Refactoring via Program Slicing and Sliding

Ran Ettinger
Programming Tools Group
University of Oxford

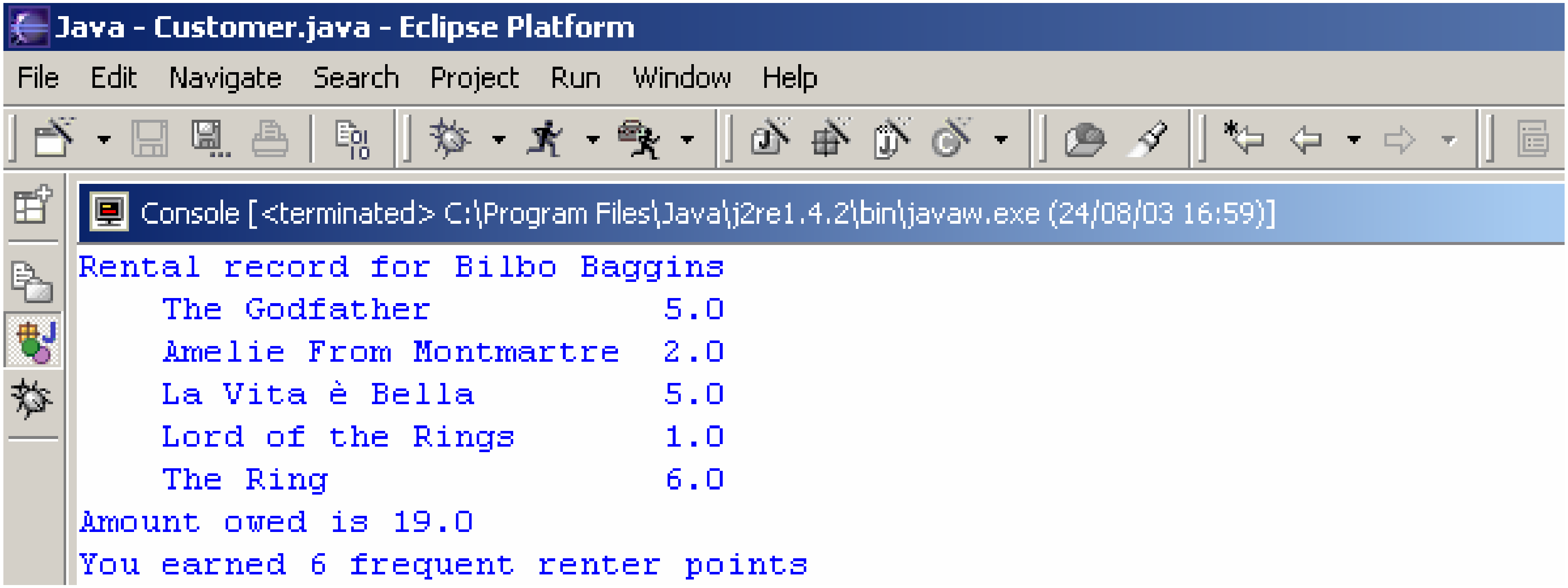
In the Advanced Software Tools research seminar
Tel Aviv University
30 October, 2006

Refactoring enables
iterative and incremental
software development

The gap:

