PEMROGRAMAN LANJUT

Code Smells: Coupler

Oleh
Tri Hadiah Muliawati
Politeknik Elektronika Negeri Surabaya
2021

Politeknik Elektronika Negeri Surabaya
Departemen Teknik Informatika dan Komputer

Coupler

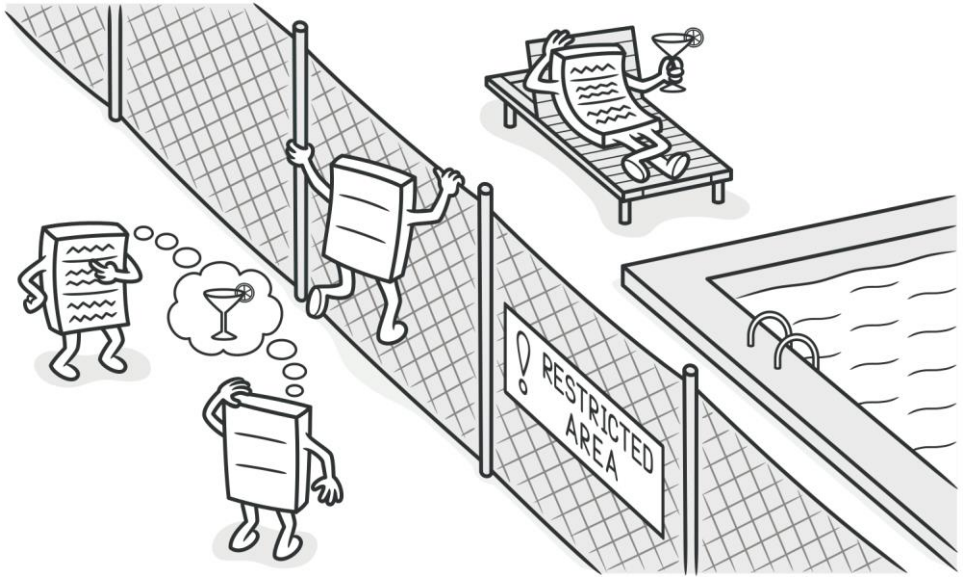
All the smells in this group contribute to excessive coupling between classes or show what happens if coupling is replaced by excessive delegation

Coupler

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

Feature Envy

- A method accesses the data of another object more than its own data.

