l'image d'accueil
        splash.setVisible(true);

        // Creation d'un objet de la classe "Window"
        Window wind = new Window();

        // Le programme ne fait rien pour que l'image s'affiche durant un
temps minimum
        try {
            Thread.sleep(1291);
        } catch (InterruptedException e) {
            e.printStackTrace();
        } // catch (InterruptedException e)

        // Destruction de l'objet "splash"
        splash.dispose();

        // Affichage de la fenetre principale
        wind.setVisible(true);

    } // public static void main(String[] args)
} // public class Launcher
```

### 7.3.2. SplashScreen.java

```
import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.Font;

import javax.swing.ImageIcon;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;

public class SplashScreen extends JFrame {

    private static final long serialVersionUID = 1L;

    // Creation des elements
    JPanel panneau = new JPanel();
    JPanel panneauCentre = new JPanel();
    JPanel panneauSud = new JPanel();
    JLabel etiquetteImage = new JLabel(new
ImageIcon(this.getToolkit().createImage(getClass().getResource("/images/Logo.png")
)));
    JLabel etiquetteInfo = new JLabel("Travail de maturite au Gymnase d'Yverdon
de Tinh Di David Tang en 2011");

    public SplashScreen() {

        // Parametrage de la fenetre
        setSize(450, 450);
        setUndecorated(true);
        setIconImage(new
ImageIcon(getClass().getResource("/images/Logo.png")).getImage());
        setTitle("BleuSolve");
        setLocationRelativeTo(null);
        setContentPane(panneau);
        panneau.add(panneauCentre, BorderLayout.CENTER);
        panneau.add(panneauSud, BorderLayout.SOUTH);
        panneauCentre.add(etiquetteImage);
        panneauSud.add(etiquetteInfo);
        etiquetteInfo.setForeground(Color.BLUE);
        etiquetteInfo.setFont(new Font("Calibri", Font.BOLD, 14));

    } // public SplashScreen()

} // public class SplashScreen extends Frame
```

### 7.3.3. Window.java

```

import java.awt.BorderLayout;
import java.awt.Desktop;
import java.awt.Dimension;
import java.awt.FlowLayout;
import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.WindowAdapter;
import java.awt.event.WindowEvent;
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.net.URI;
import java.net.URISyntaxException;

import javax.swing.BorderFactory;
import javax.swing.ImageIcon;
import javax.swing.JButton;
import javax.swing.JComboBox;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JMenu;
import javax.swing.JMenuBar;
import javax.swing.JMenuItem;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;
import javax.swing.JTextField;

class Window extends JFrame {

    private static final long serialVersionUID = 1L;

    // Creation des elements

    // Creation des Panneaux
    JPanel panneau = new JPanel();
    JScrollPane panneauDefil = new JScrollPane(panneau);
    JPanel panneauMePrNord = new JPanel();
    JPanel panneauMePrCentre = new JPanel();
    JPanel panneauSud = new JPanel();
    JPanel panneauMeEqNord = new JPanel();
    JPanel panneauMeEqCentre = new JPanel();
    JPanel panneauReEqNord = new JPanel();
    JPanel panneauReEq1Centre = new JPanel();
    JPanel panneauReEq1CentreNord = new JPanel();
    JPanel panneauReEq1CentreCentre = new JPanel();
    JScrollPane panneauDefilReEq1CentreCentre = new
JScrollPane(panneauReEq1CentreCentre);
    JPanel panneauReEqCentreSud = new JPanel();
    JPanel panneauReEq2Centre = new JPanel();
    JPanel panneauReEq2CentreNord = new JPanel();
    JPanel panneauReEq2CentreCentre = new JPanel();
    JScrollPane panneauDefilReEq2CentreCentre = new
JScrollPane(panneauReEq2CentreCentre);

```

```

    JPanel panneauReEq3Centre = new JPanel();
    JPanel panneauReEq3CentreNord = new JPanel();
    JPanel panneauReEq3CentreCentre = new JPanel();
    JScrollPane panneauDefilReEq3CentreCentre = new
JScrollPane(panneauReEq3CentreCentre);
    JPanel panneauMeSyNord = new JPanel();
    JPanel panneauMeSyCentre = new JPanel();
    JPanel panneauReSyNord = new JPanel();
    JPanel panneauReSy2Centre = new JPanel();
    JPanel panneauReSy2CentreNord = new JPanel();
    JPanel panneauReSy2CentreCentre = new JPanel();
    JScrollPane panneauDefilReSy2CentreCentre = new
JScrollPane(panneauReSy2CentreCentre);
    JPanel panneauReSyCentreSud = new JPanel();
    JPanel panneauMeCoNord = new JPanel();
    JPanel panneauMeCoCentre = new JPanel();
    JPanel panneauReCoNord = new JPanel();
    JPanel panneauReCo0Centre = new JPanel();
    JPanel panneauReCo0CentreNord = new JPanel();
    JPanel panneauReCo0CentreCentre = new JPanel();
    JScrollPane panneauDefilReCo0CentreCentre = new
JScrollPane(panneauReCo0CentreCentre);
    JPanel panneauReCoCentreSud = new JPanel();
    JPanel panneauReCo1Centre = new JPanel();
    JPanel panneauReCo1CentreNord = new JPanel();
    JPanel panneauReCo1CentreCentre = new JPanel();
    JScrollPane panneauDefilReCo1CentreCentre = new
JScrollPane(panneauReCo1CentreCentre);
    JPanel panneauMeDiNord = new JPanel();
    JPanel panneauMeDiCentre = new JPanel();
    JPanel panneauReDiNord = new JPanel();
    JPanel panneauReDi1Centre = new JPanel();
    JPanel panneauReDi1CentreNord = new JPanel();
    JPanel panneauReDi1CentreCentre = new JPanel();
    JScrollPane panneauDefilReDi1CentreCentre = new
JScrollPane(panneauReDi1CentreCentre);
    JPanel panneauReDiCentreSud = new JPanel();
    JPanel panneauMeExNord = new JPanel();
    JPanel panneauMeExCentre = new JPanel();
    JPanel panneauReExNord = new JPanel();
    JPanel panneauReEx1Centre = new JPanel();
    JPanel panneauReEx1CentreNord = new JPanel();
    JPanel panneauReEx1CentreCentre = new JPanel();
    JScrollPane panneauDefilReEx1CentreCentre = new
JScrollPane(panneauReEx1CentreCentre);
    JPanel panneauReExCentreSud = new JPanel();

    // Creation des messages d'instruction
    JLabel messageMePr1 = new JLabel("Bonjour et bienvenue dans BleuSolve !",
JLabel.CENTER);
    JLabel messageMePr2 = new JLabel("Ce programme va vous aider a resoudre des
equations tout en vous expliquant comment parvenir au resultat.", JLabel.CENTER);
    JLabel messageMePr3 = new JLabel("Veuillez choisir l'une des activites
proposees ci-dessous :", JLabel.CENTER);
    JLabel messageMeEq = new JLabel("Veuillez choisir le type d'equation a
resoudre :");
    JLabel messageReEq = new JLabel("Veuillez completer l'equation a resoudre
:");

```

```
JLabel messageMeSy = new JLabel("Veuillez choisir le type de systeme  
d'equations a resoudre :");  
JLabel messageReSy = new JLabel("Veuillez completer les equations a resoudre  
:");  
JLabel messageMeCo = new JLabel("Veuillez choisir le type d'equation a  
resoudre :");  
JLabel messageReCo = new JLabel("Veuillez completer l'equation a resoudre  
:");  
JLabel messageMeDi = new JLabel("Veuillez choisir l'activite que vous  
souhaitez faire :");  
JLabel messageReDi = new JLabel("Veuillez completer les champs de texte :");  
JLabel messageMeEx = new JLabel("Veuillez choisir l'activite que vous  
souhaitez faire :");  
JLabel messageReEx = new JLabel("Veuillez selectionner vos notes :");  
  
// Creation de la notice  
JLabel messageNotice = new JLabel("Si vous rencontrez un quelconque  
probleme, veuillez consulter la theorie ou l'aide qui se trouvent dans la barre de  
menu.");  
  
// Creation de la barre de menu  
JMenuBar barreDeMenu = new JMenuBar();  
JMenu menuFichier = new JMenu("Fichier");  
JMenu menuTheorie = new JMenu("Theorie");  
JMenu menuAllerA = new JMenu("Aller a");  
JMenu sousMenuMeEq = new JMenu("Resoudre une equation a une inconnue");  
JMenu sousMenuMeSy = new JMenu("Resoudre un systeme d'equations a plusieurs  
inconnues");  
JMenu sousMenuMeCo = new JMenu("Nombre complexe");  
JMenu sousMenuMeDi = new JMenu("Divers");  
JMenu sousMenuMeEx = new JMenu("Extras");  
JMenu menuAide = new JMenu("Aide");  
JMenuItem objetMenuEffacerPDF = new JMenuItem("Effacer tous les fichiers PDF  
crees");  
JMenuItem objetMenuRetourMenuPrincipal = new JMenuItem("Retourner au menu  
principal");  
JMenuItem objetMenuQuitter = new JMenuItem("Quitter BleuSolve");  
JMenuItem objetMenuFormuleReEq2 = new JMenuItem("Demonstration de la formule  
de resolution des equations du deuxieme degre a une inconnue");  
JMenuItem objetMenuCramer = new JMenuItem("Formule de Cramer");  
JMenuItem objetMenuHorner = new JMenuItem("Methode de Horner");  
JMenuItem objetMenuSignes = new JMenuItem("Regle des signes");  
JMenuItem objetMenuComplexe = new JMenuItem("Theorie sur les nombres  
complexes");  
JMenuItem objetSousMenuReEq1 = new JMenuItem("... du premier degre");  
JMenuItem objetSousMenuReEq2 = new JMenuItem("... du deuxieme degre");  
JMenuItem objetSousMenuReEq3 = new JMenuItem("... du troisieme degre");  
JMenuItem objetSousMenuReSy2 = new JMenuItem("avec deux inconnues");  
JMenuItem objetSousMenuReCo0 = new JMenuItem("Racine carree");  
JMenuItem objetSousMenuReCo1 = new JMenuItem("Equation du premier degre a  
une inconnue");  
JMenuItem objetSousMenuReDi1 = new JMenuItem("Division polynomiale  
(Horner)");  
JMenuItem objetSousMenuReEx1 = new JMenuItem("Calculateur de moyenne");  
JMenuItem objetMenuManuel = new JMenuItem("Manuel de l'utilisateur");  
JMenuItem objetMenuContact = new JMenuItem("Contacter le developpeur");  
JMenuItem objetMenuBlog = new JMenuItem("Blog de BleuSolve");  
JMenuItem objetMenuPageFacebook = new JMenuItem("Page Facebook de  
BleuSolve");
```

```
JMenuItem objetMenuAPropos = new JMenuItem("A propos de BleuSolve");

// Creation des liens vers les fichiers PDF dans l'archive
InputStream pdfDansLArchiveReEq2 =
getClass().getClassLoader().getResourceAsStream("pdf/ReEq2.pdf");
InputStream pdfDansLArchiveCramer =
getClass().getClassLoader().getResourceAsStream("pdf/Cramer.pdf");
InputStream pdfDansLArchiveHorner =
getClass().getClassLoader().getResourceAsStream("pdf/Horner.pdf");
InputStream pdfDansLArchiveSignes =
getClass().getClassLoader().getResourceAsStream("pdf/Signes.pdf");
InputStream pdfDansLArchiveManuel =
getClass().getClassLoader().getResourceAsStream("pdf/Manuel.pdf");
InputStream pdfDansLArchiveComplexe =
getClass().getClassLoader().getResourceAsStream("pdf/Complexe.pdf");

// Creation des fichiers PDF
File pdfCreeReEq2 = new File("BleuSolve - Demonstration de la formule de
resolution des equations du deuxieme degre a une inconnue.pdf");
File pdfCreeCramer = new File("BleuSolve - Formule de Cramer.pdf");
File pdfCreeHorner = new File("BleuSolve - Methode de Horner.pdf");
File pdfCreeSignes = new File("BleuSolve - Regle des signes.pdf");
File pdfCreeManuel = new File("BleuSolve - Manuel de l'utilisateur.pdf");
File pdfCreeComplexe = new File("BleuSolve - Theorie sur les nombres
complexes.pdf");

// Creation des boutons
JButton boutonMePr1 = new JButton("Resoudre une equation a une inconnue");
JButton boutonMePr2 = new JButton("Resoudre un systeme de plusieurs
equations a plusieurs inconnues.");
JButton boutonMePr3 = new JButton("Nombre complexe");
JButton boutonMePr8 = new JButton("Divers");
JButton boutonMePr9 = new JButton("Extras");
JButton boutonMeEq1 = new JButton("Resoudre une equation du premier degre a
une inconnue");
JButton boutonMeEq2 = new JButton("Resoudre une equation du deuxieme degre a
une inconnue");
JButton boutonMeEq3 = new JButton("Resoudre une equation du troisieme degre
a une inconnue (avec une solution entiere comprise entre -10 et 10)");
JButton boutonMeSy2 = new JButton("Resoudre un systeme de deux equations a
deux inconnues");
JButton boutonMeCo0 = new JButton("Calculer les racines carrees d'un nombre
complexe");
JButton boutonMeCo1 = new JButton("Resoudre une equation du premier degre a
une inconnue");
JButton boutonMeDi1 = new JButton("Division polynomiale (avec la methode de
Horner)");
JButton boutonMeEx1 = new JButton("Calculateur de moyenne");

JButton boutonMenuPrincipalMeEq = new JButton("Retourner au menu
principal");
JButton boutonMenuPrincipalReEq = new JButton("Retourner au menu
principal");
JButton boutonMenuPrincipalMeSy = new JButton("Retourner au menu
principal");
JButton boutonMenuPrincipalReSy = new JButton("Retourner au menu
principal");
JButton boutonMenuPrincipalMeCo = new JButton("Retourner au menu
principal");
```

```
        JButton boutonMenuPrincipalReCo = new JButton("Retourner au menu principal");
        JButton boutonMenuPrincipalMeDi = new JButton("Retourner au menu principal");
        JButton boutonMenuPrincipalReDi = new JButton("Retourner au menu principal");
        JButton boutonMenuPrincipalMeEx = new JButton("Retourner au menu principal");
        JButton boutonMenuPrincipalReEx = new JButton("Retourner au menu principal");

        JButton boutonMenuEquation = new JButton("Resoudre un autre type d'equation");
        JButton boutonMenuSysteme = new JButton("Resoudre un autre type de systeme");
        JButton boutonMenuComplexe = new JButton("Resoudre un autre type d'equation");
        JButton boutonMenuDivers = new JButton("Faire une autre activite");
        JButton boutonMenuExtras = new JButton("Faire une autre activite");

        JButton boutonQuitterMePr = new JButton("Quitter BleuSolve");
        JButton boutonQuitterMeEq = new JButton("Quitter BleuSolve");
        JButton boutonQuitterMeSy = new JButton("Quitter BleuSolve");
        JButton boutonQuitterMeCo = new JButton("Quitter BleuSolve");
        JButton boutonQuitterMeDi = new JButton("Quitter BleuSolve");
        JButton boutonQuitterMeEx = new JButton("Quitter BleuSolve");

        // Creation des boutons "Resoudre"
        JButton boutonResoudreReEq1 = new JButton("Resoudre");
        JButton boutonResoudreReEq2 = new JButton("Resoudre");
        JButton boutonResoudreReEq3 = new JButton("Resoudre");
        JButton boutonResoudreReSy2 = new JButton("Resoudre");
        JButton boutonResoudreReCo0 = new JButton("Resoudre");
        JButton boutonResoudreReCo1 = new JButton("Resoudre");
        JButton boutonResoudreReDi1 = new JButton("Calculer");
        JButton boutonResoudreReEx1 = new JButton("Calculer");

        // Creation des boutons "Effacer"
        JButton boutonEffacerReEq1 = new JButton("Effacer");
        JButton boutonEffacerReEq2 = new JButton("Effacer");
        JButton boutonEffacerReEq3 = new JButton("Effacer");
        JButton boutonEffacerReSy2 = new JButton("Effacer");
        JButton boutonEffacerReCo0 = new JButton("Effacer");
        JButton boutonEffacerReCo1 = new JButton("Effacer");
        JButton boutonEffacerReDi1 = new JButton("Effacer");
        JButton boutonEffacerReEx1 = new JButton("Effacer");

        // Creation des variables
        double a;
        double b;
        double c;
        double d;
        double e;
        double f;
        double g;
        double h;
        double r;
        double t;
        double a1;
```

```

double b1;
double c1;
double a2;
double b2;
double c2;
int x;
int nombreNotes;
double sommeNotes;
double moyenneArrondie;

// Creation des champs de texte
JTextField champReEq1_A = new JTextField(3);
JTextField champReEq1_B = new JTextField(3);
JTextField champReEq2_A = new JTextField(3);
JTextField champReEq2_B = new JTextField(3);
JTextField champReEq2_C = new JTextField(3);
JTextField champReEq3_A = new JTextField(3);
JTextField champReEq3_B = new JTextField(3);
JTextField champReEq3_C = new JTextField(3);
JTextField champReEq3_D = new JTextField(3);
JTextField champReSy2_A = new JTextField(3);
JTextField champReSy2_B = new JTextField(3);
JTextField champReSy2_C = new JTextField(3);
JTextField champReSy2_D = new JTextField(3);
JTextField champReSy2_E = new JTextField(3);
JTextField champReSy2_F = new JTextField(3);
JTextField champReCo0_A = new JTextField(3);
JTextField champReCo0_B = new JTextField(3);
JTextField champReCo1_A = new JTextField(3);
JTextField champReCo1_B = new JTextField(3);
JTextField champReCo1_C = new JTextField(3);
JTextField champReCo1_D = new JTextField(3);
JTextField champReDi1_A = new JTextField(3);
JTextField champReDi1_B = new JTextField(3);
JTextField champReDi1_C = new JTextField(3);
JTextField champReDi1_D = new JTextField(3);
JTextField champReDi1_E = new JTextField(3);
JTextField champReDi1_F = new JTextField(3);
JTextField champReDi1_G = new JTextField(3);
JTextField champReDi1_H = new JTextField(3);

