

La programmation événementielle (courte introduction)

Lors de la programmation séquentielle les instructions s'exécutent séquentiellement dans l'ordre définis par le programmeur entre le début du programme et la fin de celui-ci. Dans la programmation événementielle, au contraire, les instructions sont le plus souvent exécutées en réponse à des événements (provoqués par le système ou l'utilisateur). Le programmeur doit donc lier les "fonctionnalités" à des événements. Par exemple, lorsque l'utilisateur clique sur le bouton d'impression, le code permettant l'impression d'un document s'exécute. Le programmeur n'a aucun moyen de savoir quand ce bouton sera cliqué (peut-être jamais ?).

Lors de l'exécution d'un code événementiel, un "système d'écoute" des événements est démarré. Comme par exemple une boucle « sans fin ». Dès que le système d'écoute perçoit un événement, il appelle alors le code qui lui a été associé par le programmeur. Il faut bien sûr indiquer au système quel type d'événement il doit écouter.

Il y a deux grandes catégories d'événements. Les événements provoqués par l'utilisateur (un click de souris, une frappe au clavier, etc.) ou les événements provoqués par le système (l'application démarre, la fenêtre se ferme, une erreur est survenue, ...). Le programmeur doit donc se poser les questions suivantes en plus de celles qu'il se pose lors de la programmation impérative :

- Quoi ? Quel type d'événement déclenche le code associé (clique, touche du clavier,...)
- Où ? Où cet événement aura lieu (bouton, fenêtre, etc.)

Les 3 événements les plus fréquents dans les applications sont :

- l'événement utilisateur de clique (sur un bouton par exemple)
- l'événement utilisateur de frappe sur le clavier
- l'événement système du chargement de l'application (utile à l'initialisation)

Pour lier des codes à des événements, la plupart des langages de programmation utilise le principe des « écouteurs » (listeners). On ajoute un « écouteur » sur un certain type d'événement, et on indique la fonctionnalité qui lui est associée. Le système de gestion des événements du langage (le moteur événementiel) se chargera alors d'appeler la fonctionnalité lorsque l'événement se produira. Une des difficultés de la programmation événementielle par rapport à la programmation séquentielle est de bien comprendre que les codes ne s'exécutent pas dans l'ordre et qu'il est donc plus difficile de gérer la communication entre les différents objets. De plus, le moteur événementiel de la plupart des langages gère le parallélisme, c'est-à-dire qu'il est capable d'exécuter plusieurs codes en parallèles si plusieurs événements interviennent en même temps (par exemple un click sur un bouton déclenche deux différentes « actions », ces deux actions s'exécuteront quasi simultanément).