BTS MEC – OUTILS DE PROGRAMMATION

Fiche n°1 – Gérer ses données avec un tableur

*Cette fiche fait partie d’un ensemble de documents destinés à faciliter l’appropriation des outils de programmation par les enseignants du BTS MEC suite à la réforme mise en œuvre depuis septembre 2021.*

*Cette fiche présente le développement d’un petit programme permettant de relier des données entre différents tableaux créés au sein d’un tableur. L’exemple considéré ici consiste à relier un sous-détail de prix à un DPGF.*

Figure 1 - Logo de LibreOffice Calc et Microsoft Excel

*Auteur :*

*Vincent Lefort - Lycée Cantau Anglet*

*vincent.lefort@ac-bordeaux.fr*

*Fichiers fournis :*

* *DPGF-SDP.ods*
* *DPGF-SDP.xlsm*

***SOMMAIRE***

[1 Présentation du contexte 2](#_Toc106113169)

[2 Automatisation du report de prix unitaires à l’aide d’un algorithme 3](#_Toc106113170)

[3 Automatisation du report de prix unitaires à l’aide des formules du type « RECHERCHE » 4](#_Toc106113171)

[ANNEXE 1 – CODE EN LANGAGE VBA 5](#_Toc106113172)

[ANNEXE 2 – CODE EN LANGAGE ooBASIC 6](#_Toc106113173)

# Présentation du contexte

Les logiciels de type tableurs sont souvent utilisés pour stocker et manipuler des données sous forme d’un ou de plusieurs tableaux. Dans le domaine du BTP, même si certaines entreprises font appel à des logiciels spécifiques d’études de prix, il reste encore beaucoup d’entreprises qui utilisent des tableurs pour établir par exemple leurs Décompositions du Prix Global et Forfaitaire (DPGF) et leurs Sous-Détail de Prix (SDP).

DPGF et SDP traitent des mêmes parties d’ouvrages, mais avec un niveau de détail différent. Il est donc nécessaire de faire communiquer ces tableaux, le plus souvent en récupérant les prix unitaires calculés dans le SDP afin de les exploiter dans le DPGF.

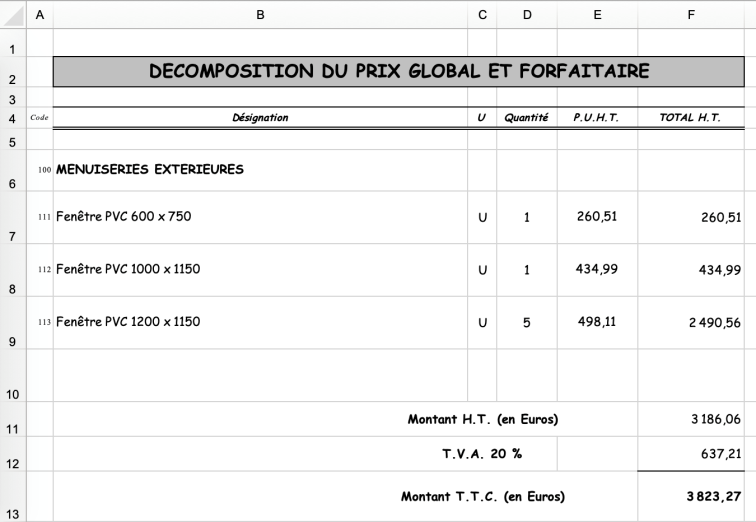
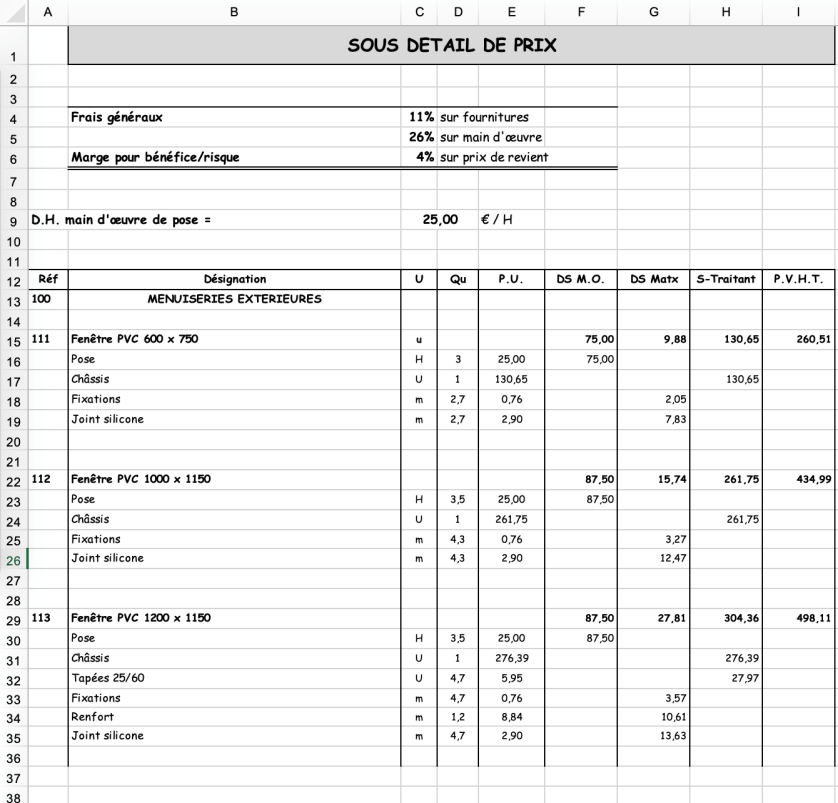
 

