Conception objet en Java avec BlueJ

une approche interactive

7. Conception des classes

Comment écrire des classes faciles à comprendre, à maintenir et réutilisables

David J. Barnes, Michael Kölling

version française: Patrice Moreaux

Rédigé avec OpenOffice.org

Concepts étudiés

- Conception dirigée par les responsabilités
- Couplage
- Cohésion
- Réingénierie

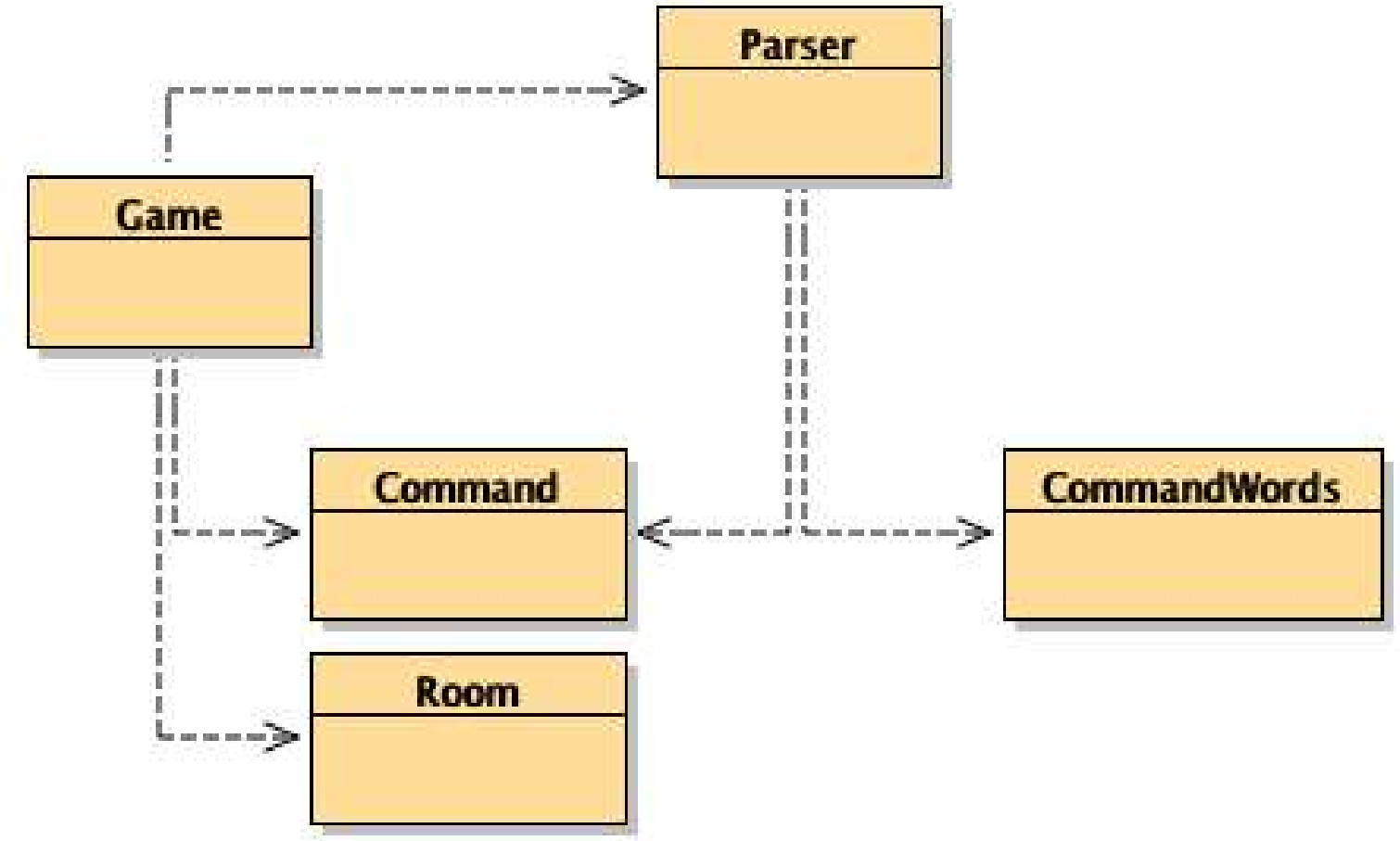
Les logiciels changent

- Le logiciel n'est pas semblable à un roman écrit une fois pour toutes.
- Le logiciel est étendu, corrigé, maintenu, porté, adapté,…
- Le travail est réalisé par des personnes différentes au cours du temps (souvent des dizaines d'années)

Évoluer ou mourir !

