SIO1 C#

Table des matières

[I. Les structures de contrôle 1](#_Toc83042216)

[1. Les structures conditionnelles 2](#_Toc83042217)

[a. Les opérateurs de comparaison 2](#_Toc83042218)

[b. Le Si … Alors : if…then 2](#_Toc83042219)

[c. Si … Alors … Sinon : if … then … else 3](#_Toc83042220)

[d. Imbrications des if 4](#_Toc83042221)

[e. Selon : select case … 4](#_Toc83042222)

# Les structures de contrôle

En programmation procédurale l'ordre des instructions est primordial.

Le processeur exécute les instructions dans l'ordre dans lequel elles apparaissent dans le programme. On dit que l'exécution est séquentielle.

Une fois que le programme a fini une instruction, il passe à la suivante. Tant qu'une instruction n'est pas terminée, il attend avant de continuer. Par exemple, une instruction de saisie va attendre que l'utilisateur rentre une valeur au clavier avant de continuer.

Parfois, il est nécessaire que le processeur n'exécute pas toutes les instructions, ou encore qu'il recommence plusieurs fois les mêmes instructions. Pour cela, il faudra casser la séquence. C'est le rôle des structures de contrôle.

Il existe deux grands types de structures de contrôle :

* Les structures conditionnelles vont permettre de n'exécuter certaines instructions que sous certaines conditions
* Les structures répétitives, encore appelées boucles, vont permettre de répéter des instructions un certain nombre de fois, sous certaines conditions

## Les structures conditionnelles

Les structures conditionnelles permettent d'exécuter des instructions différentes en fonction de certaines conditions. Une condition (encore appelée expression conditionnelle ou logique) est évaluée, c'est à dire qu'elle est jugée vrai ou fausse.

### Les **opérateurs de comparaison**

Une condition se construit grâce à des **opérateurs de comparaison**. On dénombre plusieurs opérateurs de comparaisons, les plus courants sont :

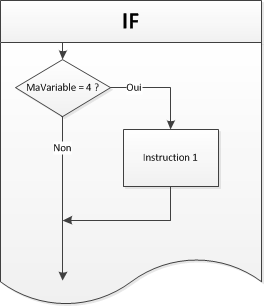
| **Opérateur** | **Description** |
| --- | --- |
| == | Egalité |
| != | Différence |
| > | Su périeur à |
| < | Inférieur à |
| >= | Supérieur ou égal |
| <= | Inférieur ou égal |
| && | ET logique |
| || | OU logique |
| ! | Négation |

