

# Generation Gap

# Generation Gap

- Auteur : John Vlissides
- <http://www.research.ibm.com/designpatterns/pubs/gg.html>

# Contexte

- Un outil qui génère du code automatiquement.
- Les programmeurs modifient le code pour ajouter ou modifier le comportement.

# Problème

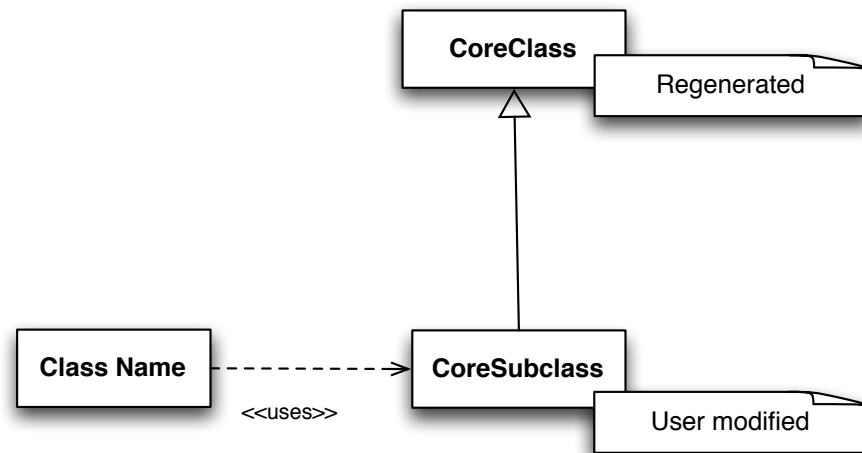
- Comment empêcher que la régénération subséquent efface les modifications du programmeur?

# Forces

- Besoin d'accès contrôlé au code généré.
- Une approche par commentaires est peu esthétique et non vérifiable.
- Trouver les différences et fusionner automatiquement est trop incertain.

# Solution

- Utiliser l'héritage pour séparer le code généré des modifications.



# Conséquences

- Les modifications sont séparées du code générée.
- Les modifications peuvent avoir un accès privilégié.
- La régénération n'implique pas la modification du code.
- Le nombre de classes est doublé.
- L'intégration dans des hiérarchies de classe existantes est plus complexe.

# Generation Gap