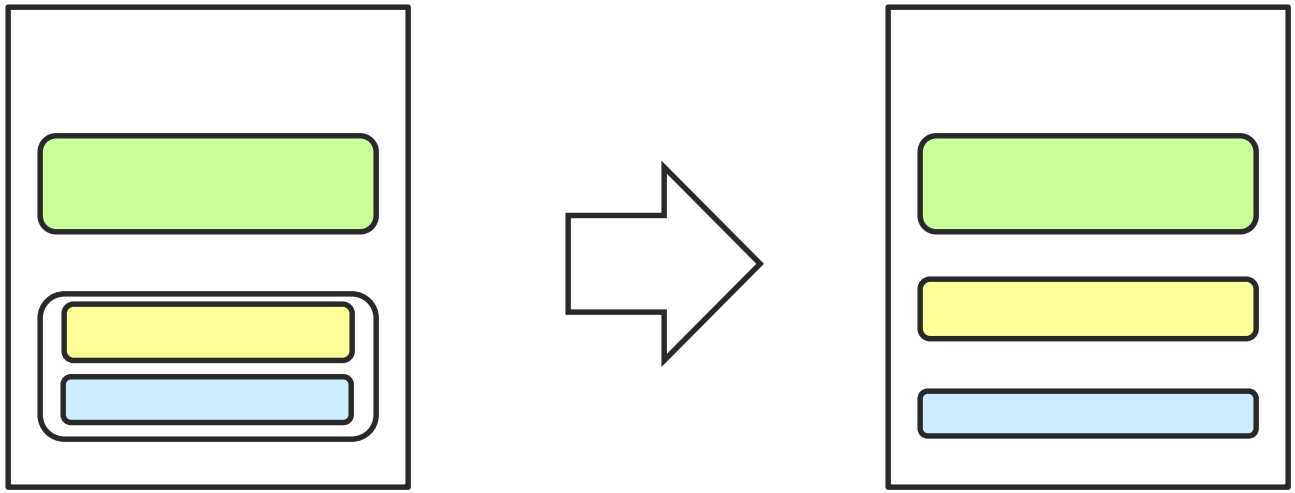
Refactoring Examples

Eduardo Figueiredo

http://www.dcc.ufmg.br/~figueiredo

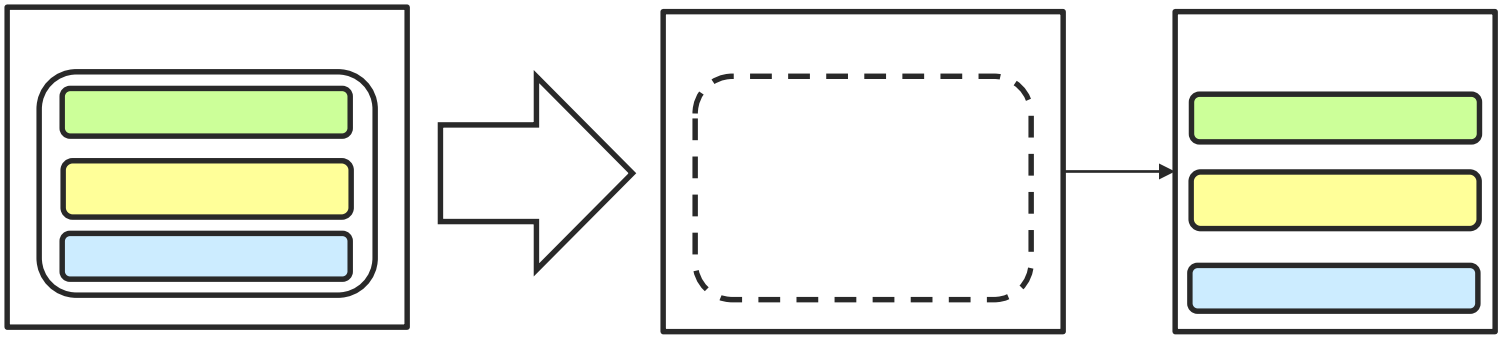
Extract Method

- You have a code fragment (part of a method) that can be grouped together
  - Turn the fragment into a new method whose name explains its purpose



