

Engenharia de Software II

Refatoração - Parte 1

Prof. André Hora
DCC/UFMG
2019.1

Categorias de Manutenção

- Manutenção corretiva
- **Manutenção preventiva**
- Manutenção adaptativa
- Manutenção perfectiva

Manutenção Preventiva

- Modificações no software para prevenir potenciais problemas no futuro
- Lida com o deterioramento de estruturas, previne falhas e melhora a manutenabilidade
- Torna os programas mais fáceis de entender e facilita trabalhos de manutenção futuros

Exemplo



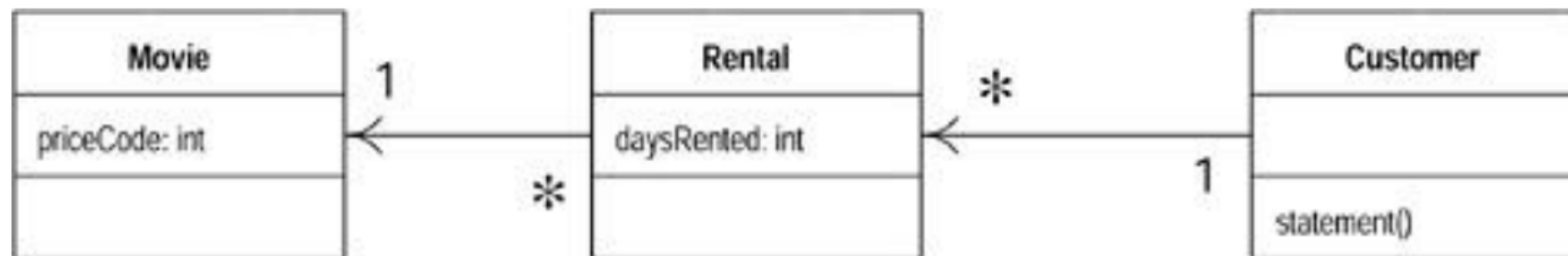
- Refatoração por exemplo
- Programa ilustrativo no contexto de uma Locadora de DVDs
- Calcula e imprime a fatura de um cliente em uma locadora de DVDs

Locadora de DVDs

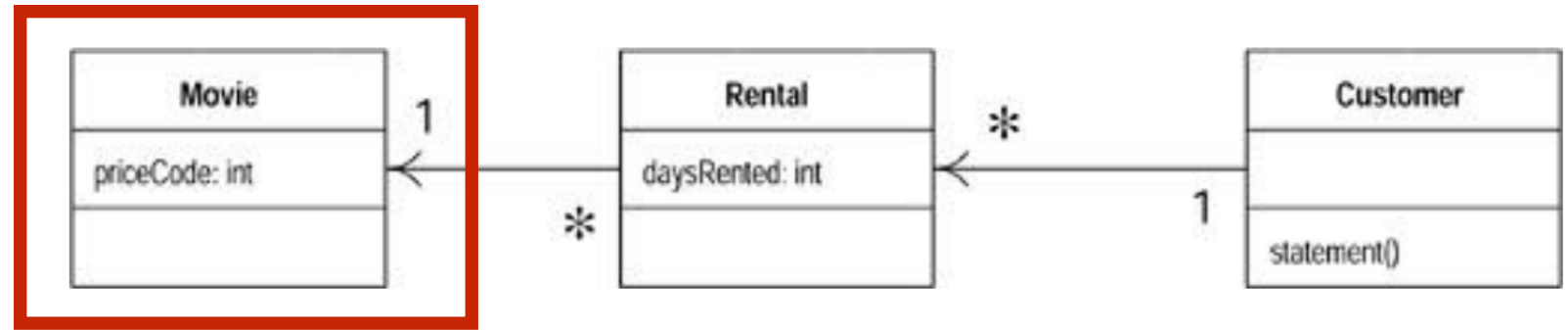
- Calcula e imprime a fatura de um cliente em uma locadora de DVDs
 - **Entrada:** quais filmes foram alugados e por quanto tempo
 - **Saída:** valores parciais (depende do tempo alugado e do tipo do filme), total & pontuação acumulada
- Tipos de filme: Regular, Children e New Release
- Pontuação: varia de acordo com o tipo do filme

Diagrama de Classes

- Classes: Movie, Rental e Customer



Movie



```
public class Movie {

    public static final int CHILDREN = 2;
    public static final int REGULAR = 0;
    public static final int NEW_RELEASE = 1;

    private String title;
    private int priceCode;

    public Movie(String title, int priceCode) {
        this.title = title;
        this.priceCode = priceCode;
    }

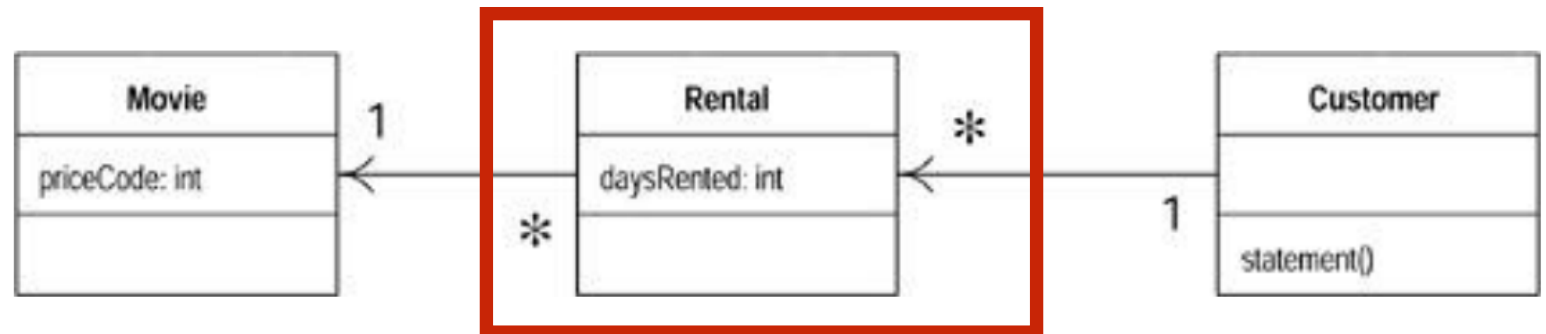
    public String getTitle() {
        return title;
    }

    public int getPriceCode() {
        return priceCode;
    }

    public void setPriceCode(int priceCode) {
        this.priceCode = priceCode;
    }

}
```

Rental



```
public class Rental {

    private Movie movie;
    private int daysRented;

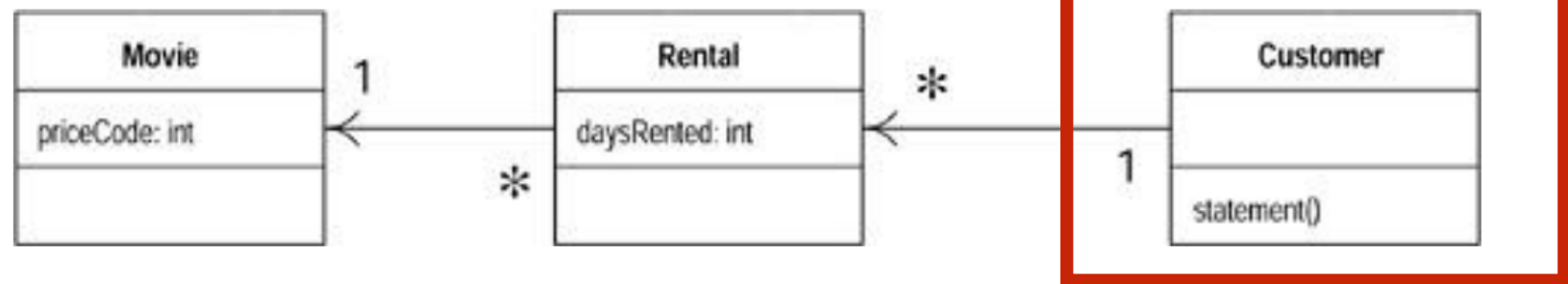
    public Rental(Movie movie, int daysRented) {
        this.movie = movie;
        this.daysRented = daysRented;
    }

    public Movie getMovie() {
        return movie;
    }

    public int getDaysRented() {
        return daysRented;
    }

}
```


Customer (1)



```
public class Customer {

    private String name;
    private List<Rental> rentals = new ArrayList<Rental>();

    public Customer(String name) {
        this.name = name;
    }

    public String getName() {
        return name;
    }

    public void addRental(Rental rental) {
        rentals.add(rental);
    }

    ...
}
```

