

Exercice 1

```
package initial;
public class Inverse
{
    public static void main(String[] args)
    {
        ConsoleTexte mc=new ConsoleTexte();
        mc.ecritLn("Ecris un nombre et je calculerai son inverse");
        double valeur=mc.getValeur();
        if (valeur!=0){
            mc.ecritLn("Je peux calculer l'inverse de ce nombre non nul");
            mc.ecritLn("L'inverse de "+valeur+" est "+1/valeur);
        }
    }
}
```

Compléter ce programme pour qu'il affiche comme précédemment l'inverse de la valeur saisie si elle est non nulle, ou "je ne peux pas calculer l'inverse de 0" quand l'utilisateur entre une valeur nulle.

```
package initial;
public class Inverse
{
    public static void main(String[] args)
    {
        ConsoleTexte mc=new ConsoleTexte();
        mc.ecritLn("Ecris un nombre et je calculerai son inverse");
        double valeur=mc.getValeur();
        if (valeur!=0){
            mc.ecritLn("Je peux calculer l'inverse de ce nombre non nul");
            mc.ecritLn("L'inverse de "+valeur+" est "+1/valeur);
        }
        else mc.ecritLn("Je ne peux pas calculer l'inverse de 0");
    }
}
```

On ajoute une clause else (sur fond vert).

Attention : bien penser à enlever le point-virgule à la ligne précédente (le point-virgule "fermant" le bloc qui commence à "if ..." se trouve maintenant à la fin de la ligne ajoutée.)

Exercice 3

Déclarer une variable de type double nommée nombre et lui affecter la valeur 1.

Ecrire une ligne de commande qui multiplie nombre par 4 et affecte le résultat à nombre.

Afficher la valeur de nombre. Modifier le programme à l'aide d'une boucle pour qu'il affiche toutes les puissances de 4 inférieures ou égales à 1000.

```
public class PuissancesDe4 {
public static void main(String[] args) {
    double nombre=1;//Déclare une variable de type double nommée nombre et lui affecte la valeur 1.
    while (nombre<1000){ // les instructions suivantes jusqu'à "}" seront exécutées tant que nombre reste inférieur à 1000
        System.out.println(nombre);//Affiche la valeur de nombre
        nombre=nombre*4;//multiplie nombre par 4 et affecte le résultat à nombre
    }
}
}
```

Exercice 4

En vous inspirant du programme de l'exercice précédent, écrire un programme qui affiche tous les multiples de 4 inférieurs ou égaux à 1000.

```
public class MultiplesDe4 {
public static void main(String[] args) {
    double nombre=0;//Déclare une variable de type double nommée nombre et lui affecte la valeur 1.
    while (nombre<1000){ // les instructions suivantes jusqu'à "}" seront exécutées tant que nombre reste inférieur à 1000
        System.out.println(nombre); //Affiche la valeur de nombre
        Nombre=nombre+4; //ajoute 4 à nombre et affecte le résultat à nombre
    }
}
}
```

Exercice 5

System.currentTimeMillis() permet d'obtenir l'heure du système en millisecondes, dans une variable de type long. Ecrire un programme Pause contenant une boucle qui dure 5 seconde.

(suggestion de réalisation : lire la valeur initiale de l'heure puis lancer une boucle qui tourne tant que l'heure est inférieure à l'heure initiale augmentée de 5000)

```
public class Pause {
public static void main(String[] args) {
    System.out.println("Pause..."); //facultatif
    long t=System.currentTimeMillis(); //lit la valeur initiale de l'heure
}
```

```
        while (System.currentTimeMillis()<t+5000){
            //cette boucle ne fait rien. Elle tourne simplement tant que l'heure est inférieure à t+5000
            }
        System.out.println("Fini!"); //facultatif
    }
}
```

Exercice 6

En utilisant une boucle for, afficher les 10 premiers éléments de la table de 7

```
package initial;
public class Table7 {
public static void main(String[] args) {
    ConsoleTexte mc=new ConsoleTexte();
    for (int i=0;i<10*7;i=i+7){
        mc.ecritLn(i);
    }
    // ou bien
    for (int i=0;i<10;i=i+1){
        mc.ecritLn(i*7);
    }
}
```

Exercice 7

En utilisant une boucle for, afficher 10 caractères identiques sur une même ligne, puis sur une même colonne

```
package initial;
public class LigneColonne {
public static void main(String[] args) {
    ConsoleTexte mc=new ConsoleTexte();

    // ligne de 10 caractères identiques :
    for (int i=0;i<10;i=i+1){
        mc.ecrit('@'); // on écrit 10 fois le caractère sans aller à la ligne
    }

    // colonne de 10 caractères identiques :
    for (int i=0;i<10;i=i+1){
        mc.ecritLn('@'); // on écrit 10 fois le caractère en allant à la ligne
    }
}
```

```
}  
}  
}
```

Exercice 8

Utiliser deux boucles for imbriquées pour positionner des caractères identiques selon un rectangle plein

```
package initial;  
public class Rectangle {  
    public static void main(String[] args) {  
        ConsoleTexte mc=new ConsoleTexte();  
        for (int i=0;i<5;i=i+1){ // effectue 5 fois les instructions suivantes jusqu'au deuxième prochain "}"  
            for (int j=0;j<10;j++){ // effectue 10 fois les instructions suivantes, jusqu'au prochain "}"  
                mc.ecrit('@'); // ecrit un caractère sans aller à la ligne  
            }  
            mc.ecritLn(""); // ecrit une chaine vide et passe à la ligne suivante  
        }  
    }  
}
```

Exercice 9

Utiliser deux boucles for imbriquées pour positionner des caractères identiques selon un triangle rectangle plein.

```
package initial;  
public class TriangleRectangle {  
  
    public static void main(String[] args) {  
        ConsoleTexte mc=new ConsoleTexte();  
        for (int i=0;i<5;i=i+1){  
            for (int j=0;j<2*i+1;j++){  
                mc.ecrit('@');  
            }  
            mc.ecritLn("");  
        }  
    }  
}
```

La logique est la même qu'au programme de l'exercice 8 mais le nombre de caractères écrits sur une même ligne dépend de i , il change donc à chaque ligne.