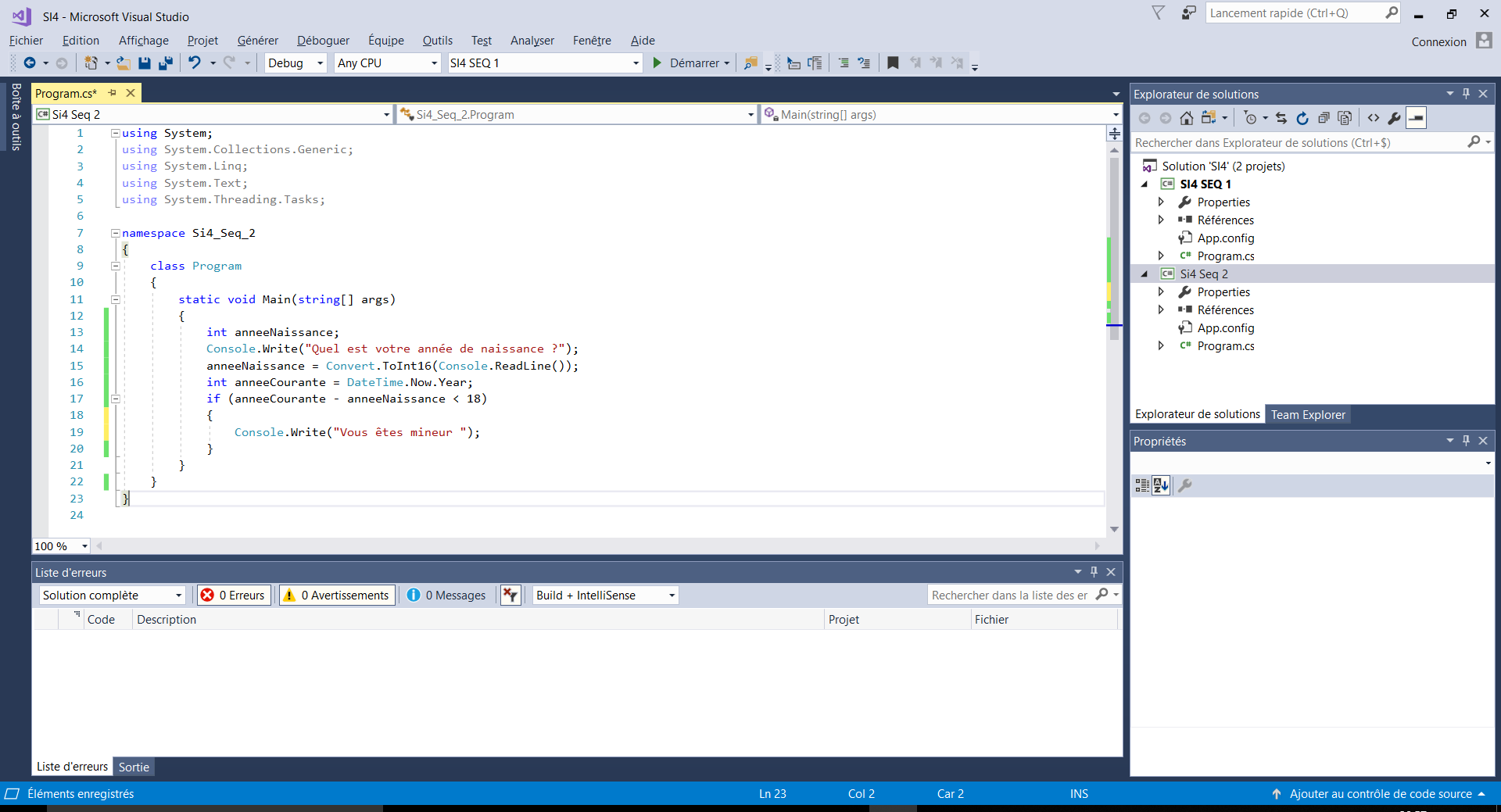
### Le Si … Alors : if…then

Si la condition est vraie, un traitement (une ou plusieurs instructions) est réalisée si la condition est fausse, le programme va continuer normalement.

Exemple : Ecrire un programme qui affiche si une variable vaut 4 ou non.

Représentation graphique :





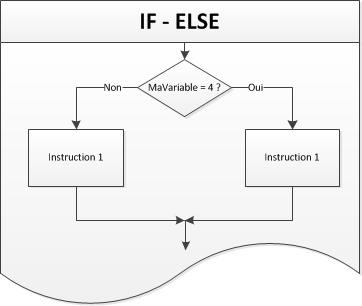
Application :