Customer (2)

```

public String statement() {
    double totalAmount = 0;
    int frequentRenterPoints = 0;

    String result = "Rental record for " + getName() + "\n";
    for (Rental rental : rentals) {
        double amount = 0;
        switch (rental.getMovie().getPriceCode()) {
            case Movie.REGULAR:
                amount += 2;
                if (rental.getDaysRented() > 2)
                    amount += (rental.getDaysRented() - 2) * 1.5;
                break;
            case Movie.NEW_RELEASE:
                amount += rental.getDaysRented() * 3;
                break;
            case Movie.CHILDREN:
                amount += 1.5;
                if (rental.getDaysRented() > 3)
                    amount += (rental.getDaysRented() - 3) * 1.5;
                break;
        }

        // add frequent renter points
        frequentRenterPoints++;
        // add bonus for a two day new release rental
        if (rental.getMovie().getPriceCode() == Movie.NEW_RELEASE && rental.getDaysRented() > 1)
            frequentRenterPoints++;

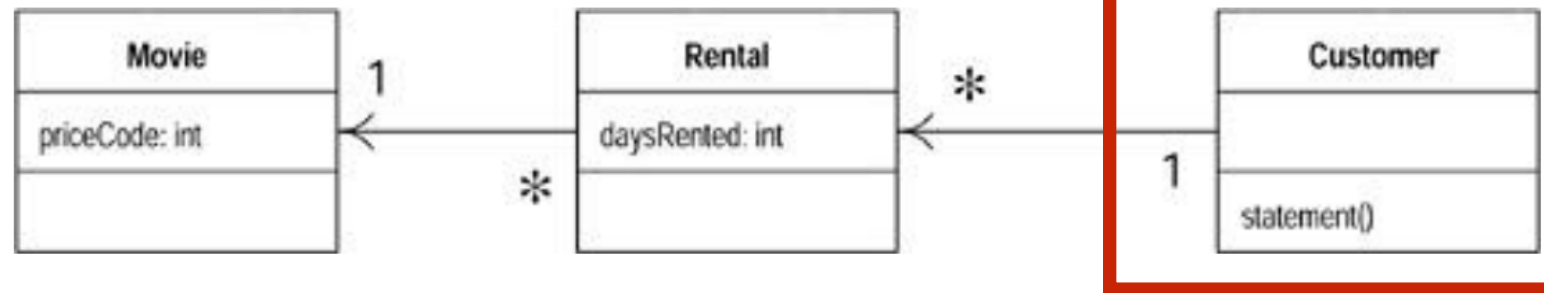
        // show figures for this rental
        result += "\t" + rental.getMovie().getTitle() + "\t" + String.valueOf(amount) + "\n";

        totalAmount += amount;
    }

    result += "Amount owed is " + String.valueOf(totalAmount) + "\n";
    result += "You earned " + String.valueOf(frequentRenterPoints) + " frequent renter points";

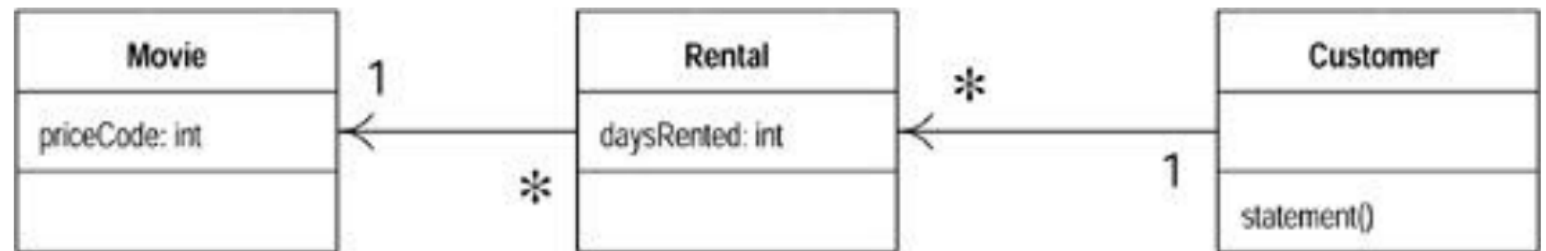
    return result;
}

```



Customer (2)

```
public String statement() {
    double totalAmount = 0;
    int frequentRenterPoints = 0;
```



```
String result = "Rental record for " + getName() + "\n";
```

```
for (Rental rental : rentals) {
```

```
    double amount = 0;
```

```
    switch (rental.getMovie().getPriceCode()) {
```

```
        case Movie.REGULAR:
```

```
            amount += 2;
```

```
            if (rental.getDaysRented() > 2)
```

```
                amount += (rental.getDaysRented() - 2) * 1.5;
```

```
            break;
```

```
        case Movie.NEW_RELEASE:
```

```
            amount += rental.getDaysRented() * 3;
```

```
            break;
```

```
        case Movie.CHILDREN:
```

```
            amount += 1.5;
```

```
            if (rental.getDaysRented() > 3)
```

```
                amount += (rental.getDaysRented() - 3) * 1.5;
```

```
            break;
```

```
    }
```

```
    // add frequent renter points
```

```
    frequentRenterPoints++;
```

```
    // add bonus for a two day new release rental
```

```
    if (rental.getMovie().getPriceCode() == Movie.NEW_RELEASE && rental.getDaysRented() > 1)
```

```
        frequentRenterPoints++;
```

```
    // show figures for this rental
```

```
    result += "\t" + rental.getMovie().getTitle() + "\t" + String.valueOf(amount) + "\n";
```

```
    totalAmount += amount;
```

```
}
```

```
result += "Amount owed is " + String.valueOf(totalAmount) + "\n";
```

```
result += "You earned " + String.valueOf(frequentRenterPoints) + " frequent renter points";
```

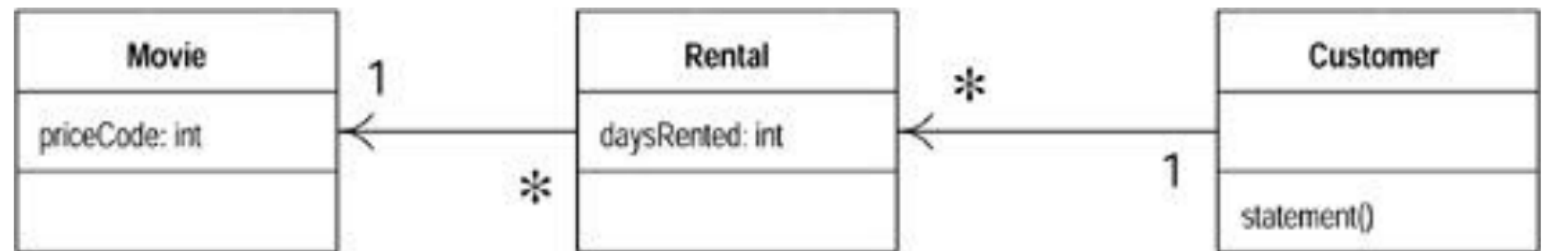
```
return result;
```

```
}
```

```
}
```

Calcula preço parcial

Customer (2)



```

public String statement() {
    double totalAmount = 0;
    int frequentRenterPoints = 0;

    String result = "Rental record for " + getName() + "\n";
    for (Rental rental : rentals) {
        double amount = 0;
        switch (rental.getMovie().getPriceCode()) {
            case Movie.REGULAR:
                amount += 2;
                if (rental.getDaysRented() > 2)
                    amount += (rental.getDaysRented() - 2) * 1.5;
                break;
            case Movie.NEW_RELEASE:
                amount += rental.getDaysRented() * 3;
                break;
            case Movie.CHILDREN:
                amount += 1.5;
                if (rental.getDaysRented() > 3)
                    amount += (rental.getDaysRented() - 3) * 1.5;
                break;
        }

        // add frequent renter points
        frequentRenterPoints++;
        // add bonus for a two day new release rental
        if (rental.getMovie().getPriceCode() == Movie.NEW_RELEASE && rental.getDaysRented() > 1)
            frequentRenterPoints++;

        // show figures for this rental
        result += "\t" + rental.getMovie().getTitle() + "\t" + String.valueOf(amount) + "\n";

        totalAmount += amount;
    }

    result += "Amount owed is " + String.valueOf(totalAmount) + "\n";
    result += "You earned " + String.valueOf(frequentRenterPoints) + " frequent renter points";

    return result;
}

```

Calcula preço total

Customer (2)

```
public String statement() {
    double totalAmount = 0;
    int frequentRenterPoints = 0;
```

```
String result = "Rental record for " + getName() + "\n";
```

```
for (Rental rental : rentals) {
```

```
    double amount = 0;
```

```
    switch (rental.getMovie().getPriceCode()) {
```

```
        case Movie.REGULAR:
```

```
            amount += 2;
```

```
            if (rental.getDaysRented() > 2)
```

```
                amount += (rental.getDaysRented() - 2) * 1.5;
```

```
            break;
```

```
        case Movie.NEW_RELEASE:
```

```
            amount += rental.getDaysRented() * 3;
```

```
            break;
```

```
        case Movie.CHILDREN:
```

```
            amount += 1.5;
```

```
            if (rental.getDaysRented() > 3)
```

```
                amount += (rental.getDaysRented() - 3) * 1.5;
```

```
            break;
```

```
    }
```

```
    // add frequent renter points
```

```
    frequentRenterPoints++;
```

```
    // add bonus for a two day new release rental
```

```
    if (rental.getMovie().getPriceCode() == Movie.NEW_RELEASE && rental.getDaysRented() > 1)
```

```
        frequentRenterPoints++;
```

```
    // show figures for this rental
```

```
    result += "\t" + rental.getMovie().getTitle() + "\t" + String.valueOf(amount) + "\n";
```

```
    totalAmount += amount;
```

```
}
```

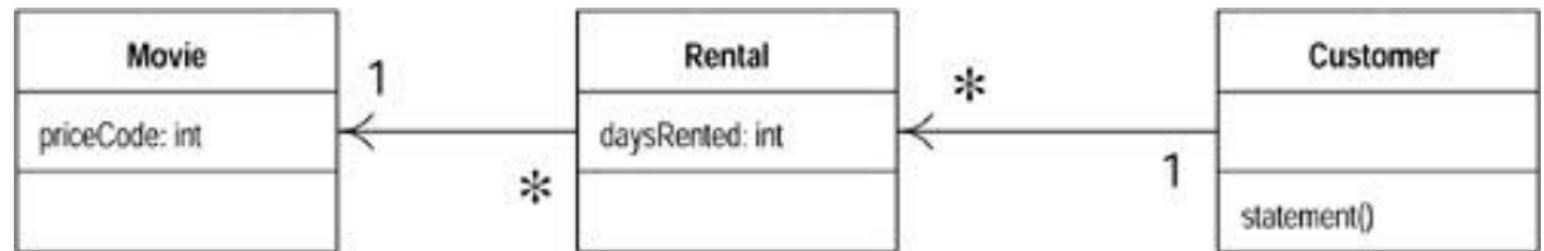
```
result += "Amount owed is " + String.valueOf(totalAmount) + "\n";
```

```
result += "You earned " + String.valueOf(frequentRenterPoints) + " frequent renter points";
```

```
return result;
```

```
}
```

```
}
```



