DESIGN PATTERNS - FAMILLE

Patrons de création

Choisir qui construit les objets, comment les initialiser et quelles identités partager.

Par Ali ZAINOUL · Cours gratuit

Ce que vous allez apprendre

- Comparer les intentions des patrons de cette famille.
- Choisir un premier exemple selon le problème rencontré.

Prérequis. Connaître les classes, les objets, les fonctions et les exceptions Python.

Le besoin commun

Choisir qui construit les objets, comment les initialiser et quelles identités partager.

La classe concrète varie-t-elle ? Faut-il une famille cohérente, une construction par étapes, une copie ou une identité partagée ?

Commencez par le contrat du client. Décrivez les entrées, le résultat, les erreurs attendues et ce qui doit pouvoir changer. Le nom d'un patron aide ensuite à discuter la solution et ses compromis.

Les patrons à étudier

- **2. Singleton**

  Mémoriser une identité partagée et mesurer les limites de sa portée.

- **3. Abstract Factory**

Créer une famille cohérente de produits sans choisir leurs classes dans le client.

- **4. Builder**

  Assembler un objet par étapes et détacher le résultat avant de réutiliser le constructeur.

- **5. Factory Method**

  Distinguer la sélection d'une fabrique simple de la création redéfinie par une sous-classe.

- **6. Prototype**

  Créer une variante à partir d'un exemplaire existant en choisissant ce qui reste partagé.

Version web : https://www.clearcode.fr/courses/programming/design-patterns/creation