

Comment concevoir une solution informatique - Exemple

Ceci est un exemple illustrant quelques points de la théorie exposée dans le document « Comment concevoir une solution informatique ».

1. Bien **analyser** et **comprendre** le problème (pas la solution !) :

On me demande de créer une application pour résoudre l'équation du 2^{ème} degré. Je devrai appliquer la méthode vue en cours de math pour calculer les 2 racines.

2. Réaliser **à la main sur papier**, les étapes nécessaires à trouver le résultat demandé :

Créez une application permettant de résoudre une équation du 2^{ème} degré

Exemple : $5x^2 - 3x + 4 = 0$, combien vaut x ?
 $\begin{matrix} a & b & c \end{matrix}$

$$\Delta = b^2 - 4ac = (-3)^2 - 4 \cdot 5 \cdot 4 = -71$$

Si $\Delta < 0$ alors pas de solution réelle

2^{ème} exemple : $2y^2 + 3y - 4 = 0$
 $\begin{matrix} a & b & c \end{matrix}$

$$\Delta = b^2 - 4ac = 3^2 - 4 \cdot 2 \cdot (-4) = 41$$

Si $\Delta > 0$ alors 2 solutions :

$$y_{1,2} = \frac{-b \pm \sqrt{\Delta}}{2a}$$

$$y_1 = \frac{-3 + \sqrt{41}}{2 \cdot 2} = 0,851$$

$$y_2 = \frac{-3 - \sqrt{41}}{2 \cdot 2} = -2,351$$

3^{ème} exemple : $z^2 + 4z + 4 = 0$
 $\begin{matrix} a & b & c \end{matrix}$

$$\Delta = b^2 - 4ac = 4^2 - 4 \cdot 1 \cdot 4 = 0$$

Si $\Delta = 0$ alors 1 solution :

$$z = -\frac{b}{2a} = -\frac{4}{2 \cdot 1} = -2$$

3. **Analyser** les étapes réalisées à la main :

J'ai déterminé les coefficients a , b et c

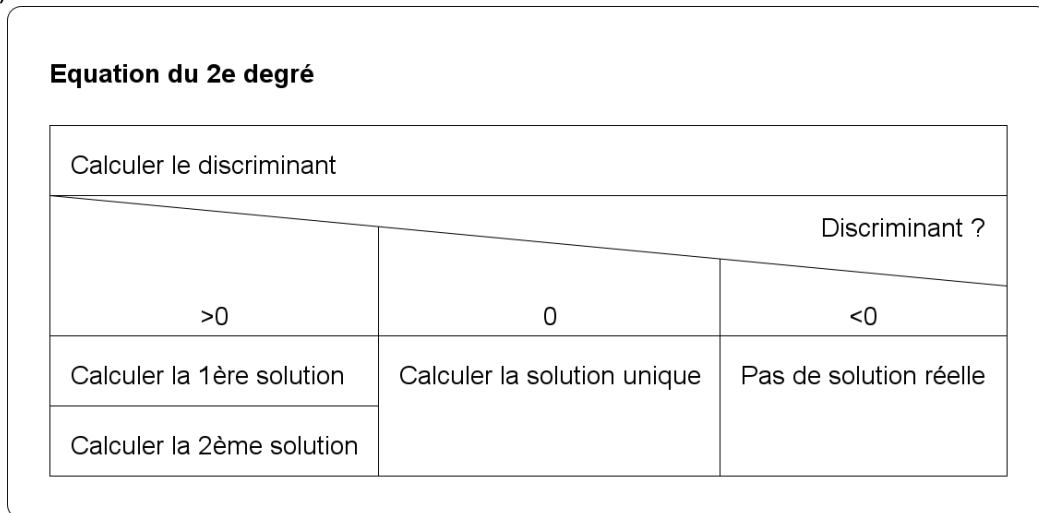
Ensuite j'ai calculé le discriminant selon une formule connue

Selon que le discriminant est positif, nul ou négatif, j'ai appliqué la bonne formule pour calculer le résultat, qui peut être soit 2 racines, soit 1 racine, soit aucune racine réelle

