Le débuggueur – Bloc F

Le débugueur

Le débuggeur est un outil interne permettant au programmeur d'être en immersion dans le code **en temps réel**.

Wikipedia:

« *Il permet d'exécuter le programme pas-à-pas — c'est-à-dire le plus souvent ligne par ligne —, d'afficher la valeur des variables à tout moment et de mettre en place des points d'arrêt sur des conditions ou sur des lignes du programme.* «

Source: https://fr.wikipedia.org/wiki/D%C3%A9bogueur

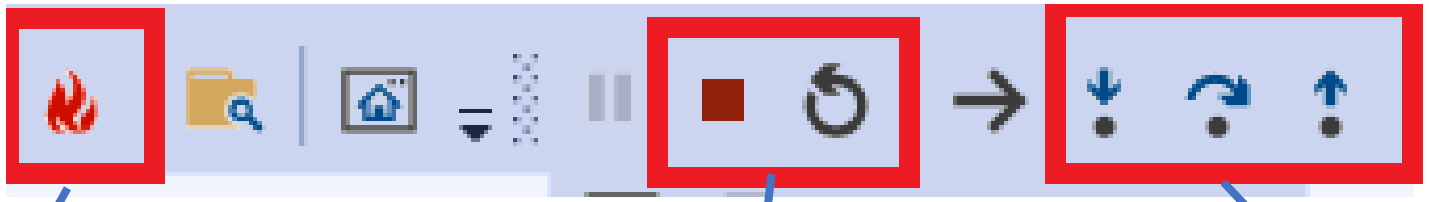
Le débugueur

Objectifs d’apprentissage:

Les breakpoints	Le curseur
Add Watch	Step Into (F11)
Step Over (F10)	Step Out (Shift + F11)
Stop debugging	Apply code change