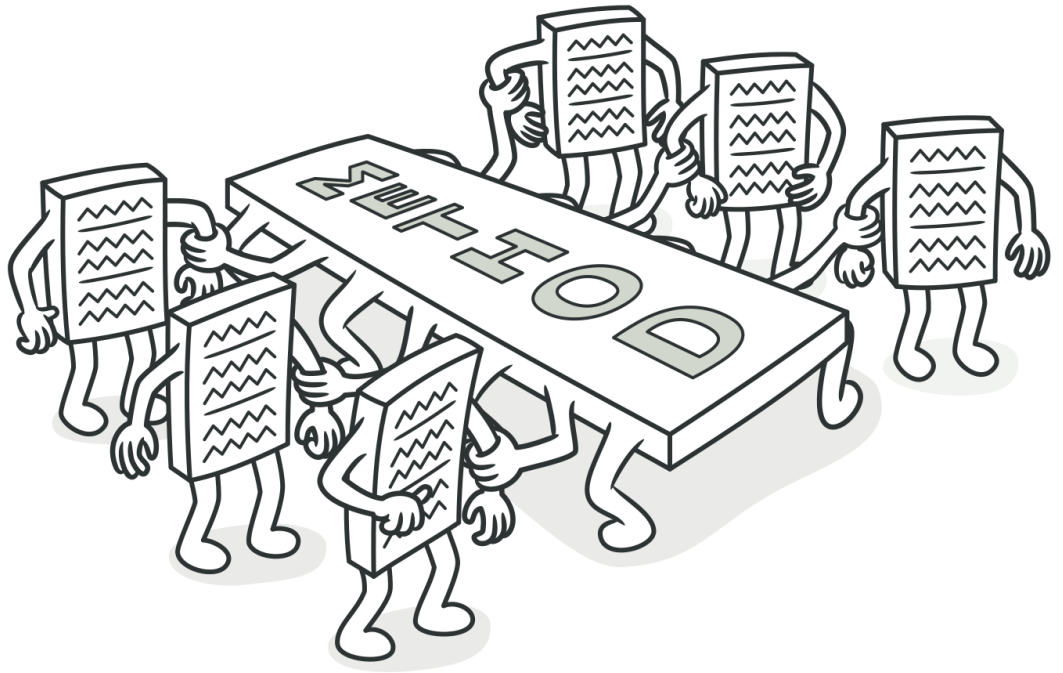


Feature Envy: Refactoring

- **Move Method:** If a method clearly should be moved to another place.
- **Extract Method:** to move the part of a method which accesses the data of another object.
- If a method uses functions from several other classes:
  - Determine which class contains most of the data used.
  - Place the method in this class along with the other data.
  - Alternatively, use **Extract Method** to split the method into several parts that can be placed in different places in different classes.

Inappropriate Intimacy

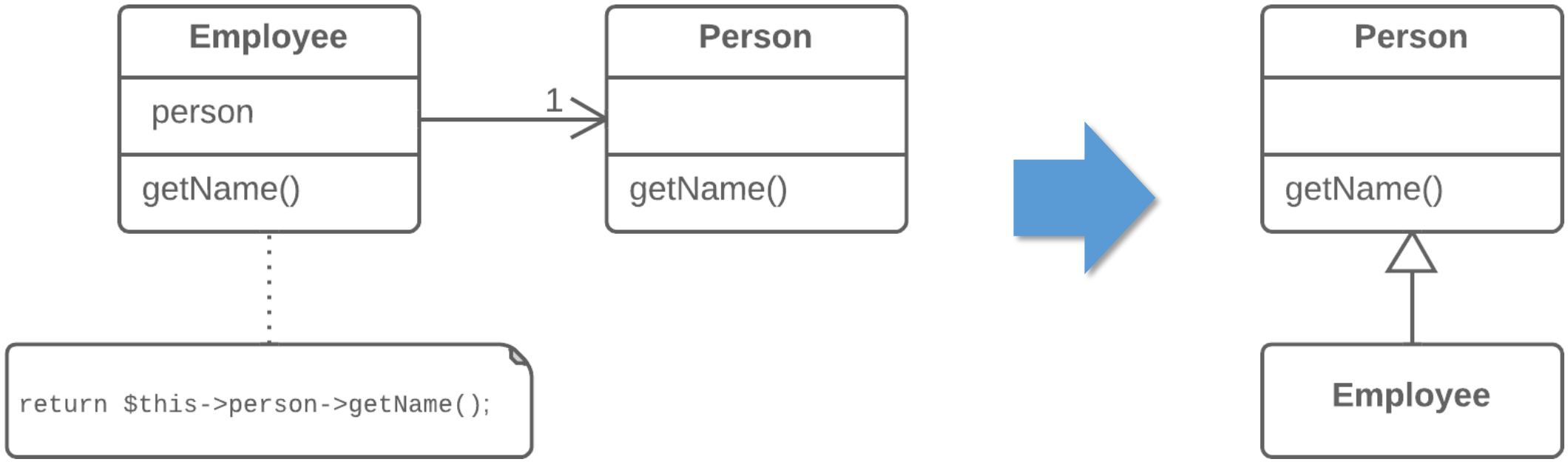
- One class uses the internal fields and methods of another class