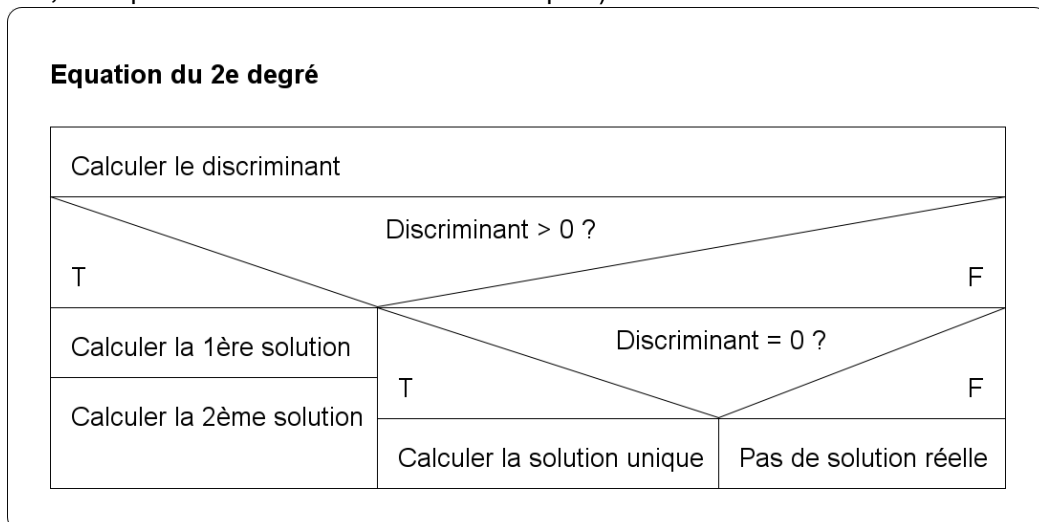
4. Décrire en **mots-clés** une **démarche** à appliquer à n'importe quel cas pour atteindre le résultat demandé :

- 1) Demander les valeurs des coefficients a, b et c
- 2) Calculer le discriminant
- 3) Si le discriminant est positif, alors calculer les 2 racines
 Si le discriminant est nul, alors calculer la racine unique
 Si le discriminant est négatif, alors il n'y a pas de racine réelle

5. Concevoir une solution sous forme d'**algorithme** (structogramme [GNS](#) ou [JSP](#), [pseudo-code](#), etc.) :



6. On peut **affiner** ensuite le structogramme pour tenir compte des **spécificités du langage** utilisé pour le codage qui suivra (p. ex. la structure **switch case** n'étant pas très efficace en C# V7.3, il est préférable d'utiliser des **if** imbriqués) :



7. **Tester** le fonctionnement de l'algorithme (structogramme) **sur papier** en passant par tous les chemins possibles

8. Traduire cet algorithme sous forme d'un **programme** en un langage de programmation adéquat (**commenter** le programme pour qu'il soit compréhensible)

Bien utilisé¹, le logiciel *Structorizer* permet de générer le code source en **C#** (la structure et commentaires) qui correspond au structogramme et de copier la partie qui nous intéresse :

Equation du 2e degré

Calculer le discriminant		
Discriminant > 0 ?		
T	F	
Calculer la 1ère solution	Discriminant = 0 ?	
	T	F
Calculer la 2ème solution	Calculer la solution unique	
	Pas de solution réelle	

```
// Generated by Structorizer 3.32-34

/// <summary>
/// Equation du 2e degré
/// </summary>
public static void Equation_du_2e_degre()
{
    // TODO: Check and accomplish variable declarations:

    // Calculer le discriminant
    ?;
    // Discriminant > 0 ?
    if (?)
    {
        // Calculer la 1ère solution
        ?;
        // Calculer la 2ème solution
        ?;
    }
    // Discriminant = 0 ?
    else if (?)
    {
        // Calculer la solution unique
        ?;
    }
    else
    {
        // Pas de solution réelle
        ?;
    }
}
```

Et la coller dans la solution *Application Windows Forms (.NET Framework)* écrite en **VC#** :

The screenshot shows the Visual Studio IDE with the following components:

- Code Editor:** Displays the C# code for `Form1`. The logic for solving the quadratic equation is pasted from the Structorizer output, enclosed in a red box. It includes comments in French and uses `?` as placeholders for variable declarations.
- Design View:** Shows a Windows Form with a single button labeled "Calculer".
- Solution Explorer:** Shows the project structure with files like `Form1.cs`, `Program.cs`, and `App.config`.
- Properties Window:** Shows the properties for the `Form1` control, including accessibility and appearance settings.

Il reste ensuite à écrire du code correspondant aux commentaires, partout où il y a des points d'interrogation

¹ Voir document « Utiliser Structorizer pour faciliter le passage au code »

9. Faire traduire le programme du langage évolué au langage compréhensible par le processeur (compiler)
10. Faire exécuter ce programme par le processeur (débuguer)
11. Tester le programme en remplissant un protocole