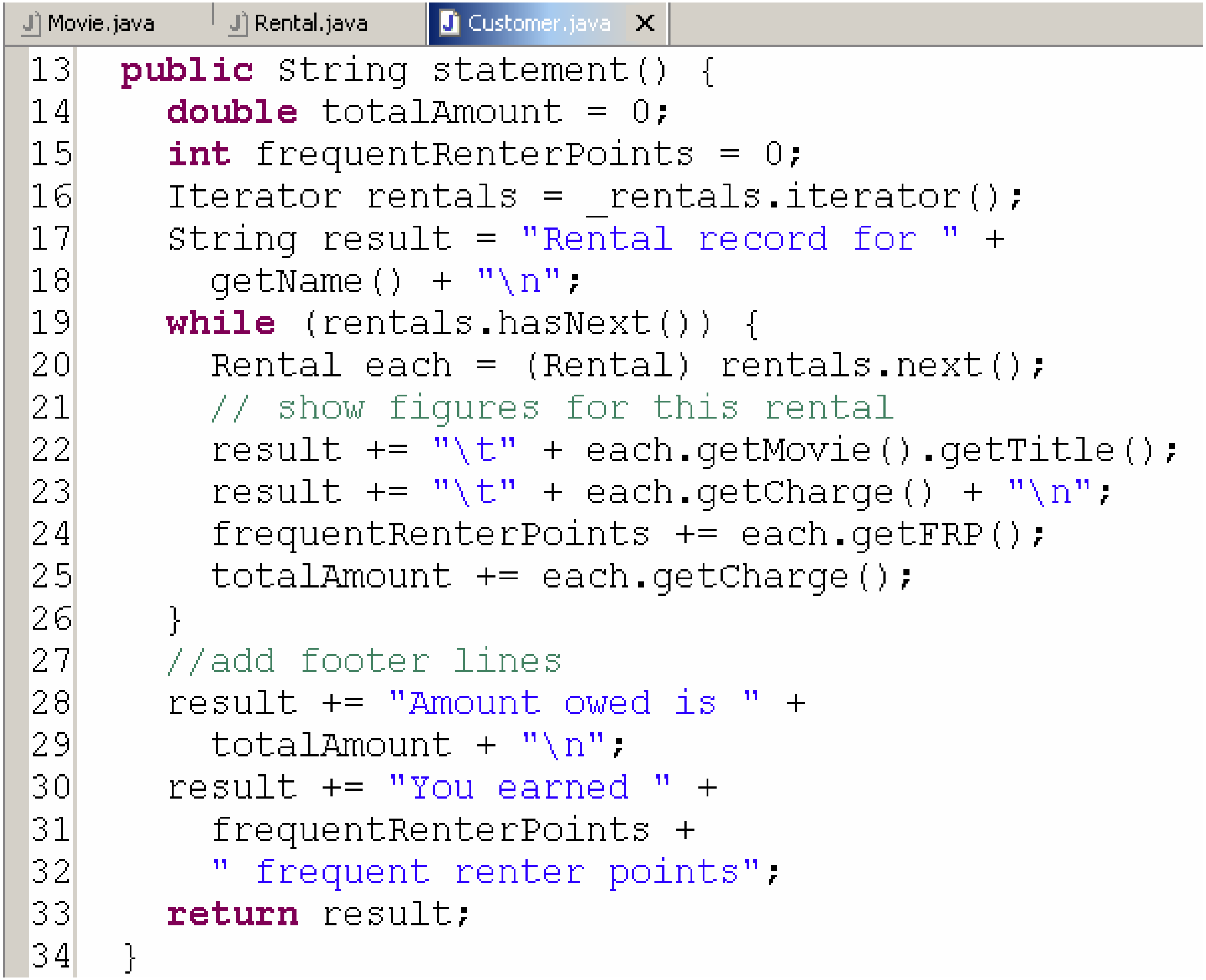
Refactoring tools are important but weak