Figure 2 - Établissement d'un DPGF et d'un SDP à l'aide d'un tableur

Cette communication entre tableaux peut être réalisée manuellement (l’opérateur fait un copier/coller des valeurs), mais cela se révèle souvent fastidieux. Afin d’automatiser le transfert d’information entre tableaux, il se révèle indispensable **d’attribuer un code à chaque partie d’ouvrage** afin de permettre à l’ordinateur de récupérer les prix unitaires calculés dans le SDP pour un code donné, afin de les retranscrire dans le DPGF pour les lignes correspondant au même code.

La codification des ouvrages permettant de créer ce lien peut être celle du CCTP, ou bien une codification issue des systèmes de classification basés sur des normes (Uniformat, Uniclass, etc.), ou encore une codification en lien avec les propriétés des objets composant la maquette numérique (type IFC de l’objet, etc.).

Nous proposons ici d’établir le programme permettant récupérer les prix unitaires calculés dans le SDP afin de les retranscrire dans le DPGF.

# Automatisation du report de prix unitaires à l’aide d’un algorithme

Nous disposons donc de 2 tableaux :

* Un SDP, qui a permis pour chaque type d’objet d’établir un prix unitaire suite à une étude détaillée,
* Un DPGF, qui va permettre de déterminer le prix global et forfaitaire des travaux à partir des prix unitaires issus du SDP et des quantités.

L’objectif est de reporter les prix du SDP vers le DPGF en s’appuyant sur un code caractérisant de façon unique chaque prix.

L’algorithme proposé est le suivant :

* Lecture du code du prix situé à la ligne « i » du DPGF (on commence à la première ligne du DPGF) **=> CodeDPGF**
  + Lecture du code du prix situé à la ligne « j » du SDP (on commence à la première ligne du SDP) **=> CodeSDP**
    - Si **CodeDPGF = Code SDP, alors :**
      * on reporte le prix unitaire de la ligne j du SDP à la ligne i du DPGF
      * on va à la ligne suivante du DPGF et on reprendra la lecture du SDP à la première ligne
    - Si **CodeDPGF # Code SDP, alors :**
      * on va à la ligne suivante du SDP
* Une fois arrivée à la dernière ligne du DPGF l’algorithme est terminé.

Cet algorithme est traduit par la suite dans les annexes et dans les documents joints :

* en langage VBA afin d’exploitable sous forme d’une « macro » dans le logiciel Microsoft Excel
* en langage ooBasic afin d’exploitable sous forme d’une « macro » dans le logiciel LibreOffice

Il nécessite :

* de savoir manipuler des variables,
* de comprendre le fonctionnement d’une boucle,
* de comprendre le fonctionnement d’une structure conditionnelle.

Il peut être exploité avec les étudiants en leur donnant cette base et en leur demandant de l’adapter pour transférer d’autres propriétés (par exemple des caractéristiques environnementales présentes dans un catalogue, ou des caractéristiques règlementaires…).