**Contém
regras de negócio**

Customer (2)

```
public String statement() {
    double totalAmount = 0;
    int frequentRenterPoints = 0;
```

```
    String result = "Rental record for " + getName() + "\n";
    for (Rental rental : rentals) {
        double amount = 0;
        switch (rental.getMovie().getPriceCode()) {
            case Movie.REGULAR:
                amount += 2;
                if (rental.getDaysRented() > 2)
                    amount += (rental.getDaysRented() - 2) * 1.5;
                break;
            case Movie.NEW_RELEASE:
                amount += rental.getDaysRented() * 3;
                break;
            case Movie.CHILDREN:
                amount += 1.5;
                if (rental.getDaysRented() > 3)
                    amount += (rental.getDaysRented() - 3) * 1.5;
                break;
        }
    }
```

```
    // add frequent renter points
    frequentRenterPoints++;
    // add bonus for a two day new release rental
    if (rental.getMovie().getPriceCode() == Movie.NEW_RELEASE && rental.getDaysRented() > 1)
        frequentRenterPoints++;
```

```
    // show figures for this rental
    result += "\t" + rental.getMovie().getTitle() + "\t" + String.valueOf(amount) + "\n";
```

```
    totalAmount += amount;
```

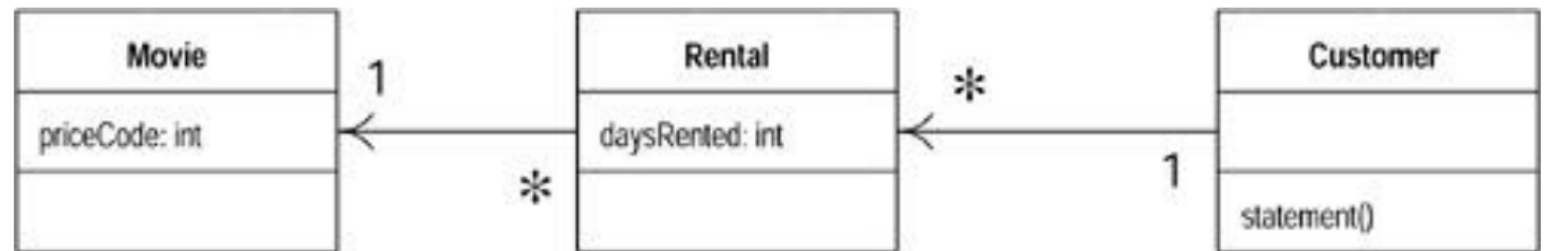
```
}
```

```
result += "Amount owed is " + String.valueOf(totalAmount) + "\n";
result += "You earned " + String.valueOf(frequentRenterPoints) + " frequent renter points";
```

```
return result;
```

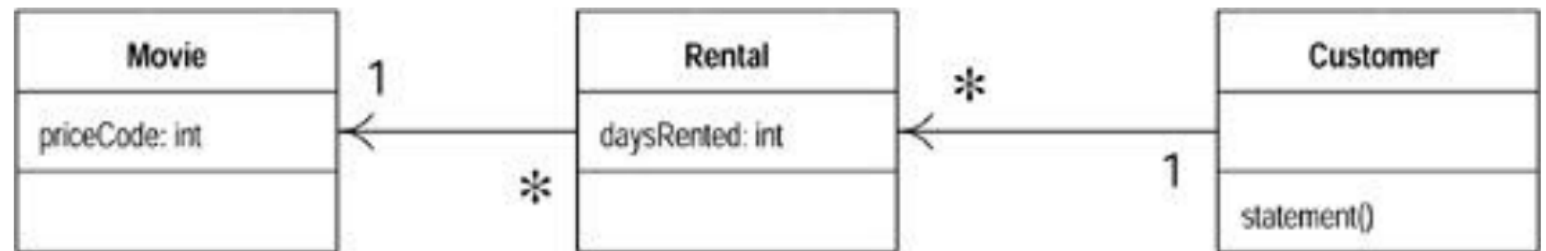
```
}
```

```
}
```



**Calcula
pontuação**

Customer (2)



```

public String statement() {
    double totalAmount = 0;
    int frequentRenterPoints = 0;

    String result = "Rental record for " + getName() + "\n";
    for (Rental rental : rentals) {
        double amount = 0;
        switch (rental.getMovie().getPriceCode()) {
            case Movie.REGULAR:
                amount += 2;
                if (rental.getDaysRented() > 2)
                    amount += (rental.getDaysRented() - 2) * 1.5;
                break;
            case Movie.NEW_RELEASE:
                amount += rental.getDaysRented() * 3;
                break;
            case Movie.CHILDREN:
                amount += 1.5;
                if (rental.getDaysRented() > 3)
                    amount += (rental.getDaysRented() - 3) * 1.5;
                break;
        }

        // add frequent renter points
        frequentRenterPoints++;
        // add bonus for a two day new release rental
        if (rental.getMovie().getPriceCode() == Movie.NEW_RELEASE && rental.getDaysRented() > 1)
            frequentRenterPoints++;

        // show figures for this rental
        result += "\t" + rental.getMovie().getTitle() + "\t" + String.valueOf(amount) + "\n";
        totalAmount += amount;
    }

    result += "Amount owed is " + String.valueOf(totalAmount) + "\n";
    result += "You earned " + String.valueOf(frequentRenterPoints) + " frequent renter points";

    return result;
}

```

Gera relatório

Customer (2)

```
public String statement() {
    double totalAmount = 0;
    int frequentRenterPoints = 0;

    String result = "Rental record for " + getName() + "\n";
    for (Rental rental : rentals) {
        double amount = 0;
        switch (rental.getMovie().getPriceCode()) {
            case Movie.REGULAR:
                amount += 2;
                if (rental.getDaysRented() > 2)
                    amount += (rental.getDaysRented() - 2) * 1.5;
                break;
            case Movie.NEW_RELEASE:
                amount += rental.getDaysRented() * 3;
                break;
            case Movie.CHILDREN:
                amount += 1.5;
                if (rental.getDaysRented() > 3)
                    amount += (rental.getDaysRented() - 3) * 1.5;
                break;
        }

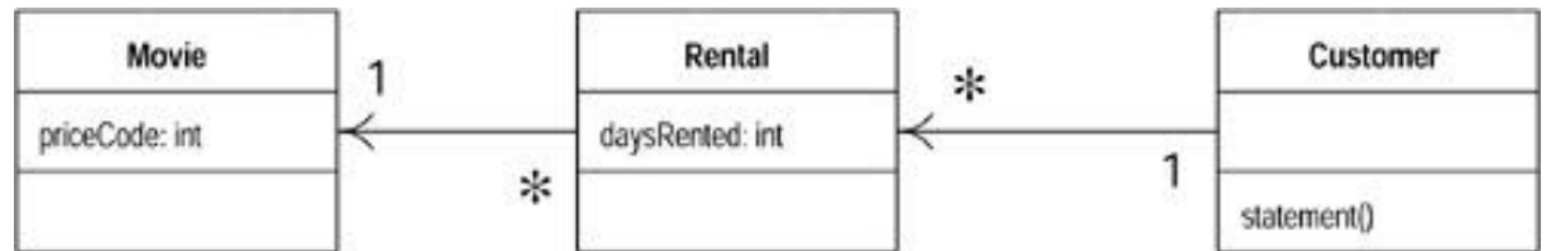
        // add frequent renter points
        frequentRenterPoints++;
        // add bonus for a two day new release rental
        if (rental.getMovie().getPriceCode() == Movie.NEW_RELEASE &&
            rental.getDaysRented() > 2)
            frequentRenterPoints++;

        // show figures for this rental
        result += "\t" + rental.getMovie().getTitle() + "\t" + String.valueOf(amount) + "\n";

        totalAmount += amount;
    }

    result += "Amount owed is " + String.valueOf(totalAmount) + "\n";
    result += "You earned " + String.valueOf(frequentRenterPoints) + " frequent renter points";

    return result;
}
```



- **Contém regras de negócio**
- **Calcula preço parcial**
- **Gera relatório**
- **Calcula pontuação**
- **Calcula preço total**

Possíveis problemas...

- Qual a sua opinião sobre esse código?

Possíveis problemas...

- Qual a sua opinião sobre esse código?
 - Código não é bem projetado
 - Código não é OO
 - Para um programa simples, não importa muito
 - Mas se esse código é parte de um sistema complexo, temos um problema
- Problema: o método `statement()` faz muita coisa
 - Essas tarefas devem ser feitas por outras classes

Possíveis problemas...

- Mesmo com todos esses problemas o programa funciona...

Possíveis problemas...