Source: https://fr.wikipedia.org/wiki/D%C3%A9bogueur

Le débugueur

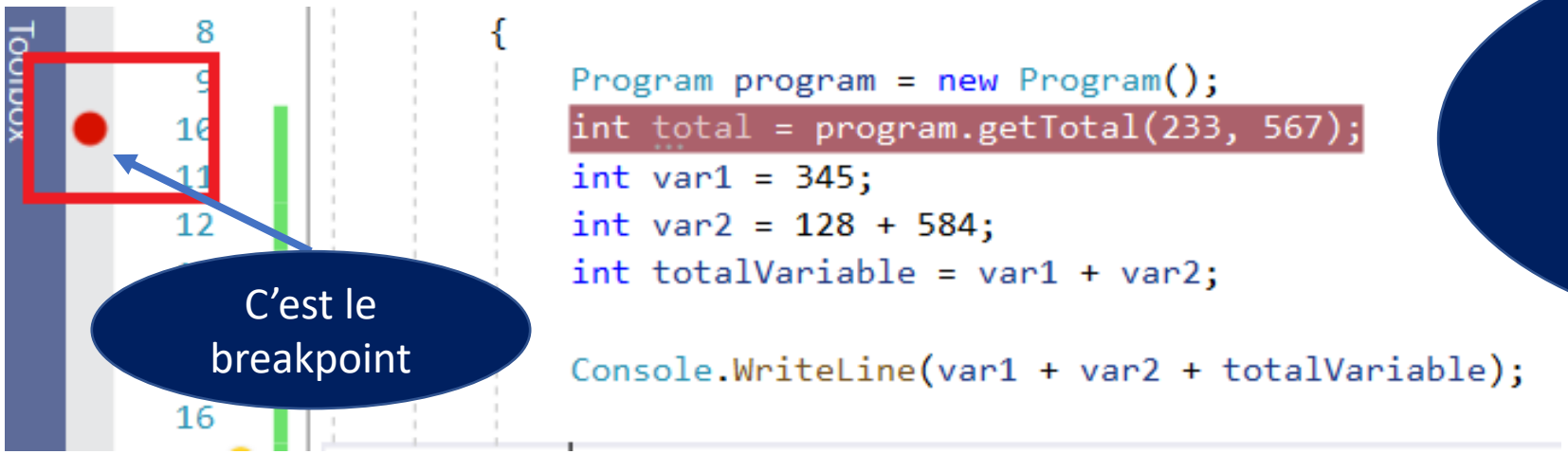


Appliquer les changements en temps réel

Arrêter et redémarrer le programme

1- Step Into
2- Step Over
3- Step Out

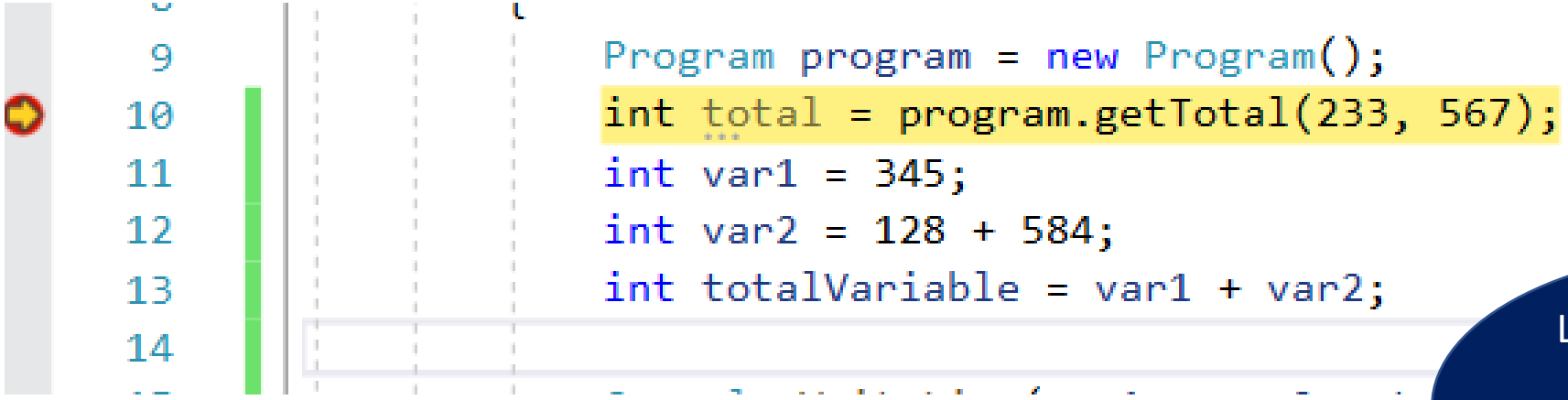
Les points d'arrêt



L'ajout d'un point d'arrêt (ou breakpoint en anglais) se fait simplement en cliquant dans la ligne grise à l'endroit désirée.

Les points d’arrêt

Lors de l’exécution, le système va arrêter au point d’arrêt et attendre vos instruction.

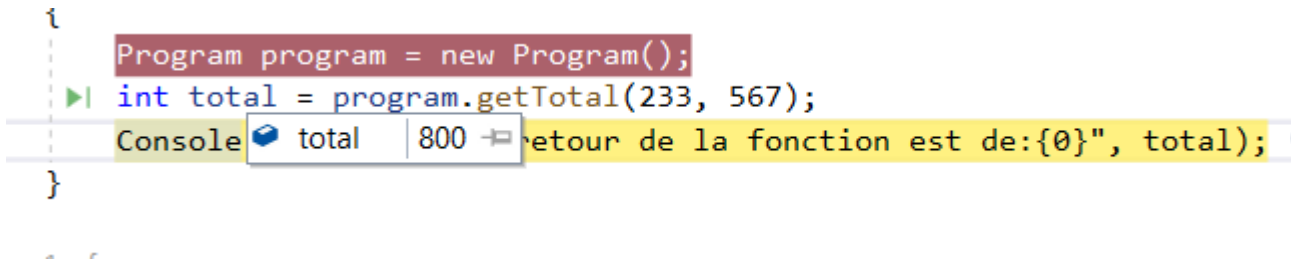


Le système a arrêté à la ligne 10 comme nous l’avons spécifié à l’aide du breakpoint

Le curseur

Le débugueur permet au programmeur d'observer et de contrôler le déroulement d'un programme. Il peut ainsi observer, arrêter ou mettre en pause les instructions de l'application.