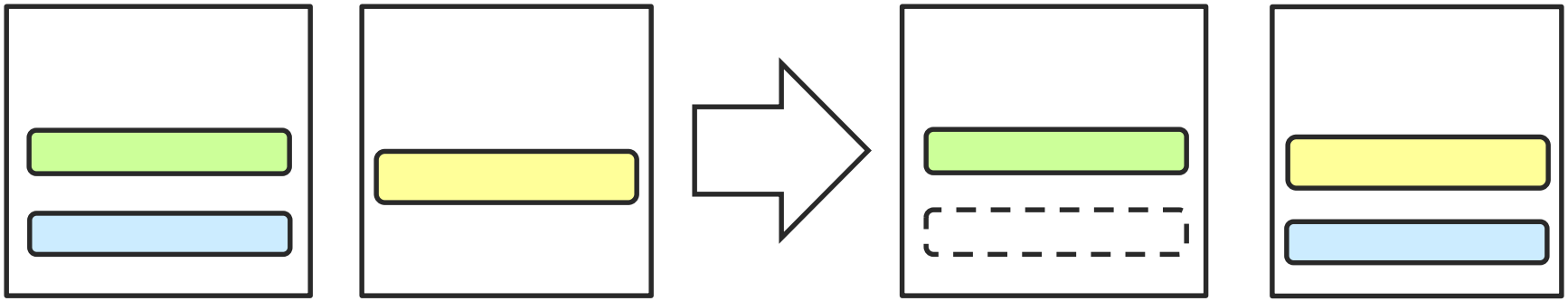
Replace Method with Method Object

- You have a long method that uses local variables in such a way that you cannot apply Extract Method
  - Turn the method into its own object
  - All local variables become fields
  - Decompose the method into other methods



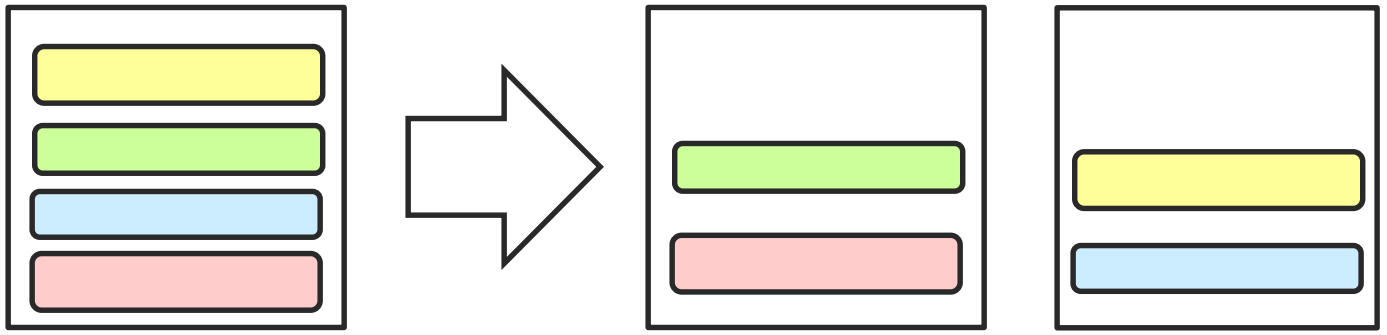
Move Method

- A method is using more features of another class than its own class
  - Create a new method with a similar body in the class it uses most
  - Either remove or turn the old method into a delegation