Example: Fowler’s video-store

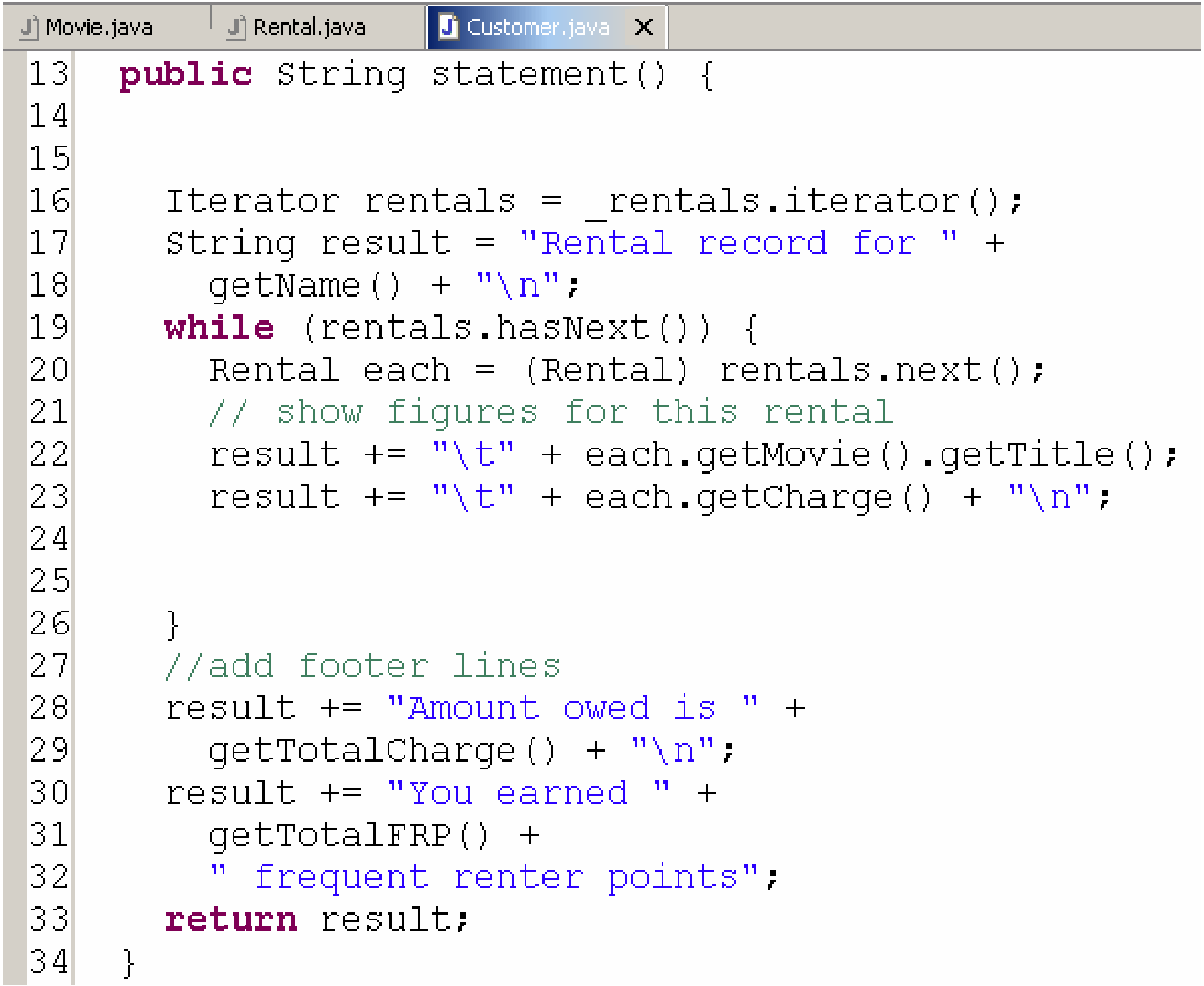


Console [<terminated> C:\Program Files\Java\j2re1.4.2\bin\javaw.exe (24/08/03 16:59)]

```
Rental record for Bilbo Baggins
	The Godfather	5.0
	Amelie From Montmartre	2.0
	La Vita è Bella	5.0
	Lord of the Rings	1.0
	The Ring	6.0
Amount owed is 19.0
You earned 6 frequent renter points
```