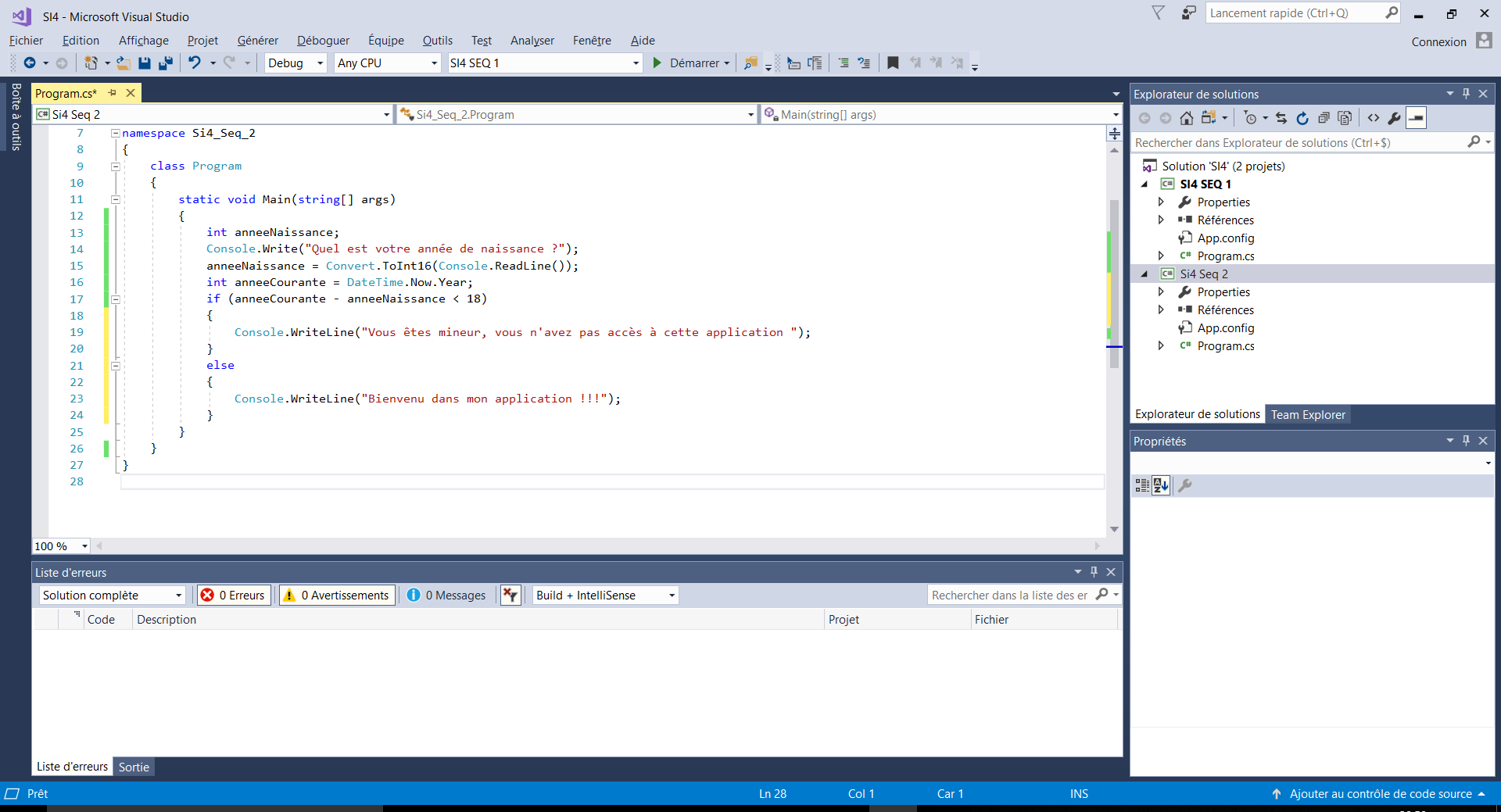
Ecrire un programme qui fait saisir à l’utilisateur la référence, le prix unitaire et la quantité commandée d’un produit. Afficher un résumé des données saisies, si le montant total est supérieur à 1000€, faire 10% de réduction : afficher le montant de la réduction et le montant après celle-ci.

### Si … Alors … Sinon : if … then … else

Si la condition est vraie, un traitement (une ou plusieurs instructions) est réalisée, si la condition est fausse un autre traitement est réalisé.

Représentation graphique :





Application :

Faire saisir le nom du candidat au bts et sa moyenne au BTS, afficher s’il a obtenu son BTS ou lui offrir un tour gratuit sinon !

Faire saisir un nombre et afficher s’il est positif ou nul.