Inappropriate Intimacy: Refactoring

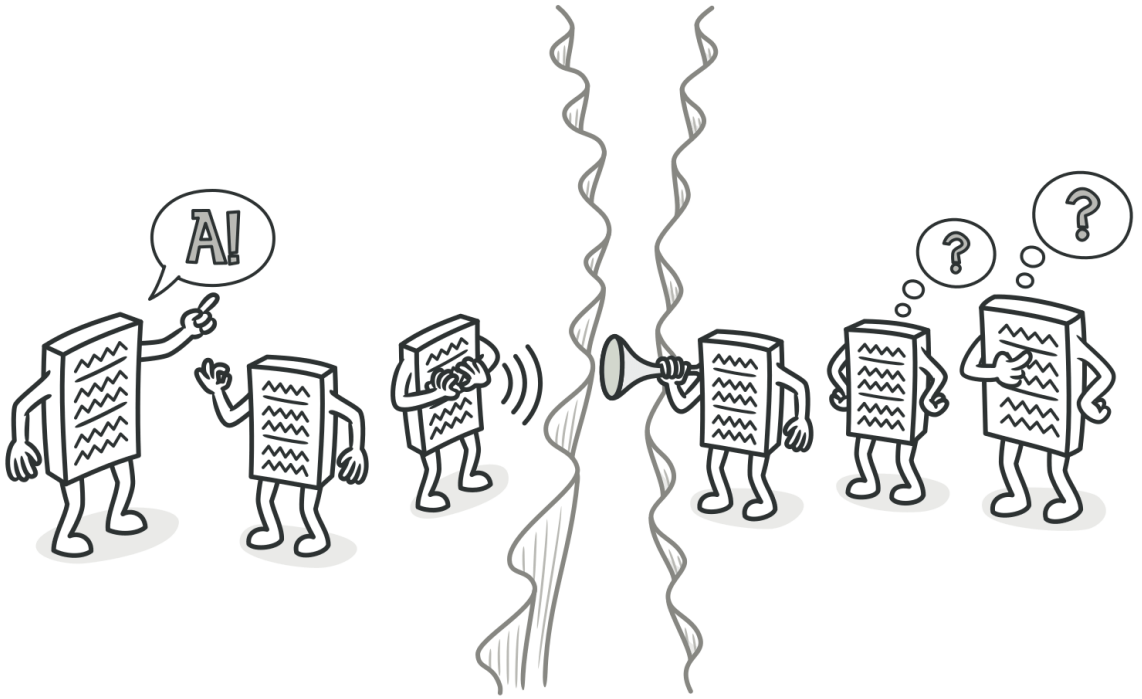
- **Move Method** and **Move Field:** If the original class doesn't need these method and field, move those parts of original class to the class in which those parts are used.
- **Replace Delegation with Inheritance:** If those classes can be turned into subclass and superclass.