- Mesmo com todos esses problemas o programa funciona...
- ...até decidirmos **alterar** o código para adicionar uma nova funcionalidade, corrigir um bug, etc
- Sistemas com projetos fracos, são difíceis de alterar

Alteração 1:

Fatura em HTML

- Fatura deve ser impressa em HTML para que o sistema possa rodar na web
- Solução ruim: criar o método `htmlStatement()` a partir de `statement()`

Customer

```
public String htmlStatement() {
    double totalAmount = 0;
    int frequentRenterPoints = 0;
```

```
String result = "<H1>Rental record for " + getName() + "</H1>\n";
```

```
for (Rental rental : rentals) {
    double amount = 0;
    switch (rental.getMovie().getPriceCode()) {
        case Movie.REGULAR:
            amount += 2;
            if (rental.getDaysRented() > 2)
                amount += (rental.getDaysRented() - 2) * 1.5;
            break;
        case Movie.NEW_RELEASE:
            amount += rental.getDaysRented() * 3;
            break;
        case Movie.CHILDREN:
            amount += 1.5;
            if (rental.getDaysRented() > 3)
                amount += (rental.getDaysRented() - 3) * 1.5;
            break;
    }
```

```
// add frequent renter points
```

```
frequentRenterPoints++;
```

```
// add bonus for a two day new release rental
```

```
if (rental.getMovie().getPriceCode() == Movie.NEW_RELEASE && rental.getDaysRented() > 1)
    frequentRenterPoints++;
```

```
// show figures for this rental
```

```
result += "\t" + rental.getMovie().getTitle() + "\t" + String.valueOf(amount) + "<BR>\n";
```

```
totalAmount += amount;
```

```
}
```

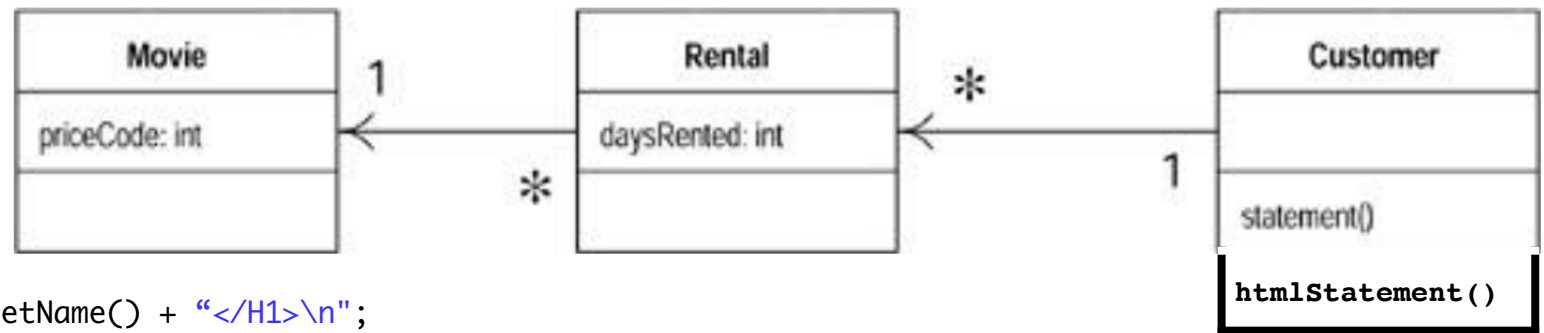
```
result += "<P>Amount owed is <EM>" + String.valueOf(totalAmount) + "</EM> "</P>\n";
```

```
result += "<P>You earned <EM>" + String.valueOf(frequentRenterPoints) + "</EM> frequent renter points</P>";
```

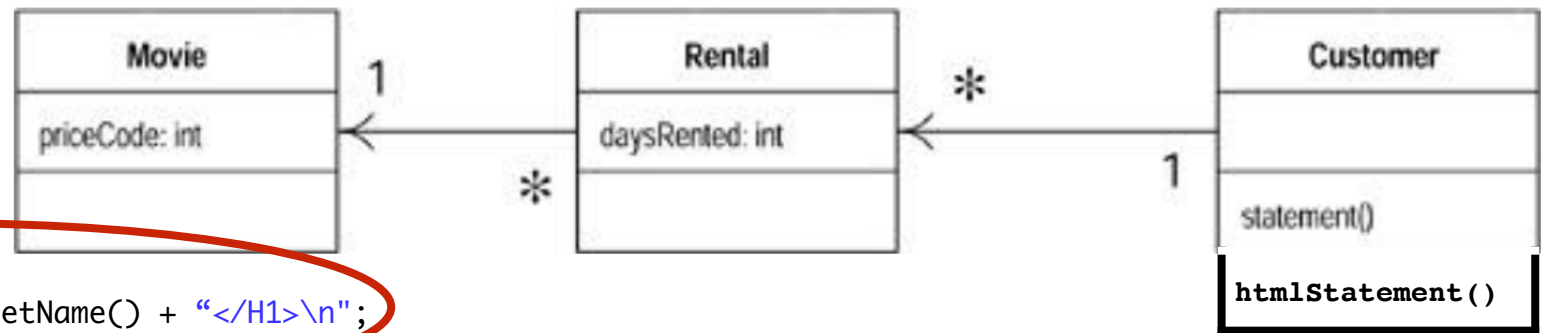
```
return result;
```

```
}
```

```
}
```



Customer



```

public String htmlStatement() {
    double totalAmount = 0;
    int frequentRenterPoints = 0;

    String result = "<H1>Rental record for " + getName() + "</H1>\n";
    for (Rental rental : rentals) {
        double amount = 0;
        switch (rental.getMovie().getPriceCode()) {
            case Movie.REGULAR:
                amount += 2;
                if (rental.getDaysRented() > 2)
                    amount += (rental.getDaysRented() - 2) * 1.5;
                break;
            case Movie.NEW_RELEASE:
                amount += rental.getDaysRented() * 3;
                break;
            case Movie.CHILDREN:
                amount += 1.5;
                if (rental.getDaysRented() > 3)
                    amount += (rental.getDaysRented() - 3) * 1.5;
                break;
        }

        // add frequent renter points
        frequentRenterPoints++;
        // add bonus for a two day new release rental
        if (rental.getMovie().getPriceCode() == Movie.NEW_RELEASE && rental.getDaysRented() > 1)
            frequentRenterPoints++;

        // show figures for this rental
        result += "\t" + rental.getMovie().getTitle() + "\t" + String.valueOf(amount) + "<BR>\n";

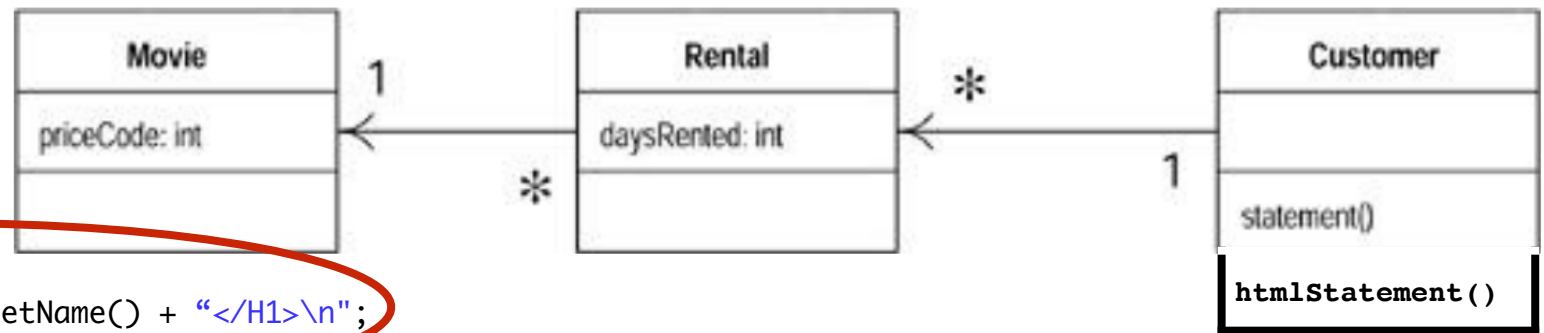
        totalAmount += amount;
    }

    result += "<P>Amount owed is <EM>" + String.valueOf(totalAmount) + "</EM> </P>\n";
    result += "<P>You earned <EM>" + String.valueOf(frequentRenterPoints) + "</EM> frequent renter points</P>";

    return result;
}

```

Customer



```

public String htmlStatement() {
    double totalAmount = 0;
    int frequentRenterPoints = 0;

    String result = "<H1>Rental record for " + getName() + "</H1>\n";
    for (Rental rental : rentals) {
        double amount = 0;
        switch (rental.getMovie().getPriceCode()) {
            case Movie.REGULAR:
                amount += 2;
                if (rental.getDaysRented() > 2)
                    amount += (rental.getDaysRented() - 2) * 5;
                break;
            case Movie.NEW_RELEASE:
                amount += rental.getDaysRented() * 3;
                break;
            case Movie.CHILDREN:
                amount += 1.5;
                if (rental.getDaysRented() > 3)
                    amount += (rental.getDaysRented() - 3) * 1.5;
                break;
        }

        // add frequent renter points
        frequentRenterPoints++;
        // add bonus for a two day new release rental
        if (rental.getMovie().getPriceCode() == Movie.NEW_RELEASE && rental.getDaysRented() > 1)
            frequentRenterPoints++;

        // show figures for this rental
        result += "\t" + rental.getMovie().getTitle() + "\t" + String.valueOf(amount) + "<BR>\n";

        totalAmount += amount;
    }

    result += "<P>Amount owed is <EM>" + String.valueOf(totalAmount) + "</EM> </P>\n";
    result += "<P>You earned <EM>" + String.valueOf(frequentRenterPoints) + "</EM> frequent renter points</P>";

    return result;
}

```


- O que ocorre quando as regras de cobrança são alteradas?
- Temos que alterar os métodos `statement()` e `htmlStatement()`
- O problema de copiar e colar código vem quando precisamos alterar o código

Alteração 2:

Novos Tipos de Filmes

- Os tipos dos filmes podem mudar
 - Tipos hoje: Regular, Children e New Releases
 - Tipos amanhã: ?
- Importante: essa alteração afeta diretamente a forma de cobrança e o cálculo das pontuações