// Creation des etiquettes
JLabel etiquetteReEq1_1 = new JLabel("\u00b7x +");
JLabel etiquetteReEq1_2 = new JLabel("= 0");
JLabel etiquetteReEq2_1 = new JLabel("\u00b7x\u00b2 +");
JLabel etiquetteReEq2_2 = new JLabel("\u00b7x +");
JLabel etiquetteReEq2_3 = new JLabel("= 0");
JLabel etiquetteReEq3_1 = new JLabel("\u00b7x\u00b3 +");
JLabel etiquetteReEq3_2 = new JLabel("\u00b7x\u00b2 +");
JLabel etiquetteReEq3_3 = new JLabel("\u00b7x +");
JLabel etiquetteReEq3_4 = new JLabel("= 0");
JLabel etiquetteReSy2_1 = new JLabel("\u2460 : ");
JLabel etiquetteReSy2_2 = new JLabel("\u00b7x +");
JLabel etiquetteReSy2_3 = new JLabel("\u00b7y = ");
JLabel etiquetteReSy2_4 = new JLabel("\u2461 : ");
JLabel etiquetteReSy2_5 = new JLabel("\u00b7x +");
JLabel etiquetteReSy2_6 = new JLabel("\u00b7y = ");
JLabel etiquetteReCo0_1 = new JLabel("x\u00b2 = ");
JLabel etiquetteReCo0_2 = new JLabel("+");

```

```

JLabel etiquetteReCo0_3 = new JLabel("\u00b7i");
JLabel etiquetteReCo1_1 = new JLabel("(");
JLabel etiquetteReCo1_2 = new JLabel("+");
JLabel etiquetteReCo1_3 = new JLabel("\u00b7i)\u00b7x +");
JLabel etiquetteReCo1_4 = new JLabel("+");
JLabel etiquetteReCo1_5 = new JLabel("\u00b7i = 0");
JLabel etiquetteReDi1_0 = new JLabel("(");
JLabel etiquetteReDi1_1 = new JLabel("\u00b7x\u2076 +");
JLabel etiquetteReDi1_2 = new JLabel("\u00b7x\u2075 +");
JLabel etiquetteReDi1_3 = new JLabel("\u00b7x\u2074 +");
JLabel etiquetteReDi1_4 = new JLabel("\u00b7x\u00b3 +");
JLabel etiquetteReDi1_5 = new JLabel("\u00b7x\u00b2 +");
JLabel etiquetteReDi1_6 = new JLabel("\u00b7x +");
JLabel etiquetteReDi1_7 = new JLabel("") : (x +);
JLabel etiquetteReDi1_8 = new JLabel(")");

// Creation des tableaux
int [] tableauReEq3 = {0, 1, -1, 2, -2, 3, -3, 4, -4, 5, -5, 6, -6, 7, -7,
8, -8, 9, -9, 10, -10};
String [] tableauReEx1 = {"--", "6", "5.5", "5", "4.5", "4", "3.5", "3",
"2.5", "2", "1.5", "1"};

// Creation des boites combo
JComboBox boiteComboReEx1_1 = new JComboBox(tableauReEx1);
JComboBox boiteComboReEx1_2 = new JComboBox(tableauReEx1);
JComboBox boiteComboReEx1_3 = new JComboBox(tableauReEx1);
JComboBox boiteComboReEx1_4 = new JComboBox(tableauReEx1);
JComboBox boiteComboReEx1_5 = new JComboBox(tableauReEx1);
JComboBox boiteComboReEx1_6 = new JComboBox(tableauReEx1);
JComboBox boiteComboReEx1_7 = new JComboBox(tableauReEx1);
JComboBox boiteComboReEx1_8 = new JComboBox(tableauReEx1);
JComboBox boiteComboReEx1_9 = new JComboBox(tableauReEx1);
JComboBox boiteComboReEx1_10 = new JComboBox(tableauReEx1);
JComboBox boiteComboReEx1_11 = new JComboBox(tableauReEx1);
JComboBox boiteComboReEx1_12 = new JComboBox(tableauReEx1);

// Creation de la zone de texte pour les resultats
JTextArea resultatReEq1Expl = new JTextArea("");
JTextArea resultatReEq1Calc = new JTextArea("");
JTextArea resultatReEq2Expl = new JTextArea("");
JTextArea resultatReEq2Calc = new JTextArea("");
JTextArea resultatReEq3Expl = new JTextArea("");
JTextArea resultatReEq3Calc = new JTextArea("");
JTextArea resultatReSy2Expl = new JTextArea("");
JTextArea resultatReSy2Calc = new JTextArea("");
JTextArea resultatReCo0Expl = new JTextArea("");
JTextArea resultatReCo0Calc = new JTextArea("");
JTextArea resultatReCo1Expl = new JTextArea("");
JTextArea resultatReCo1Calc = new JTextArea("");
JTextArea resultatReDi1 = new JTextArea("");
JTextArea resultatReEx1 = new JTextArea("");

// Creation de la chaine de caracteres pour le resultat
String caractereResultatReEq3Expl;
String caractereResultatReEq3Calc;
String caractereResultatReCo0Expl;
String caractereResultatReCo0Calc;

// Creation des mises en page

```

```

FlowLayout miseEnPageMenu = new FlowLayout(FlowLayout.CENTER, 20000, 10);
FlowLayout miseEnPageResultat = new FlowLayout(FlowLayout.CENTER, 0, 0);

public Window() {

//      Paramétrage de la fenêtre et de la barre de menu

// Parametres de la fenetre principale
setSize(800, 450);
setIconImage(new
ImageIcon(getClass().getResource("/images/Logo.png")).getImage());
setTitle("BleuSolve - Menu principal");
setLocationRelativeTo(null);
setExtendedState(MAXIMIZED_BOTH);

// Fermeture de la fenetre (croix rouge)
setDefaultCloseOperation(JFrame.DO_NOTHING_ON_CLOSE);
this.addWindowListener(new WindowAdapter() {
    public void windowClosing(WindowEvent event) {

        if (JOptionPane.showConfirmDialog(new JFrame(),
            "Etes-vous sur de vouloir quitter BleuSolve
et \n" +
            "supprimer tous les fichiers PDF crees ?",
            "BleuSolve - Quitter",
            JOptionPane.YES_NO_OPTION) ==
JOptionPane.YES_OPTION) {
                pdfCreeReEq2.delete();
                pdfCreeCramer.delete();
                pdfCreeHorner.delete();
                pdfCreeSignes.delete();
                pdfCreeComplexe.delete();
                pdfCreeManuel.delete();
                System.exit(0);
            }
        } // public void windowClosing(WindowEvent event) {
}); // this.addWindowListener(new WindowAdapter() {

// Ajout des composants a fenetre principale
setJMenuBar(barreDeMenu);
setContentPane(panneauDefil);

// Definition de la taille minimale du "panneau de fond"
panneau.setPreferredSize(new Dimension(780, 386));

// Ajout de la barre de menu
barreDeMenu.add(menuFichier);
menuFichier.add(objetMenuEffacerPDF);
menuFichier.add(objetMenuRetourMenuPrincipal);
menuFichier.add(objetMenuQuitter);
barreDeMenu.add(menuTheorie);
menuTheorie.add(objetMenuFormuleReEq2);
menuTheorie.add(objetMenuCramer);
menuTheorie.add(objetMenuHorner);
menuTheorie.add(objetMenuSignes);
menuTheorie.add(objetMenuComplexe);
barreDeMenu.add(menuAllerA);
menuAllerA.add(sousMenuMeEq);

```

```

sousMenuMeEq.add(objetSousMenuReEq1);
sousMenuMeEq.add(objetSousMenuReEq2);
sousMenuMeEq.add(objetSousMenuReEq3);
menuAllerA.add(sousMenuMeSy);
sousMenuMeSy.add(objetSousMenuReSy2);
menuAllerA.add(sousMenuMeCo);
sousMenuMeCo.add(objetSousMenuReCo0);
sousMenuMeCo.add(objetSousMenuReCo1);
menuAllerA.add(sousMenuMeDi);
sousMenuMeDi.add(objetSousMenuReDi1);
menuAllerA.add(sousMenuMeEx);
sousMenuMeEx.add(objetSousMenuReEx1);
barreDeMenu.add(menuAide);
menuAide.add(objetMenuManuel);
menuAide.add(objetMenuContact);
menuAide.add(objetMenuBlog);
menuAide.add(objetMenuPageFacebook);
menuAide.add(objetMenuAPropos);

// Definitions des actions quand on clique sur les boutons de la
barre de menu
objetMenuEffacerPDF.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {

        pdfCreeReEq2.delete();
        pdfCreeCramer.delete();
        pdfCreeHorner.delete();
        pdfCreeSignes.delete();
        pdfCreeComplexe.delete();
        pdfCreeManuel.delete();

    } // public void actionPerformed(ActionEvent arg0)
}); // objetMenuEffacerPDF.addActionListener(new ActionListener()

objetMenuRetourMenuPrincipal.addActionListener(new
EcouteurBoutonMenuPrincipal());

objetMenuQuitter.addActionListener(new EcouteurBoutonQuitter());

objetMenuFormuleReEq2.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {

        try {
            // Extraction du PDF qui se situe dans l'archive
            FileOutputStream fos = new
FileOutputStream(pdfCreeReEq2);
            while (pdfDansLArchiveReEq2.available() > 0) {
                fos.write(pdfDansLArchiveReEq2.read());
            } // while (pdfDansLArchiveReEq2.available()
> 0)

            fos.close();
            // Ouverture du PDF
            Desktop.getDesktop().open(pdfCreeReEq2);
        } catch (IOException e) {
            e.printStackTrace();
        } // catch (IOException e)

    } // public void actionPerformed(ActionEvent arg0)
}); // objetMenuFormuleReEq2.addActionListener(new ActionListener()

```

```
objetMenuCramer.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {

        try {
            // Extraction du PDF qui se situe dans l'archive
            FileOutputStream fos = new
FileOutputStream(pdfCreeCramer);
            while (pdfDansLArchiveCramer.available() > 0) {
                fos.write(pdfDansLArchiveCramer.read());
            } // while (pdfDansLArchiveCramer.available()
> 0)

            fos.close();
            // Ouverture du PDF
            Desktop.getDesktop().open(pdfCreeCramer);
        } catch (IOException e) {
            e.printStackTrace();
        } // catch (IOException e)

    } // public void actionPerformed(ActionEvent arg0)
}); // objetMenuCramer.addActionListener(new ActionListener()

objetMenuHorner.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {

        try {
            // Extraction du PDF qui se situe dans l'archive
            FileOutputStream fos = new
FileOutputStream(pdfCreeHorner);
            while (pdfDansLArchiveHorner.available() > 0) {
                fos.write(pdfDansLArchiveHorner.read());
            } // while (pdfDansLArchiveHorner.available()
> 0)

            fos.close();
            // Ouverture du PDF
            Desktop.getDesktop().open(pdfCreeHorner);
        } catch (IOException e) {
            e.printStackTrace();
        } // catch (IOException e)

    } // public void actionPerformed(ActionEvent arg0)
}); // objetMenuHorner.addActionListener(new ActionListener()

objetMenuSignes.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {

        try {
            // Extraction du PDF qui se situe dans l'archive
            FileOutputStream fos = new
FileOutputStream(pdfCreeSignes);
            while (pdfDansLArchiveSignes.available() > 0) {
                fos.write(pdfDansLArchiveSignes.read());
            } // while (pdfDansLArchiveSignes.available()
> 0)

            fos.close();
            // Ouverture du PDF
            Desktop.getDesktop().open(pdfCreeSignes);
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
});
```

```

        } // catch (IOException e)
    } // public void actionPerformed(ActionEvent arg0)
}); // objetMenuSignes.addActionListener(new ActionListener()

objetMenuComplexe.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {

        try {
            // Extraction du PDF qui se situe dans l'archive
            FileOutputStream fos = new
FileOutputStream(pdfCreeComplexe);
            while (pdfDansLArchiveComplexe.available() > 0) {
                fos.write(pdfDansLArchiveComplexe.read());
            } // while
(pdfDansLArchiveComplexe.available() > 0)
            fos.close();
            // Ouverture du PDF
            Desktop.getDesktop().open(pdfCreeComplexe);
        } catch (IOException e) {
            e.printStackTrace();
        } // catch (IOException e)

    } // public void actionPerformed(ActionEvent arg0)
}); // objetMenuSignes.addActionListener(new ActionListener()

// Affectation de l'accès rapide
objetSousMenuReEq1.addActionListener(new EcouteurBoutonReEq1());
objetSousMenuReEq2.addActionListener(new EcouteurBoutonReEq2());
objetSousMenuReEq3.addActionListener(new EcouteurBoutonReEq3());
objetSousMenuReSy2.addActionListener(new EcouteurBoutonReSy2());
objetSousMenuReCo0.addActionListener(new EcouteurBoutonReCo0());
objetSousMenuReCo1.addActionListener(new EcouteurBoutonReCo1());
objetSousMenuReDi1.addActionListener(new EcouteurBoutonReDi1());
objetSousMenuReEx1.addActionListener(new EcouteurBoutonReEx1());

objetMenuManuel.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {

        try {
            // Extraction du PDF qui se situe dans l'archive
            FileOutputStream fos = new
FileOutputStream(pdfCreeManuel);
            while (pdfDansLArchiveManuel.available() > 0) {
                fos.write(pdfDansLArchiveManuel.read());
            } // while (pdfDansLArchiveManuel.available()
> 0)

            fos.close();
            // Ouverture du PDF
            Desktop.getDesktop().open(pdfCreeManuel);
        } catch (IOException e) {
            e.printStackTrace();
        } // catch (IOException e)

    } // public void actionPerformed(ActionEvent arg0)
}); // objetMenuManuel.addActionListener(new ActionListener()

objetMenuContact.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {

```

```

        //Ouvrir le logiciel de messagerie par default
        try {
            Desktop.getDesktop().mail(new URI("mailto",
" david.tang@hotmail.ch?subject=BleuSolve", null));
        } catch (IOException e) {
            e.printStackTrace();
        } catch (URISyntaxException e) {
            e.printStackTrace();
        } // catch (URISyntaxException e)

    } // public void actionPerformed(ActionEvent arg0)
}); // objetMenuContact.addActionListener(new ActionListener()

objetMenuBlog.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {

        try {
            Desktop.getDesktop().browse(new
URI("http://bleusolve.wordpress.com/"));
        } catch (IOException e) {
            e.printStackTrace();
        } catch (URISyntaxException e) {
            e.printStackTrace();
        } // catch (URISyntaxException e) {

    } // public void actionPerformed(ActionEvent arg0)
}); // objetMenuBlog.addActionListener(new ActionListener()

objetMenuPageFacebook.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {

        try {
            Desktop.getDesktop().browse(new
URI("http://www.facebook.com/BleuSolve"));
        } catch (IOException e) {
            e.printStackTrace();
        } catch (URISyntaxException e) {
            e.printStackTrace();
        } // catch (URISyntaxException e) {

    } // public void actionPerformed(ActionEvent arg0)
}); // objetMenuPageFacebook.addActionListener(new ActionListener()

objetMenuAPropos.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {

        JOptionPane.showMessageDialog(null, "BleuSolve a ete
concu pour un Travail de Maturite en Mathematiques au Gymnase d'Yverdon en 2011.
\n" +
            "Date : du 2 juin 2011 au 19 novembre 2011
\n" +
            "Developpeur : Tinh Di David Tang \n" +
            "Version : 1.0.0 \n" +
            "\n" +
            "Note : \n" +
            "- Pour des raisons de compatibilite durant
le Travail de Maturite, tous les accents ont ete enleves. \n" +

```

```

        "- BleuSolve a ete optimise pour Windows,
il se peut qu'il y ait quelques petits problemes de compatibilite avec d'autres
systemes d'exploitation. \n" +
        "Le developpeur s'excuse prealablement pour
tous les desagregements occasionnes. \n" +
        "\n" +
        "Remerciements :\n" +
        "- a Monsieur Pierre Breguet, pour avoir
propose et suivi le theme \"Programmation en Java\" pour le travail de Maturite
\n" +
        "- a Monsieur Olivier Desponds, pour ses
conseils \n" +
        "- a David Wagnieres, pour avoir dessine le
logo actuel de BleuSolve \n" +
        "- a Guillaume Guenat, pour avoir dessine
les premiers logo et icone de BleuSolve \n" +
        "- aux Alpha-testeurs : Cedric Favre,
Guillaume Guenat, Adrien Jeckelmann, Thomas Mizraji, Lindor Shala et David
Wagnieres \n" +
        "- aux Beta-testeurs : Quentin Jaquier,
Bastien Morleo, Martin Quartier, Damien Scantamburlo, Robin Walther et tous les
Alpha-testeurs \n" +
        "- et a tous ceux qui ont pris la peine
d'essayer BleuSolve. ");
    } // public void actionPerformed(ActionEvent arg0)
}); // objetMenuAPropos.addActionListener(new ActionListener()

// Paramétrage et fixation des éléments

// Fixation des messages d'instruction
panneauMePrNord.setLayout(new GridLayout(3, 1));
panneauMePrNord.add(messageMePr1);
panneauMePrNord.add(messageMePr2);
panneauMePrNord.add(messageMePr3);
panneauMeEqNord.add(messageMeEq);
panneauReEqNord.add(messageReEq);
panneauMeSyNord.add(messageMeSy);
panneauReSyNord.add(messageReSy);
panneauMeCoNord.add(messageMeCo);
panneauReCoNord.add(messageReCo);
panneauMeDiNord.add(messageMeDi);
panneauReDiNord.add(messageReDi);
panneauMeExNord.add(messageMeEx);
panneauReExNord.add(messageReEx);

// Notice
panneauSud.add(messageNotice);

// Fixation des panneaux au menu principal
panneau.setLayout(new BorderLayout());
panneau.add(panneauMePrNord, BorderLayout.NORTH);
panneau.add(panneauMePrCentre, BorderLayout.CENTER);
panneau.add(panneauSud, BorderLayout.SOUTH);

// Ajout des boutons aux panneaux de menu
panneauMePrCentre.add(boutonMePr1);
panneauMePrCentre.add(boutonMePr2);

```

```
panneauMePrCentre.add(boutonMePr3);
panneauMePrCentre.add(boutonMePr8);
panneauMePrCentre.add(boutonMePr9);
panneauMePrCentre.add(boutonQuitterMePr);
panneauMeEqCentre.add(boutonMeEq1);
panneauMeEqCentre.add(boutonMeEq2);
panneauMeEqCentre.add(boutonMeEq3);
panneauMeEqCentre.add(boutonMenuPrincipalMeEq);
panneauMeEqCentre.add(boutonQuitterMeEq);
panneauMeSyCentre.add(boutonMeSy2);
panneauMeSyCentre.add(boutonMenuPrincipalMeSy);
panneauMeSyCentre.add(boutonQuitterMeSy);
panneauMeCoCentre.add(boutonMeCo0);
panneauMeCoCentre.add(boutonMeCo1);
panneauMeCoCentre.add(boutonMenuPrincipalMeCo);
panneauMeCoCentre.add(boutonQuitterMeCo);
panneauMeDiCentre.add(boutonMeDi1);
panneauMeDiCentre.add(boutonMenuPrincipalMeDi);
panneauMeDiCentre.add(boutonQuitterMeDi);
panneauMeExCentre.add(boutonMeEx1);
panneauMeExCentre.add(boutonMenuPrincipalMeEx);
panneauMeExCentre.add(boutonQuitterMeEx);

