TD 5 : structures (éléments de correction)

Exercice 1 : Date

```
Soit structure Date
{
        champ annee : entier
        champ mois : entier
        champ jour : entier
}
```

Algorithme 2 : Créer une date sans vérification
Principe : Créer un objet Date et initialiser les trois champs avec les entrées.

```
Entrée :        xJour : entier
                xMois : entier
                xAnnee : entier
Local :         /
Sortie :        date : Date
Début
|       Soit date : Date
|       date.jour = xJour
|       date.mois = xMois
|       date.annee = xAnnee
Fin
```

Algorithme 3 : Afficher une date au format JJ/MM/AAAA
Principe : Afficher un chaîne de caractères formatée en insérant les trois champs dans l'ordre.
Préconditions : la date n'est pas null

```
Entrée :        xDate : structure Date
Local :         /
Sortie :        void
Début
|       Afficher("Date : {xDate.jour}/{xDate.mois}/{xDate.annee}")
Fin
```

Algorithme 4 : Créer une date avec vérification de validité
Principe : Vérifier que la date est valide sur le même principe que le TD1, exercice 6, si oui initialiser les champs de la structure, sinon erreur.

```
Entrée :        xJour : entier
                xMois : entier
                xAnnee : entier
Local :         estBissextile : booléen
                nombreJoursDansMois : entier
Sortie :        date : structure Date
Début
|       Soit date : structure Date
|       Soit estBissextile : booléen = faux
|       Soit nombreJoursDansMois : entier = 31
|       Si (xAnnee < 0) Alors
|       |       Afficher("Erreur, année {xAnnee} invalide !")
|       Sinon
|       |       Si (xAnnee divisible par 4 et non divisible par 100) ou xAnnee divisible par 400 Alors
|       |       |       estBissextile = vrai
|       |       Fin Si
|       |       Si (xMois < 0 ou xMois > 12) Alors
|       |       |       Afficher("Erreur, mois {xMois} invalide !")
|       |       Sinon
|       |       |       Si xMois == 4 ou xMois == 6 ou xMois == 9 ou xMois == 11 Alors
|       |       |       |       nombreJoursDansMois = 30
|       |       |       Sinon Si xMois == 2 Alors
|       |       |       |       Si estBissextile == vrai Alors
|       |       |       |       |       nombreJoursDansMois = 29
|       |       |       |       Sinon
|       |       |       |       |       nombreJoursDansMois = 28
|       |       |       |       Fin Si
|       |       |       |       Si xJour < 0 ou xJour > nombreJoursDansMois Alors
|       |       |       |       |       Afficher("Erreur, jour {xJour} invalide !")
|       |       |       |       Sinon
|       |       |       |       |       date.annee = xAnnee
|       |       |       |       |       date.mois = xMois
|       |       |       |       |       date.jour = xJour
|       |       |       |       Fin Si
|       |       |       Fin Si
|       |       Fin Si
|       Fin Si
Fin
```

Algorithme 5 : Comparer deux dates et afficher ≤ ≥ ou =
Principe : Comparer d'abord les années, puis les mois et enfin les jours. S'arrêter dès que l'un des champs de la première date est différent du même champ dans la seconde. Si les trois sont identiques alors les dates sont identiques.
Préconditions : Les deux dates sont valides, initialisées et non null
Entrée : xDate1 : structure Date
xDate2 : structure Date
Local : /
Sortie : void

```
Début
|       Si xDate1.annee > xDate2.annee Alors
|       |       Afficher("xDate1 > xDate2")
|       Sinon Si xDate1.annee < xDate2.annee Alors
|       |       Afficher("xDate1 < xDate2")
|       Sinon
|       |       Si xDate1.mois > xDate2.mois Alors
|       |       |       Afficher("xDate1 > xDate2")
|       |       Sinon Si xDate1.mois < xDate2.mois Alors
|       |       |       Afficher("xDate1 < xDate2")
|       |       Sinon
|       |       |       Si xDate1.jour > xDate2.jour Alors
|       |       |       |       Afficher("xDate1 > xDate2")
|       |       |       Sinon Si xDate1.jour < xDate2.jour Alors
|       |       |       |       Afficher("xDate1 < xDate2")
|       |       |       Sinon
|       |       |       |       Afficher("xDate1 = xDate2")
|       |       |       Fin Si
|       |       Fin Si
|       Fin Si
Fin
```