Refatoração é Fundamental

- Quando é preciso adicionar nova funcionalidade mas o código não está estruturado:
 - **Primeiro**: refatorar o código para que seja mais fácil adicionar a nova funcionalidade
 - **Depois**: adicionar a nova funcionalidade
- OBS: antes que iniciar qualquer refatoração, é importante ter um **conjunto sólido de testes**

Refatorando

- Principal problema: tamanho de `statement ()`
 - Códigos menores são mais fáceis de entender e gerenciar
- Primeiro passo: quebrar métodos longos e mover as partes para classes melhores
 - Usar as refatorações: Extract Method e Move Method

Customer.statement() - extraindo cálculo da locação

Customer

```
public String statement() {
    double totalAmount = 0;
    int frequentRenterPoints = 0;

    String result = "Rental record for " + getName() + "\n";
    for (Rental rental : rentals) {
        double amount = 0;
        switch (rental.getMovie().getPriceCode()) {
            case Movie.REGULAR:
                amount += 2;
                if (rental.getDaysRented() > 2)
                    amount += (rental.getDaysRented() - 2) * 1.5;
                break;
            case Movie.NEW_RELEASE:
                amount += rental.getDaysRented() * 3;
                break;
            case Movie.CHILDREN:
                amount += 1.5;
                if (rental.getDaysRented() > 3)
                    amount += (rental.getDaysRented() - 3) * 1.5;
                break;
        }

        // add frequent renter points
        frequentRenterPoints++;
        // add bonus for a two day new release rental
        if (rental.getMovie().getPriceCode() == Movie.NEW_RELEASE &&
            rental.getDaysRented() > 1) frequentRenterPoints++;

        // show figures for this rental
        result += "\t" + rental.getMovie().getTitle() + "\t" +
            String.valueOf(amount) + "\n";
        totalAmount += amount;
    }

    result += "Amount owed is " + String.valueOf(totalAmount) + "\n";
    result += "You earned " + String.valueOf(frequentRenterPoints) +
        " frequent renter points";
    return result;
}
```

Customer.statement() - extraindo cálculo da locação

Customer

```
public String statement() {
    double totalAmount = 0;
    int frequentRenterPoints = 0;

    String result = "Rental record for " + getName() + "\n";
    for (Rental rental : rentals) {
        double amount = 0;
        switch (rental.getMovie().getPriceCode()) {
            case Movie.REGULAR:
                amount += 2;
                if (rental.getDaysRented() > 2)
                    amount += (rental.getDaysRented() - 2) * 1.5;
                break;
            case Movie.NEW_RELEASE:
                amount += rental.getDaysRented() * 3;
                break;
            case Movie.CHILDREN:
                amount += 1.5;
                if (rental.getDaysRented() > 3)
                    amount += (rental.getDaysRented() - 3) * 1.5;
                break;
        }

        // add frequent renter points
        frequentRenterPoints++;
        // add bonus for a two day new release rental
        if (rental.getMovie().getPriceCode() == Movie.NEW_RELEASE &&
            rental.getDaysRented() > 1) frequentRenterPoints++;

        // show figures for this rental
        result += "\t" + rental.getMovie().getTitle() + "\t" +
            String.valueOf(amount) + "\n";
        totalAmount += amount;
    }

    result += "Amount owed is " + String.valueOf(totalAmount) + "\n";
    result += "You earned " + String.valueOf(frequentRenterPoints) +
        " frequent renter points";
    return result;
}
```

Customer

```
public String statement() {
    double totalAmount = 0;
    int frequentRenterPoints = 0;

    String result = "Rental record for " + getName() + "\n";
    for (Rental rental : rentals) {
        double amount = 0;

        amount = getCharge(rental);

        // add frequent renter points
        frequentRenterPoints++;
        // add bonus for a two day new release rental
        if (rental.getMovie().getPriceCode() == Movie.NEW_RELEASE &&
            rental.getDaysRented() > 1) frequentRenterPoints++;

        // show figures for this rental
        result += "\t" + rental.getMovie().getTitle() + "\t" +
            String.valueOf(amount) + "\n";
        totalAmount += amount;
    }

    result += "Amount owed is " + String.valueOf(totalAmount) + "\n";
    result += "You earned " + String.valueOf(frequentRenterPoints) +
        " frequent renter points";
    return result;
}

private double getCharge(Rental rental) {
    double amount = 0;
    switch (rental.getMovie().getPriceCode()) {
        case Movie.REGULAR:
            amount += 2;
            if (rental.getDaysRented() > 2)
                amount += (rental.getDaysRented() - 2) * 1.5;
            break;
        case Movie.NEW_RELEASE:
            amount += rental.getDaysRented() * 3;
            break;
        case Movie.CHILDREN:
            amount += 1.5;
            if (rental.getDaysRented() > 3)
                amount += (rental.getDaysRented() - 3) * 1.5;
            break;
    }
    return amount;
}
```

Customer.statement() - extraindo cálculo da locação

Customer

```
public String statement() {
    double totalAmount = 0;
    int frequentRenterPoints = 0;

    String result = "Rental record for " + getName() + "\n";
    for (Rental rental : rentals) {
        double amount = 0;
        switch (rental.getMovie().getPriceCode()) {
            case Movie.REGULAR:
                amount += 2;
                if (rental.getDaysRented() > 2)
                    amount += (rental.getDaysRented() - 2) * 1.5;
                break;
            case Movie.NEW_RELEASE:
                amount += rental.getDaysRented() * 3;
                break;
            case Movie.CHILDREN:
                amount += 1.5;
                if (rental.getDaysRented() > 3)
                    amount += (rental.getDaysRented() - 3) * 1.5;
                break;
        }

        // add frequent renter points
        frequentRenterPoints++;
        // add bonus for a two day new release rental
        if (rental.getMovie().getPriceCode() == Movie.NEW_RELEASE &&
            rental.getDaysRented() > 1) frequentRenterPoints++;

        // show figures for this rental
        result += "\t" + rental.getMovie().getTitle() + "\t" +
            String.valueOf(amount) + "\n";
        totalAmount += amount;
    }

    result += "Amount owed is " + String.valueOf(totalAmount) + "\n";
    result += "You earned " + String.valueOf(frequentRenterPoints) +
        " frequent renter points";
    return result;
}
```

Customer

```
public String statement() {
    double totalAmount = 0;
    int frequentRenterPoints = 0;

    String result = "Rental record for " + getName() + "\n";
    for (Rental rental : rentals) {
        double amount = 0;

        amount = getCharge(rental);

        // add frequent renter points
        frequentRenterPoints++;
        // add bonus for a two day new release rental
        if (rental.getMovie().getPriceCode() == Movie.NEW_RELEASE &&
            rental.getDaysRented() > 1) frequentRenterPoints++;

        // show figures for this rental
        result += "\t" + rental.getMovie().getTitle() + "\t" +
            String.valueOf(amount) + "\n";
        totalAmount += amount;
    }

    result += "Amount owed is " + String.valueOf(totalAmount) + "\n";
    result += "You earned " + String.valueOf(frequentRenterPoints) +
        " frequent renter points";
    return result;
}

private double getCharge(Rental rental) {
    double amount = 0;
    switch (rental.getMovie().getPriceCode()) {
        case Movie.REGULAR:
            amount += 2;
            if (rental.getDaysRented() > 2)
                amount += (rental.getDaysRented() - 2) * 1.5;
            break;
        case Movie.NEW_RELEASE:
            amount += rental.getDaysRented() * 3;
            break;
        case Movie.CHILDREN:
            amount += 1.5;
            if (rental.getDaysRented() > 3)
                amount += (rental.getDaysRented() - 3) * 1.5;
            break;
    }
    return amount;
}
```

- Refatoração deve ser feita em pequenos passos
- Se erros forem cometidos, fica mais fácil de encontrar o problema
- Sempre rodar os testes após refatorar

Movendo getCharge() de Customer para Rental

Customer

```
private double getCharge(Rental rental) {  
    double amount = 0;  
    switch (rental.getMovie().getPriceCode()) {  
        case Movie.REGULAR:  
            amount += 2;  
            if (rental.getDaysRented() > 2)  
                amount += (rental.getDaysRented() - 2) * 1.5;  
            break;  
        case Movie.NEW_RELEASE:  
            amount += rental.getDaysRented() * 3;  
            break;  
        case Movie.CHILDREN:  
            amount += 1.5;  
            if (rental.getDaysRented() > 3)  
                amount += (rental.getDaysRented() - 3) * 1.5;  
            break;  
    }  
    return amount;  
}
```

Rental

```
public double getCharge() {  
    double amount = 0;  
    switch (getMovie().getPriceCode()) {  
        case Movie.REGULAR:  
            amount += 2;  
            if (getDaysRented() > 2)  
                amount += (getDaysRented() - 2) * 1.5;  
            break;  
        case Movie.NEW_RELEASE:  
            amount += getDaysRented() * 3;  
            break;  
        case Movie.CHILDREN:  
            amount += 1.5;  
            if (getDaysRented() > 3)  
                amount += (getDaysRented() - 3) * 1.5;  
            break;  
    }  
    return amount;  
}
```

Movendo getCharge() de Customer para Rental

Customer

```
private double getCharge(Rental rental) {  
    double amount = 0;  
    switch (rental.getMovie().getPriceCode()) {  
        case Movie.REGULAR:  
            amount += 2;  
            if (rental.getDaysRented() > 2)  
                amount += (rental.getDaysRented() - 2) * 1.5;  
            break;  
        case Movie.NEW_RELEASE:  
            amount += rental.getDaysRented() * 3;  
            break;  
        case Movie.CHILDREN:  
            amount += 1.5;  
            if (rental.getDaysRented() > 3)  
                amount += (rental.getDaysRented() - 3) * 1.5;  
            break;  
    }  
    return amount;  
}
```