// Definition de l'emplacement des boutons
panneauMePrCentre.setLayout(miseEnPageMenu) ;
panneauMeEqCentre.setLayout(miseEnPageMenu) ;
panneauMeSyCentre.setLayout(miseEnPageMenu) ;
panneauMeCoCentre.setLayout(miseEnPageMenu) ;
panneauMeDiCentre.setLayout(miseEnPageMenu) ;
panneauMeExCentre.setLayout(miseEnPageMenu) ;

// Definition de la mise en page des resultats
panneauReEq1CentreCentre.setLayout(miseEnPageResultat);
panneauReEq2CentreCentre.setLayout(miseEnPageResultat);
panneauReEq3CentreCentre.setLayout(miseEnPageResultat);
panneauReSy2CentreCentre.setLayout(miseEnPageResultat);
panneauReCo0CentreCentre.setLayout(miseEnPageResultat);
panneauReCo1CentreCentre.setLayout(miseEnPageResultat);
panneauReDi1CentreCentre.setLayout(miseEnPageResultat);
panneauReEx1CentreCentre.setLayout(miseEnPageResultat);

// Ajout des resultats aux panneaux
panneauReEq1CentreCentre.add(resultatReEq1Expl);
panneauReEq1CentreCentre.add(resultatReEq1Calc);
panneauReEq2CentreCentre.add(resultatReEq2Expl);
panneauReEq2CentreCentre.add(resultatReEq2Calc);
panneauReEq3CentreCentre.add(resultatReEq3Expl);
panneauReEq3CentreCentre.add(resultatReEq3Calc);
panneauReSy2CentreCentre.add(resultatReSy2Expl);
panneauReSy2CentreCentre.add(resultatReSy2Calc);
panneauReCo0CentreCentre.add(resultatReCo0Expl);
panneauReCo0CentreCentre.add(resultatReCo0Calc);
panneauReCo1CentreCentre.add(resultatReCo1Expl);
panneauReCo1CentreCentre.add(resultatReCo1Calc);
panneauReDi1CentreCentre.add(resultatReDi1);
panneauReEx1CentreCentre.add(resultatReEx1);

// Rendre les resultats ineditable
resultatReEq1Expl.setEditable(false);
```

```
    resultatReEq1Calc.setEditable(false);
    resultatReEq2Expl.setEditable(false);
    resultatReEq2Calc.setEditable(false);
    resultatReEq3Expl.setEditable(false);
    resultatReEq3Calc.setEditable(false);
    resultatReSy2Expl.setEditable(false);
    resultatReSy2Calc.setEditable(false);
    resultatReCo0Expl.setEditable(false);
    resultatReCo0Calc.setEditable(false);
    resultatReCo1Expl.setEditable(false);
    resultatReCo1Calc.setEditable(false);
    resultatReDi1.setEditable(false);
    resultatReEx1.setEditable(false);

    // Creation des titres pour les panneaux
    panneauMePrNord.setBorder(BorderFactory.createTitledBorder("Message
de bienvenue"));

    panneauMeEqNord.setBorder(BorderFactory.createTitledBorder("Instruction"));
    panneauMeSyNord.setBorder(BorderFactory.createTitledBorder("Instruction"));
    panneauMeCoNord.setBorder(BorderFactory.createTitledBorder("Instruction"));
    panneauMeDiNord.setBorder(BorderFactory.createTitledBorder("Instruction"));
    panneauMeExNord.setBorder(BorderFactory.createTitledBorder("Instruction"));

    panneauMePrCentre.setBorder(BorderFactory.createTitledBorder("Activites
proposees"));

    panneauMeEqCentre.setBorder(BorderFactory.createTitledBorder("Activites
proposees"));

    panneauMeSyCentre.setBorder(BorderFactory.createTitledBorder("Activites
proposees"));

    panneauMeCoCentre.setBorder(BorderFactory.createTitledBorder("Activites
proposees"));

    panneauMeDiCentre.setBorder(BorderFactory.createTitledBorder("Activites
proposees"));

    panneauMeExCentre.setBorder(BorderFactory.createTitledBorder("Activites
proposees"));

    panneauReEqNord.setBorder(BorderFactory.createTitledBorder("Instruction"));
    panneauReSyNord.setBorder(BorderFactory.createTitledBorder("Instruction"));
    panneauReCoNord.setBorder(BorderFactory.createTitledBorder("Instruction"));
    panneauReDiNord.setBorder(BorderFactory.createTitledBorder("Instruction"));
    panneauReExNord.setBorder(BorderFactory.createTitledBorder("Instruction"));

    panneauReEq1CentreNord.setBorder(BorderFactory.createTitledBorder("Saisie de
la donnee"));
```

```
    panneauReEq2CentreNord.setBorder(BorderFactory.createTitledBorder("Saisie de la donnee"));

    panneauReEq3CentreNord.setBorder(BorderFactory.createTitledBorder("Saisie de la donnee"));

    panneauReSy2CentreNord.setBorder(BorderFactory.createTitledBorder("Saisie de la donnee"));

    panneauReCo0CentreNord.setBorder(BorderFactory.createTitledBorder("Saisie de la donnee"));

    panneauReCo1CentreNord.setBorder(BorderFactory.createTitledBorder("Saisie de la donnee"));

    panneauReDi1CentreNord.setBorder(BorderFactory.createTitledBorder("Saisie de la donnee"));

    panneauReEx1CentreNord.setBorder(BorderFactory.createTitledBorder("Saisie de la donnee"));

    panneauDefilReEq1CentreCentre.setBorder(BorderFactory.createTitledBorder("Resultat"));

    panneauDefilReEq2CentreCentre.setBorder(BorderFactory.createTitledBorder("Resultat"));

    panneauDefilReEq3CentreCentre.setBorder(BorderFactory.createTitledBorder("Resultat"));

    panneauDefilReSy2CentreCentre.setBorder(BorderFactory.createTitledBorder("Resultat"));

    panneauDefilReCo0CentreCentre.setBorder(BorderFactory.createTitledBorder("Resultat"));

    panneauDefilReCo1CentreCentre.setBorder(BorderFactory.createTitledBorder("Resultat"));

    panneauDefilReDi1CentreCentre.setBorder(BorderFactory.createTitledBorder("Resultat"));

    panneauDefilReEx1CentreCentre.setBorder(BorderFactory.createTitledBorder("Resultat"));

    panneauReEqCentreSud.setBorder(BorderFactory.createTitledBorder("Retour aux menus"));

    panneauReSyCentreSud.setBorder(BorderFactory.createTitledBorder("Retour aux menus"));

    panneauReCoCentreSud.setBorder(BorderFactory.createTitledBorder("Retour aux menus"));

    panneauReDiCentreSud.setBorder(BorderFactory.createTitledBorder("Retour aux menus"));
```

```
panneauReExCentreSud.setBorder(BorderFactory.createTitledBorder("Retour aux menus"));
panneauSud.setBorder(BorderFactory.createTitledBorder("Note"));

// Parametrage des panneaux de defilement
panneauDefil.getVerticalScrollBar().setUnitIncrement(12);

panneauDefilReEq1CentreCentre.getVerticalScrollBar().setUnitIncrement(12);
panneauDefilReEq2CentreCentre.getVerticalScrollBar().setUnitIncrement(12);
panneauDefilReEq3CentreCentre.getVerticalScrollBar().setUnitIncrement(12);
panneauDefilReSy2CentreCentre.getVerticalScrollBar().setUnitIncrement(12);
panneauDefilReCo0CentreCentre.getVerticalScrollBar().setUnitIncrement(12);
panneauDefilReCo1CentreCentre.getVerticalScrollBar().setUnitIncrement(12);
panneauDefilReDi1CentreCentre.getVerticalScrollBar().setUnitIncrement(12);
panneauDefilReEx1CentreCentre.getVerticalScrollBar().setUnitIncrement(12);
panneauDefil.getHorizontalScrollBar().setUnitIncrement(12);

panneauDefilReEq1CentreCentre.getHorizontalScrollBar().setUnitIncrement(12);
panneauDefilReEq2CentreCentre.getHorizontalScrollBar().setUnitIncrement(12);
panneauDefilReEq3CentreCentre.getHorizontalScrollBar().setUnitIncrement(12);
panneauDefilReSy2CentreCentre.getHorizontalScrollBar().setUnitIncrement(12);
panneauDefilReCo0CentreCentre.getHorizontalScrollBar().setUnitIncrement(12);
panneauDefilReCo1CentreCentre.getHorizontalScrollBar().setUnitIncrement(12);
panneauDefilReDi1CentreCentre.getHorizontalScrollBar().setUnitIncrement(12);
panneauDefilReEx1CentreCentre.getHorizontalScrollBar().setUnitIncrement(12);

// Parametrage des boites combo
boiteComboReEx1_1.setMaximumRowCount(12);
boiteComboReEx1_2.setMaximumRowCount(12);
boiteComboReEx1_3.setMaximumRowCount(12);
boiteComboReEx1_4.setMaximumRowCount(12);
boiteComboReEx1_5.setMaximumRowCount(12);
boiteComboReEx1_6.setMaximumRowCount(12);
boiteComboReEx1_7.setMaximumRowCount(12);
boiteComboReEx1_8.setMaximumRowCount(12);
boiteComboReEx1_9.setMaximumRowCount(12);
boiteComboReEx1_10.setMaximumRowCount(12);
boiteComboReEx1_11.setMaximumRowCount(12);
boiteComboReEx1_12.setMaximumRowCount(12);

// Ajout des boutons dans les panneaux de resolution
panneauReEqCentreSud.add(boutonMenuEquation);
panneauReEqCentreSud.add(boutonMenuPrincipalReq);
panneauReSyCentreSud.add(boutonMenuSysteme);
```

```
panneauReSyCentreSud.add(boutonMenuPrincipalReSy);
panneauReCoCentreSud.add(boutonMenuComplexe);
panneauReCoCentreSud.add(boutonMenuPrincipalReCo);
panneauReDiCentreSud.add(boutonMenuDivers);
panneauReDiCentreSud.add(boutonMenuPrincipalReDi);
panneauReExCentreSud.add(boutonMenuExtras);
panneauReExCentreSud.add(boutonMenuPrincipalReEx);

// Fixation des panneaux de resolution
panneauReEq1CentreNord.add(champReEq1_A);
panneauReEq1CentreNord.add(etiquetteReEq1_1);
panneauReEq1CentreNord.add(champReEq1_B);
panneauReEq1CentreNord.add(etiquetteReEq1_2);
panneauReEq1CentreNord.add(boutonResoudreReEq1);
panneauReEq1CentreNord.add(boutonEffacerReEq1);

panneauReEq2CentreNord.add(champReEq2_A);
panneauReEq2CentreNord.add(etiquetteReEq2_1);
panneauReEq2CentreNord.add(champReEq2_B);
panneauReEq2CentreNord.add(etiquetteReEq2_2);
panneauReEq2CentreNord.add(champReEq2_C);
panneauReEq2CentreNord.add(etiquetteReEq2_3);
panneauReEq2CentreNord.add(boutonResoudreReEq2);
panneauReEq2CentreNord.add(boutonEffacerReEq2);

panneauReEq3CentreNord.add(champReEq3_A);
panneauReEq3CentreNord.add(etiquetteReEq3_1);
panneauReEq3CentreNord.add(champReEq3_B);
panneauReEq3CentreNord.add(etiquetteReEq3_2);
panneauReEq3CentreNord.add(champReEq3_C);
panneauReEq3CentreNord.add(etiquetteReEq3_3);
panneauReEq3CentreNord.add(champReEq3_D);
panneauReEq3CentreNord.add(etiquetteReEq3_4);
panneauReEq3CentreNord.add(boutonResoudreReEq3);
panneauReEq3CentreNord.add(boutonEffacerReEq3);

panneauReSy2CentreNord.setLayout(new GridLayout(2, 7));
panneauReSy2CentreNord.add(etiquetteReSy2_1);
panneauReSy2CentreNord.add(champReSy2_A);
panneauReSy2CentreNord.add(etiquetteReSy2_2);
panneauReSy2CentreNord.add(champReSy2_B);
panneauReSy2CentreNord.add(etiquetteReSy2_3);
panneauReSy2CentreNord.add(champReSy2_C);
panneauReSy2CentreNord.add(boutonResoudreReSy2);
panneauReSy2CentreNord.add(etiquetteReSy2_4);
panneauReSy2CentreNord.add(champReSy2_D);
panneauReSy2CentreNord.add(etiquetteReSy2_5);
panneauReSy2CentreNord.add(champReSy2_E);
panneauReSy2CentreNord.add(etiquetteReSy2_6);
panneauReSy2CentreNord.add(champReSy2_F);
panneauReSy2CentreNord.add(boutonEffacerReSy2);

panneauReCo0CentreNord.add(etiquetteReCo0_1);
panneauReCo0CentreNord.add(champReCo0_A);
panneauReCo0CentreNord.add(etiquetteReCo0_2);
panneauReCo0CentreNord.add(champReCo0_B);
panneauReCo0CentreNord.add(etiquetteReCo0_3);
panneauReCo0CentreNord.add(boutonResoudreReCo0);
panneauReCo0CentreNord.add(boutonEffacerReCo0);
```

```
panneauReCo1CentreNord.add(etiquetteReCo1_1);
panneauReCo1CentreNord.add(champReCo1_A);
panneauReCo1CentreNord.add(etiquetteReCo1_2);
panneauReCo1CentreNord.add(champReCo1_B);
panneauReCo1CentreNord.add(etiquetteReCo1_3);
panneauReCo1CentreNord.add(champReCo1_C);
panneauReCo1CentreNord.add(etiquetteReCo1_4);
panneauReCo1CentreNord.add(champReCo1_D);
panneauReCo1CentreNord.add(etiquetteReCo1_5);
panneauReCo1CentreNord.add(boutonResoudreReCo1);
panneauReCo1CentreNord.add(boutonEffacerReCo1);
```

```
panneauReDi1CentreNord.add(etiquetteReDi1_0);
panneauReDi1CentreNord.add(champReDi1_A);
panneauReDi1CentreNord.add(etiquetteReDi1_1);
panneauReDi1CentreNord.add(champReDi1_B);
panneauReDi1CentreNord.add(etiquetteReDi1_2);
panneauReDi1CentreNord.add(champReDi1_C);
panneauReDi1CentreNord.add(etiquetteReDi1_3);
panneauReDi1CentreNord.add(champReDi1_D);
panneauReDi1CentreNord.add(etiquetteReDi1_4);
panneauReDi1CentreNord.add(champReDi1_E);
panneauReDi1CentreNord.add(etiquetteReDi1_5);
panneauReDi1CentreNord.add(champReDi1_F);
panneauReDi1CentreNord.add(etiquetteReDi1_6);
panneauReDi1CentreNord.add(champReDi1_G);
panneauReDi1CentreNord.add(etiquetteReDi1_7);
panneauReDi1CentreNord.add(champReDi1_H);
panneauReDi1CentreNord.add(etiquetteReDi1_8);
panneauReDi1CentreNord.add(boutonResoudreReDi1);
panneauReDi1CentreNord.add(boutonEffacerReDi1);
```

```
panneauReEx1CentreNord.add(boiteComboReEx1_1);
panneauReEx1CentreNord.add(boiteComboReEx1_2);
panneauReEx1CentreNord.add(boiteComboReEx1_3);
panneauReEx1CentreNord.add(boiteComboReEx1_4);
panneauReEx1CentreNord.add(boiteComboReEx1_5);
panneauReEx1CentreNord.add(boiteComboReEx1_6);
panneauReEx1CentreNord.add(boiteComboReEx1_7);
panneauReEx1CentreNord.add(boiteComboReEx1_8);
panneauReEx1CentreNord.add(boiteComboReEx1_9);
panneauReEx1CentreNord.add(boiteComboReEx1_10);
panneauReEx1CentreNord.add(boiteComboReEx1_11);
panneauReEx1CentreNord.add(boiteComboReEx1_12);
panneauReEx1CentreNord.add(boutonResoudreReEx1);
panneauReEx1CentreNord.add(boutonEffacerReEx1);
```

```
// Affectation des actions aux boutons

boutonMePr1.addActionListener(new EcouteurBoutonMenuEquation());

boutonMeEq1.addActionListener(new EcouteurBoutonReEq1());

boutonResoudreReEq1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {

        // Affectation des valeurs aux variables
```

```

        try {
            a =
Double.valueOf(champReEq1_A.getText()).doubleValue();
        } catch (NumberFormatException e) {
            a = 0;
            champReEq1_A.setText("0");
        } // catch (NumberFormatException e)
        try {
            b =
Double.valueOf(champReEq1_B.getText()).doubleValue();
        } catch (NumberFormatException e) {
            b = 0;
            champReEq1_B.setText("0");
        } // catch (NumberFormatException e)

// Resultat
if (a == 0 && b == 0) {
    resultatReEq1Expl.setText(
        "L'equation a resoudre est : \n" +
        "Il existe une infinite de solutions,
car tous les nombres multiplies par \"0\" donne \"0\". ");
    resultatReEq1Calc.setText(
        a + "\u00b7x + " + b + " = 0 \n");
} else if (a == 0 && b != 0) {
    resultatReEq1Expl.setText(
        "L'equation a resoudre est : \n" +
        "Il n'y a aucune solution, car tous
les nombres multiplies par \"0\" donne \"0\". ");
    resultatReEq1Calc.setText(
        a + "\u00b7x + " + b + " = 0 \n");
} else if (a != 0 && b == 0) {
    resultatReEq1Expl.setText(
        "L'equation a resoudre est : \n" +
        "Etant donne que seul \"0\" multiplie
par un nombre peut donner \"0\". ");
    resultatReEq1Calc.setText(
        a + "\u00b7x + " + b + " = 0 \n" +
        "\u21d2 x = 0");
} else {
    resultatReEq1Expl.setText(
        "L'equation a resoudre est : \n" +
        "On isole la partie qui contient
l'inconnue : \n" +
        "\n" +
        "Puis on divise le tout par : " + a +
        "Et on trouve la reponse :");
    resultatReEq1Calc.setText(
        a + "\u00b7x + " + b + " = 0 \n" +
        a + "\u00b7x + " + b + " = 0 \t || +
+ (-b) + "\n" +
        a + "\u00b7x = " + (-b) + " \t || :
+ a + "\n" +
        "\u21d2 x = " + (-b/a));
} // else

    resultatReEq1Expl.setCaretPosition(0);
} // public void actionPerformed(ActionEvent arg0)
}); // boutonResoudreReEq1.addActionListener(new ActionListener()

```

