



Normes de programmation

420-D17-JO

1. Normes d'édition du code Pascal

A compter de maintenant, vous devrez respecter les normes d'édition du code pascal données ici pendant toute l'année scolaire.

Ces normes (qui pourraient être différentes) visent à augmenter la lisibilité du code. Les programmes que vous écrirez au travail seront éventuellement lus par des patrons, par des collègues et aussi par vous-mêmes.

Il est important que ce code soit facile à lire, non seulement pour des raisons esthétiques, mais aussi pour des raisons d'efficacité lorsque des modifications doivent être apportées à du code six mois ou deux ans après sa création.

Voici un résumé des normes d'édition des programmes sources fixées par le département d'informatique.

Norme	Explication et exemple	
Ne pas dépasser la marge de droite.	La marge de droite devrait être visible dans l'éditeur de code à la colonne 80 . Cette marge et sa visibilité sont contrôlées dans Options - Environnement - Affichage éditeur . Dans la mesure du possible, ne dépassez pas cette marge pour ne pas avoir de problème lors de l'impression du code pascal.	
Écrire le code en minuscules.	Des études ont démontré qu'un texte tout écrit en majuscules est plus difficile à lire et donc à comprendre qu'un texte écrit en minuscules ou avec une combinaison de minuscules et de majuscules. Les textes que nous lisons dans la vie courante sont écrits de cette façon. Ce n'est pas un hasard.	
	Faire	Ne pas faire
	Var Car: char; Nbr: integer; Majeur: boolean;	VAR CAR: CHAR; NBR: INTEGER; MAJEUR: BOOLEAN;
Écrire les identificateurs de constante en majuscules.	Malgré la norme précédente, il est souhaitable de pouvoir distinguer au premier coup d'œil un identificateur de constante d'un identificateur de variable. Dans EleveParGpe := NbrEleve div NBR_GPE; nous pouvons savoir tout de suite que NBR_GPE est un identificateur de constante et que les deux autres identificateurs sont des variables.	
Indenter le code.	L'indentation du code permet de faire ressortir des blocs logiques.	
	Faire	Ne pas faire
	Var Car: char; Nbr: integer; Majeur: boolean;	Var Car: char; Nbr: integer; Majeur: boolean;
Faire des paragraphes.	begin i := i + 1; j := j + 1; end;	begin i := i + 1; j := j + 1; end;
	Lorsque cela s'applique faire des paragraphes dans les instructions en passant une ligne vide pour faire ressortir les blocs logiques.	
Aérer le code.	Laisser un espace après une virgule, un deux-points ou un point virgule. Laisser un espace devant et après un := ou un opérateur (+, -, *, /, =, <>, etc).	
	Faire	Ne pas faire
	Var i, j, k: integer; begin i := i + j * k; end;	Var i,j,k:integer; begin i:=i+j*k; end;

2. Les commentaires

- Le texte d'un commentaire est ignoré par le compilateur. Ces commentaires s'adressent donc aux programmeurs et ont pour but de faciliter la compréhension du code lors de sa lecture. Dans le code Pascal nous pouvons écrire des commentaires entre des accolades «{» et «}» ou entre «(*» et «*)» ou // pour une ligne.
- Dans chaque unité et dans le .dpr**, vous indiquez en commentaire vos nom et prénom et la date de création. Plus tard vous ajouterez aussi dans le .«dpr» l'objectif de l'application (à quoi sert ce programme) et dans chacun des «units» un brève description du rôle de cette unité.

<pre> program Project1; {Auteur: France Beaudoin Date: 28 septembre 2010 Objectif: On écrit ici l'objectif général du programme} uses </pre>	<pre> unit Unit2; {Auteur: France Beaudoin Date: 28 septembre 2010} Rôle : On écrit ici le rôle de l'unité associé à la fiche. } interface </pre>
---	--

- Pour chaque déclaration de «const», de «type» et de «var» que **vous** faites, vous ajoutez un commentaire à la **droite** de la déclaration.

```

var
    NbCopier: integer; {Nombre de fois que l'utilisateur clique
                        sur le bouton Copier.}
    NbEffacer: integer; {Nombre de fois que l'utilisateur clique sur
                        le bouton Effacer.}

```

- Lorsque cela est pertinent (ça l'est dans 99% des cas), commentez un gestionnaire d'événement en donnant l'objectif de cette méthode (son but, son rôle). Utilisez dans ces commentaires **des verbes d'action**. Ils sont en caractères gras dans l'exemple qui suit. Les commentaires des déclarations ne contiennent pas ce genre de verbe. Écrivez cet objectif **en dessous de l'en-tête de la méthode**.

```

procedure TfchPri.cmdCopierClick (Sender: TObject);

{Objectif:  Concaténer dans la boîte de texte Destination, le
           texte de la boîte de texte Destination avec celui de
           la boîte de texte Source.  Effacer alors le texte de
           la boîte de texte Source et lui donner le focus. }

begin
    txtDestination.Text := txtDestination.Text + txtSource.Text;
    txtSource.Clear;
    txtSource.SetFocus;
end;

```

- Il arrive souvent (dans des cas plus complexes qu'ici) que l'objectif seul ne suffise pas à rendre le code facile à comprendre. Dans ces cas, il faut ajouter des commentaires dans les traitements. L'objectif de la méthode est un commentaire **général**. Les commentaires dans les traitements concernent les **détails** de l'implantation, c'est-à-dire comment le code de ce gestionnaire atteint son objectif. Il faut ici aussi utiliser des verbes d'action dans les commentaires. **Ces commentaires sont placés au-dessus des instructions commentées et non à leur droite comme pour les déclarations.**

```
begin
  {Récupérer le pointeur sur la boîte de texte qui a reconnu l'événement.}
  RefEdit := Sender as TEdit;

  {Afficher la longueur de la chaîne de la boîte de texte active.}
  txtLongueur.Text := IntToStr(Lenght(RefEdit.Text));

  ....
  ....
end;
```

Attention

A partir de maintenant, tout travail de programmation non commenté (ou commenté *pro forma*) ne sera pas corrigé et méritera la note zéro.

3. Nomenclature des composants

Tous les composants (contrôles) d'une feuille doit avoir un nom significatif commençant par un préfixe qui exprime le type du composant. Le préfixe est en minuscule. En voici une liste :

Composant (dans l'EDI)	Nom	Préfixe
ListBox	Boîte de liste	lst
ComboBox	Boîte composée	cmb
EditBox ou TextBox ou Memo	Boîte de texte	txt
RadioButton	Bouton d'option	opt
CheckBox	Case à cocher	cch
PageControl ou TabControl	Onglet	ong
TrackBar	Potentiomètre	ptm
UpDown	Bouton haut-bas	bhb
Button	Bouton de commande	cmd
SpeedButton	Bouton outil	otl
Image	Image	img
Label	Étiquette	etq
ScrollBar ou vScrollBar	Barre de défilement	dfl
Panel	Panneau	pnn
ProgressBar	Barre de progression	prg
Menu	Menu	mnu
PopupMenu	Menu contextuel	mnc
StringGrid	Grille de chaînes	grl
DialogBox	Boîte de dialogue	dlg

France Beaudoin