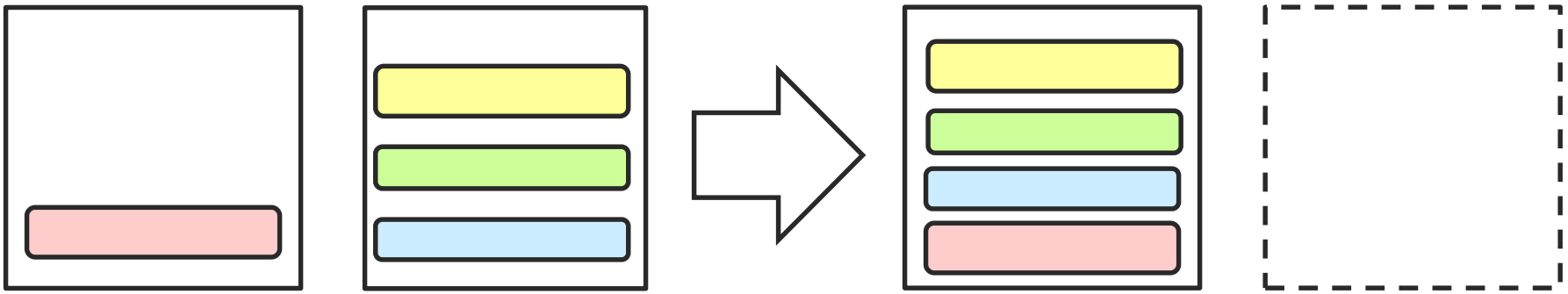
Extract Class

- You have one class doing work that should be done by two
  - Create a new class
  - Move the relevant fields and methods from the old class into the new class



Inline Class

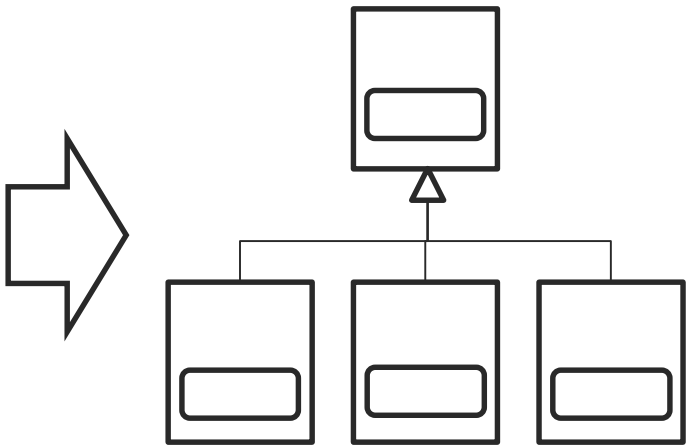
- A class isn’t doing very much
  - Move all its features into another class
  - Delete the empty class



Replace Conditional with Polymorphism

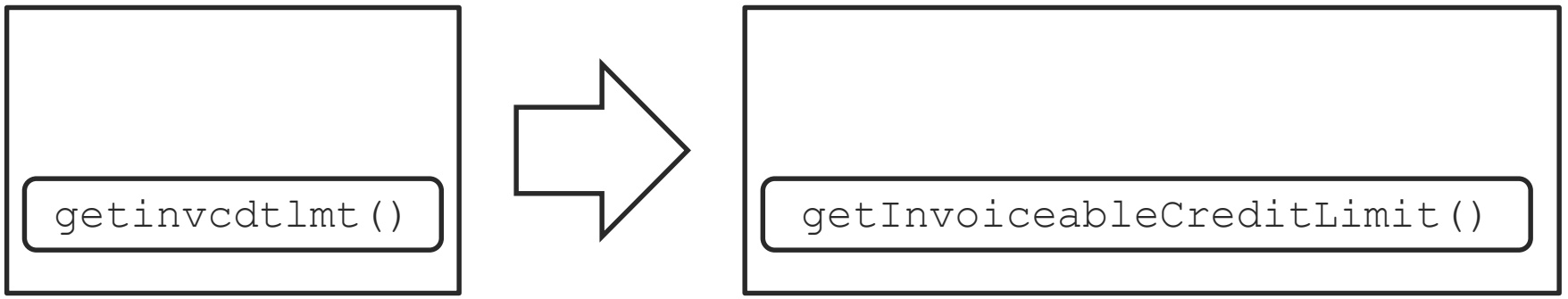
- You have a conditional that chooses different behavior depending on the type
  - Move each leg of the conditional to a overriding method in a subclass
  - Make the original method abstract

```
double speed() {
  switch(type) {
    case(EUROPEAN)
      return getBaseSpeed();
    case(AFRICAN)
      return getBaseSpeed()*nunCoconuts;
    case(NORWEGIAN)
      return getBaseSpeed()-loadFactor();
}}
```



