

MAC5715

Tópicos Avançados em Programação
Orientada a Objeto
Proposta de Anti-Padrão

Vinicius Gama Pinheiro e Mário Henrique Cruz Torres

Nome do Anti-Padrão: Árvore Genealógica

Diálogo-evidência:

- O que faz essa classe?
- Ah, ela herda vários métodos da superclasse, e outros que estão definidos na superclasse dessa superclasse... Mas esta última na verdade herda de duas outras classes, então... o que você perguntou mesmo?
- Esquece... Este outro método está definido onde?
- Hum, me lembro que ele foi declarado na superclasse X, porém não lembro se ainda utilizamos esta classe X pois não sei ao certo o que esta também está herdando. Eu não mexeria nisso.”

Forma Geral:

É encontrado em sistemas cujas classes modelam entidades que possuem muitas especializações aninhadas, e/ou utilizam fortemente herança múltipla. Geralmente é acarretado pelo mal uso da característica de herança por desenvolvedores com pouca experiência em sistemas orientados a objeto. O sistema acaba contando com classes que herdam muitos métodos, os quais tem o comportamento mais variado possível, o que acarreta na perda de coesão de várias classes, em um forte acoplamento no sistema e uma enorme dificuldade em compreender a função de cada classe. Algumas classes podem eventualmente conter código herdado que não tem mais serventia, mas que permanece desapercebido nas classes superiores. Ou então podem exis-

tir classes que estendem funcionalidades que não são mais úteis ao sistema, como um subsistema de acesso a disquetes (num sistema que não utilize mais disquetes) por exemplo, e mesmo assim permanecerem por anos no código.

Sintomas e Consequências:

- classes com muitos métodos herdados implementados em diversos níveis hierárquicos.

- herança múltipla utilizada desmedidamente.

- grande profundidade e largura da árvore de herança.

- dificuldade em saber onde um determinado método é definido.

- dificuldade em saber quais atributos são necessários para o funcionamento de quais métodos.

- devido ao forte acoplamento, é muito difícil manter o sistema, sem perder muito tempo "desvendando" as relações entre classes, métodos, atributos.

Solução refatorada:

- se o sistema não for muito grande é interessante traçar um mapa de todas classes do sistema sem se preocupar com seus relacionamentos.

- adota-se uma abordagem bottom-up, inicia-se o sistema pela sua primeira linha, e procura-se pela primeira instanciação de um objeto.

- a partir deste momento é preciso olhar o código da classe deste objeto e começar a ver as dependências e associações desta classe, marcando-as no mapa de classes.

- se ao final houver classes não marcadas no mapa, estas são possíveis classes 'aposentadas' e deve-se removê-las transportando sua funcionalidade para a classe pai.

- substituir mecanismos de herança por mecanismos que buscam um maior desacoplamento como delegação, e utilizar padrões de projeto como decorator ou visitor.

- a depender do tamanho do sistema e da falta de ferramentas auxiliares, este trabalho pode demandar várias semanas.