Rental

```
public double getCharge() {  
    double amount = 0;  
    switch (getMovie().getPriceCode()) {  
        case Movie.REGULAR:  
            amount += 2;  
            if (getDaysRented() > 2)  
                amount += (getDaysRented() - 2) * 1.5;  
            break;  
        case Movie.NEW_RELEASE:  
            amount += getDaysRented() * 3;  
            break;  
        case Movie.CHILDREN:  
            amount += 1.5;  
            if (getDaysRented() > 3)  
                amount += (getDaysRented() - 3) * 1.5;  
            break;  
    }  
    return amount;  
}
```

Atualizando Customer.statement()

Customer

```
public String statement() {
    double totalAmount = 0;
    int frequentRenterPoints = 0;

    String result = "Rental record for " + getName() + "\n";
    for (Rental rental : rentals) {

        double amount = getCharge(rental);

        // add frequent renter points
        frequentRenterPoints++;
        // add bonus for a two day new release rental
        if (rental.getMovie().getPriceCode() == Movie.NEW_RELEASE &&
            rental.getDaysRented() > 1) frequentRenterPoints++;

        // show figures for this rental
        result += "\t" + rental.getMovie().getTitle() + "\t" +
            String.valueOf(amount) + "\n";
        totalAmount += amount;
    }

    result += "Amount owed is " + String.valueOf(totalAmount) + "\n";
    result += "You earned " + String.valueOf(frequentRenterPoints) +
        " frequent renter points";
    return result;
}
```

Customer

```
public String statement() {
    double totalAmount = 0;
    int frequentRenterPoints = 0;

    String result = "Rental record for " + getName() + "\n";
    for (Rental rental : rentals) {

        double amount = rental.getCharge();

        // add frequent renter points
        frequentRenterPoints++;
        // add bonus for a two day new release rental
        if (rental.getMovie().getPriceCode() == Movie.NEW_RELEASE &&
            rental.getDaysRented() > 1) frequentRenterPoints++;

        // show figures for this rental
        result += "\t" + rental.getMovie().getTitle() + "\t" +
            String.valueOf(amount) + "\n";
        totalAmount += amount;
    }

    result += "Amount owed is " + String.valueOf(totalAmount) + "\n";
    result += "You earned " + String.valueOf(frequentRenterPoints) +
        " frequent renter points";
    return result;
}
```

Atualizando Customer.statement()

Customer

```
public String statement() {
    double totalAmount = 0;
    int frequentRenterPoints = 0;

    String result = "Rental record for " + getName() + "\n";
    for (Rental rental : rentals) {

        double amount = getCharge(rental);

        // add frequent renter points
        frequentRenterPoints++;
        // add bonus for a two day new release rental
        if (rental.getMovie().getPriceCode() == Movie.NEW_RELEASE &&
            rental.getDaysRented() > 1) frequentRenterPoints++;

        // show figures for this rental
        result += "\t" + rental.getMovie().getTitle() + "\t" +
            String.valueOf(amount) + "\n";
        totalAmount += amount;
    }

    result += "Amount owed is " + String.valueOf(totalAmount) + "\n";
    result += "You earned " + String.valueOf(frequentRenterPoints) +
        " frequent renter points";
    return result;
}
```

Customer

```
public String statement() {
    double totalAmount = 0;
    int frequentRenterPoints = 0;

    String result = "Rental record for " + getName() + "\n";
    for (Rental rental : rentals) {

        double amount = rental.getCharge();

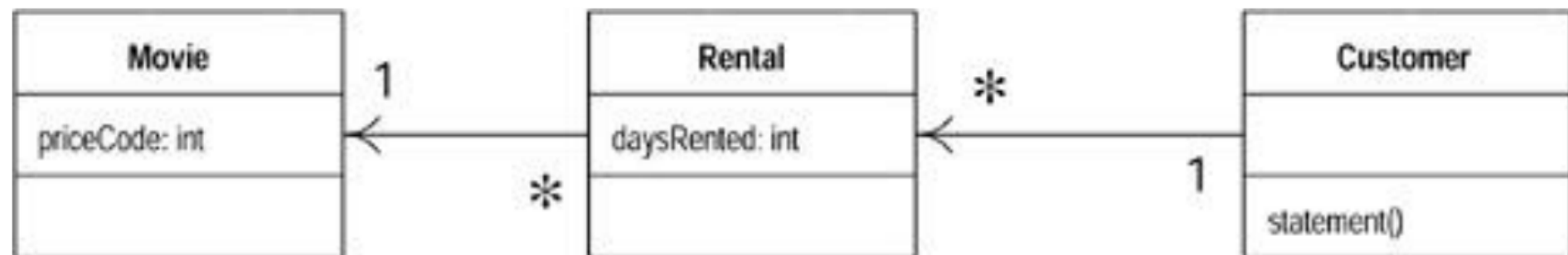
        // add frequent renter points
        frequentRenterPoints++;
        // add bonus for a two day new release rental
        if (rental.getMovie().getPriceCode() == Movie.NEW_RELEASE &&
            rental.getDaysRented() > 1) frequentRenterPoints++;

        // show figures for this rental
        result += "\t" + rental.getMovie().getTitle() + "\t" +
            String.valueOf(amount) + "\n";
        totalAmount += amount;
    }

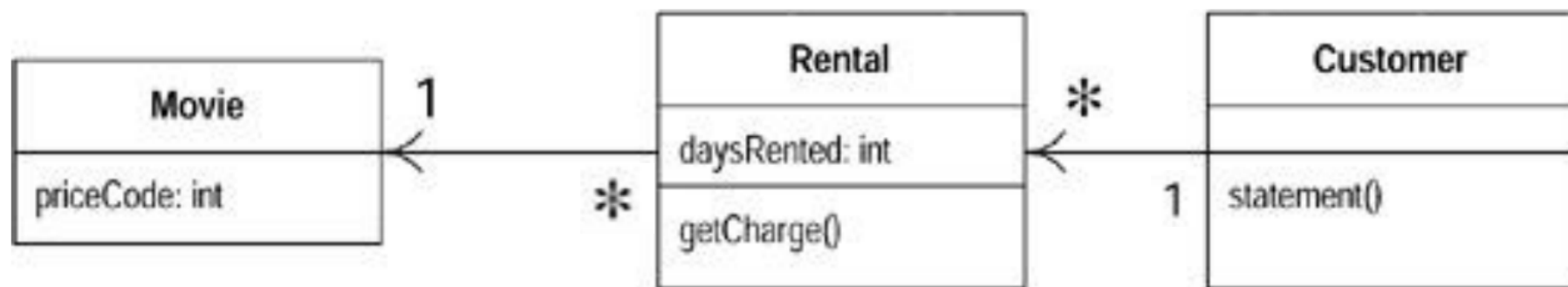
    result += "Amount owed is " + String.valueOf(totalAmount) + "\n";
    result += "You earned " + String.valueOf(frequentRenterPoints) +
        " frequent renter points";
    return result;
}
```

Diagramas de Classes

antes



depois



Customer.statement() - extraindo cálculo da pontuação

Customer

```
public String statement() {
    double totalAmount = 0;
    int frequentRenterPoints = 0;

    String result = "Rental record for " + getName() + "\n";
    for (Rental rental : rentals) {

        double amount = rental.getCharge();

        // add frequent renter points
        frequentRenterPoints++;
        // add bonus for a two day new release rental
        if (rental.getMovie().getPriceCode() == Movie.NEW_RELEASE &&
            rental.getDaysRented() > 1) frequentRenterPoints++;

        // show figures for this rental
        result += "\t" + rental.getMovie().getTitle() + "\t" +
            String.valueOf(amount) + "\n";
        totalAmount += amount;
    }

    result += "Amount owed is " + String.valueOf(totalAmount) + "\n";
    result += "You earned " + String.valueOf(frequentRenterPoints) +
        " frequent renter points";
    return result;
}
```

Customer

Rental

Customer.statement() - extraindo cálculo da pontuação

Customer

```
public String statement() {
    double totalAmount = 0;
    int frequentRenterPoints = 0;

    String result = "Rental record for " + getName() + "\n";
    for (Rental rental : rentals) {

        double amount = rental.getCharge();

        // add frequent renter points
        frequentRenterPoints++;
        // add bonus for a two day new release rental
        if (rental.getMovie().getPriceCode() == Movie.NEW_RELEASE &&
            rental.getDaysRented() > 1) frequentRenterPoints++;

        // show figures for this rental
        result += "\t" + rental.getMovie().getTitle() + "\t" +
            String.valueOf(amount) + "\n";
        totalAmount += amount;
    }

    result += "Amount owed is " + String.valueOf(totalAmount) + "\n";
    result += "You earned " + String.valueOf(frequentRenterPoints) +
        " frequent renter points";
    return result;
}
```

Customer

```
public String statement() {
    double totalAmount = 0;
    int frequentRenterPoints = 0;

    String result = "Rental record for " + getName() + "\n";
    for (Rental rental : rentals) {

        double amount = rental.getCharge();

        frequentRenterPoints += rental.getFrequentRenterPoints();

        // show figures for this rental
        result += "\t" + rental.getMovie().getTitle() + "\t" +
            String.valueOf(amount) + "\n";
        totalAmount += amount;
    }

    result += "Amount owed is " + String.valueOf(totalAmount) + "\n";
    result += "You earned " + String.valueOf(frequentRenterPoints) +
        " frequent renter points";
    return result;
}
```

Rental

```
public int getFrequentRenterPoints() {
    if (getMovie().getPriceCode() == Movie.NEW_RELEASE && getDaysRented() > 1)
        return 2;
    return 1;
}
```