```

boutonEffacerReEq1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {

        champReEq1_A.setText("");
        champReEq1_B.setText("");
        resultatReEq1Expl.setText("");
        resultatReEq1Calc.setText("");

    } // public void actionPerformed(ActionEvent arg0)
}); // boutonEffacerReEq1.addActionListener(new ActionListener()

boutonMeEq2.addActionListener(new EcouteurBoutonReEq2());

boutonResoudreReEq2.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {

        // Affectation des valeurs aux variables
        try {
            a =
Double.valueOf(champReEq2_A.getText()).doubleValue();
        } catch (NumberFormatException e) {
            a = 0;
            champReEq2_A.setText("0");
        } // catch (NumberFormatException e)
        try {
            b =
Double.valueOf(champReEq2_B.getText()).doubleValue();
        } catch (NumberFormatException e) {
            b = 0;
            champReEq2_B.setText("0");
        } // catch (NumberFormatException e)
        try {
            c =
Double.valueOf(champReEq2_C.getText()).doubleValue();
        } catch (NumberFormatException e) {
            c = 0;
            champReEq2_C.setText("0");
        } // catch (NumberFormatException e)

        // Resultat
        if (a == 0) {
            resultatReEq2Expl.setText(
                "L'equation a resoudre est : \n" +
                "Ceci est une equation du premier
degre, veuillez consulter l'application ad hoc dans ce logiciel.");
            resultatReEq2Calc.setText(
                a + "\u00b7x\u00b2 + " + b + "\u00b7x
+ " + c + " = 0 \n");
        } else if (((b*b)-4*a*c) < 0) {
            resultatReEq2Expl.setText(
                "L'equation a resoudre est : \n" +
                "Le discriminant de votre equation
(b\u00b2 - 4\u00b7a\u00b7c) etant negatif, il n'existe pas de solution reelle pour
cette equation. \n" +
                "Nous allons neanmoins la resoudre
dans l'ensemble des nombre complexe : \n" +
                "Etant donne que l'equation est de la
forme \n" +

```



```

(Math.sqrt(-b*b+4*a*c)/(2*a)) + "\u00b7i ou x = " + ((-b)/(2*a)) + " + " + -
(Math.sqrt(-b*b+4*a*c)/(2*a)) + "\u00b7i \n" +
"          2\u00b7" + a + "\n" +
"\n" +
"\n" +
a + "\u00b7x\u00b2 + " + b + "\u00b7x
+ " + c + " = 0 \t || +" + (-c) + "\n" +
a + "\u00b7x\u00b2 + " + b + "\u00b7x
= " + (-c) + " \t || : " + a + "\n" +
" x\u00b2 + " + b/a + "\u00b7x = " +
(-c/a) + " \t || +" + (b*b)/(4*a*a) + "\n" +
" x\u00b2 + " + b/a + "\u00b7x + " +
(b*b)/(4*a*a) + " = " + -c/a + "+" + (b*b)/(4*a*a) + " \t || factorisation de la
partie de gauche et simplification de la partie de droite \n" +
"\n" +
"(x + " + (b/(2*a)) + ") \u00b2 = " +
(b*b - 4*a*c)/(4*a*a) + " \t || Multiplication par \"-i\u00b2\" (qui est egal a 1)
a la partie de droite\n" +
"          \"(x + " + (b/(2*a)) + ") \u00b2 = " +
-(b*b - 4*a*c)/(4*a*a) + "\u00b7i \u00b2 \t || Racine des deux parties\n" +
" x + " + (b/(2*a)) + " = \u00b7(" +
(Math.sqrt(-(b*b - 4*a*c))/(4*a*a)) + "\u00b7i)\n" +
"\n" +
" x + " + (b/(2*a)) + " = " +
(Math.sqrt(-(b*b - 4*a*c))/(4*a*a)) + "\u00b7i \t || +" + (-b/(2*a)) + "\n" +
" \u021d2 x = " + -(b/(2*a)) + " + " +
(Math.sqrt(-(b*b - 4*a*c)))/(2*a) + "\u00b7i \n" +
" x + " + (b/(2*a)) + " = " + -
(Math.sqrt(-(b*b - 4*a*c))/(4*a*a)) + "\u00b7i \t || +" + (-b/(2*a)) + "\n" +
" \u021d2 x = " + -(b/(2*a)) + " + " + -
(Math.sqrt(-(b*b - 4*a*c)))/(2*a) + "\u00b7i");
} else if (((b*b)-4*a*c) > 0 && a == 1 && ((-
b+Math.sqrt(b*b-4*a*c))/(2*a)) == (double)(int)((-b+Math.sqrt(b*b-4*a*c))/(2*a))
&& ((-b-Math.sqrt(b*b-4*a*c))/(2*a)) == (double)(int)((-b-Math.sqrt(b*b-
4*a*c))/(2*a))) {
    e = ((-b+Math.sqrt(b*b-4*a*c))/(2*a));
    f = ((-b-Math.sqrt(b*b-4*a*c))/(2*a));
    resultatReEq2Expl.setText(
        "L'equation a resoudre est : \n" +
        "Et elle est facilement factorisable.
\n" +
        "Il faut juste trouver deux nombres
qui, quand on les additionne donnent \" + b + "\"" et quand on les multiplient
donnent \" + c + "\". \n" +
        "Ces deux nombres sont donc : \n" +
        "Comme vous etes surement encore
sceptique, on va developper le resultat de cette factorisation : \n" +
        "Puis le simplifier : \n" +
        "Et on retombe sur l'equation de
depart, ce qui veut dire que notre factorisation est juste : \n" +
        "Bon, maintenant, reprenons cette
equation vu precedemment : \n" +
        "L'equation est verifiee si \"(x + " +
-e + ")\" et/ou \"(x + " + -f + ")\" vaut \"0\". \n" +
        "Les reponses sont donc : \n" +
        "\n" +

```



```

+ "\n" +
"\n" +
4\u00b7a\u00b7c) \n" +
\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500 \n" +
+ b + "\u00b2 - 4\u00b7" + a + "\u00b7" + c + ") \n" +
\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500 \u21d2 x = " + ((-b+Math.sqrt(b*b-
4*a*c))/(2*a)) + " ou x = " + ((-b-Math.sqrt(b*b-4*a*c))/(2*a)) + "\n" +
"
2\u00b7" + a + "\n" +
"\n" +
a + "\u00b7x\u00b2 + " + b + "\u00b7x
a + "\u00b7x\u00b2 + " + b + "\u00b7x
"x\u00b2 + " + b/a + "\u00b7x = " +
"x\u00b2 + " + b/a + "\u00b7x + " +
(b*b)/(4*a*a) + " = " + -c/a + "+" + (b*b)/(4*a*a) + " \t || factorisation de la
partie de gauche et simplification de la partie de droite \n" +
"(x + " + (b/(2*a)) + ") \u00b2 = " +
(b*b - 4*a*c)/(4*a*a) + " \t || Racine carree des deux membres\n" +
"x + " + (b/(2*a)) + " = \u00b1 " +
Math.sqrt((b*b - 4*a*c)/(4*a*a)) + "\n" +
"\n" +
"x + " + (b/(2*a)) + " = " +
Math.sqrt((b*b - 4*a*c)/(4*a*a)) + " \t || +" + -(b/(2*a)) + "\n" +
"x = " + Math.sqrt((b*b -
4*a*c)/(4*a*a)) + " - " + (b/(2*a)) + "\n" +
"\u21d2 x = " + (Math.sqrt((b*b -
4*a*c)/(4*a*a)))-((b/(2*a)))) + "\n" +
"x + " + (b/(2*a)) + " = " + -
Math.sqrt((b*b - 4*a*c)/(4*a*a)) + " \t || +" + -(b/(2*a)) + "\n" +
"x = -" + Math.sqrt((b*b -
4*a*c)/(4*a*a)) + " - " + (b/(2*a)) + "\n" +
"\u21d2 x = " + (-Math.sqrt((b*b -
4*a*c)/(4*a*a)))-((b/(2*a)))));
} else if (((b*b)-4*a*c) > 0) {
    resultatReEq2Expl.setText(
        "L'equation a resoudre est : \n" +
        "Etant donne que l'equation est de la
forme \n" +
"\n" +
"vous pouvez simplement utiliser la
formule de resolution des equations du deuxieme degre : \n" +
"\n" +
"\n" +
"Ce qui donnerait avec les valeurs
que vous avez entrees :\n" +
"\n" +

```



```

(x + " + (b/(2*a)) + ") \u00b2 = " +
(b*b - 4*a*c)/(4*a*a) + " \t || Racine carree des deux membres\n" +
"x + " + (b/(2*a)) + " = \u00b1 " +
Math.sqrt((b*b - 4*a*c)/(4*a*a)) + "\n" +
"\n" +
"x + " + (b/(2*a)) + " = " +
Math.sqrt((b*b - 4*a*c)/(4*a*a)) + " \t || +" + -(b/(2*a)) + "\n" +
"x = " + Math.sqrt((b*b -
4*a*c)/(4*a*a)) + " - " + (b/(2*a)) + "\n" +
"\u21d2 x = " + (Math.sqrt((b*b -
4*a*c)/(4*a*a))-(b/(2*a))) + "\n" +
"x + " + (b/(2*a)) + " = " + -
Math.sqrt((b*b - 4*a*c)/(4*a*a)) + " \t || +" + -(b/(2*a)) + "\n" +
"x = -" + Math.sqrt((b*b -
4*a*c)/(4*a*a)) + " - " + (b/(2*a)) + "\n" +
"\u21d2 x = " + (-Math.sqrt((b*b -
4*a*c)/(4*a*a))-(b/(2*a)))));
} else if ((b*b)-4*a*c) == 0 && a == 1 && (-b/(2*a)) ==
(double)(int)(-b/(2*a)) {
e = (-b/(2*a));
resultatReEq2Expl.setText(
"L'equation a resoudre est : \n" +
"Et elle est facilement factorisable.
\n" +
"Il faut juste trouver deux nombres
qui, quand on les additionne donnent \" + b + "\" et quand on les multiplient
donnent \" + c + "\". \n" +
"Ces deux nombres (qui sont egaux)
sont donc : \n" +
"Comme vous etes surement encore
sceptique, on va developper le resultat de cette factorisation : \n" +
"Puis le simplifier : \n" +
"Et on retombe sur l'equation de
depart, ce qui veut dire que notre factorisation est juste : \n" +
"Bon, maintenant, reprenons cette
equation vu precedemment : \n" +
"-e + ")\" vaut \"0\". \n" +
"L'equation est verifiee si \"(x + " +
"La reponse est donc : \n" +
"\n" +
"Mais si vous avez des difficultes
avec la factorisation, vous pouvez toujours utiliser la methode classique : \n" +
"Etant donne que l'equation est de la
forme \n" +
"\n" +
"vous pouvez simplement utiliser la
formule de resolution des equations du deuxieme degre : \n" +
"\n" +
"\n" +
"Ce qui donnerait avec les valeurs
que vous avez entrees : \n" +
"\n" +
"Mais nous allons resoudre sans
passer directement par la formule (on va donc effectuer une application numerique
de la demonstration de la formule) : \n" +
" Tout d'abord, nous allons passer \"
+ c + "\" de l'autre cote de l'equation : \n" +

```

```

+
l'egalite un terme pour que la partie de gauche devienne factorisable (de la forme
\"a\u00b2 + 2\u00b7a\u00b7b + b\u00b2\") : \n" +
partie de gauche (en \"(a + b)\u00b2\") et simplifier la partie de droite : \n" +
: \n" +
l'inconnue : \n" +
: ");
+ " + c + " = 0 \n" +
= 0 \n" +
"\u00b7x + " + (e*e) + " = 0 \n" +
(e*e) + " = 0 \n" +
= 0 \n" +
\n" +
4\u00b7a\u00b7c) \n" +
\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500 \n" +
+ b + "\u00b2 - 4\u00b7" + a + "\u00b7" + c + ") \n" +
\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500 \u21d2 x = " + (-b/(2*a)) + "\n" +
+ " + c + " = 0 \t || +" + (-c) + "\n" +
= " + (-c) + " \t || : " + a + "\n" +
(-c/a) + " \t || +" + (b*b)/(4*a*a) + "\n" +
(b*b)/(4*a*a) + " = " + -c/a + "+" + (b*b)/(4*a*a) + " \t || factorisation de la
partie de gauche et simplification de la partie de droite \n" +
(b*b - 4*a*c)/(4*a*a) + " \t || Racine carree des deux membres\n" +
Math.sqrt((b*b - 4*a*c)/(4*a*a)) + " \t || +" + -(b/(2*a)) + "\n" +
" Puis diviser par \" + a + "\" : \n"
" Et on ajoute des deux cotes de
" On peut donc a present factoriser la
partie de gauche (en \"(a + b)\u00b2\") et simplifier la partie de droite : \n" +
" Et on met le tout a la racine carree
" On isole la partie qui contient
" Et on trouve la solution recherchee
resultatReEq2Calc.setText(
a + "\u00b7x\u00b2 + " + b + "\u00b7x
\n" +
\n" +
-e + " et " + -e + "\n" +
"(x + " + -e + ") \u00b7(x + " + -e + ")
"x\u00b2 + " + -e + "\u00b7x + " + -e +
"x\u00b2 + " + (-e-e) + "\u00b7x + " +
"(x + " + -e + ") \u00b7(x + " + -e + ")
\n" +
\u21d2 x = " + e + "\n" +
\n" +
\n" +
"a\u00b7x\u00b2 + b\u00b7x + c = 0,
" -b \u00b7 \u221a(b\u00b2 -
"x =
" 2\u00b7a \n" +
" + (-b) + " \u00b7 \u221a(
"x =
\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500 \u21d2 x = " + (b/(2*a)) + "\n" +
\n" +
a + "\u00b7x\u00b2 + " + b + "\u00b7x
a + "\u00b7x\u00b2 + " + b + "\u00b7x
"x\u00b2 + " + b/a + "\u00b7x = " +
"x\u00b2 + " + b/a + "\u00b7x + " +
" \t || factorisation de la
"(x + " + (b/(2*a)) + ") \u00b2 = " +
" \t || Racine carree des deux membres\n" +
"x + " + (b/(2*a)) + " = " +
" \t || +" + -(b/(2*a)) + "\n" +

```



```

        "x\u00b2 + " + b/a + "\u00b7x + " +
(b*b)/(4*a*a) + " = " + -c/a + "+" + (b*b)/(4*a*a) + " \t || factorisation de la
partie de gauche et simplification de la partie de droite \n" +
        "(x + " + (b/(2*a)) + ")\u00b2 = " +
(b*b - 4*a*c)/(4*a*a) + " \t || Racine carree des deux membres\n" +
        "x + " + (b/(2*a)) + " = " +
Math.sqrt((b*b - 4*a*c)/(4*a*a)) + " \t || +" + -(b/(2*a)) + "\n" +
        "\u21d2 x = " + (Math.sqrt((b*b -
4*a*c)/(4*a*a))-((b/(2*a))));
    } // else if ((b*b)-4*a*c) == 0

    resultatReEq2Expl.setCaretPosition(0);

} // public void actionPerformed(ActionEvent arg0)
}); // boutonResoudreReEq2.addActionListener(new ActionListener()

boutonEffacerReEq2.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {

        champReEq2_A.setText("");
        champReEq2_B.setText("");
        champReEq2_C.setText("");
        resultatReEq2Expl.setText("");
        resultatReEq2Calc.setText("");

    } // public void actionPerformed(ActionEvent arg0)
}); // boutonEffacerReEq2.addActionListener(new ActionListener()

boutonMeEq3.addActionListener(new EcouteurBoutonReEq3());

boutonResoudreReEq3.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {

        // Affectation des valeurs aux variables
        try {
            a =
Double.valueOf(champReEq3_A.getText()).doubleValue();
        } catch (NumberFormatException e) {
            a = 0;
            champReEq3_A.setText("0");
        } // catch (NumberFormatException e)
        try {
            b =
Double.valueOf(champReEq3_B.getText()).doubleValue();
        } catch (NumberFormatException e) {
            b = 0;
            champReEq3_B.setText("0");
        } // catch (NumberFormatException e)
        try {
            c =
Double.valueOf(champReEq3_C.getText()).doubleValue();
        } catch (NumberFormatException e) {
            c = 0;
            champReEq3_C.setText("0");
        } // catch (NumberFormatException e)
        try {
            d =
Double.valueOf(champReEq3_D.getText()).doubleValue();
        } catch (NumberFormatException e) {

```

```

        d = 0;
        champReEq3_D.setText("0");
    } // catch (NumberFormatException e)

    // Resultat
    if (a == 0 && b == 0 && c == 0 && d == 0) {
        resultatReEq3Expl.setText(
            "L'equation a resoudre est : \n" +
            "Il existe une infinite de solutions,
car tous les nombres multiplies par \"0\" donne \"0\". ");
        resultatReEq3Calc.setText(
            a + "\u00b7x\u00b3 + " + b +
            "\u00b7x\u00b2 + " + c + "\u00b7x + " + d + " = 0 \n");
    } else if (a == 0 && b == 0 && c == 0 && d != 0) {
        resultatReEq3Expl.setText(
            "L'equation a resoudre est : \n" +
            "Il n'y a aucune solution, car tous
les nombres multiplies par \"0\" donne \"0\". ");
        resultatReEq3Calc.setText(
            a + "\u00b7x\u00b3 + " + b +
            "\u00b7x\u00b2 + " + c + "\u00b7x + " + d + " = 0 \n");
    } else if (a == 0 && b != 0 && c != 0 && d != 0) {
        resultatReEq3Expl.setText(
            "L'equation a resoudre est : \n" +
            "Ceci est une equation du premier
degre, veuillez consulter l'application ad hoc dans ce logiciel.");
        resultatReEq3Calc.setText(
            a + "\u00b7x\u00b3 + " + b +
            "\u00b7x\u00b2 + " + c + "\u00b7x + " + d + " = 0 \n");
    } else if (a != 0 && b != 0 && c != 0 && d != 0) {
        resultatReEq3Expl.setText(
            "L'equation a resoudre est : \n" +
            "Ceci est une equation du deuxieme
degre, veuillez consulter l'application ad hoc dans ce logiciel.");
        resultatReEq3Calc.setText(
            a + "\u00b7x\u00b3 + " + b +
            "\u00b7x\u00b2 + " + c + "\u00b7x + " + d + " = 0 \n");
    } else {
        try {
            caractereResultatReEq3Expl = "" +
                "L'equation a resoudre est :
\n" +
                "Nous allons tout d'abord
trouver la solution \"simple\" qui est un nombre entier compris entre -10 et 10 (y
compris). \n" +
                "Pour cela, nous allons
essayer de remplacer l'inconnue x par 0, 1, -1, 2, -2, [...], 10, -10 jusqu'a ce
que le resultat de l'equation donne bien zero : \n";
            caractereResultatReEq3Calc = "" +
                a + "\u00b7x\u00b3 + " + b +
                "\u00b7x\u00b2 + " + c + "\u00b7x + " + d + " = 0 \n\n";

            for (int indice = 0;
(a*Math.pow(tableauReEq3[indice], 3)+b*Math.pow(tableauReEq3[indice],
2)+c*tableauReEq3[indice]+d) != 0; indice++) {
                caractereResultatReEq3Expl += "Avec x
= " + tableauReEq3[indice] + " : \n";
                caractereResultatReEq3Calc += a +
                "\u00b7" + Math.pow(tableauReEq3[indice], 3) + " + " + b + "\u00b7" +

```