- Il n'y a que deux possibilités pour le logiciel:
  - être continuellement maintenu
  - mourir.
- Le logiciel qui ne peut pas être maintenu sera jeté.

Le monde de Zuul



Qualité du code

Deux concepts importants pour la qualité du code:

- Le couplage
- La cohésion

Couplage

- Le couplage fait référence aux liens entre deux unités distinctes d'un programme.
- Deux classes qui dépendent chacune fortement des détails de l'autre sont dites *fortement couplées*.
- Nous devons rechercher un couplage faible.

Couplage faible

Le couplage faible permet de:

- comprendre le fonctionnement d'une classe sans en analyser d'autres ;
- modifier une classe sans affecter les autres.
- donc: améliore la maintenabilité.

Cohésion

- La cohésion fait référence au nombre et la diversité des tâches dont une unité est responsable.
- Si chaque unité est responsable d'une seule tâche logique, nous dirons qu'elle est *fortement cohérente*.
- La cohésion concerne les classes et les méthodes.
- Nous devons rechercher une forte cohésion.

Forte cohésion

La forte cohésion facilite:

- la compréhension de ce que fait une classe ou une méthode;
- l'emploi d'identificateurs significatifs;
- la ré-utilisation de classe ou de méthode.

Cohésion de méthodes

- Une méthode doit être responsable d'une et une seule tâche bien définie.

Cohésion de classes

- Une classe doit représenter une seule entité bien définie.

Duplication de code

La duplication de code

- est un indicateur de mauvaise conception,
- rend la maintenance plus difficile,
- peut conduire à l'introduction d'erreurs lors de la maintenance.

Conception dirigée par les responsabilités

- Question: où devons-nous introduire une nouvelle méthode (dans quelle classe)?
- Chaque classe doit être responsable de la manipulation de ses données.
- La classe qui possède des données A doit être responsable des traitements sur ces données A.
- La conception guidée par les responsabilités (CGR) induit un couplage faible.

Rendre les modifications locales

- L'un des buts de la réduction du couplage et de la CGR est de rendre les modifications locales.
- Quand une modification est nécessaire, elle doit concerner le moins de classes possible.

Penser le futur

- En concevant une classe nous devons penser aux modifications qui risquent de se produire dans le futur.
- Nous devons chercher à faciliter ces modifications.

Réingénierie

- Lors de la maintenance d'une classe, on ajoute souvent du code.
- Les classes et les méthodes tendent à « grossir »
- De temps en temps la conception des classes et des méthodes devrait être revue pour maintenir la cohésion et le couplage faible: c'est la *réingénierie*.

Réingénierie et tests

- Ne pas faire d'autres modifications de code que celles dues à la réingénirie pendant la phase de changement de conception.
- Réaliser d'abord et uniquement les changements de réingénierie sans changer les fonctionnalités.
- Tester avant et après la réingénierie pour vérifier que rien n'est modifié.

Les questions récurrentes de conception

Les réponses aux questions:

- quelle doit être la taille d'une classe?
- quelle doit être la longueur d'une méthode?
- s'expriment en termes de cohésion et couplage.

Conseils de conception

- Une méthode est trop longue si elle effectue plus d'une tâche logique.
- Une classe est trop complexe si elle représente plus d'une entité logique.
- Remarque: cela laisse beaucoup de liberté au concepteur…

Résumé (1)

- Les programmes changent perpétuellement.
- Qualité de code signifie beaucoup plus qu'une exécution correcte à un moment donné.
- Le code doit être compréhensible et « maintenable ».

Résumé (2)

- Le code de qualité évite les duplications, présente une forte cohésion et un faible couplage.
- Le style de codage (commentaires, nommage, présentation, etc.) est aussi important.
- Il y a une grande différence en quantité de travail pour modifier du code faiblement structuré et du code bien structuré.

Sommaire général