# Automatisation du report de prix unitaires à l’aide des formules du type « RECHERCHE »

Les tableurs proposent également une formule directement exploitable nommée « RECHERCHE » afin d’effectuer une action semblable :

Extrait de la documentation Microsoft Excel :

*« La fonction RECHERCHE, une des fonctions de recherche et de référence, permet d’effectuer une recherche dans une ligne ou une colonne et de renvoyer une valeur à partir de la même position dans une seconde ligne ou colonne.*

*Par exemple, supposons que vous connaissez le numéro de référence d’une pièce automobile, mais que vous ignorez son prix. Vous pouvez utiliser la fonction RECHERCHE pour renvoyer le prix dans la cellule H2 lorsque vous entrez le numéro de référence de la pièce automobile dans la cellule H1. »*

*Syntaxe : RECHERCHE(valeur\_cherchée; vecteur\_recherche; [vecteur\_résultat])*

* *valeur\_cherchée Obligatoire. Valeur que la fonction RECHERCHE recherche dans le premier vecteur. L’argument valeur\_cherchée peut être un nombre, du texte, une valeur logique ou un nom ou une référence à une valeur.*
* *vecteur\_recherche Obligatoire. Plage de cellules qui contient une seule ligne ou colonne. Les valeurs de l’argument vecteur\_recherche peuvent être du texte, des nombres ou des valeurs logiques.*
* *vecteur\_résultat Facultatif. Plage qui contient une seule ligne ou colonne. L’argument vecteur\_résultat doit être de même dimension que l’argument vecteur\_recherche. Il doit également être de la même taille.*

Le tableau Excel donné en annexe présente également cette approche afin de permettre aux élèves de comparer.

Cette approche est cependant limitée : par exemple les valeurs vecteur\_recherche doivent être placées en ordre croissant : ..., -2, -1, 0, 1, 2..., A-Z, FAUX, VRAI ; Sinon, la recherche risque de ne pas renvoyer la valeur correcte. La fonction ne fait pas de distinction entre les majuscules et les minuscules…

Le fait de développer une macro avec les élèves leur permet donc de bien s’approprier le fonctionnement de l’outil et de l’adapter à leurs besoins.

ANNEXE 1 – CODE EN LANGAGE VBA

Sub Remplir\_PU\_BPU\_selonDQE()

*' EFFACER LE CONTENU DES DESIGNATIONS DANS LA FEUILLE "DPGF"*

Range(Sheets("DPGF").Cells(8, 2), Sheets("DPGF").Cells(2500, 2)).Select *'désignations*

Selection.ClearContents

Range(Sheets("DPGF").Cells(8, 5), Sheets("DPGF").Cells(2500, 5)).Select *'PU*

Selection.ClearContents

*' BOUCLE SUR LES CODES SAISIS DANS LA FEUILLE DE DEVIS*

For i = 6 To 1000 *' "i" est le numéro de la ligne lue dans le DPGF*

CodeDPGF = Sheets("DPGF").Cells(i, 1) *' "Num\_Chap" est le code lu à la ligne i*

*' BOUCLE SUR LES CODES SAISIS DANS LA FEUILLE DES SOUS-DETAILS*

For j = 12 To 1000 *' "j" est le numéro de la ligne lue dans le SDP*

CodeSDP = Sheets("SDP").Cells(j, 1)

If CodeSDP = CodeDPGF Then *'si on retrouve le meme code dans la feuille des sous-détails à la ligne "j"...*

Sheets("DPGF").Cells(i, 3) = Sheets("SDP").Cells(j, 2) *'on copie l'intitulé lu dans le sous-détail vers le devis*

Sheets("DPGF").Cells(i, 6) = Sheets("SDP").Cells(j, 11) *'on copie le PU lu dans le sous-détail vers le devis*

Exit For *'et on sort de la lecture des codes de sous-détails*

End If

Next j *'si on a pas trouvé le code à la ligne "j" de la feuille des sous-détails, on va à la ligne j+1*

Next i *' si on a traité ligne i du devis, on va à la ligne i+1*

End Sub

ANNEXE 2 – CODE EN LANGAGE ooBASIC

La programmation en ooBASIC est une programmation organisée en objets : c’est une syntaxe plus puissante (mais un peu plus compliquée) que le VBA. Les objets manipulés sont comme une boite ou « capsule » qui contient des données ou « attributs » et des petites fonctions ou « méthodes » qui y sont associées.