```

Math.pow(tableauReEq3[indice], 2) + " + " + c + "\u00b7" + tableauReEq3[indice] +
" + " + d + " = " + (a*Math.pow(tableauReEq3[indice],
3)+b*Math.pow(tableauReEq3[indice], 2)+c*tableauReEq3[indice]+d) + " \u2260
\u21d2 \u2717 \n";
        x = tableauReEq3[indice + 1];
    } // for (int indice = 0;
(a*Math.pow(tableauReEq3[indice], 3)+b*Math.pow(tableauReEq3[indice],
2)+c*tableauReEq3[indice]+d) != 0; indice++)

    // Solution "simple"
    caractereResultatReEq3Expl += "" +
        "Avec x = " + x + " : \n" +
        "\u21d2 La solution \"simple\"

est donc : " + x + ". \n" +
        "Nous pouvons donc reecrire
l'equation differemment en le developpant etant donne qu'on connait deja une
solution : (x + " + (-x) + ") \u00b7(...) = 0 \n" +
        "Il ne reste plus qu'a trouver
\"(...)\". Il faut donc diviser (" + a + "\u00b7x\u00b3 +" + b + "\u00b7x\u00b2 +"
+ c + "\u00b7x +" + d + ") par " + "(x -" + x + "). \n" +
        "Pour cela, nous allons
utiliser la methode de Horner (qui est expliquee dans le menu \"Theorie\") : \n" +
        "\n" +
        "\n" +
        "On obtient donc : \n" +
        "Il ne nous reste plus qu'a
resoudre l'equation du deuxieme degre avec l'application ad hoc qui se trouve dans
ce logiciel pour trouver la ou les solution(s) manquante(s). ";
        caractereResultatReEq3Calc += "" +
            a + "\u00b7" + Math.pow(x, 3)
+ " + " + b + "\u00b7" + Math.pow(x, 2) + " + " + c + "\u00b7" + x + " + " + d + "
= " + (a*Math.pow(x, 3)+b*Math.pow(x, 2)+c*x+d) + " = 0 \u21d2 \u2713 \n" +
            "\n" +
            "\n" +
            "\n" +
            "\t" + a + "\t" + b + "\t" + c
+ "\t" + d + "\n" +
            x + "\t\t" + a*x + "\t" +
            ((b+a*x)*x) + "\t" + ((c+(b+a*x)*x)*x) + "\n" +
            "\t" + a + "\t" + (b+a*x) +
            "\t" + (c+(b+a*x)*x) + "\t" + (d+(c+(b+a*x)*x)*x) + "\n" +
            "(x + " + (-x) + ") \u00b7(" + a
+ "\u00b7x\u00b2 +" + (b+a*x) + "\u00b7x +" + (c+(b+a*x)*x) + ") = 0 \n";

        // Absence de solution "simple"
    } catch (ArrayIndexOutOfBoundsException e) {
        caractereResultatReEq3Expl += "" +
            "Il n'existe pas de solution
entiere comprise entre -10 et 10 pour l'equation que vous avez entree !";
    } // catch (ArrayIndexOutOfBoundsException e)

    // Ajout de la chaine de caractere au resultat

resultatReEq3Expl.setText(caractereResultatReEq3Expl);

resultatReEq3Calc.setText(caractereResultatReEq3Calc);

    } // else

```

```

        resultatReEq3Expl.setCaretPosition(0);
    } // public void actionPerformed(ActionEvent arg0)
}); // boutonResoudreReEq3.addActionListener(new ActionListener()

boutonEffacerReEq3.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {

        champReEq3_A.setText("");
        champReEq3_B.setText("");
        champReEq3_C.setText("");
        champReEq3_D.setText("");
        resultatReEq3Expl.setText("");
        resultatReEq3Calc.setText("");

    } // public void actionPerformed(ActionEvent arg0)
}); // boutonEffacerReEq3.addActionListener(new ActionListener()

boutonMenuPrincipalMeEq.addActionListener(new
EcouteurBoutonMenuPrincipal());
boutonQuitterMeEq.addActionListener(new EcouteurBoutonQuitter());
boutonMenuPrincipalReEq.addActionListener(new
EcouteurBoutonMenuPrincipal());
boutonMenuEquation.addActionListener(new
EcouteurBoutonMenuEquation());

boutonMePr2.addActionListener(new EcouteurBoutonMenuSysteme());

boutonMeSy2.addActionListener(new EcouteurBoutonReSy2());

boutonResoudreReSy2.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {

        // Affectation des valeurs aux variables
        try {
            a1 =
Double.valueOf(champReSy2_A.getText()).doubleValue();
        } catch (NumberFormatException e) {
            a1 = 0;
            champReSy2_A.setText("0");
        } // catch (NumberFormatException e)
        try {
            b1 =
Double.valueOf(champReSy2_B.getText()).doubleValue();
        } catch (NumberFormatException e) {
            b1 = 0;
            champReSy2_B.setText("0");
        } // catch (NumberFormatException e)
        try {
            c1 =
Double.valueOf(champReSy2_C.getText()).doubleValue();
        } catch (NumberFormatException e) {
            c1 = 0;
            champReSy2_C.setText("0");
        } // catch (NumberFormatException e)
        try {
            a2 =
Double.valueOf(champReSy2_D.getText()).doubleValue();

```

```

    } catch (NumberFormatException e) {
        a2 = 0;
        champReSy2_D.setText("0");
    } // catch (NumberFormatException e)
    try {
        b2 =
Double.valueOf(champReSy2_E.getText()).doubleValue();
    } catch (NumberFormatException e) {
        b2 = 0;
        champReSy2_E.setText("0");
    } // catch (NumberFormatException e)
    try {
        c2 =
Double.valueOf(champReSy2_F.getText()).doubleValue();
    } catch (NumberFormatException e) {
        c2 = 0;
        champReSy2_F.setText("0");
    } // catch (NumberFormatException e)

    // Calcul du plus grand diviseur commun
    if (b1 != 0 && b2 != 0) {
        a = b1;
        b = b2;
        do {
            r = a % b;
            if (r != 0) {
                a = b;
                b = r;
            }
        } while (r != 0);
        // PGDC = b
    } // if (b1 != 0 && b2 != 0)

    // Resultat
    if (a1 == 0 && b1 == 0 && c1 == 0 && a2 == 0 && b2 == 0
&& c2 == 0) {
        resultatReSy2Expl.setText(
            "Le systeme a resoudre est : \n" +
            "\n" +
            "Il existe une infinite de solution :
n'importe quel nombre multiplie par 0 donne 0 et 0 + 0 = 0. Donc, tous les nombres
sont des solutions du systeme. ");
        resultatReSy2Calc.setText(
            "\u2460 : " + a1 + "\u00b7x + " + b1 +
            "\u2461 : " + a2 + "\u00b7x + " + b2 +
            "\u00b7y = " + c1 + "\n" +
            "\u00b7y = " + c2 + "\n");
    } else if (a1 == 0 && a2 == 0 && b1 != 0 && b2 != 0 &&
(c1/b1) == (c2/b2)) {
        resultatReSy2Expl.setText(
            "Le systeme a resoudre est : \n" +
            "\n" +
            "Il existe une infinite de solution
pour \"x\" puisqu'on ne le prend pas en compte dans l'equation. \n" +
            "Mais cela ne nous empeche pas de
trouver \"y\" : \n");
        resultatReSy2Calc.setText(
            "\u2460 : " + a1 + "\u00b7x + " + b1 +
            "\u00b7y = " + c1 + "\n" +

```

```

        "\u2461 : " + a2 + "\u00b7x +" + b2 +
        "\n" +
        b1 + "\u00b7y = " + c1 + " \t || : " +
        "\u21d2 y = " + (c1/b1));
    } else if (a1 != 0 && a2 != 0 && b1 == 0 && b2 == 0 &&
(c1/a1) == (c2/a2)) {
        resultatReSy2Expl.setText(
            "Le systeme a resoudre est : \n" +
            "\n" +
            "Il existe une infinite de solution
pour \"y\" puisqu'on ne le prend pas en compte dans l'equation. \n" +
            "Mais cela ne nous empeche pas de
trouver \"x\" : \n");
        resultatReSy2Calc.setText(
            "\u2460 : " + a1 + "\u00b7x +" + b1 +
            "\u2461 : " + a2 + "\u00b7x +" + b2 +
            "\n" +
            a1 + "\u00b7x = " + c1 + " \t || : " +
            "\u21d2 x = " + (c1/a1));
    } else if (a1 == 0 && a2 == 0 && b1 != 0 && b2 != 0 &&
(c1/b1) != (c2/b2)) {
        resultatReSy2Expl.setText(
            "Le systeme a resoudre est : \n" +
            "\n" +
            "Il existe une infinite de solution
pour \"x\" puisqu'on ne le prend pas en compte dans l'equation. \n" +
            "\"y\" n'a pas de solution, car il ne
peut pas avoir deux valeurs en meme temps. ");
        resultatReSy2Calc.setText(
            "\u2460 : " + a1 + "\u00b7x +" + b1 +
            "\u2461 : " + a2 + "\u00b7x +" + b2 +
            "\n");
    } else if (a1 != 0 && a2 != 0 && b1 == 0 && b2 == 0 &&
(c1/a1) != (c1/a2)) {
        resultatReSy2Expl.setText(
            "Le systeme a resoudre est : \n" +
            "\n" +
            "Il existe une infinite de solution
pour \"y\" puisqu'on ne le prend pas en compte dans l'equation. \n" +
            "\"x\" n'a pas de solution, car il ne
peut pas avoir deux valeurs en meme temps. ");
        resultatReSy2Calc.setText(
            "\u2460 : " + a1 + "\u00b7x +" + b1 +
            "\u2461 : " + a2 + "\u00b7x +" + b2 +
            "\n");
    } else if ((a1*b2/b) == (a2*b1/b) && (b1*b2/b) ==
(b2*b1/b) && (c1*b2/b) == (c2*b1/b)) {
        resultatReSy2Expl.setText(
            "Le systeme a resoudre est : \n" +
            "\n" +

```

```

"Ce systeme contient deux fois la
meme equation (si on dessine ces deux droites sur un graphique, elles seront
confondues). \n" +
"Mais cela ne nous empeche pas de le
resoudre (en utilisant la premiere equation) : \n" +
"On isole la partie qui contient
\"x\" : \n" +
"On divise le tout par le facteur
multiplicatif de \"x\" : \n" +
"Et on trouve \"x\" : \n" +
"Maintenant, on reprend cette
equation et on sort la variable \"y\" : \n" +
"On isole la partie qui contient
\"y\" : \n" +
"On divise le tout par le facteur
multiplicatif de \"y\" : \n" +
"Et on trouve \"y\" : \n" +
"Remarquez qu'on trouve l'equation
d'une droite. \n" +
"Si on refait les memes calculs pour
la deuxieme equations, on obtiendra le meme resultat, \n" +
"puisque les deux equations sont
multiples l'un a l'autre. Vous n'avez qu'a essayer !\n" +
"Notez aussi que les reponses sont
toutes les valeurs qui sont situees sur la droite. ");
resultatReSy2Calc.setText(
"\u00b7y = " + c1 + "\n" +
"\u00b7y = " + c2 + "\n" +
+ c1 + " \t || +" + -b1 + "\u00b7y \n" +
"\u00b7y \t || : " + a1 + "\n" +
b1/a1) + "\u00b7y \n" +
"\u00b7y \t || +" + (b1/a1) + "\u00b7y -x \n" +
-x \t || : " + (b1/a1) + "\n" +
+ " + (c1/b1) + "\n" +
"\u2460 : " + a1 + "\u00b7x +" + b1 +
"\u2461 : " + a2 + "\u00b7x +" + b2 +
"\n" +
"\n" +
a1 + "\u00b7x +" + b1 + "\u00b7y = "
a1 + "\u00b7x = " + c1 + " +" + -b1 +
"\u21d2 x = " + (c1/a1) + " +" + (-
"\n" +
"x = " + (c1/a1) + " +" + (-b1/a1) +
(b1/a1) + "\u00b7y = " + (c1/a1) + "
"\u21d2 y = " + (-a1/b1) + "\u00b7x
"\n" +
"\n" +
"\n");
} else if (((a1*b2)-(a2*b1)) == 0) {
resultatReSy2Expl.setText(
"Le systeme a resoudre est : \n" +
"\n" +
"Calculons tout d'abord le
determinant suivant (c.f. La formule de Cramer) : \n" +
"\n" +
"\n" +
"Etant donne que le determinant vaut
\"0\", il n'existe pas de solution a ce systeme (c.f. La formule de Cramer. ");
resultatReSy2Calc.setText(

```

```

"\u00b7y = " + c1 + "\n" +
"\u00b7y = " + c2 + "\n" +
\t\u2502 \n" +
\t\u2502 \n" +
a2 + "\u00b7" + b1 + ") = " + ((a1*b2)-(a2*b1)) + "\n");
} else if (a1 == 0 && a2 != 0 && b1 != 0 && b2 != 0) {
    resultatReSy2Expl.setText(
        "Le systeme a resoudre est : \n" +
        "\n" +
        "Pour ce type de systeme, il est
fortement conseille de resoudre avec la methode de substitution : \n" +
        "Prenons la premiere equation et
trouvons \"y\" : \n" +
        "\n" +
        "Remplacons \"y\" par \"\" + (c1/b1) +
        "\n" +
        "\n" +
        "\n" +
        "\n" +
        "Resolution avec la formule de Cramer
: \n" +
        "Si on applique la formule de Cramer
dans ce systeme, on obtient ceci : \n" +
        "\n" +
        "\n");
    resultatReSy2Calc.setText(
        "\u2460 : " + a1 + "\u00b7x +" + b1 +
        "\u2461 : " + a2 + "\u00b7x +" + b2 +
        "\n" +
        b1 + "\u00b7y = " + c1 + " \t || : " +
        "\u21d2 y = " + (c1/b1) + "\n" +
        a2 + "\u00b7x +( " + b2 + "\u00b7" +
        a2 + "\u00b7x +" + ((c1/b1)*b2) + " =
" + c2 + " \t || +" + ((-c1/b1)*b2) + "\n" +

```

```

c1/b1)*b2)+c2) + " \t || : " + a2 + "\n" +
c1/b1)*b2)+c2)/a2) + "\n" +

b1 + " \t\u2502 \n" +
b2 + " \t\u2502 \n" +

\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500 \n" +
b1 + " \t\u2502 \n" +
b2 + " \t\u2502 \n" +
+
+
(c2*b1)/((a1*b2)-(a2*b1))) + "\n" +

(c1) + " \t\u2502 \n" +
(c2) + " \t\u2502 \n" +

"\n" +
+
(a2*(c1))/((a1*b2)-(a2*b1)));
} else if (a1 != 0 && a2 == 0 && b1 != 0 && b2 != 0) {
    resultatReSy2Expl.setText(
        "Le systeme a resoudre est : \n" +
        "\n" +
        "Pour ce type de systeme, il est
fortement conseille de resoudre avec la methode de substitution : \n" +
        "Prenons la deuxieme equation et
trouvons \"y\" : \n" +

        "\" dans la premiere equation : \n" +

a2 + "\u00b7x = " + (((-
"\u21d2 x = " + ((((-
"\n" +
"\n" +
" \u2502 \t " + c1 + " \t " +
" \u2502 \t " + c2 + " \t " +
"x =
\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500 \n" +
" \u2502 \t " + a1 + " \t " +
" \u2502 \t " + a2 + " \t " +
" " + ((c1*b2)-(c2*b1)) + "\n"
"x = \u2500\u2500\u2500\u2500 \n" +
" " + ((a1*b2)-(a2*b1)) + "\n"
"\u21d2 x = " + (((c1*b2)-
"\n" +
" \u2502 \t " + a1 + " \t " +
" \u2502 \t " + a2 + " \t " +
"y =
\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500 \n" +
" \u2502 \t " + a1 + " \t " +
" \u2502 \t " + a2 + " \t " +
" " + ((a1*(c2))-(a2*(c1))) +
"y = \u2500\u2500\u2500\u2500 \n" +
" " + ((a1*b2)-(a2*b1)) + "\n"
"\u21d2 y = " + (((a1*(c2))-
"Le systeme a resoudre est : \n" +
"\n" +
"Pour ce type de systeme, il est
fortement conseille de resoudre avec la methode de substitution : \n" +
"Prenons la deuxieme equation et
trouvons \"y\" : \n" +

        "\" dans la premiere equation : \n" +

a2 + "\u00b7x = " + (((-
"\u21d2 x = " + ((((-
"\n" +
"\n" +
" \u2502 \t " + c1 + " \t " +
" \u2502 \t " + c2 + " \t " +
"x =
\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500 \n" +
" \u2502 \t " + a1 + " \t " +
" \u2502 \t " + a2 + " \t " +
" " + ((c1*b2)-(c2*b1)) + "\n"
"x = \u2500\u2500\u2500\u2500 \n" +
" " + ((a1*b2)-(a2*b1)) + "\n"
"\u21d2 x = " + (((c1*b2)-
"\n" +
" \u2502 \t " + a1 + " \t " +
" \u2502 \t " + a2 + " \t " +
"y =
\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500 \n" +
" \u2502 \t " + a1 + " \t " +
" \u2502 \t " + a2 + " \t " +
" " + ((a1*(c2))-(a2*(c1))) +
"y = \u2500\u2500\u2500\u2500 \n" +
" " + ((a1*b2)-(a2*b1)) + "\n"
"\u21d2 y = " + (((a1*(c2))-

```