Ainsi, il est possible d'observer les variables et leur contenu pendant le flow d'exécution. Ceci permet de facilement déterminer les causes des défaillances.



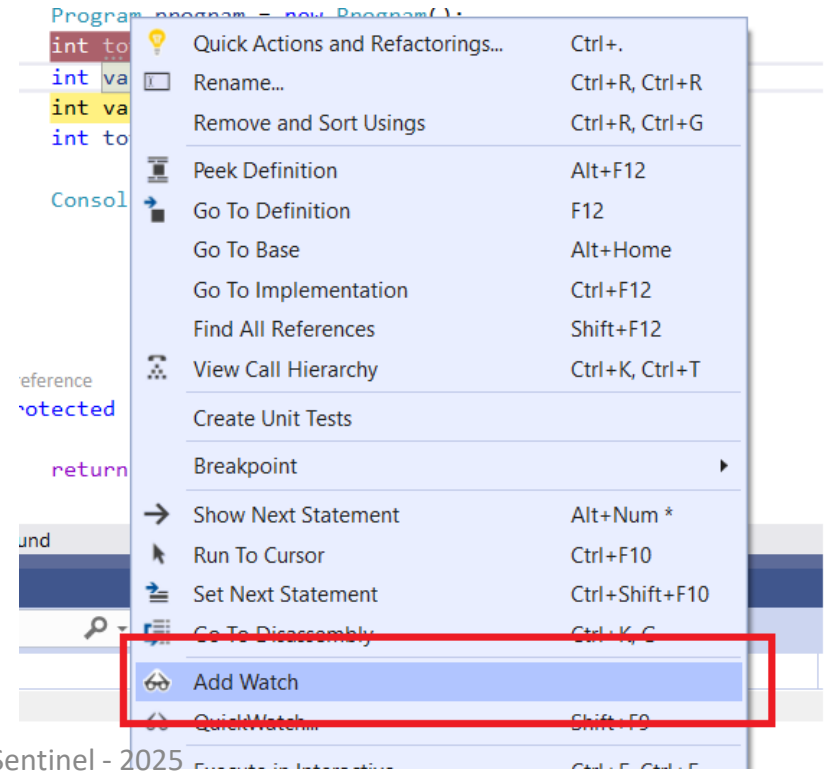
Dans l'exemple on observe la valeur de la variable total. En positionnant la souris sur la variable le débugueur nous informe que la valeur actuelle est de 800

Utiliser l'inspecteur

Les outils *Add Watch* permettent d'ajouter des surveillances sur plusieurs éléments. Ainsi, nous pouvons avoir une vision d'ensemble et examiner les éléments voulus au fur et à mesure que l'exécution se déroule.