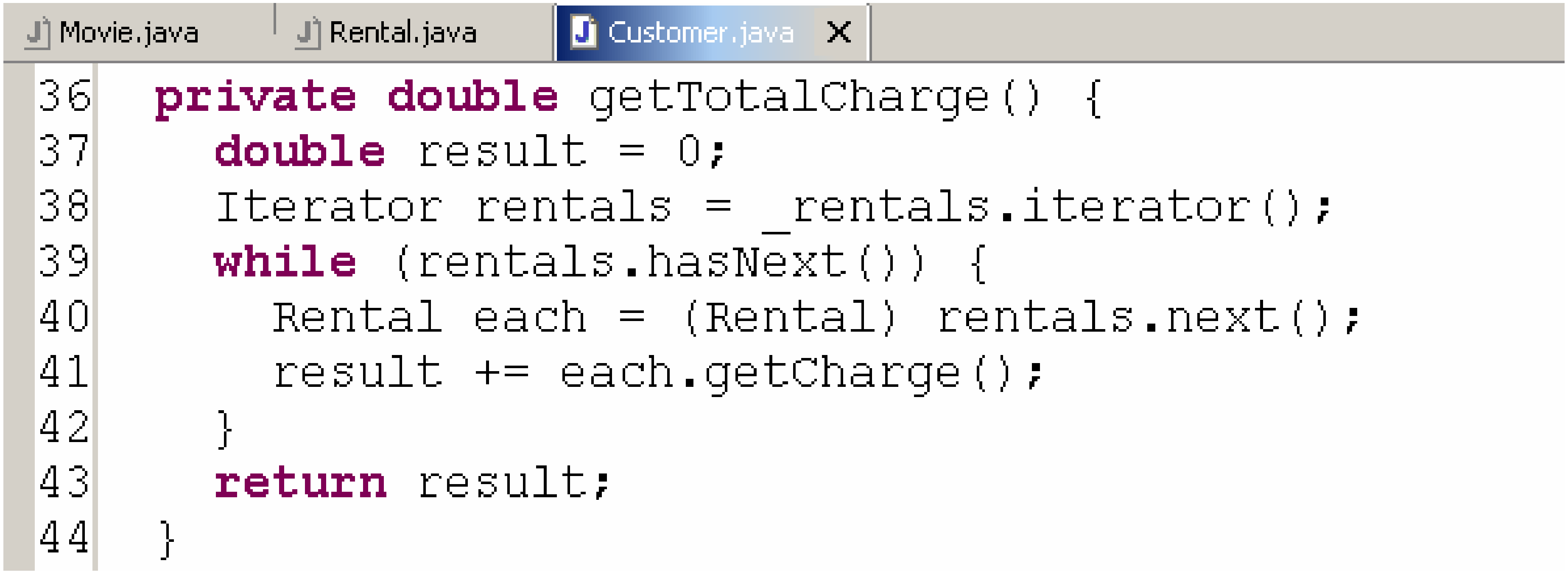
```
  public String statement() {
    double totalAmount = 0;
    int frequentRenterPoints = 0;
    Iterator rentals = _rentals.iterator();
    String result = "Rental record for " +
      getName() + "\n";
    while (rentals.hasNext()) {
      Rental each = (Rental) rentals.next();
      // show figures for this rental
      result += "\t" + each.getMovie().getTitle();
      result += "\t" + each.getCharge() + "\n";
      frequentRenterPoints += each.getFRP();
      totalAmount += each.getCharge();
    }
    //add footer lines
    result += "Amount owed is " +
      totalAmount + "\n";
    result += "You earned " +
      frequentRenterPoints +
      " frequent renter points";
    return result;
  }
```