```

a1 + "\n" +
(c1/a1) + ") = " + c2 + "\n" +
" + c2 + " \t || +" + ((-c1/a1)*a2) + "\n" +
c1/a1)*a2)+c2) + " \t || :" + b2 + "\n" +
c1/a1)*a2)+c2)/b2) + "\n" +

b1 + " \t\u2502 \n" +
b2 + " \t\u2502 \n" +

\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500 \n" +
b1 + " \t\u2502 \n" +
b2 + " \t\u2502 \n" +
+
+
(c2*b1))/((a1*b2)-(a2*b1))) + "\n" +

(c1) + " \t\u2502 \n" +
(c2) + " \t\u2502 \n" +

\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500 \n" +
b1 + " \t\u2502 \n" +
b2 + " \t\u2502 \n" +
"\n" +
+
(a2*(c1))/((a1*b2)-(a2*b1)));
} else if (a1 != 0 && a2 != 0 && b1 != 0 && b2 == 0) {
    resultatReSy2Expl.setText(
        "Le systeme a resoudre est : \n" +
        "\n" +
        "Pour ce type de systeme, il est
fortement conseille de resoudre avec la methode de substitution : \n" +
        "Prenons la deuxieme equation et
trouvons \"x\" : \n" +

```

```

"\n" +
a1 + "\u00b7x = " + c1 + " \t || :" +

```

```

"\u21d2 x = " + (c1/a1) + "\n" +
b2 + "\u00b7y +(" + a2 + "\u00b7" +

```

```

b2 + "\u00b7y +" + ((c1/a1)*a2) + " =
b2 + "\u00b7y = " + (((-

```

```

"\u21d2 y = " + ((((-

```

```

"\n" +
"\n" +

```

```

" \u2502 \t " + c1 + " \t " +
" \u2502 \t " + c2 + " \t " +

```

```

"x =

```

```

" \u2502 \t " + a1 + " \t " +
" \u2502 \t " + a2 + " \t " +

```

```

" + ((c1*b2)-(c2*b1)) + "\n"

```

```

"x = \u2500\u2500\u2500\u2500 \n" +
" + ((a1*b2)-(a2*b1)) + "\n"

```

```

"\u21d2 x = " + (((c1*b2)-

```

```

"\n" +
" \u2502 \t " + a1 + " \t " +
" \u2502 \t " + a2 + " \t " +

```

```

" + ((a1*(c2))-(a2*(c1))) +

```

```

"y = \u2500\u2500\u2500\u2500 \n" +
" + ((a1*b2)-(a2*b1)) + "\n"

```

```

"\u21d2 y = " + (((a1*(c2))-

```

```


```

```


```

```


```

```


```

```


```

```


```











```

boutonQuitterMeSy.addActionListener(new EcouteurBoutonQuitter());
boutonMenuPrincipalReSy.addActionListener(new
EcouteurBoutonMenuPrincipal());
boutonMenuSysteme.addActionListener(new EcouteurBoutonMenuSysteme());

boutonMePr3.addActionListener(new EcouteurBoutonMenuComplexe());

boutonMeCo0.addActionListener(new EcouteurBoutonReCo0());

boutonResoudreReCo0.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {

        // Affectation des valeurs aux variables
        try {
            c =
Double.valueOf(champReCo0_A.getText()).doubleValue();
        } catch (NumberFormatException e) {
            c = 0;
            champReCo0_A.setText("0");
        } // catch (NumberFormatException e)
        try {
            d =
Double.valueOf(champReCo0_B.getText()).doubleValue();
        } catch (NumberFormatException e) {
            d = 0;
            champReCo0_B.setText("0");
        } // catch (NumberFormatException e)

        // Resultat
        r = Math.sqrt((c*c)+(d*d));
        if (c == 0 && d == 0) {
            caractereResultatReCo0Expl = "" +
                "Le nombre que vous avez entre est :
\n" +
                "La racine carree de 0 donne 0. ";
            caractereResultatReCo0Calc = "" +
                c + " + " + d + "\u00b7i\n";
        } else {
            caractereResultatReCo0Expl = "" +
                "L'equation que vous avez entree est
: \n" +
                "Posons \"x = a +b\u00b7i\" \n" +
                "\"x\" etant le resultat recherche.
\n" +
                "Calculons \"x\u00b2\" : \n" +
                "Donc \n" +
                "On separe la partie reelle de la
partie imaginaire : \n" +
                "Partie reelle : \n" +
                "Partie imaginaire : \n" +
                "\n" +
                "On s'occupe d'abord de la partie
imaginaire : \n" +
                "Cherchons \"b\" : \n" +
                "\n" +
                "\n" +
                "\n" +

```

```

reelle : \n" +
"/(2\u00b7a)\n" : \n" +
supprimer les fractions : \n" +
fait un changement de variable pour obtenir une equation du deuxieme degre : \n" +
du deuxieme degre grace a la formule : \n" +
"a\u00b2\n" : \n" +
solutions etant donne qu'on l'a mis a la racine carree) : \n" +
l'equation precedente pour trouver "b\n" : \n" +
l'equation precedente pour trouver "b\n" : \n" +
carree en passant par la forme polaire de "x\u00b2\n". \n" +
+ c + " + " + d + "\u00b7i\n", \n" +
module : \n";
caractereResultatReCo0Calc = "" +
"x\u00b2 = " + c + " + " + d +
"\n" +
"\n" +
"x\u00b2 = (a + b\u00b7i)\u00b2 =
a\u00b2 + 2\u00b7a\u00b7b\u00b7i + b\u00b2\u00b7i\u00b2 = a\u00b2
+ 2\u00b7a\u00b7b\u00b7i - b\u00b2 \n" +
"a\u00b2 + 2\u00b7a\u00b7b\u00b7i -
"\n" +
"a\u00b2 - b\u00b2 = " + c + "\n" +
"2\u00b7a\u00b7b\u00b7i = " + d +
"\n" +
"\n" +
"2\u00b7a\u00b7b = " + d + " \t ||
"b = " + d + "/(2\u00b7a) \n" +
"\n" +
"\n" +
"a\u00b2 - b\u00b2 = " + c + "\n" +
"a\u00b2 + (" + (-d) +
"/(2\u00b7a)\u00b2 = " + c + "\n" +

```



```

t = Math.PI+Math.atan(d/c);
caractereResultatReCo0Expl += " " +
    "Et un de ses arguments : \n"
+
    "Mais \"\u03b1\" n'est pas
l'angle recherche, \n" +
    "il faut encore lui ajouter
\" \u03c0\" puisqu'il se situe dans le troisieme ou quatrieme quadrant et qu'on a
utilise la tangente pour le trouver : \n" +
    "(Faites un dessin si vous
n'en etes pas convaincu) \n";
    caractereResultatReCo0Calc += " " +
        "tan\u03b1 = " + d + "/" + c +
" = " + (d/c) + " \u21d2 \u03b1 = arctan(" + (d/c) + ") = " + Math.atan(d/c) + "
[rad] \n" +
        "\n" +
        "\u21d2 \u03b8 = \u03b1
+\u03c0 = " + Math.atan(d/c) + " +\u03c0 = " + t + " [rad] = " + (t/Math.PI) +
"\u00b7\u03c0 [rad] = " + Math.toDegrees(t) + " [\u00b0] \n" +
        "\n";
    } else if (c > 0) {
        t = Math.atan(d/c);
        caractereResultatReCo0Expl += " " +
            "Et un de ses arguments : \n";
        caractereResultatReCo0Calc += " " +
            "tan\u03b8 = " + d + "/" + c +
" = " + (d/c) + " \u21d2 \u03b8 = arctan(" + (d/c) + ") = " + t + " [rad] = " +
(t/Math.PI) + "\u00b7\u03c0 [rad] = " + Math.toDegrees(t) + " [\u00b0] \n";
    } // else if (c > 0)
    caractereResultatReCo0Expl += " " +
        "Donc, sa forme polaire est : \n" +
        "Maintenant, posons que : \n" +
        "Donc \n" +
        "On trouve donc r' : \n" +
        "Et \u03b8' : \n" +
        "En notation algébrique, \"x\" vaut
donc : \n" +
        "Pour trouver la premiere solution,
on dit que \"k = 0\" : \n" +
        "\n" +
        "Et on retransforme ce nombre en
notation algebrique : x = r\u00b7cos\u03b8 + r\u00b7sin(\u03b8)\u00b7i : \n" +
        "Mais il existe encore une deuxieme
solution avec \"k = 1\" : \n" +
        "\n" +
        "Et on retransforme ce deuxieme
nombre en notation algebrique : \n" +
        "Si on continue avec \"k = 3\", \"k =
4\", [...] ou \"k = -1\", \"k = -2\", [...], \n" +
        "on retombe sur les memes solutions
puisque qu'on rajoute une ou plusieurs fois un tour complet (\u00b7\u03c0) a
l'argument. \n" +
        "Il existe donc deux solutions pour
cette equation. ";
        caractereResultatReCo0Calc += " " +
            "[ " + r + " ; " + t + " +
k\u00b7\u00b7\u03c0 ] \t | k \u2208 \u2124 \n" +
            "x = [ r' ; \u03b8' ] \n" +

```

```

r'\u00b2 ; 2\u00b7\u03b8' ] = [ " + r + " ; " + t + " + k\u00b7\u03c0 ] \n"
+
Math.sqrt(r) + "\n" +
k\u00b7\u03c0)/2 = " + (t/2) + " + k\u00b7\u03c0 \n" +
" + k\u00b7\u03c0 ] \n" +
" + k\u00b7\u03c0 ] \t || k = 0 \n" +
" ] \n" +
"\u00b7cos(" + (t/2) + ") +" + Math.sqrt(r) + "\u00b7sin(" + (t/2) + ")\u00b7i = "
+ (Math.sqrt(r)*Math.cos(t/2)) + " +" + (Math.sqrt(r)*Math.sin(t/2)) + "\u00b7i
\n" +
" + k\u00b7\u03c0 ] \t || k = 1 \n" +
((t/2)+Math.PI) + " ] \n" +
"\u00b7cos(" + ((t/2)+Math.PI) + ") +" + Math.sqrt(r) + "\u00b7sin(" +
((t/2)+Math.PI) + ")\u00b7i = " + (Math.sqrt(r)*Math.cos((t/2)+Math.PI)) + " +" +
(Math.sqrt(r)*Math.sin((t/2)+Math.PI)) + "\u00b7i \n" +
"\n" +
"\n" +
"\u21d2 x\u2081 = " +
(Math.sqrt(r)*Math.cos(t/2)) + " +" + (Math.sqrt(r)*Math.sin(t/2)) + "\u00b7i et
x\u2082 = " + (Math.sqrt(r)*Math.cos((t/2)+Math.PI)) + " +" +
(Math.sqrt(r)*Math.sin((t/2)+Math.PI)) + "\u00b7i";
} // else

resultatReCo0Expl.setText(caractereResultatReCo0Expl);
resultatReCo0Calc.setText(caractereResultatReCo0Calc);

resultatReCo0Expl.setCaretPosition(0);

} // public void actionPerformed(ActionEvent arg0)
}); // boutonResoudreReCo0.addActionListener(new ActionListener()

boutonEffacerReCo0.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {

        champReCo0_A.setText("");
        champReCo0_B.setText("");
        resultatReCo0Expl.setText("");
        resultatReCo0Calc.setText("");

    } // public void actionPerformed(ActionEvent arg0)
}); // boutonEffacerReCo0.addActionListener(new ActionListener()

boutonMeCo1.addActionListener(new EcouteurBoutonReCo1());

boutonResoudreReCo1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {

        // Affectation des valeurs aux variables
        try {

```

```

        a1 =
Double.valueOf(champReCo1_A.getText()).doubleValue();
    } catch (NumberFormatException e) {
        a1 = 0;
        champReCo1_A.setText("0");
    } // catch (NumberFormatException e)
    try {
        b1 =
Double.valueOf(champReCo1_B.getText()).doubleValue();
    } catch (NumberFormatException e) {
        b1 = 0;
        champReCo1_B.setText("0");
    } // catch (NumberFormatException e)
    try {
        a2 =
Double.valueOf(champReCo1_C.getText()).doubleValue();
    } catch (NumberFormatException e) {
        a2 = 0;
        champReCo1_C.setText("0");
    } // catch (NumberFormatException e)
    try {
        b2 =
Double.valueOf(champReCo1_D.getText()).doubleValue();
    } catch (NumberFormatException e) {
        b2 = 0;
        champReCo1_D.setText("0");
    } // catch (NumberFormatException e)

    // Resultat
    if (a1 == 0 && b1 == 0 && a2 == 0 && b2 == 0) {
        resultatReCo1Expl.setText(
            "L'equation a resoudre est : \n" +
            "Il existe une infinite de solutions,
car tous les nombres multiplies par 0 donne 0.");
        resultatReCo1Calc.setText(
            "(" + a1 + " + " + b1 +
"\u00b7i)\u00b7x +" + a2 + " +" + b2 + "\u00b7i = 0 \n");
    } else if (a1 == 0 && b1 == 0 && a2 != 0 && b2 != 0) {
        resultatReCo1Expl.setText(
            "L'equation a resoudre est : \n" +
            "Il n'y a aucune solution, car aucun
nombre multiplie par 0 ne peut donner un autre nombre que 0.");
        resultatReCo1Calc.setText(
            "(" + a1 + " + " + b1 +
"\u00b7i)\u00b7x +" + a2 + " +" + b2 + "\u00b7i = 0 \n");
    } else if (a1 != 0 && b1 != 0 && a2 == 0 && b2 == 0) {
        resultatReCo1Expl.setText(
            "L'equation a resoudre est : \n" +
            "Etant donne que seul \"0\" multiplie
par un nombre peut donner \"0\". ");
        resultatReCo1Calc.setText(
            "(" + a1 + " + " + b1 +
"\u00b7i)\u00b7x +" + a2 + " +" + b2 + "\u00b7i = 0 \n" +
            "\u21d2 x = 0");
    } else {
        resultatReCo1Expl.setText(
            "L'equation a resoudre est : \n" +
            "On commence par isoler l'inconnue
\"x\" : \n" +

```

```

        "\n" +
        "Dans la fraction, on multiplie le
numerateur et le denominateur par le conjugué du denominateur : \n" +
        "\n" +
        "\n" +
        "On simplifie le numerateur et le
denominateur (Notez que le denominateur est un produit remarquable) : \n" +
        "\n" +
        "\n" +
        "On simplifie la fraction (Notez que
\"i\u00b2 = -1\") : \n" +
        "\n" +
        "\n" +
        "\n" +
        "\n" +
        "\n" +
        "Et on trouve : \n" +
        "\n");
        resultatReCo1Calc.setText(
        "(" + a1 + " + " + b1 +
        "\u00b7x + " + a2 + " + " + b2 + "\u00b7i = 0 \n" +
        "(" + a1 + " + " + b1 +
        "\u00b7x + " + a2 + " + " + b2 + "\u00b7i = 0 \t || + " + (-a2) + " + " + (-
b2) + "\u00b7i \n" +
        "(" + a1 + " + " + b1 +
        "\u00b7x = " + (-a2) + " + " + (-b2) + "\u00b7i \t || :(" + a1 + " + " + b1
+ "\u00b7i) \n" +
        "          " + (-a2) + " + " + (-b2) +
        "\u00b7i \t || \u00b7(" + a1 + " + " + (-b1) + "\u00b7i) \n" +
        "x =
\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500
00\u2500\u2500 \n" +
        "          " + a1 + " + " + b1 +
        "\u00b7i \t || \u00b7(" + a1 + " + " + (-b1) + "\u00b7i) \n" +
        "          (" + (-a2) + " + " + (-b2)
+ "\u00b7i)\u00b7(" + a1 + " + " + (-b1) + "\u00b7i) \n" +
        "x =
\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500
00\u2500\u2500 \n" +
        "          (" + a1 + " + " + b1 +
        "\u00b7i)\u00b7(" + a1 + " + " + (-b1) + "\u00b7i) \n" +
        "          " + (-a2*a1) + " + " +
(a2*b1) + "\u00b7i + " + (-b2*a1) + "\u00b7i + " + (b2*b1) + "\u00b7i\u00b2 \n" +
        "x =
\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500
00\u2500\u2500 \n" +
        "          " + (a1*a1) + " + " + (-
b1*b1) + "\u00b7i\u00b2 \n" +
        "          " + (-a2*a1) + " + " +
((a2*b1)-(b2*a1)) + "\u00b7i + " + (-b2*b1) + "\n" +
        "x =
\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500
00\u2500\u2500 \n" +
        "          " + (a1*a1) + " + " +
(b1*b1) + "\n" +
        "          " + ((-a2*a1)-
(b2*b1)) + " + " + ((a2*b1)-(b2*a1)) + "\u00b7i \n" +
        "\u21d2 x =
\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500\u2500
00\u2500\u2500 = " + (((-

```

```

a2*a1)-(b2*b1))/((a1*a1)+(b1*b1))) + " +" + (((a2*b1)-(b2*a1))/((a1*a1)+(b1*b1)))
+ "\u00b7i \n" +
" " +
((a1*a1)+(b1*b1)));
    } // else

    resultatReCo1Expl.setCaretPosition(0);
} // public void actionPerformed(ActionEvent arg0)
}); // boutonResoudreReCo1.addActionListener(new ActionListener()

boutonEffacerReCo1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {

        champReCo1_A.setText("");
        champReCo1_B.setText("");
        champReCo1_C.setText("");
        champReCo1_D.setText("");
        resultatReCo1Expl.setText("");
        resultatReCo1Calc.setText("");

    } // public void actionPerformed(ActionEvent arg0)
}); // boutonEffacerReCo1.addActionListener(new ActionListener()

boutonMenuPrincipalMeCo.addActionListener(new
EcouteurBoutonMenuPrincipal());
boutonQuitterMeCo.addActionListener(new EcouteurBoutonQuitter());
boutonMenuPrincipalReCo.addActionListener(new
EcouteurBoutonMenuPrincipal());
boutonMenuComplexe.addActionListener(new
EcouteurBoutonMenuComplexe());

boutonMePr8.addActionListener(new EcouteurBoutonMenuDivers());

boutonMeDi1.addActionListener(new EcouteurBoutonReDi1());

boutonResoudreReDi1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {

        // Affectation des valeurs aux variables
        try {
            a =
Double.valueOf(champReDi1_A.getText()).doubleValue();
        } catch (NumberFormatException e) {
            a = 0;
            champReDi1_A.setText("0");
        } // catch (NumberFormatException e)
        try {
            b =
Double.valueOf(champReDi1_B.getText()).doubleValue();
        } catch (NumberFormatException e) {
            b = 0;
            champReDi1_B.setText("0");
        } // catch (NumberFormatException e)
        try {
            c =
Double.valueOf(champReDi1_C.getText()).doubleValue();
        } catch (NumberFormatException e) {

```