Customer.statement() - extraindo cálculo da pontuação

Customer

```
public String statement() {
    double totalAmount = 0;
    int frequentRenterPoints = 0;

    String result = "Rental record for " + getName() + "\n";
    for (Rental rental : rentals) {

        double amount = rental.getCharge();

        // add frequent renter points
        frequentRenterPoints++;
        // add bonus for a two day new release rental
        if (rental.getMovie().getPriceCode() == Movie.NEW_RELEASE &&
            rental.getDaysRented() > 1) frequentRenterPoints++;

        // show figures for this rental
        result += "\t" + rental.getMovie().getTitle() + "\t" +
            String.valueOf(amount) + "\n";
        totalAmount += amount;
    }

    result += "Amount owed is " + String.valueOf(totalAmount) + "\n";
    result += "You earned " + String.valueOf(frequentRenterPoints) +
        " frequent renter points";
    return result;
}
```

Customer

```
public String statement() {
    double totalAmount = 0;
    int frequentRenterPoints = 0;

    String result = "Rental record for " + getName() + "\n";
    for (Rental rental : rentals) {

        double amount = rental.getCharge();

        frequentRenterPoints += rental.getFrequentRenterPoints();

        // show figures for this rental
        result += "\t" + rental.getMovie().getTitle() + "\t" +
            String.valueOf(amount) + "\n";
        totalAmount += amount;
    }

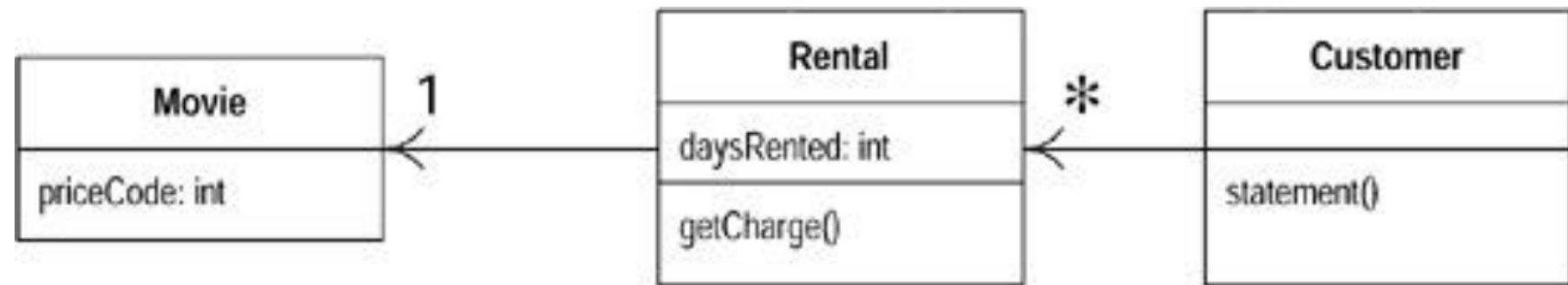
    result += "Amount owed is " + String.valueOf(totalAmount) + "\n";
    result += "You earned " + String.valueOf(frequentRenterPoints) +
        " frequent renter points";
    return result;
}
```

Rental

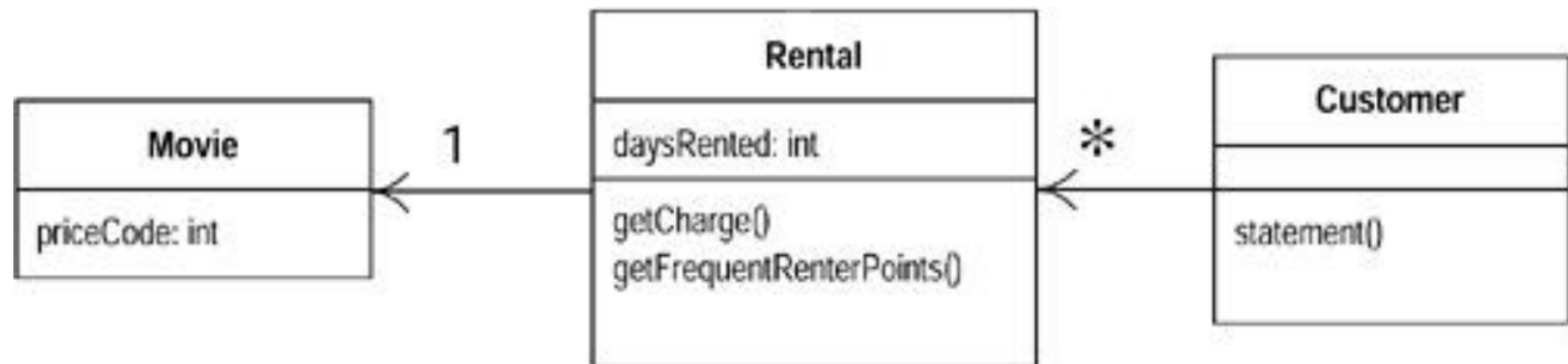
```
public int getFrequentRenterPoints() {
    if (getMovie().getPriceCode() == Movie.NEW_RELEASE && getDaysRented() > 1)
        return 2;
    return 1;
}
```


Diagramas de Classes

antes



depois



Mais duas refatorações em statement()

- Extraíndo o total da locação
- Extraíndo o total de pontuação
- Refatoração: Replace Temp with Query

Customer.statement() - extraindo o total da locação

Customer

```
public String statement() {
    double totalAmount = 0;
    int frequentRenterPoints = 0;

    String result = "Rental record for " + getName() + "\n";
    for (Rental rental : rentals) {

        double amount = rental.getCharge();

        frequentRenterPoints += rental.getFrequentRenterPoints();

        // show figures for this rental
        result += "\t" + rental.getMovie().getTitle() + "\t" +
            String.valueOf(amount) + "\n";
        totalAmount += amount;
    }

    result += "Amount owed is " + String.valueOf(totalAmount) + "\n";
    result += "You earned " + String.valueOf(frequentRenterPoints) +
        " frequent renter points";
    return result;
}
```

Customer

```
public String statement() {

    int frequentRenterPoints = 0;

    String result = "Rental record for " + getName() + "\n";
    for (Rental rental : rentals) {

        double amount = rental.getCharge();

        frequentRenterPoints += rental.getFrequentRenterPoints();

        // show figures for this rental
        result += "\t" + rental.getMovie().getTitle() + "\t" +
            String.valueOf(amount) + "\n";
    }

    result += "Amount owed is " + String.valueOf(getTotalCharge()) + "\n";
    result += "You earned " + String.valueOf(frequentRenterPoints) +
        " frequent renter points";
    return result;
}

private double getTotalCharge() {
    double total = 0;
    for (Rental rental : rentals)
        total += rental.getCharge();
    return total;
}
```

Customer.statement() - extraindo o total da locação

Customer

```
public String statement() {  
    double totalAmount = 0;  
    int frequentRenterPoints = 0;  
  
    String result = "Rental record for " + getName() + "\n";  
    for (Rental rental : rentals) {  
  
        double amount = rental.getCharge();  
  
        frequentRenterPoints += rental.getFrequentRenterPoints();  
  
        // show figures for this rental  
        result += "\t" + rental.getMovie().getTitle() + "\t" +  
            String.valueOf(amount) + "\n";  
        totalAmount += amount;  
    }  
  
    result += "Amount owed is " + String.valueOf(totalAmount) + "\n";  
    result += "You earned " + String.valueOf(frequentRenterPoints) +  
        " frequent renter points";  
    return result;  
}
```

Customer

```
public String statement() {  
  
    int frequentRenterPoints = 0;  
  
    String result = "Rental record for " + getName() + "\n";  
    for (Rental rental : rentals) {  
  
        double amount = rental.getCharge();  
  
        frequentRenterPoints += rental.getFrequentRenterPoints();  
  
        // show figures for this rental  
        result += "\t" + rental.getMovie().getTitle() + "\t" +  
            String.valueOf(amount) + "\n";  
    }  
  
    result += "Amount owed is " + String.valueOf(getTotalCharge()) + "\n";  
    result += "You earned " + String.valueOf(frequentRenterPoints) +  
        " frequent renter points";  
    return result;  
}
```

```
private double getTotalCharge() {  
    double total = 0;  
    for (Rental rental : rentals)  
        total += rental.getCharge();  
    return total;  
}
```

Customer.statement() - extraindo o total da locação

Customer

```
public String statement() {  
    double totalAmount = 0;  
    int frequentRenterPoints = 0;  
  
    String result = "Rental record for " + getName() + "\n";  
    for (Rental rental : rentals) {  
  
        double amount = rental.getCharge();  
  
        frequentRenterPoints += rental.getFrequentRenterPoints();  
  
        // show figures for this rental  
        result += "\t" + rental.getMovie().getTitle() + "\t" +  
            String.valueOf(amount) + "\n";  
        totalAmount += amount;  
    }  
  
    result += "Amount owed is " + String.valueOf(totalAmount) + "\n";  
    result += "You earned " + String.valueOf(frequentRenterPoints) +  
        " frequent renter points";  
    return result;  
}
```

**O programa perdeu em desempenho,
porém ganhou em manutenibilidade,
legibilidade e testabilidade**

Customer

```
public String statement() {  
  
    int frequentRenterPoints = 0;  
  
    String result = "Rental record for " + getName() + "\n";  
    for (Rental rental : rentals) {  
  
        double amount = rental.getCharge();  
  
        frequentRenterPoints += rental.getFrequentRenterPoints();  
  
        // show figures for this rental  
        result += "\t" + rental.getMovie().getTitle() + "\t" +  
            String.valueOf(amount) + "\n";  
    }  
  
    result += "Amount owed is " + String.valueOf(getTotalCharge()) + "\n";  
    result += "You earned " + String.valueOf(frequentRenterPoints) +  
        " frequent renter points";  
    return result;  
}  
  
private double getTotalCharge() {  
    double total = 0;  
    for (Rental rental : rentals)  
        total += rental.getCharge();  
    return total;  
}
```