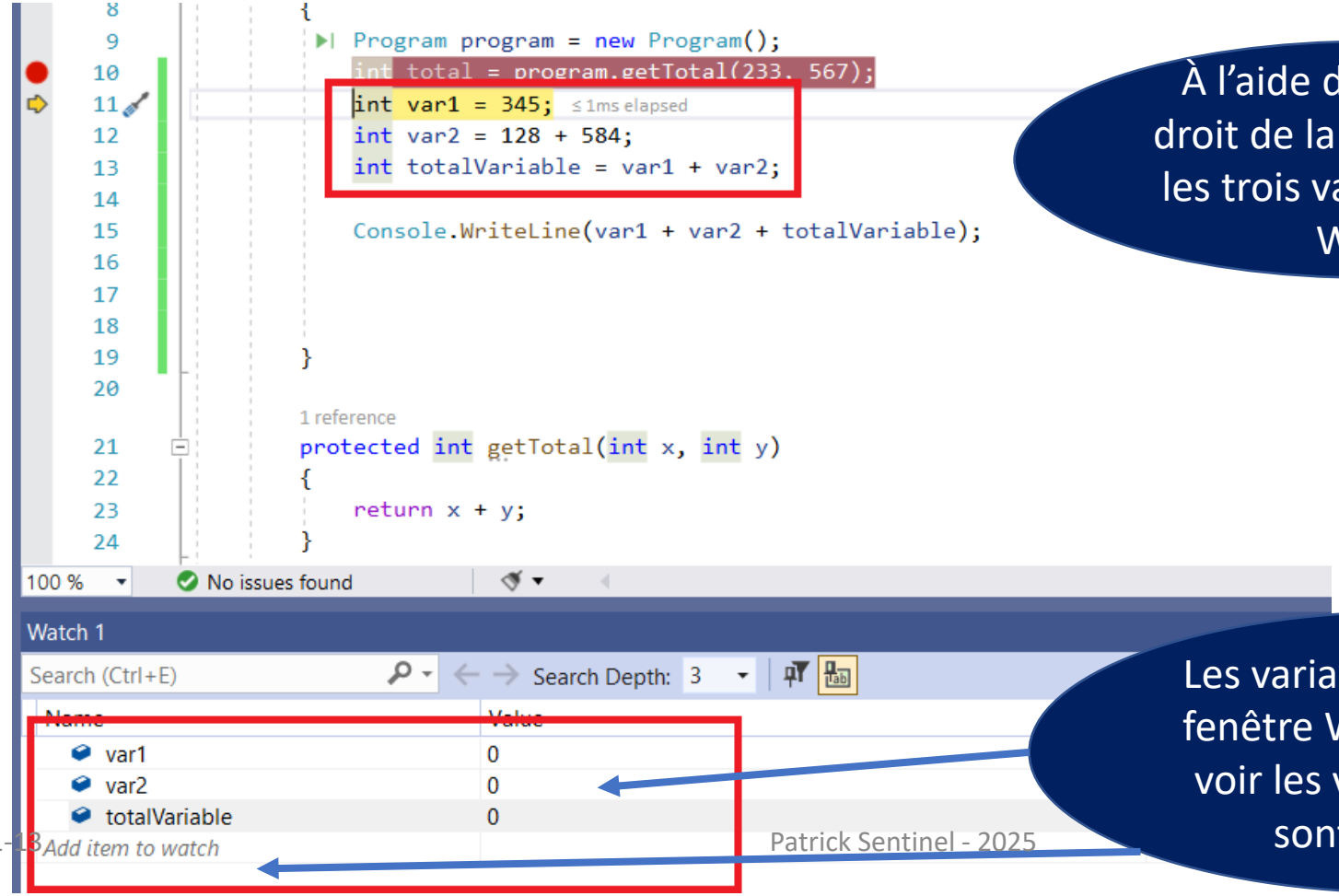
Utilisation de Add Watch:

1- Positionnez le cuseur sur une variable
2- Bouton de droit de la souris
3- Cliquez sur Add Watch



