

CODE SMELLS



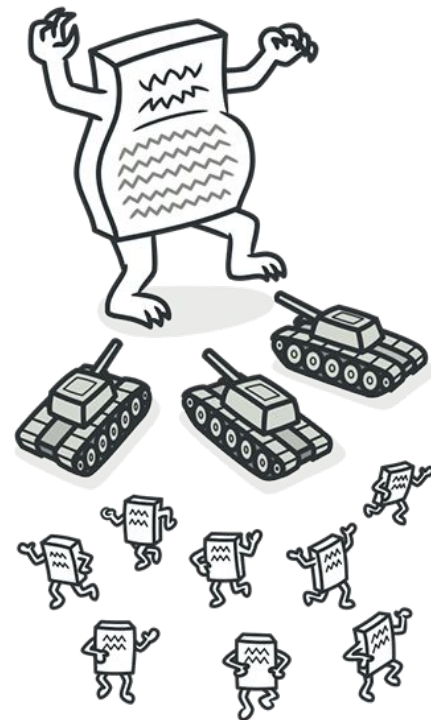
Cosa sono i CODE SMELLS?



Bloaters

I code smells di tipologia bloaters sono:

- Long Method
- Large Class
- Primitive Obsession
- Long Parameter List
- Data Clumps



Bloaters - Long Method

Un metodo contiene troppe righe di codice



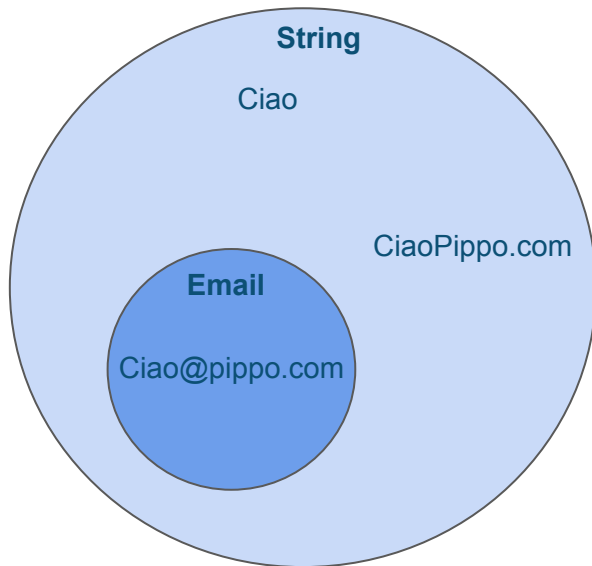
Bloaters - Large Class

Una classe che contiene molti campi, metodi, linee di codice...



Bloaters - Primitive Obsession

Si utilizzano in maniera eccessiva oggetti primitivi



Bloaters - Long Parameter List

Vengono passati molti parametri ad una funzione



Bloaters - Data Clumps

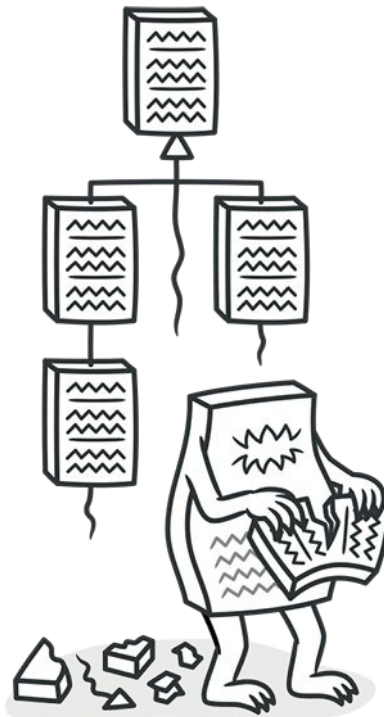
Diverse parti del codice contengono lo stesso identico gruppi di variabili



Object-Orientation Abusers

I code smells di Object-Orientation Abusers sono:

- Alternative Classes with Different Interfaces
- Refused Bequest
- Switch Statements
- Temporary Field



Object-Orientation Abusers - Alternative Classes with Different Interfaces

Due classi eseguono funzioni identiche ma hanno nomi di metodi diversi



Object-Orientation Abusers - Refused Bequest

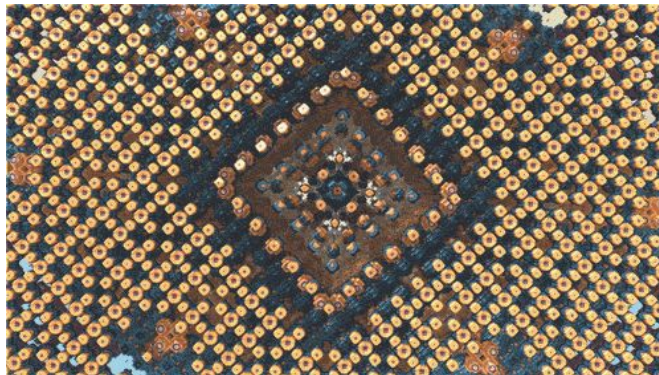
Se una sottoclasse utilizza solo alcuni dei metodi e delle proprietà ereditati dai suoi genitori, la gerarchia è sbilanciata