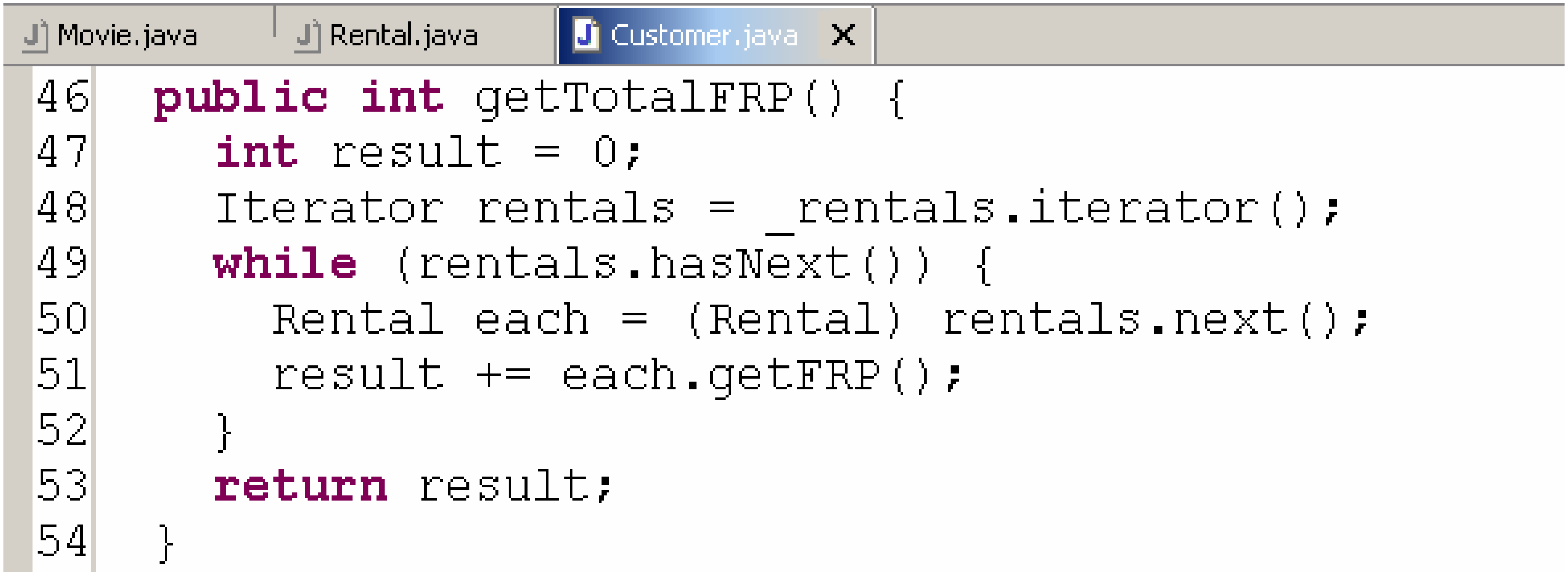
```
public String statement() {


  Iterator rentals = _rentals.iterator();
  String result = "Rental record for " +
    getName() + "\n";
  while (rentals.hasNext()) {
    Rental each = (Rental) rentals.next();
    // show figures for this rental
    result += "\t" + each.getMovie().getTitle();
    result += "\t" + each.getCharge() + "\n";


  }
  //add footer lines
  result += "Amount owed is " +
    getTotalCharge() + "\n";
  result += "You earned " +
    getTotalFRP() +
    " frequent renter points";
  return result;
}
```



```java
private double getTotalCharge() {
  double result = 0;
  Iterator rentals = _rentals.iterator();
  while (rentals.hasNext()) {
    Rental each = (Rental) rentals.next();
    result += each.getCharge();
  }
  return result;
}
```



```java
public int getTotalFRP() {
  int result = 0;
  Iterator rentals = _rentals.iterator();
  while (rentals.hasNext()) {
    Rental each = (Rental) rentals.next();
    result += each.getFRP();
  }
  return result;
}
```

Programmers use slices when refactoring

Challenge:

Automatic slice-extraction

Approach:

Program Sliding

Live sliding examples…

Potential implications of slides and sliding:

Refactoring and beyond

Thanks!