Utiliser l'inspecteur

Exemple d'exécution Add Watch

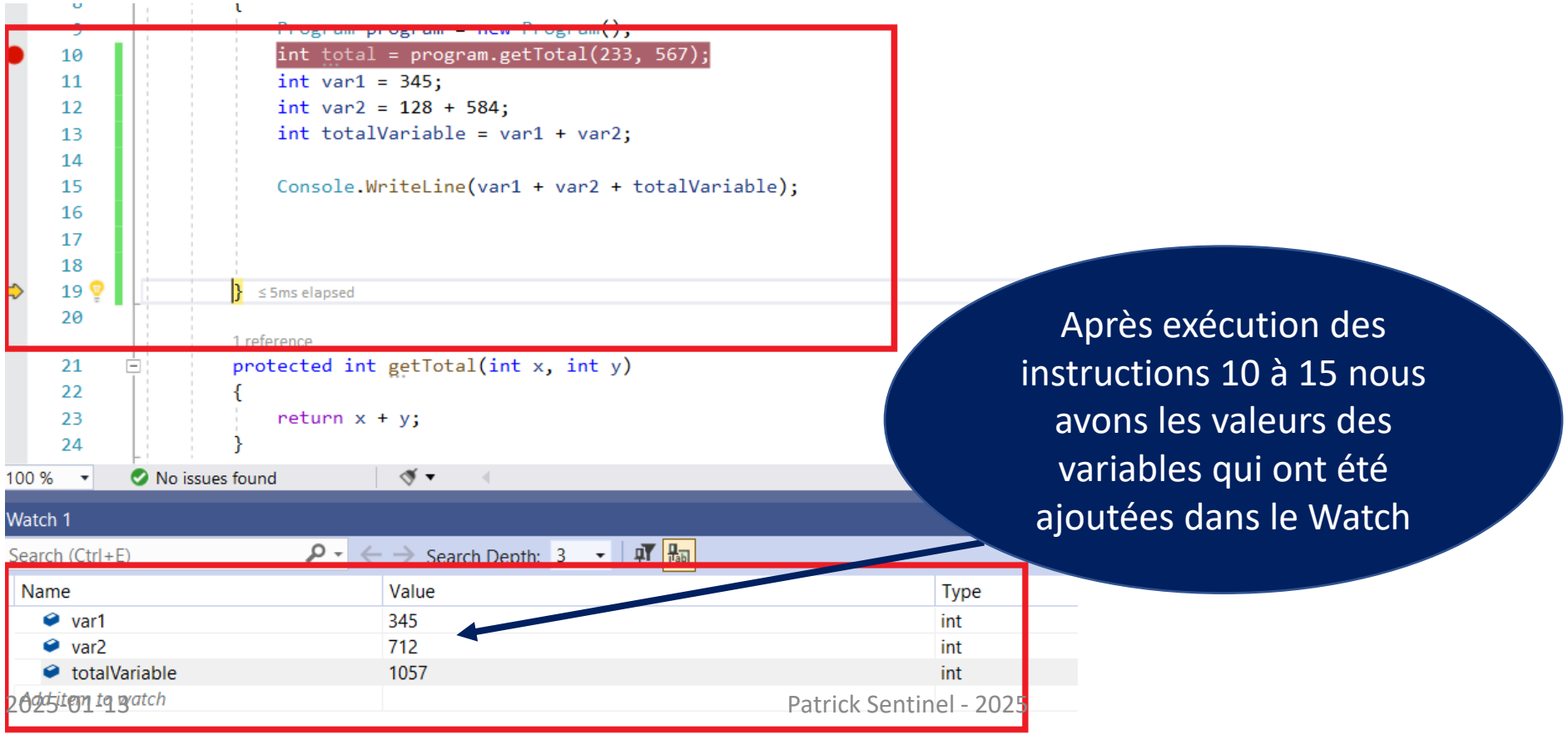


À l'aide du bouton de droit de la souris ajouter les trois variables à Add Watch

Les variables sont dans la fenêtre Watch et on peut voir les valeurs qui leurs sont attribuées