```

        c = 0;
        champReDi1_C.setText("0");
    } // catch (NumberFormatException e)
    try {
        d =
Double.valueOf(champReDi1_D.getText()).doubleValue();
    } catch (NumberFormatException e) {
        d = 0;
        champReDi1_D.setText("0");
    } // catch (NumberFormatException e)
    try {
        e =
Double.valueOf(champReDi1_E.getText()).doubleValue();
    } catch (NumberFormatException err) {
        e = 0;
        champReDi1_E.setText("0");
    } // catch (NumberFormatException e)
    try {
        f =
Double.valueOf(champReDi1_F.getText()).doubleValue();
    } catch (NumberFormatException e) {
        f = 0;
        champReDi1_F.setText("0");
    } // catch (NumberFormatException e)
    try {
        g =
Double.valueOf(champReDi1_G.getText()).doubleValue();
    } catch (NumberFormatException e) {
        g = 0;
        champReDi1_G.setText("0");
    } // catch (NumberFormatException e)
    try {
        h = -
Double.valueOf(champReDi1_H.getText()).doubleValue();
    } catch (NumberFormatException e) {
        h = 0;
        champReDi1_H.setText("0");
    } // catch (NumberFormatException e)

    // Resultat
    resultatReDi1.setText("Il est preferable d'utiliser la
methode de Horner pour ce cas de division polynomiale : \n" +
        "\n" +
        "\t" + a + "\t" + b + "\t" + c + "\t" + d +
        "\t" + e + "\t" + f + "\t" + g + "\n" +
        h + "\t\t" + a*h + "\t" + ((b+a*h)*h) +
        "\t" + ((c+(b+a*h)*h)*h) + "\t" + ((d+(c+(b+a*h)*h)*h)*h) + "\t" +
        ((e+(d+(c+(b+a*h)*h)*h)*h)*h) + "\t" + ((f+(e+(d+(c+(b+a*h)*h)*h)*h)*h) + "\n"
+
        "\t" + a + "\t" + (b+a*h) + "\t" +
        (c+(b+a*h)*h) + "\t" + (d+(c+(b+a*h)*h)*h) + "\t" + (e+(d+(c+(b+a*h)*h)*h)*h) +
        "\t" + (f+(e+(d+(c+(b+a*h)*h)*h)*h)*h) + "\t" +
        (g+(f+(e+(d+(c+(b+a*h)*h)*h)*h)*h)*h) + "\n" +
        "\n" +
        "Donc la \"reponse\" est : " + a +
        "\u00b7\u2075 + (b+a*h) + \u00b7\u2074 + (c+(b+a*h)*h) + \u00b7\u00b3
+ (d+(c+(b+a*h)*h)*h) + \u00b7\u00b2 + (e+(d+(c+(b+a*h)*h)*h)*h) +
\u00b7 + (f+(e+(d+(c+(b+a*h)*h)*h)*h)*h) + "\n" +

```

```

                                "Et il reste : " +
(g+(f+(e+(d+(c+(b+a*h)*h)*h)*h)*h)*h) + "\n" +
                                "Donc : " + a + "\u00b7\u2076 +" + b +
"\u00b7\u2075 +" + c + "\u00b7\u2074 +" + d + "\u00b7\u00b3 +" + e +
"\u00b7\u00b2 +" + f + "\u00b7x +" + g + " = (x +" + h + ")\u00b7(" + a +
"\u00b7\u2075 +" + (b+a*h) + "\u00b7\u2074 +" + (c+(b+a*h)*h) + "\u00b7\u00b3
+" + (d+(c+(b+a*h)*h)*h) + "\u00b7\u00b2 +" + (e+(d+(c+(b+a*h)*h)*h)*h) +
"\u00b7x +" + (f+(e+(d+(c+(b+a*h)*h)*h)*h)*h) + ") +" +
(g+(f+(e+(d+(c+(b+a*h)*h)*h)*h)*h));

                                resultatReDi1.setCaretPosition(0);

                                } // public void actionPerformed(ActionEvent arg0)
}); // boutonResoudreReDi1.addActionListener(new ActionListener()

boutonEffacerReDi1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {

        champReDi1_A.setText("");
        champReDi1_B.setText("");
        champReDi1_C.setText("");
        champReDi1_D.setText("");
        champReDi1_E.setText("");
        champReDi1_F.setText("");
        champReDi1_G.setText("");
        champReDi1_H.setText("");
        resultatReDi1.setText("");

    } // public void actionPerformed(ActionEvent arg0)
}); // boutonEffacerReDi1.addActionListener(new ActionListener()

boutonMenuPrincipalMeDi.addActionListener(new
EcouteurBoutonMenuPrincipal());
boutonQuitterMeDi.addActionListener(new EcouteurBoutonQuitter());
boutonMenuPrincipalReDi.addActionListener(new
EcouteurBoutonMenuPrincipal());
boutonMenuDivers.addActionListener(new EcouteurBoutonMenuDivers());

boutonMePr9.addActionListener(new EcouteurBoutonMenuExtras());

boutonMeEx1.addActionListener(new EcouteurBoutonReEx1());

boutonResoudreReEx1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {

        // Calcul du nombre de notes inscrits et de la somme des
notes inscrits

        nombreNotes = 12;
        sommeNotes = 0;
        moyenneArrondie = 0;

        if ("--".equals(boiteComboReEx1_1.getSelectedItem()))
            nombreNotes -= 1;
        else
            sommeNotes += Double.parseDouble((String)
(boiteComboReEx1_1.getSelectedItem()));
        if ("--".equals(boiteComboReEx1_2.getSelectedItem()))
            nombreNotes -= 1;

```

```
        else
            sommeNotes += Double.parseDouble((String)
(boiteComboReEx1_2.getSelectedItem()));
            if ("--".equals(boiteComboReEx1_3.getSelectedItem()))
                nombreNotes -= 1;
            else
                sommeNotes += Double.parseDouble((String)
(boiteComboReEx1_3.getSelectedItem()));
            if ("--".equals(boiteComboReEx1_4.getSelectedItem()))
                nombreNotes -= 1;
            else
                sommeNotes += Double.parseDouble((String)
(boiteComboReEx1_4.getSelectedItem()));
            if ("--".equals(boiteComboReEx1_5.getSelectedItem()))
                nombreNotes -= 1;
            else
                sommeNotes += Double.parseDouble((String)
(boiteComboReEx1_5.getSelectedItem()));
            if ("--".equals(boiteComboReEx1_6.getSelectedItem()))
                nombreNotes -= 1;
            else
                sommeNotes += Double.parseDouble((String)
(boiteComboReEx1_6.getSelectedItem()));
            if ("--".equals(boiteComboReEx1_7.getSelectedItem()))
                nombreNotes -= 1;
            else
                sommeNotes += Double.parseDouble((String)
(boiteComboReEx1_7.getSelectedItem()));
            if ("--".equals(boiteComboReEx1_8.getSelectedItem()))
                nombreNotes -= 1;
            else
                sommeNotes += Double.parseDouble((String)
(boiteComboReEx1_8.getSelectedItem()));
            if ("--".equals(boiteComboReEx1_9.getSelectedItem()))
                nombreNotes -= 1;
            else
                sommeNotes += Double.parseDouble((String)
(boiteComboReEx1_9.getSelectedItem()));
            if ("--".equals(boiteComboReEx1_10.getSelectedItem()))
                nombreNotes -= 1;
            else
                sommeNotes += Double.parseDouble((String)
(boiteComboReEx1_10.getSelectedItem()));
            if ("--".equals(boiteComboReEx1_11.getSelectedItem()))
                nombreNotes -= 1;
            else
                sommeNotes += Double.parseDouble((String)
(boiteComboReEx1_11.getSelectedItem()));
            if ("--".equals(boiteComboReEx1_12.getSelectedItem()))
                nombreNotes -= 1;
            else
                sommeNotes += Double.parseDouble((String)
(boiteComboReEx1_12.getSelectedItem()));

        // Calcul de la moyenne arrondie
        if ((sommeNotes/nombreNotes) < 1.25)
            moyenneArrondie = 1;
        else if ((sommeNotes/nombreNotes) >= 1.25 &&
(sommeNotes/nombreNotes) < 1.75)
```

```

        moyenneArrondie = 1.5;
    else if ((sommeNotes/nombreNotes) >= 1.75 &&
(sommeNotes/nombreNotes) < 2.25)
        moyenneArrondie = 2;
    else if ((sommeNotes/nombreNotes) >= 2.25 &&
(sommeNotes/nombreNotes) < 2.75)
        moyenneArrondie = 2.5;
    else if ((sommeNotes/nombreNotes) >= 2.75 &&
(sommeNotes/nombreNotes) < 3.25)
        moyenneArrondie = 3;
    else if ((sommeNotes/nombreNotes) >= 3.25 &&
(sommeNotes/nombreNotes) < 3.75)
        moyenneArrondie = 3.5;
    else if ((sommeNotes/nombreNotes) >= 3.75 &&
(sommeNotes/nombreNotes) < 4.25)
        moyenneArrondie = 4;
    else if ((sommeNotes/nombreNotes) >= 4.25 &&
(sommeNotes/nombreNotes) < 4.75)
        moyenneArrondie = 4.5;
    else if ((sommeNotes/nombreNotes) >= 4.75 &&
(sommeNotes/nombreNotes) < 5.25)
        moyenneArrondie = 5;
    else if ((sommeNotes/nombreNotes) >= 5.25 &&
(sommeNotes/nombreNotes) < 5.75)
        moyenneArrondie = 5.5;
    else if ((sommeNotes/nombreNotes) >= 5.75)
        moyenneArrondie = 6;

    // resultat
    if (nombreNotes % 2 != 0) {
        resultatReEx1.setText(
de : " + (sommeNotes/nombreNotes) + "\n" +
        "Votre moyenne precise actuelle est
de : " + moyenneArrondie + "\n" +
        "Votre moyenne arrondie actuelle est
        "\n" +
        "Vous devrez faire au moins " +
((5.75*(nombreNotes+1))-sommeNotes) + " au prochain test pour avoir au moins 6 de
moyenne. \n" +
        "Vous devrez faire au moins " +
((5.25*(nombreNotes+1))-sommeNotes) + " au prochain test pour avoir au moins 5.5
de moyenne. \n" +
        "Vous devrez faire au moins " +
((4.75*(nombreNotes+1))-sommeNotes) + " au prochain test pour avoir au moins 5 de
moyenne. \n" +
        "Vous devrez faire au moins " +
((4.25*(nombreNotes+1))-sommeNotes) + " au prochain test pour avoir au moins 4.5
de moyenne. \n" +
        "Vous devrez faire au moins " +
((3.75*(nombreNotes+1))-sommeNotes) + " au prochain test pour avoir au moins 4 de
moyenne. \n" +
        "Vous devrez faire au moins " +
((3.25*(nombreNotes+1))-sommeNotes) + " au prochain test pour avoir au moins 3.5
de moyenne. \n" +
        "Vous devrez faire au moins " +
((2.75*(nombreNotes+1))-sommeNotes) + " au prochain test pour avoir au moins 3 de
moyenne. \n" +

```

```

        "Vous devrez faire au moins " +
        ((2.25*(nombreNotes+1))-sommeNotes) + " au prochain test pour avoir au moins 2.5
de moyenne. \n" +
        "Vous devrez faire au moins " +
        ((1.75*(nombreNotes+1))-sommeNotes) + " au prochain test pour avoir au moins 2 de
moyenne. \n" +
        "\n" +
        "Vous devrez faire au moins un total
de " + (((5.75*(nombreNotes+2))-sommeNotes)+0.25) + " aux deux prochains tests
pour avoir au moins 6 de moyenne. \n" +
        "Vous devrez faire au moins un total
de " + (((5.25*(nombreNotes+2))-sommeNotes)+0.25) + " aux deux prochains tests
pour avoir au moins 5.5 de moyenne. \n" +
        "Vous devrez faire au moins un total
de " + (((4.75*(nombreNotes+2))-sommeNotes)+0.25) + " aux deux prochains tests
pour avoir au moins 5 de moyenne. \n" +
        "Vous devrez faire au moins un total
de " + (((4.25*(nombreNotes+2))-sommeNotes)+0.25) + " aux deux prochains tests
pour avoir au moins 4.5 de moyenne. \n" +
        "Vous devrez faire au moins un total
de " + (((3.75*(nombreNotes+2))-sommeNotes)+0.25) + " aux deux prochains tests
pour avoir au moins 4 de moyenne. \n" +
        "Vous devrez faire au moins un total
de " + (((3.25*(nombreNotes+2))-sommeNotes)+0.25) + " aux deux prochains tests
pour avoir au moins 3.5 de moyenne. \n" +
        "Vous devrez faire au moins un total
de " + (((2.75*(nombreNotes+2))-sommeNotes)+0.25) + " aux deux prochains tests
pour avoir au moins 3 de moyenne. \n" +
        "Vous devrez faire au moins un total
de " + (((2.25*(nombreNotes+2))-sommeNotes)+0.25) + " aux deux prochains tests
pour avoir au moins 2.5 de moyenne. \n" +
        "Vous devrez faire au moins un total
de " + (((1.75*(nombreNotes+2))-sommeNotes)+0.25) + " aux deux prochains tests
pour avoir au moins 2 de moyenne. ");
    } else {
        resultatReEx1.setText(
de : " + (sommeNotes/nombreNotes) + "\n" +
de : " + moyenneArrondie + "\n" +
        "\n" +
        "Vous devrez faire au moins " +
        (((5.75*(nombreNotes+1))-sommeNotes)+0.25) + " au prochain test pour avoir au
moins 6 de moyenne. \n" +
        "Vous devrez faire au moins " +
        (((5.25*(nombreNotes+1))-sommeNotes)+0.25) + " au prochain test pour avoir au
moins 5.5 de moyenne. \n" +
        "Vous devrez faire au moins " +
        (((4.75*(nombreNotes+1))-sommeNotes)+0.25) + " au prochain test pour avoir au
moins 5 de moyenne. \n" +
        "Vous devrez faire au moins " +
        (((4.25*(nombreNotes+1))-sommeNotes)+0.25) + " au prochain test pour avoir au
moins 4.5 de moyenne. \n" +
        "Vous devrez faire au moins " +
        (((3.75*(nombreNotes+1))-sommeNotes)+0.25) + " au prochain test pour avoir au
moins 4 de moyenne. \n" +
        "Vous devrez faire au moins " +
        (((3.25*(nombreNotes+1))-sommeNotes)+0.25) + " au prochain test pour avoir au
moins 3.5 de moyenne. \n" +

```

```

        "Vous devrez faire au moins " +
        (((2.75*(nombreNotes+1))-sommeNotes)+0.25) + " au prochain test pour avoir au
moins 3 de moyenne. \n" +
        "Vous devrez faire au moins " +
        (((2.25*(nombreNotes+1))-sommeNotes)+0.25) + " au prochain test pour avoir au
moins 2.5 de moyenne. \n" +
        "Vous devrez faire au moins " +
        (((1.75*(nombreNotes+1))-sommeNotes)+0.25) + " au prochain test pour avoir au
moins 2 de moyenne. \n" +
        "\n" +
        "Vous devrez faire au moins un total
de " + ((5.75*(nombreNotes+2))-sommeNotes) + " aux deux prochains tests pour avoir
au moins 6 de moyenne. \n" +
        "Vous devrez faire au moins un total
de " + ((5.25*(nombreNotes+2))-sommeNotes) + " aux deux prochains tests pour avoir
au moins 5.5 de moyenne. \n" +
        "Vous devrez faire au moins un total
de " + ((4.75*(nombreNotes+2))-sommeNotes) + " aux deux prochains tests pour avoir
au moins 5 de moyenne. \n" +
        "Vous devrez faire au moins un total
de " + ((4.25*(nombreNotes+2))-sommeNotes) + " aux deux prochains tests pour avoir
au moins 4.5 de moyenne. \n" +
        "Vous devrez faire au moins un total
de " + ((3.75*(nombreNotes+2))-sommeNotes) + " aux deux prochains tests pour avoir
au moins 4 de moyenne. \n" +
        "Vous devrez faire au moins un total
de " + ((3.25*(nombreNotes+2))-sommeNotes) + " aux deux prochains tests pour avoir
au moins 3.5 de moyenne. \n" +
        "Vous devrez faire au moins un total
de " + ((2.75*(nombreNotes+2))-sommeNotes) + " aux deux prochains tests pour avoir
au moins 3 de moyenne. \n" +
        "Vous devrez faire au moins un total
de " + ((2.25*(nombreNotes+2))-sommeNotes) + " aux deux prochains tests pour avoir
au moins 2.5 de moyenne. \n" +
        "Vous devrez faire au moins un total
de " + ((1.75*(nombreNotes+2))-sommeNotes) + " aux deux prochains tests pour avoir
au moins 2 de moyenne.");
    } // else

    resultatReEx1.setCaretPosition(0);
} // public void actionPerformed(ActionEvent arg0)
}); // boutonResoudreReEx1.addActionListener(new ActionListener()

boutonEffacerReEx1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {

        boiteComboReEx1_1.setSelectedIndex(0);
        boiteComboReEx1_2.setSelectedIndex(0);
        boiteComboReEx1_3.setSelectedIndex(0);
        boiteComboReEx1_4.setSelectedIndex(0);
        boiteComboReEx1_5.setSelectedIndex(0);
        boiteComboReEx1_6.setSelectedIndex(0);
        boiteComboReEx1_7.setSelectedIndex(0);
        boiteComboReEx1_8.setSelectedIndex(0);
        boiteComboReEx1_9.setSelectedIndex(0);
        boiteComboReEx1_10.setSelectedIndex(0);
        boiteComboReEx1_11.setSelectedIndex(0);
        boiteComboReEx1_12.setSelectedIndex(0);
    }
});

```

```

        resultatReEx1.setText("");
    } // public void actionPerformed(ActionEvent arg0)
}); // boutonEffacerReEx1.addActionListener(new ActionListener()

    boutonMenuPrincipalMeEx.addActionListener(new
EcouteurBoutonMenuPrincipal());
    boutonQuitterMeEx.addActionListener(new EcouteurBoutonQuitter());
    boutonMenuPrincipalReEx.addActionListener(new
EcouteurBoutonMenuPrincipal());
    boutonMenuExtras.addActionListener(new EcouteurBoutonMenuExtras());

    // Affectation des boutons "Quitter"
    boutonQuitterMePr.addActionListener(new EcouteurBoutonQuitter());