### Imbrications des if

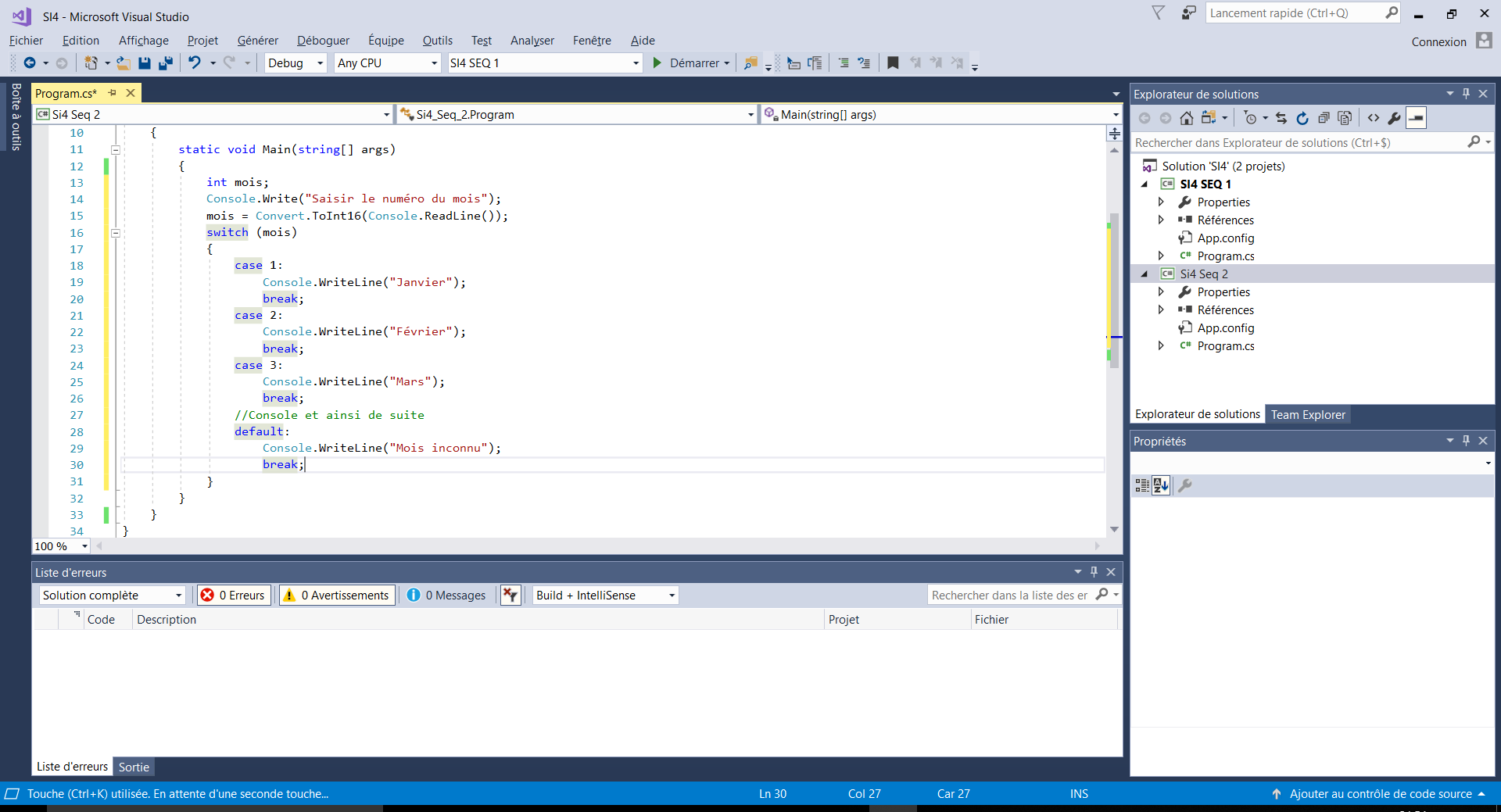
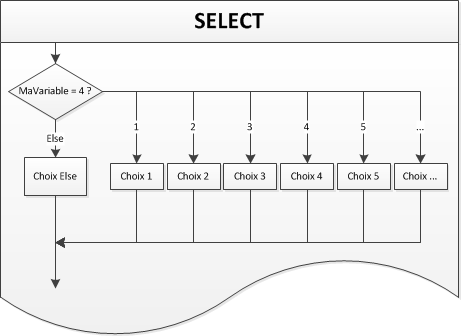
Les boucles If, Then et Else peuvent s'imbriquer, ce qui signifie qu'on peut en mettre plusieurs l'une dans l'autre.

Il faut ici respecter l'indentation afin de se repérer dans cette imbrication de si.

Application : 0 n’est positif ni négatif : modifier le programme précédent

### Selon : select case …

Nous avons vu If, ElseIf et Else, mais pour ce qui est, par exemple, du cas d'un menu, vous avez dix choix différents dans votre menu. Comment faire ?



Attention au bloc de code et la portée des variables :