Utiliser l'inspecteur

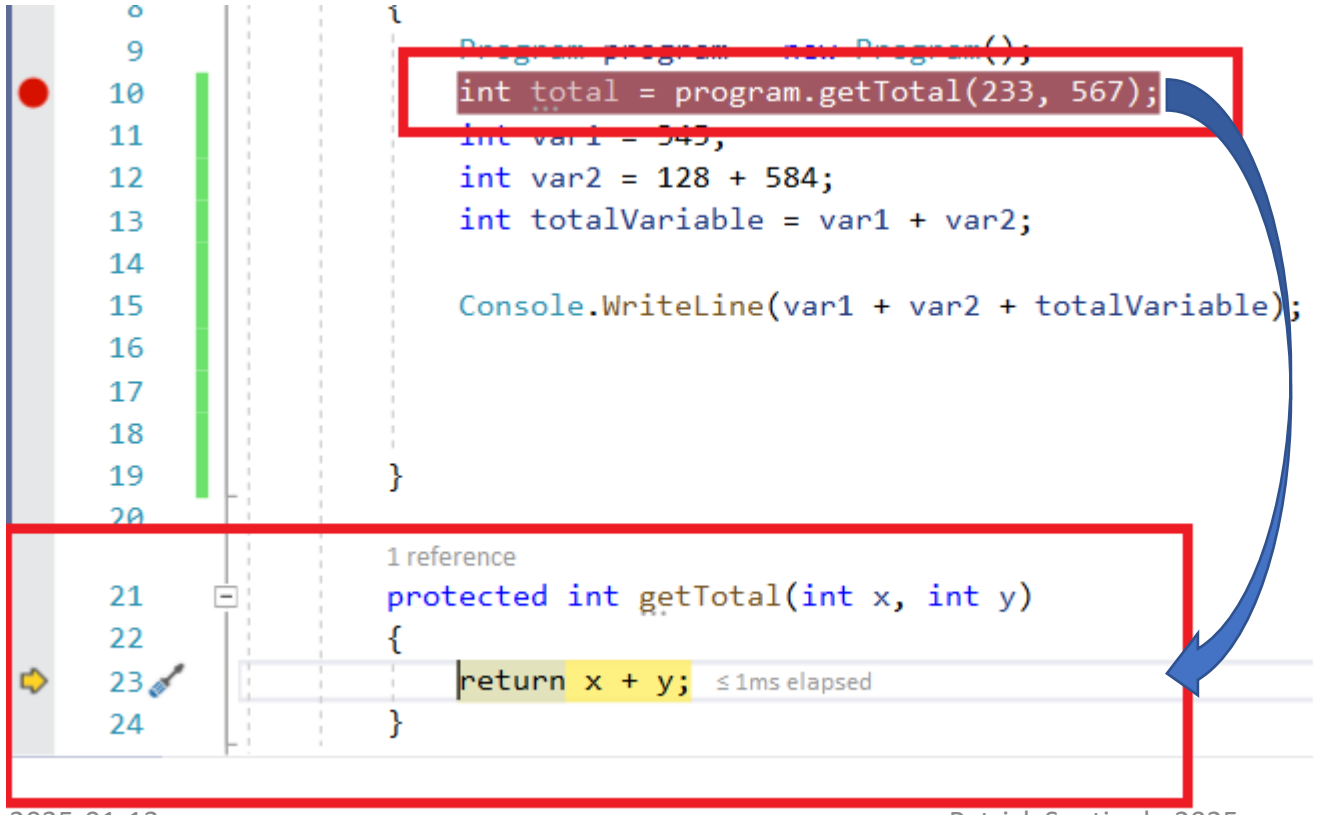
Exemple d'exécution Add Watch



Après exécution des instructions 10 à 15 nous avons les valeurs des variables qui ont été ajoutées dans le Watch

Step Into (F11)