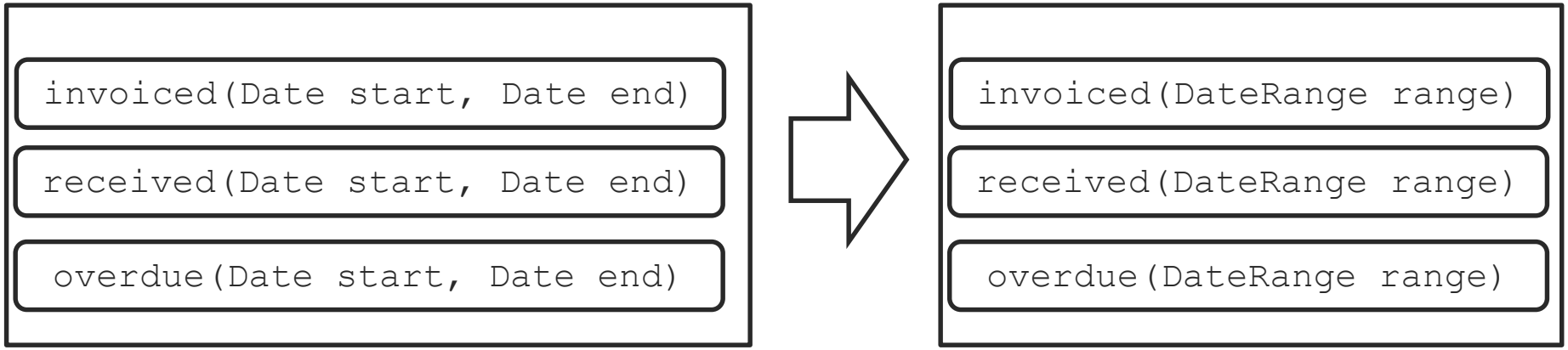
Rename Method

- The name of a method does not reveal its purpose
  - Change the name of the method



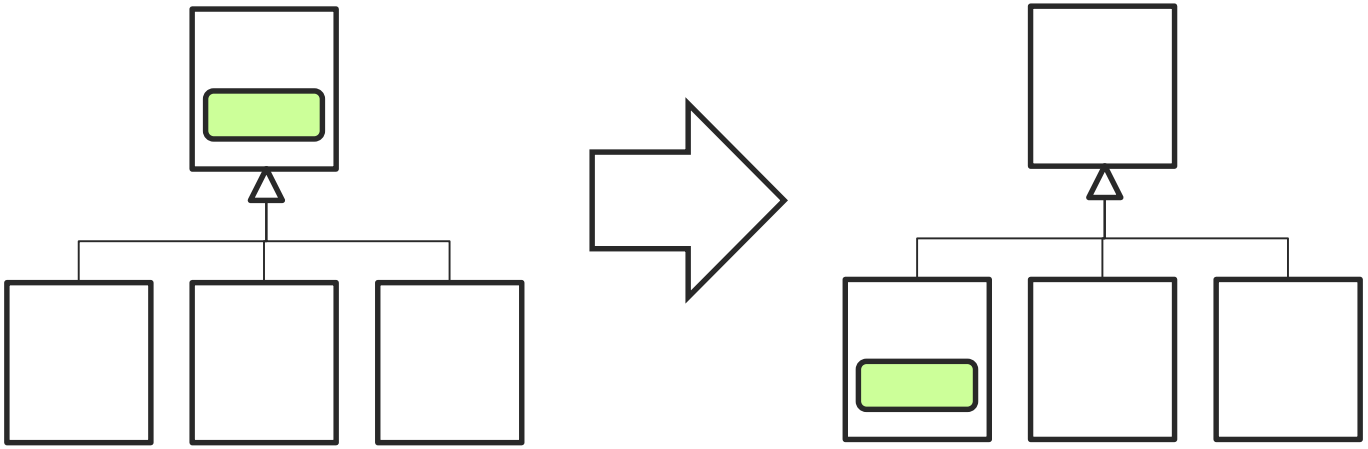
Introduce Parameter Object

- You have a group of parameters that naturally go together
  - Replace them with an object



