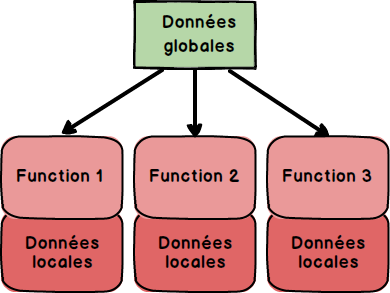
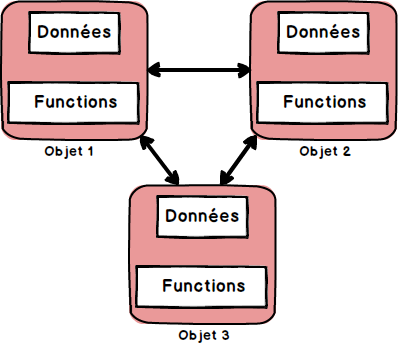
**L**a différence entre la **programmation procédurale** et la **programmation orientée objet (POO)** réside dans le fait que dans la **programmation procédurale**, les programmes sont basés sur des fonctions, et les données peuvent être facilement accessibles et modifiables, alors qu’en **programmation orientée objet**, chaque programme est constitué d’entités appelées objets, qui ne sont pas facilement accessibles et modifiables.

C’est quoi la programmation procédurale?



Dans la **programmation procédurale** le programme est divisé en petites parties appelées **procédures** ou **fonctions**. Comme son nom l’indique, la **programmation procédurale** contient une procédure étape par étape à exécuter. Ici, les problèmes sont décomposés en petites parties et ensuite, pour résoudre chaque partie, une ou plusieurs fonctions sont utilisées.

C’est quoi la Programmation Orientée Objet (POO)?



Dans la **programmation orientée objet** le programme est divisé en parties appelées **objets**.  
La programmation orientée objet est un concept de programmation qui se concentre sur l’objet plutôt que sur les actions et les données plutôt que sur la logique.

Table de comparaison

|  | **Programmation Procédurale** | **Programmation Orientée Objet** |
| --- | --- | --- |
| **Programmes** | Le programme principal est divisé en petites parties selon les fonctions. | Le programme principal est divisé en petit objet en fonction du problème. |
| **Les données** | Chaque fonction contient des données différentes. | Les données et les fonctions de chaque objet individuel agissent comme une seule unité. |
| **Permission** | Pour ajouter de nouvelles données au programme, l’utilisateur doit s’assurer que la fonction le permet. | Le passage de message garantit l’autorisation d’accéder au membre d’un objet à partir d’un autre objet. |
| **Exemples** | Pascal, Fortran | PHP5, C ++, Java. |
| **Accès** | Aucun spécificateur d’accès n’est utilisé. | Les spécificateurs d’accès public, private, et protected sont utilisés. |
| **La communication** | Les fonctions communiquent avec d’autres fonctions en gardant les règles habituelles. | Un objet communique entre eux via des messages. |
| **Contrôle des données** | La plupart des fonctions utilisent des données globales. | Chaque objet contrôle ses propres données. |
| **Importance** | Les fonctions ou les algorithmes ont plus d’importance que les données dans le programme. | Les données prennent plus d’importance que les fonctions du programme. |
| **Masquage des données** | Il n’y a pas de moyen idéal pour masquer les données. | Le masquage des données est possible, ce qui empêche l’accès illégal de la fonction depuis l’extérieur. |

Conclusion

Les failles de la **programmation procédurale** posent le besoin de la **programmation orientée objet**. La **programmation orientée objet** corrige les défauts du **programmation procédurale** en introduisant le concept «**objet**» et «**classe**». Il améliore la sécurité des données, ainsi que l’initialisation et le nettoyage automatiques des objets. La **programmation orientée objet** permet de créer plusieurs instances de l’objet sans aucune interférence.