} // public Fenetre()

// Affectation des boutons "Retour au Menu principal"
class EcouteurBoutonMenuPrincipal implements ActionListener {
    public void actionPerformed(ActionEvent e) {

        setTitle("BleuSolve - Menu principal");
        panneau.removeAll();
        panneau.add(panneauMePrNord, BorderLayout.NORTH);
        panneau.add(panneauMePrCentre, BorderLayout.CENTER);
        panneau.add(panneauSud, BorderLayout.SOUTH);
        panneau.setVisible(false);
        panneau.setVisible(true);

    } // public void actionPerformed(ActionEvent e)
} // class EcouteurBoutonMenuPrincipal implements ActionListener

// Affectation des boutons "Menu resoudre une equation a une inconnue"
class EcouteurBoutonMenuEquation implements ActionListener {
    public void actionPerformed(ActionEvent e) {

inconnue");

        panneau.removeAll();
        panneau.add(panneauMeEqNord, BorderLayout.NORTH);
        panneau.add(panneauMeEqCentre, BorderLayout.CENTER);
        panneau.add(panneauSud, BorderLayout.SOUTH);
        panneau.setVisible(false);
        panneau.setVisible(true);

    } // public void actionPerformed(ActionEvent e)
} // class EcouteurBoutonMenuEquation implements ActionListener

class EcouteurBoutonReEq1 implements ActionListener {
    public void actionPerformed(ActionEvent e) {

        // Fixation des panneaux de resolution
        panneauReEq1Centre.setLayout(new BorderLayout());
        panneauReEq1Centre.add(panneauReEq1CentreNord,
BorderLayout.NORTH);
        panneauReEq1Centre.add(panneauDefilReEq1CentreCentre,
BorderLayout.CENTER);

```

```

        panneauReEq1Centre.add(panneauReEqCentreSud,
BorderLayout.SOUTH);

        setTitle("BleuSolve - Resoudre une equation du premier degre a
une inconnue");
        panneau.removeAll();
        panneau.add(panneauReEqNord, BorderLayout.NORTH);
        panneau.add(panneauReEq1Centre, BorderLayout.CENTER);
        panneau.add(panneauSud, BorderLayout.SOUTH);
        panneau.setVisible(false);
        panneau.setVisible(true);

    } // public void actionPerformed(ActionEvent e)
}; // class EcouteurBoutonReEq1 implements ActionListener

class EcouteurBoutonReEq2 implements ActionListener {
    public void actionPerformed(ActionEvent e) {

        // Fixation des panneaux de resolution
        panneauReEq2Centre.setLayout(new BorderLayout());
        panneauReEq2Centre.add(panneauReEq2CentreNord,
BorderLayout.NORTH);
        panneauReEq2Centre.add(panneauDefilReEq2CentreCentre,
BorderLayout.CENTER);
        panneauReEq2Centre.add(panneauReEqCentreSud,
BorderLayout.SOUTH);

        setTitle("BleuSolve - Resoudre une equation du deuxieme degre a
une inconnue");
        panneau.removeAll();
        panneau.add(panneauReEqNord, BorderLayout.NORTH);
        panneau.add(panneauReEq2Centre, BorderLayout.CENTER);
        panneau.add(panneauSud, BorderLayout.SOUTH);
        panneau.setVisible(false);
        panneau.setVisible(true);

    } // public void actionPerformed(ActionEvent e)
}; // class EcouteurBoutonReEq2 implements ActionListener

class EcouteurBoutonReEq3 implements ActionListener {
    public void actionPerformed(ActionEvent e) {

        // Fixation des panneaux de resolution
        panneauReEq3Centre.setLayout(new BorderLayout());
        panneauReEq3Centre.add(panneauReEq3CentreNord,
BorderLayout.NORTH);
        panneauReEq3Centre.add(panneauDefilReEq3CentreCentre,
BorderLayout.CENTER);
        panneauReEq3Centre.add(panneauReEqCentreSud,
BorderLayout.SOUTH);

        setTitle("BleuSolve - Resoudre une equation \"simple\" du
troisieme degre a une inconnue");
        panneau.removeAll();
        panneau.add(panneauReEqNord, BorderLayout.NORTH);
        panneau.add(panneauReEq3Centre, BorderLayout.CENTER);
        panneau.add(panneauSud, BorderLayout.SOUTH);
        panneau.setVisible(false);
        panneau.setVisible(true);
    }
};

```

```

    } // public void actionPerformed(ActionEvent e)
}; // class EcouteurBoutonReEq3 implements ActionListener

// Affectation des boutons "Menu resoudre un systeme d'equations a plusieurs
inconnues"
class EcouteurBoutonMenuSysteme implements ActionListener {
    public void actionPerformed(ActionEvent e) {

        setTitle("BleuSolve - Menu resoudre un systeme d'equations a
plusieurs inconnues");
        panneau.removeAll();
        panneau.add(panneauMeSyNord, BorderLayout.NORTH);
        panneau.add(panneauMeSyCentre, BorderLayout.CENTER);
        panneau.add(panneauSud, BorderLayout.SOUTH);
        panneau.setVisible(false);
        panneau.setVisible(true);

    } // public void actionPerformed(ActionEvent e)
} // class EcouteurBoutonMenuSysteme implements ActionListener

class EcouteurBoutonReSy2 implements ActionListener {
    public void actionPerformed(ActionEvent e) {

        // Fixation des panneaux de resolution
        panneauReSy2Centre.setLayout(new BorderLayout());
        panneauReSy2Centre.add(panneauReSy2CentreNord,
BorderLayout.NORTH);
        panneauReSy2Centre.add(panneauDefilReSy2CentreCentre,
BorderLayout.CENTER);
        panneauReSy2Centre.add(panneauReSy2CentreSud,
BorderLayout.SOUTH);

        setTitle("BleuSolve - Resoudre un systeme de deux equations a
deux inconnues");
        panneau.removeAll();
        panneau.add(panneauReSyNord, BorderLayout.NORTH);
        panneau.add(panneauReSy2Centre, BorderLayout.CENTER);
        panneau.add(panneauSud, BorderLayout.SOUTH);
        panneau.setVisible(false);
        panneau.setVisible(true);

    } // public void actionPerformed(ActionEvent e)
} // class EcouteurBoutonReSy2 implements ActionListener

// Affectation des boutons "Menu Nombre complexe"
class EcouteurBoutonMenuComplexe implements ActionListener {
    public void actionPerformed(ActionEvent e) {

        setTitle("BleuSolve - Menu nombre complexe");
        panneau.removeAll();
        panneau.add(panneauMeCoNord, BorderLayout.NORTH);
        panneau.add(panneauMeCoCentre, BorderLayout.CENTER);
        panneau.add(panneauSud, BorderLayout.SOUTH);
        panneau.setVisible(false);
        panneau.setVisible(true);

    } // public void actionPerformed(ActionEvent e)
} // class EcouteurBoutonMenuComplexe implements ActionListener

```

```

class EcouteurBoutonReCo0 implements ActionListener {
    public void actionPerformed(ActionEvent e) {

        // Fixation des panneaux de resolution
        panneauReCo0Centre.setLayout(new BorderLayout());
        panneauReCo0Centre.add(panneauReCo0CentreNord,
BorderLayout.NORTH);
        panneauReCo0Centre.add(panneauDefilReCo0CentreCentre,
BorderLayout.CENTER);
        panneauReCo0Centre.add(panneauReCoCentreSud,
BorderLayout.SOUTH);

        setTitle("BleuSolve - Calculer les racines carrees d'un nombre
complexe");

        panneau.removeAll();
        panneau.add(panneauReCoNord, BorderLayout.NORTH);
        panneau.add(panneauReCo0Centre, BorderLayout.CENTER);
        panneau.add(panneauSud, BorderLayout.SOUTH);
        panneau.setVisible(false);
        panneau.setVisible(true);

    } // public void actionPerformed(ActionEvent e)
} // class EcouteurBoutonReCo0 implements ActionListener

class EcouteurBoutonReCo1 implements ActionListener {
    public void actionPerformed(ActionEvent e) {

        // Fixation des panneaux de resolution
        panneauReCo1Centre.setLayout(new BorderLayout());
        panneauReCo1Centre.add(panneauReCo1CentreNord,
BorderLayout.NORTH);
        panneauReCo1Centre.add(panneauDefilReCo1CentreCentre,
BorderLayout.CENTER);
        panneauReCo1Centre.add(panneauReCoCentreSud,
BorderLayout.SOUTH);

        setTitle("BleuSolve - Resoudre une equation du premier degre a
une inconnue (Nombre complexe)");
        panneau.removeAll();
        panneau.add(panneauReCoNord, BorderLayout.NORTH);
        panneau.add(panneauReCo1Centre, BorderLayout.CENTER);
        panneau.add(panneauSud, BorderLayout.SOUTH);
        panneau.setVisible(false);
        panneau.setVisible(true);

    } // public void actionPerformed(ActionEvent e)
} // class EcouteurBoutonReCo1 implements ActionListener

// Affectation des boutons "Divers"
class EcouteurBoutonMenuDivers implements ActionListener {
    public void actionPerformed(ActionEvent e) {

        setTitle("BleuSolve - Menu divers");
        panneau.removeAll();
        panneau.add(panneauMeDiNord, BorderLayout.NORTH);
        panneau.add(panneauMeDiCentre, BorderLayout.CENTER);
        panneau.add(panneauSud, BorderLayout.SOUTH);
        panneau.setVisible(false);
    }
}

```

```

        panneau.setVisible(true);
    } // public void actionPerformed(ActionEvent e)
} // class EcouteurBoutonMenuDivers implements ActionListener

class EcouteurBoutonReDi1 implements ActionListener {
    public void actionPerformed(ActionEvent e) {

        // Fixation des panneaux de resolution
        panneauReDi1Centre.setLayout(new BorderLayout());
        panneauReDi1Centre.add(panneauReDi1CentreNord,
BorderLayout.NORTH);
        panneauReDi1Centre.add(panneauDefilReDi1CentreCentre,
BorderLayout.CENTER);
        panneauReDi1Centre.add(panneauReDiCentreSud,
BorderLayout.SOUTH);

        setTitle("BleuSolve - Division polynomiale (avec la methode de
Horner)");

        panneau.removeAll();
        panneau.add(panneauReDiNord, BorderLayout.NORTH);
        panneau.add(panneauReDi1Centre, BorderLayout.CENTER);
        panneau.add(panneauSud, BorderLayout.SOUTH);
        panneau.setVisible(false);
        panneau.setVisible(true);

    } // public void actionPerformed(ActionEvent e)
} // class EcouteurBoutonReDi1 implements ActionListener

// Affectation des boutons "Extras"
class EcouteurBoutonMenuExtras implements ActionListener {
    public void actionPerformed(ActionEvent e) {

        setTitle("BleuSolve - Menu extras");
        panneau.removeAll();
        panneau.add(panneauMeExNord, BorderLayout.NORTH);
        panneau.add(panneauMeExCentre, BorderLayout.CENTER);
        panneau.add(panneauSud, BorderLayout.SOUTH);
        panneau.setVisible(false);
        panneau.setVisible(true);

    } // public void actionPerformed(ActionEvent e)
} // class EcouteurBoutonMenuExtras implements ActionListener

class EcouteurBoutonReEx1 implements ActionListener {
    public void actionPerformed(ActionEvent e) {

        // Fixation des panneaux de resolution
        panneauReEx1Centre.setLayout(new BorderLayout());
        panneauReEx1Centre.add(panneauReEx1CentreNord,
BorderLayout.NORTH);
        panneauReEx1Centre.add(panneauDefilReEx1CentreCentre,
BorderLayout.CENTER);
        panneauReEx1Centre.add(panneauReExCentreSud,
BorderLayout.SOUTH);

        setTitle("BleuSolve - Calculeur de moyenne");
        panneau.removeAll();
        panneau.add(panneauReExNord, BorderLayout.NORTH);

```

```
        panneau.add(panneauReEx1Centre, BorderLayout.CENTER);
        panneau.add(panneauSud, BorderLayout.SOUTH);
        panneau.setVisible(false);
        panneau.setVisible(true);

    } // public void actionPerformed(ActionEvent e)
} // class EcouteurBoutonReEx1 implements ActionListener

// Affectation des boutons "Quitter"
class EcouteurBoutonQuitter implements ActionListener {
    public void actionPerformed(ActionEvent e) {

        if (JOptionPane.showConfirmDialog(new JFrame(),
            "Etes-vous sur de vouloir quitter BleuSolve et \n"
+
            "supprimer tous les fichiers PDF crees ?",
            "BleuSolve - Quitter",
            JOptionPane.YES_NO_OPTION) ==
JOptionPane.YES_OPTION) {
            pdfCreeReEq2.delete();
            pdfCreeCramer.delete();
            pdfCreeHorner.delete();
            pdfCreeSignes.delete();
            pdfCreeComplexe.delete();
            pdfCreeManuel.delete();
            System.exit(0);
        } // if

    } // public void actionPerformed(ActionEvent e)
} // class EcouteurBoutonQuitter implements ActionListener

} // class Fenetre extends JFrame
```

## 8. Table des illustrations

Image 1 : Fichier "BleuSolve" .....	5
Image 2 : Image d'attente .....	5
Exemple 1 : Explications des menus.....	6
Image 3 : La barre de menu.....	6
Image 4 : La barre de menu : Fichier .....	7
Image 5 : La barre de menu : Théorie .....	7
Image 6 : La barre de menu : Aller à .....	7
Image 7 : La barre de menu : Aide .....	8
Exemple 2 : Explications des fenêtres de résolution .....	8
Exemple 3 : Explications des résultats .....	9
Image 8 : Menu principal.....	10
Image 9 : Menu résoudre une équation à une inconnue .....	11
Image 10 : Résoudre une équation du premier degré à une inconnue .....	11
Image 11 : Résoudre une équation du premier degré à une inconnue - Résultat.....	12
Image 12 : Résoudre une équation du deuxième degré à une inconnue .....	12
Image 13 : Résoudre une équation du deuxième degré à une inconnue – Résultat.....	13
Image 14 : Résoudre une équation du troisième degré "simple" à une inconnue.....	13
Image 15 : Résoudre une équation du troisième degré "simple" à une inconnue – Résultat	14
Image 16 : Menu résoudre un système d'équation à plusieurs inconnues .....	15
Image 17 : Résoudre un système de deux équations à deux inconnues .....	15
Image 18 : Résoudre un système de deux équations à deux inconnues – Résultat .....	16
Image 19 : Menu nombre complexe .....	17
Image 20 : Calculer les racines carrées d'un nombre complexe .....	17
Image 21 : Calculer les racines carrées d'un nombre complexe - Résultat .....	18
Image 22 : Résoudre une équation du premier degré à une inconnue (Nombre complexe)..	18
Image 23 : Résoudre une équation du premier degré à une inconnue (Nombre complexe) - Résultat.....	19
Image 24 : Menu divers.....	19
Image 25 : Division polynomiale (avec la méthode de Horner).....	20
Image 26 : Division polynomiale (avec la méthode de Horner) - Résultat .....	20
Image 27 : Menu extras.....	21
Image 28 : Calculateur de moyenne .....	21
Image 29 : Calculateur de moyenne – Résultat .....	22
Image 30 : Boîte de confirmation avant de quitter.....	22
Image 31 : Algorithme d'Euclide .....	29
Image 32 : Représentation graphique d'un nombre complexe .....	38

## 9. Bibliographie

### 9.1. Crédit iconographique

Toutes les copies d'écran proviennent du logiciel BleuSolve fonctionnant sous Windows 7.

Le logo de BleuSolve a été dessiné par David Wagnières.

<http://fr.wikipedia.org/w/index.php?title=Fichier%3APGCD.png>

[http://fr.wikipedia.org/wiki/Fichier:Complex\\_number.svg](http://fr.wikipedia.org/wiki/Fichier:Complex_number.svg)

### 9.2. Ouvrages consultés

HESS, Jean-Pierre, *Support de cours Java, comem+ heig-vd*, jean-pierre.hess@heig-vd.ch

DELANNOY, Claude, *Programmer en Java, 5<sup>e</sup> édition Java 5 et 6*, Edition EYROLLES, 2008

### 9.3. Sites internet

<http://forum.macbidouille.com/lofiversion/index.php/t124341.html>

[http://java.developpez.com/faq/gui/?page=graphique\\_fenetres\\_composants#GRAPHIQUE\\_fenetre](http://java.developpez.com/faq/gui/?page=graphique_fenetres_composants#GRAPHIQUE_fenetre)

<http://www.clubic.com/forum/programmation/lancer-un-pdf-a-partir-de-java-id492264-page1.html>

<http://www.developpez.net/forums/d105240/environnements-developpement/eclipse/eclipse-java/info-conseils-gerer-ressources/>

[http://www.developpez.net/forums/d540178/java/interfaces-graphiques-java/awt-swing/changer-iframe-exit\\_on\\_close-default/](http://www.developpez.net/forums/d540178/java/interfaces-graphiques-java/awt-swing/changer-iframe-exit_on_close-default/)

[http://www.fobec.com/CMS/java/sources/envoyer-email-avec-logiciel-par-defaut\\_974.html](http://www.fobec.com/CMS/java/sources/envoyer-email-avec-logiciel-par-defaut_974.html)

<http://www.siteduzero.com/forum-83-514640-p1-java-mettre-un-scrollbar-pour-une-liste-de-bouton.html>

<http://www.siteduzero.com/forum-83-671468-p1-lancer-un-pdf-avec-un-jar.html>

<http://www.siteduzero.com/forum-83-705581-p1-barres-de-defilement.html>

## 10. Remerciements

Je remercie chaleureusement :

- Monsieur Pierre Breguet pour avoir proposé le sujet de « Programmation en Java » et suivi ce travail de maturité
- Monsieur Olivier Desponds pour avoir donné son avis sur le logiciel et pour ses cours de mathématiques
- David Wagnières et Guillaume Guenat pour avoir dessiné les différents logos de BleuSolve
- Les alpha- et bêta-testeurs (Cédric Favre, Guillaume Guenat, Quentin Jaquier, Adrien Jeckelmann, Thomas Mizraji, Bastien Morleo, Martin Quartier, Damien Scantamburlo, Lindor Shala, David Wagnières et Robin Walther) pour avoir testé le programme, donné leurs avis et signalé les éventuels problèmes
- Tous ceux qui ont soutenu BleuSolve implicitement
- Le logiciel « Eclipse » qui facilite la programmation en Java et l'exportation de nos programmes

## 11. Liens

<http://www.facebook.com/BleuSolve>

<http://bleusolve.wordpress.com/>

Tinh Di David Tang, le samedi 19 novembre 2011