Object-Orientation Abusers - Switch Statements



Hai uno switch/if complesso



Object-Orientation Abusers - Temporary Field

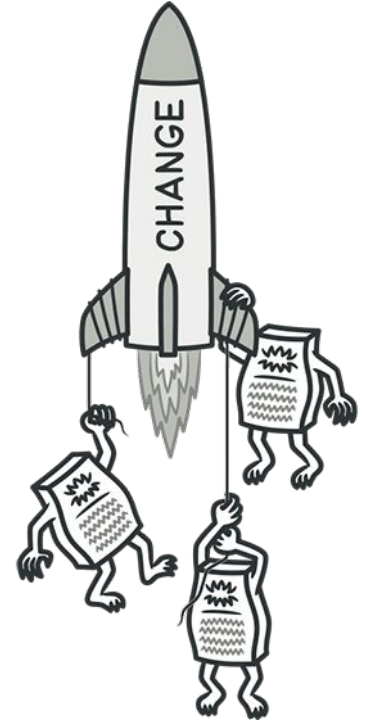
Le variabili ottengono i loro valori solo in determinate circostanze



Change Preventers

I code smells di tipologia Change Preventers sono:

- Divergent Change
- Parallel Inheritance Hierarchies
- Shotgun Surgery



Change Preventers - Divergent Change

Si verifica quando devono essere apportate molte modifiche a una singola classe



Change Preventers - Shotgun Surgery

Si verifica quando una singola modifica deve essere apportata a più classi contemporaneamente.



Change Preventers - Parallel Inheritance Hierarchies

Ogni volta che crei una sottoclasse per una classe,
ti ritrovi a dover creare una sottoclasse per un'altra classe



Dispensables

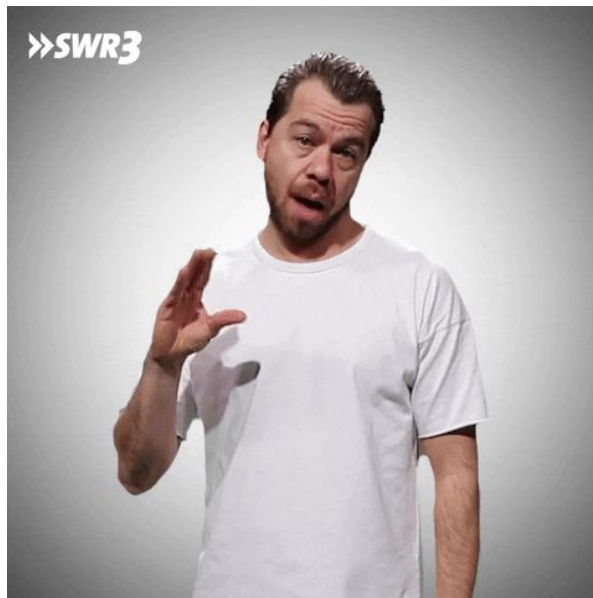
I code smells di tipologia dispensables sono:

- Comments
- Duplicate Code
- Data Class
- Dead Code
- Lazy Class
- Speculative Generality



Dispensables – Comments

Un metodo è pieno di commenti esplicativi



Dispensables - Duplicate Code

Due frammenti di codice sembrano quasi identici



Dispensables - Data Class

Una classe di dati si riferisce a una classe che contiene solo campi e metodi grezzi per accedervi



Dispensables - Dead Code

Una variabile, parametro, campo, metodo o classe non viene più utilizzata (solitamente perché obsoleta)



Dispensables - Lazy Class

Comprendere e mantenere le classi costa sempre tempo e denaro.

Pertanto, se una classe non fa abbastanza per attirare la tua attenzione, dovrebbe essere eliminata



Dispensables - Speculative Generality

È presente una classe, un metodo, un campo o un parametro inutilizzato



Couplers

I code smells di tipologia couplers sono:

- Feature Envy
- Inappropriate Intimacy
- Message Chains
- Middle Man



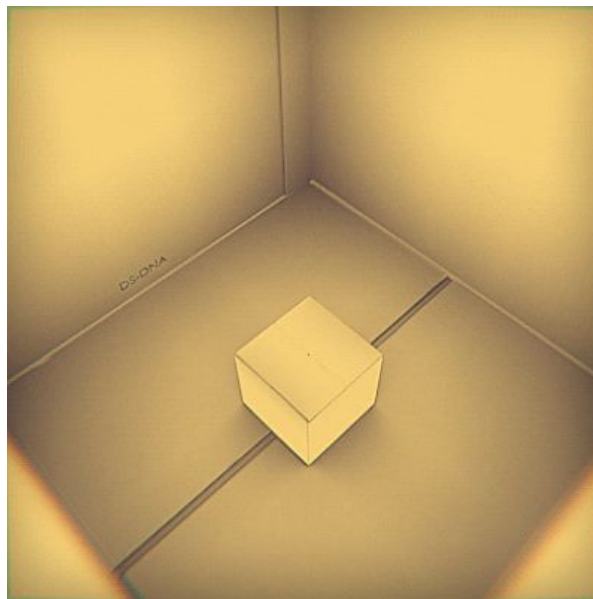
Couplers - Feature Envy

Un metodo accede ai dati di un altro oggetto più che ai propri dati



Couplers - Inappropriate Intimacy

Una classe utilizza i campi e i metodi interni di un'altra classe



Couplers - Message Chains

Nel codice vedi una serie di chiamate simili $a \rightarrow b() \rightarrow c() \rightarrow d()$



Couplers - Middle Man

Se una classe esegue solo un'azione, delegando il lavoro a un'altra classe, perché esiste?



QUESTIONS?