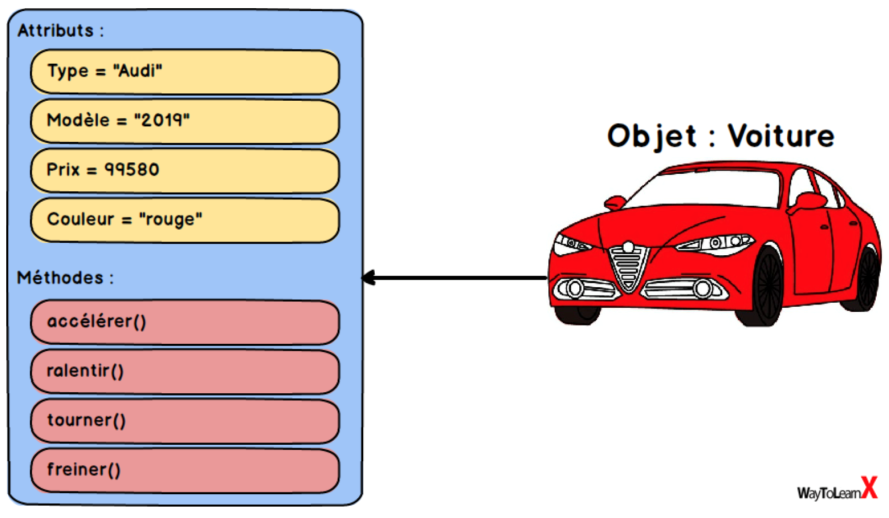


Figure - Exemple d'un objet "voiture" en programmation orientée objets (source : https://waytolearnx.com)

Les attributs ou méthodes liés à un objet sont accessibles à l’aide d’une syntaxe particulière utilisant des points. Ainsi dans l’exemple présenté dans la figure ci-dessus :

* *Voiture.Type* retournera « Audi »
* *Voiture.Couleur* retournera « rouge »
* *Voiture.freiner()* provoquera un ralentissement de la voiture (par exemple dans un jeu vidéo)…

sub Main

Dim Doc As Object

Dim Sheet\_DPGF As Object

Dim Sheet\_SDP As Object

Dim CellRange As Object

Dim Cell As Object

Dim Cell\_Designation

Dim Cell\_PU

Dim i *'"i" est le numéro de la ligne lue dans le DPGF*

Dim j *'"j" est le numéro de la ligne lue dans le SDP*

Doc = ThisComponent

Sheet\_DPGF = Doc.Sheets.getByName("DPGF") *‘permet de récupérer une feuille du document Doc, nommée « DPGF »*

Sheet\_SDP = Doc.Sheets.getByName("SDP") *‘permet de récupérer une feuille du document Doc, nommée « SDP »*

*'On efface le contenu des cellules contenant les désignations (TEXTE, FORMULE, VALEURS)*

CellRange = Sheet\_DPGF.getCellRangeByName("B7:B9")

CellRange.clearContents(\_

com.sun.star.sheet.CellFlags.STRING + \_

com.sun.star.sheet.CellFlags.FORMULA + \_

com.sun.star.sheet.CellFlags.VALUE)

*'On efface les prix unitaires*

CellRange = Sheet\_DPGF.getCellRangeByName("E7:E9")

CellRange.clearContents(\_

com.sun.star.sheet.CellFlags.STRING + \_

com.sun.star.sheet.CellFlags.FORMULA + \_

com.sun.star.sheet.CellFlags.VALUE)

For i = 6 To 1000

Cell = Sheet\_DPGF.getCellByPosition(1-1,i-1) *'Attention la cellule (0,0) est A1, la cellule (2,1) et C2*

CodeDPGF = Cell.String

For j = 13 To 1000

Cell = Sheet\_SDP.getCellByPosition(1-1,j-1*) 'Attention la cellule (0,0) est A1, la cellule (2,1) et C2*

CodeSDP = Cell.String

If CodeSDP = CodeDPGF Then

Cell\_Designation = Sheet\_SDP.getCellByPosition(2-1,j-1)

Cell\_PU = Sheet\_SDP.getCellByPosition(9-1,j-1)

Cell = Sheet\_DPGF.getCellByPosition(2-1,i-1)

Cell.String = Cell\_Designation.String

Cell = Sheet\_DPGF.getCellByPosition(5-1,i-1)

Cell.Value = Cell\_PU.Value

Exit For

End If

Next j

Next i

end sub