Case Study: Replace Delegation with Inheritance



Message Chains

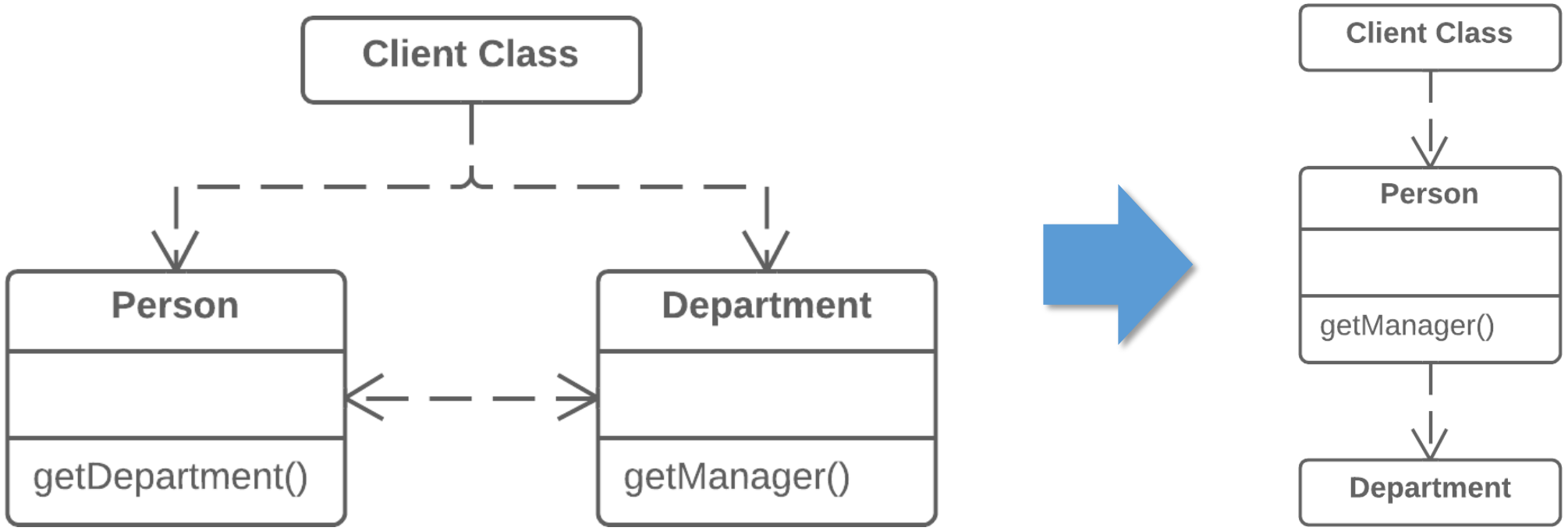
- In code you see a series of calls resembling $a->b()->c()->d()
- These chains mean that the client is dependent on navigation along the class structure.
- Any changes in these relationship require modifying the client



Message Chains: Refactoring

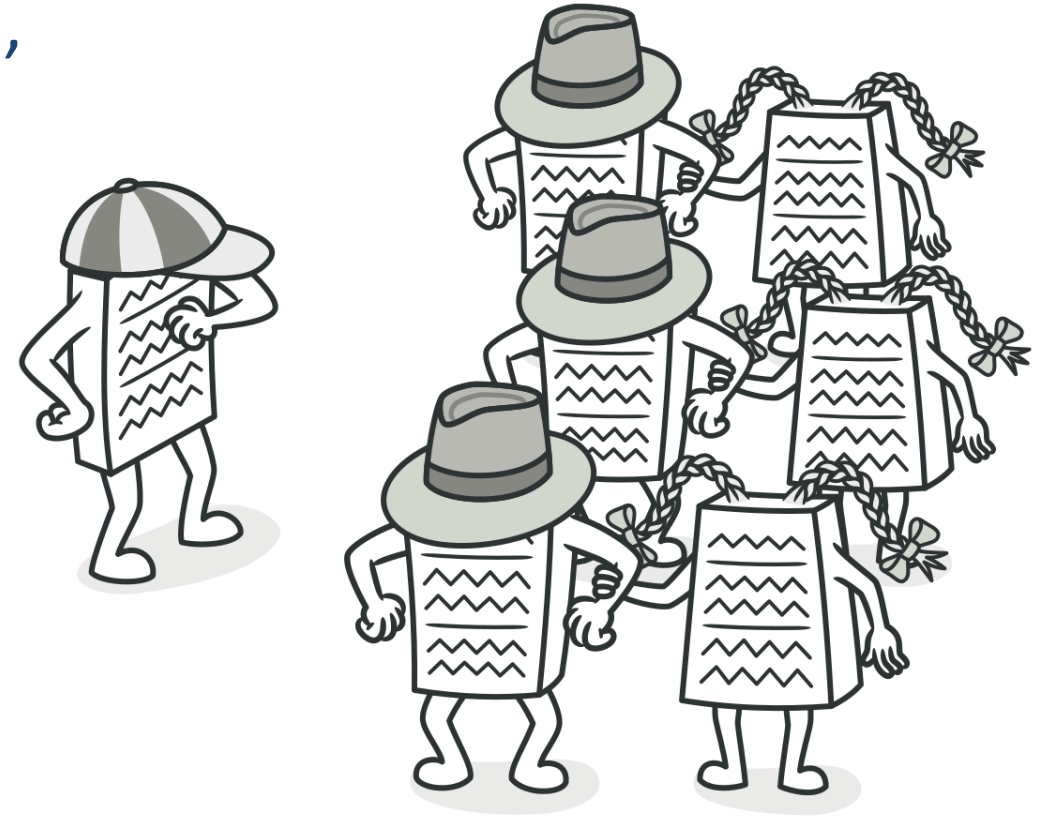
- **Hide Delegate:** To delete a message chain. Use cautiously, excessive usage of this refactoring method may cause other code smell (Middle Man).
- **Extract Method** and **Move Method:** If it makes more sense to move the functionality to the beginning of the chain.