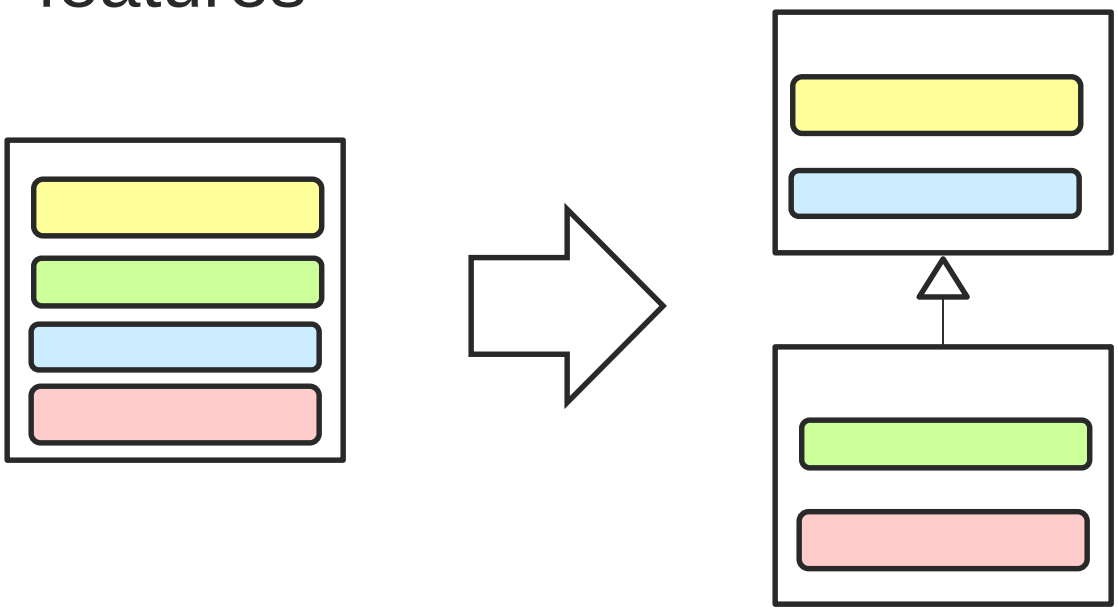
Push Down Method

- Behavior on a superclass is relevant only for some of its subclasses
  - Move it to those subclasses



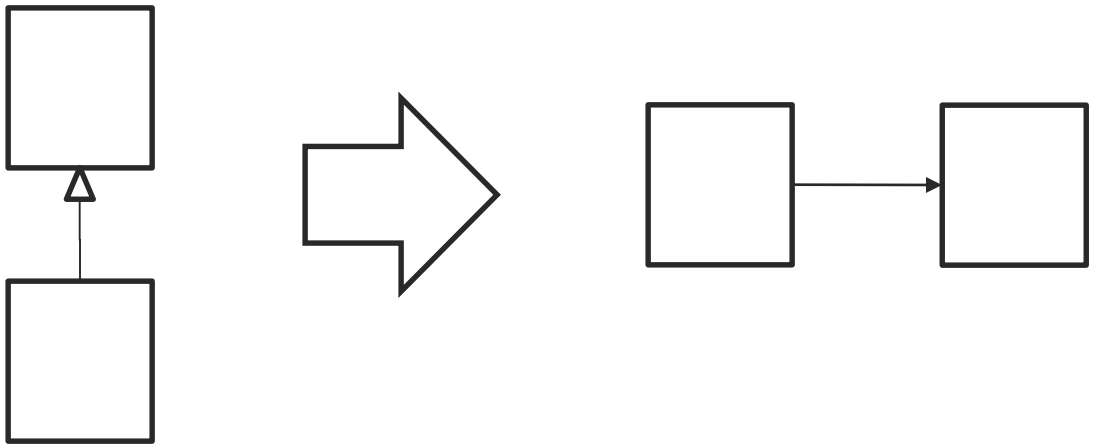
Extract Subclass

- A class has feature that are used only in some instances
  - Create a subclass for that subset of features



Replace Inheritance with Delegation

- A subclass uses only part of a superclass interface
  - Create a field for the superclass
  - Adjust methods to delegate to superclass
  - Remove the inheritance relationship



Bibliography

- Martin Fowler. **Refactoring: Improving the Design of Existing Code**. Addison-Wesley Professional, 1st edition, 1999.
  - Chapters 2, 6, 7, 8, 9, 10, 11, and 12