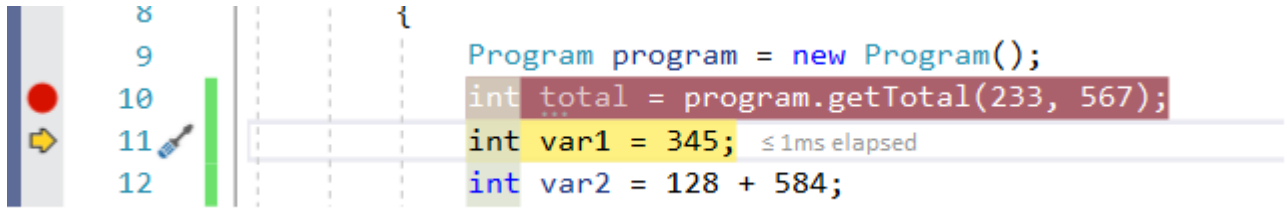
La fonction Step Into vous permet d'entrer dans un block de code.



Le Step Into (F11) vous permet d'entrer dans la fonction plutôt que de passer par-dessus.

Step Over (F10)

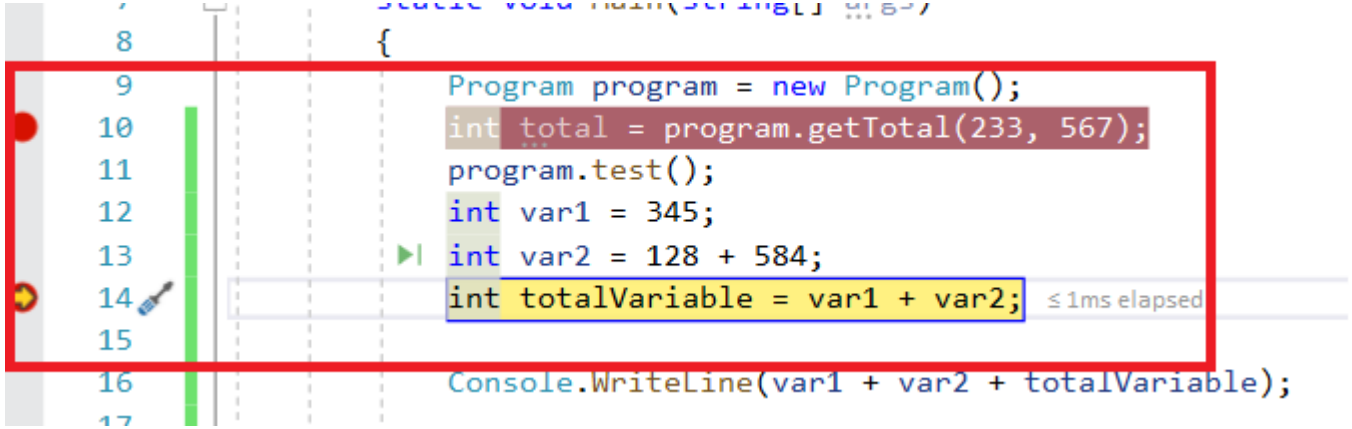
La fonction Step Over vous permet de ne pas entrer dans une fonction. S'il n'est pas nécessaire d'entrer dans la fonction pour valider le comportement de cette dernière alors il ne suffit que d'appuyer sur Step Over.



Le Step Over (F10) vous permet de ne pas entrer dans la fonction appelée à la ligne 10. ATTENTION, la ligne 10 s'exécute mais on n'entre pas à l'intérieur

Step Out (Shift + F11)

La fonction *Step Out* vous permet redonner le contrôle au programme aussi longtemps qu'il ne rencontre pas un autre point d'arrêt.



Le système a arrêté au premier *breakpoint* et en appuyant sur Step Out le système continue l'exécution jusqu'au prochain *breakpoint*. S'il n'y a pas d'autres *breakpoint* le système continue de s'exécuter normalement

Debugueur

Le débugueur sera l'un de vos principaux outils de travail. Vous ne pourrez pas vous en passer et il vous sera d'une très grande utilité!

Apprivoisez-le et utilisez-le régulièrement!

Questions ?

Écrivez-moi à info@29a.ca