



3. Ajoutez une méthode **P(double x)** pour évaluer le polynôme. On donne la valeur de x et la méthode nous donne la valeur de **P(x)**.
4. Ajoutez une méthode **getDelta()** qui calcule le discriminant d'un polynôme.
5. On veut ajouter le comportement **calculRacines()** aux instances de la classe Equation. Cette méthode doit calculer les racines et affecter leurs valeurs aux attributs r1 et r2.

Remarque : on ne s'intéresse qu'aux racines réelles. Si l'équation possède une racine double, on affecte la même valeur aux deux attributs r1 et r2.

6. Écrire une méthode **ToString** qui convertit une instance en chaîne de caractères. Cette méthode doit avoir le comportement suivant :

- Si les racines n'ont pas été calculées, ToString renvoie par exemple la chaîne de caractères :
Equation 2*x^2 + 3*x + 1 = 0. Tout réel est solution ou il n'y a pas de solution
- Si le discriminant est positif, ToString renvoie par exemple la chaîne de caractères :
Equation 2*x^2 + 3*x + 1 = 0. Il y deux solutions égales a r1 et r2 (voir exemple en bas)
- Si le discriminant est strictement négatif, ToString renvoie par exemple la chaîne de caractères:
Equation 1*x^2 + 1*x + 4 = 0. Pas de racine réelle.

Exemple d'exécution :

```
Entrer une valeur pour a : 12
Entrer une valeur pour b : 300
Entrer une valeur pour c : 12
12x^2+300x+12=0
il y deux solutions egales a -0,040064205622888 et -24,9599357943771
```

EXERCICE 4 - Le temps qui passe

On veut écrire un programme qui permet de convertir des secondes, en minutes et en heures.

1. Proposez une classe **Temps** avec une méthode **String convertir(int secondes)** qui prend en paramètre un nombre de secondes et renvoie une chaîne de caractères qui donne le nombre d'heures et de minutes représentés par ce nombre de secondes.

Par exemple, l'appel **t.convertir(4567)** doit renvoyer la chaîne "1h 16mn 7s".

2. Comment faire pour tester cette méthode ?
3. Y a-t-il un intérêt à avoir plusieurs instances de la classe Temps ?



EXERCICE 5 – Gestion d'un établissement

Un établissement d'enseignement général souhaite informatiser la gestion des professeurs et des élèves.

- 1) Créer la classe `Personne` définit par les propriétés privées suivantes : `Code`, `Nom`, `Prénom`
- 2) Ajouter à cette classe un constructeur par défaut qui initialise le code par un nombre aléatoire, le nom et le prénom par « Inconnu » et un constructeur qui permet d'initialiser toutes les propriétés.
- 3) Un professeur est une personne avec les attributs supplémentaires : `Diplôme` et `Spécialité`. Créer cette classe avec les constructeurs nécessaires.
- 4) Un élève est une personne avec les attributs supplémentaires : `Section` et `Nombre de jours d'absence (NbAbs)`. Créer cette classe avec les constructeurs nécessaires.
- 5) Ajouter la méthode `Afficher()` qui permet d'afficher les informations d'une personne. Et redéfinir cette méthodes dans les classes `Professeur` et `Elève`.
- 6) Ajouter la méthode `IncAbsence()` qui permet d'incrémenter le nombre de jours d'absence d'un élève de 1.
- 7) Dans la fonction `Main` de la classe `program`, réaliser le test de ces classes :
 - a. Créer une personne `P1` caractérisée par : `code = 1`, `Nom = Khalil` et `Prénom = Morad`
 - b. Créer un professeur `P2` caractérisée par : `code = 2`, `Nom = Lahlou`, `Prénom = Nabil`, `Diplôme = DESS` et `Spécialité = Mathématique`.
 - c. Créer un élève `P3` caractérisée par : `code = 3`, `Nom = Talal`, `Prénom = Aziz`, `Section = TDI` et `Nombre de jours d'absence = 5`
 - d. Afficher les caractéristiques des objets `P1`, `P2` et `P3`
 - e. Ajouter 2 jours d'absence à `P3`
 - f. Changer la spécialité de `P2` en `Physique` au lieu de `Mathématique`.
 - g. Afficher de nouveau les caractéristiques des objets `P1`, `P2` et `P3`