Case Study: Hide Delegate



Middle Man

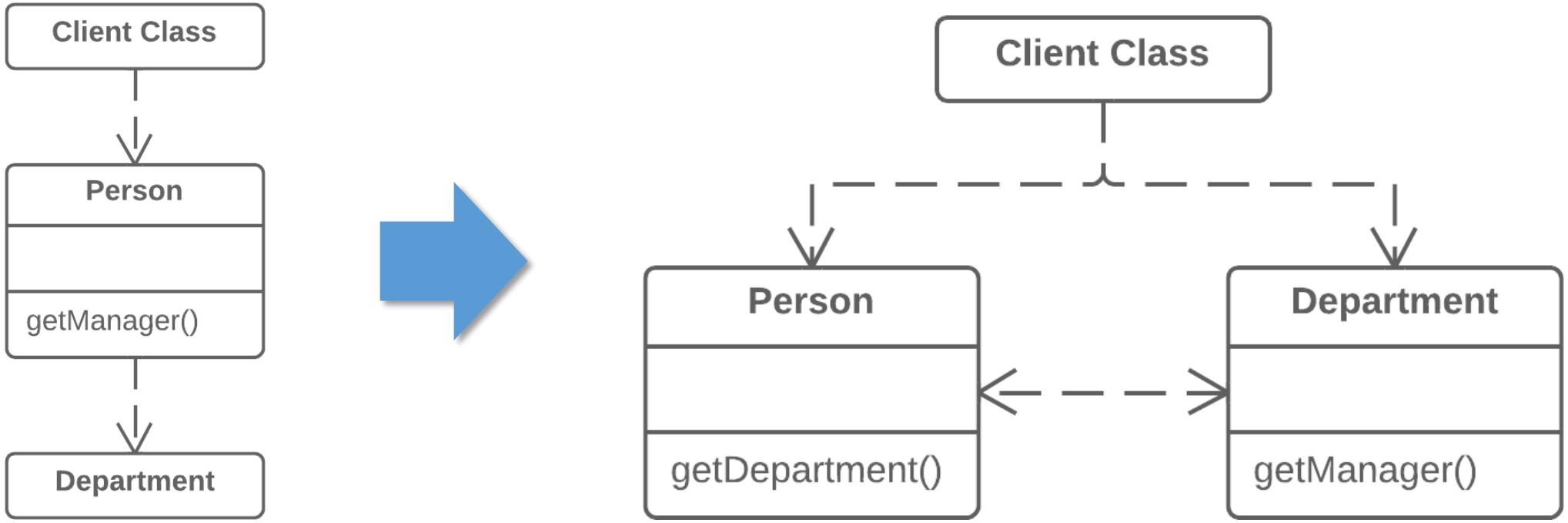
- If a class performs only one action, delegating work to another class, why does it exist at all?



Middle Man: Refactoring

- **Remove Middle Man:** Reverse the delegate hiding

Case Study: Remove Middle Man



References

- Fowler, Martin. Refactoring: Improving the Design of Existing Code. Addison-Wesley Professional, 1999.
- https://refactoring.guru/

bridge to the future

http://www.eepis-its.edu