Customer.statement() - extraíndo o total de pontuação

Customer

```
public String statement() {
    int frequentRenterPoints = 0;

    String result = "Rental record for " + getName() + "\n";
    for (Rental rental : rentals) {

        double amount = rental.getCharge();

        frequentRenterPoints += rental.getFrequentRenterPoints();

        // show figures for this rental
        result += "\t" + rental.getMovie().getTitle() + "\t" +
            String.valueOf(amount) + "\n";
    }

    result += "Amount owed is " + String.valueOf(getTotalCharge()) + "\n";
    result += "You earned " + String.valueOf(frequentRenterPoints) +
        " frequent renter points";
    return result;
}
```

Customer

```
public String statement() {

    String result = "Rental record for " + getName() + "\n";
    for (Rental rental : rentals) {

        double amount = rental.getCharge();

        // show figures for this rental
        result += "\t" + rental.getMovie().getTitle() + "\t" +
            String.valueOf(amount) + "\n";
    }

    result += "Amount owed is " + String.valueOf(getTotalCharge()) + "\n";
    result += "You earned " + String.valueOf(getTotalFrequentRenterPoints()) +
        " frequent renter points";
    return result;
}

private int getTotalFrequentRenterPoints() {
    int total = 0;
    for (Rental rental : rentals)
        total += rental.getFrequentRenterPoints();
    return total;
}
```

Customer.statement() - extraíndo o total de pontuação

Customer

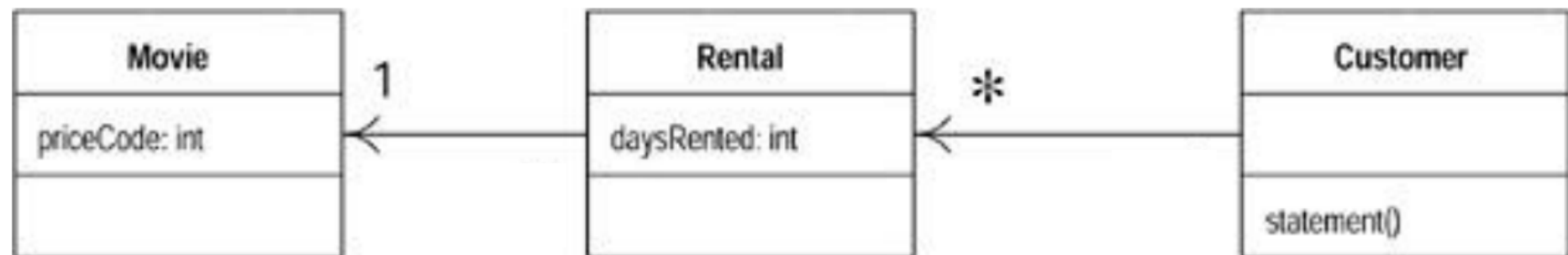
```
public String statement() {  
    int frequentRenterPoints = 0;  
  
    String result = "Rental record for " + getName() + "\n";  
    for (Rental rental : rentals) {  
  
        double amount = rental.getCharge();  
  
        frequentRenterPoints += rental.getFrequentRenterPoints();  
  
        // show figures for this rental  
        result += "\t" + rental.getMovie().getTitle() + "\t" +  
            String.valueOf(amount) + "\n";  
    }  
  
    result += "Amount owed is " + String.valueOf(getTotalCharge()) + "\n";  
    result += "You earned " + String.valueOf(frequentRenterPoints) +  
        " frequent renter points";  
    return result;  
}
```

Customer

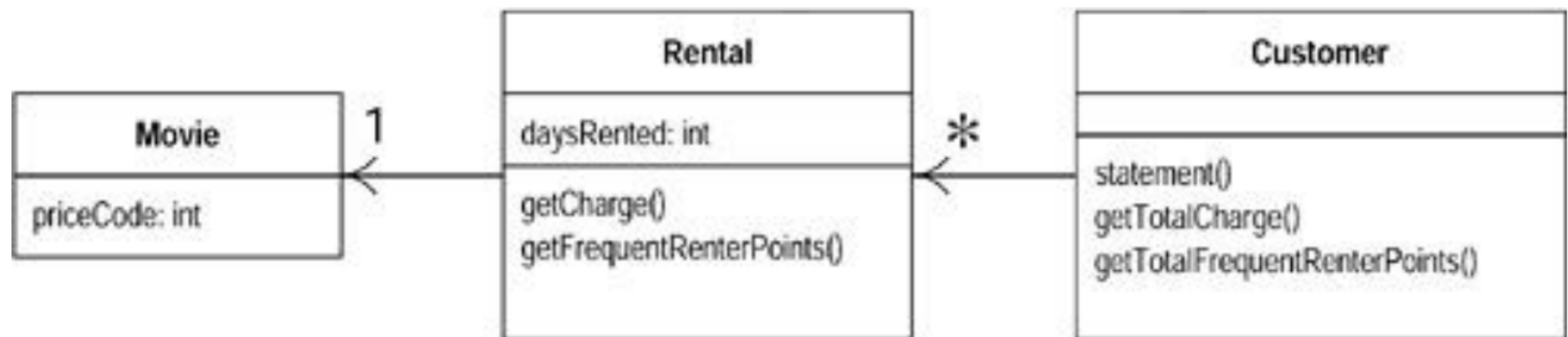
```
public String statement() {  
  
    String result = "Rental record for " + getName() + "\n";  
    for (Rental rental : rentals) {  
  
        double amount = rental.getCharge();  
  
        // show figures for this rental  
        result += "\t" + rental.getMovie().getTitle() + "\t" +  
            String.valueOf(amount) + "\n";  
    }  
  
    result += "Amount owed is " + String.valueOf(getTotalCharge()) + "\n";  
    result += "You earned " + String.valueOf(getTotalFrequentRenterPoints()) +  
        " frequent renter points";  
    return result;  
}  
  
private int getTotalFrequentRenterPoints() {  
    int total = 0;  
    for (Rental rental : rentals)  
        total += rental.getFrequentRenterPoints();  
    return total;  
}
```

Diagramas de Classes

primeiro

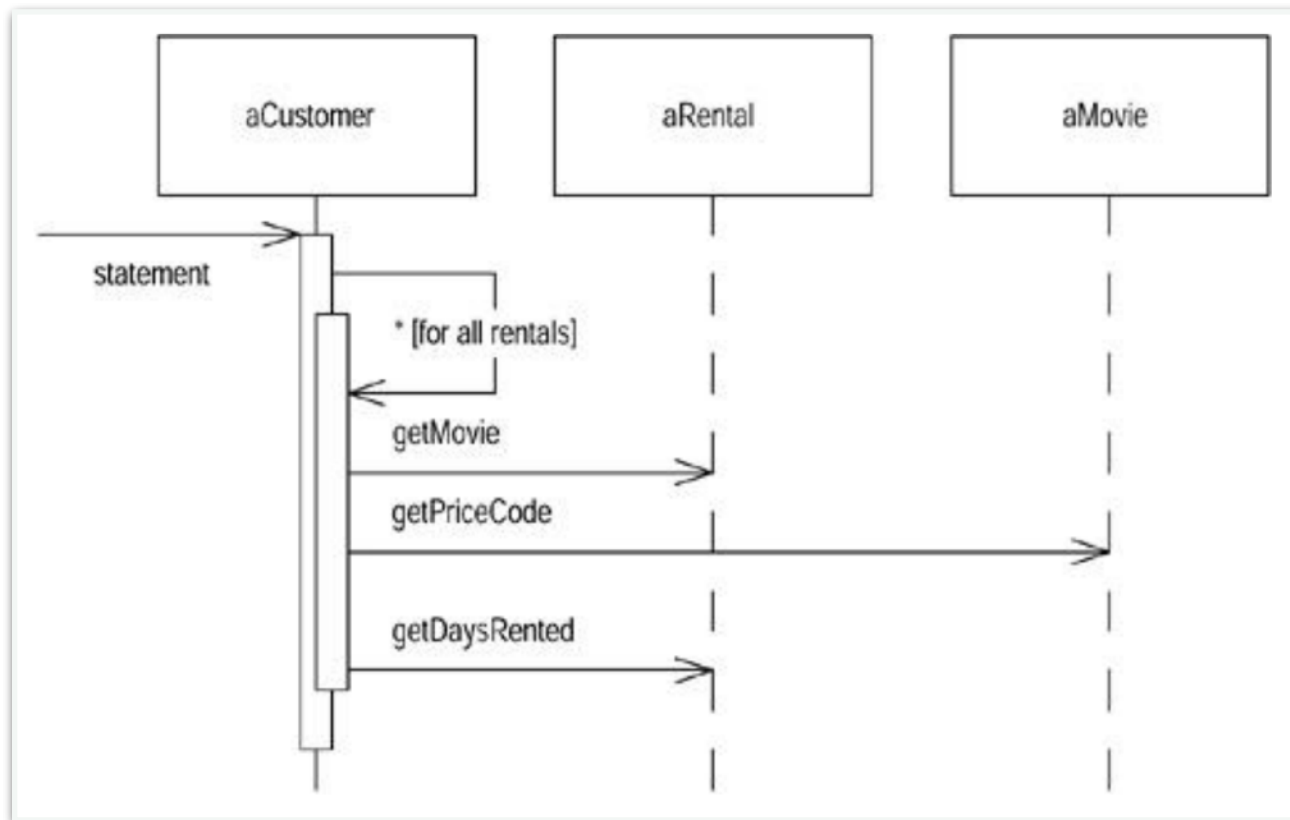


último

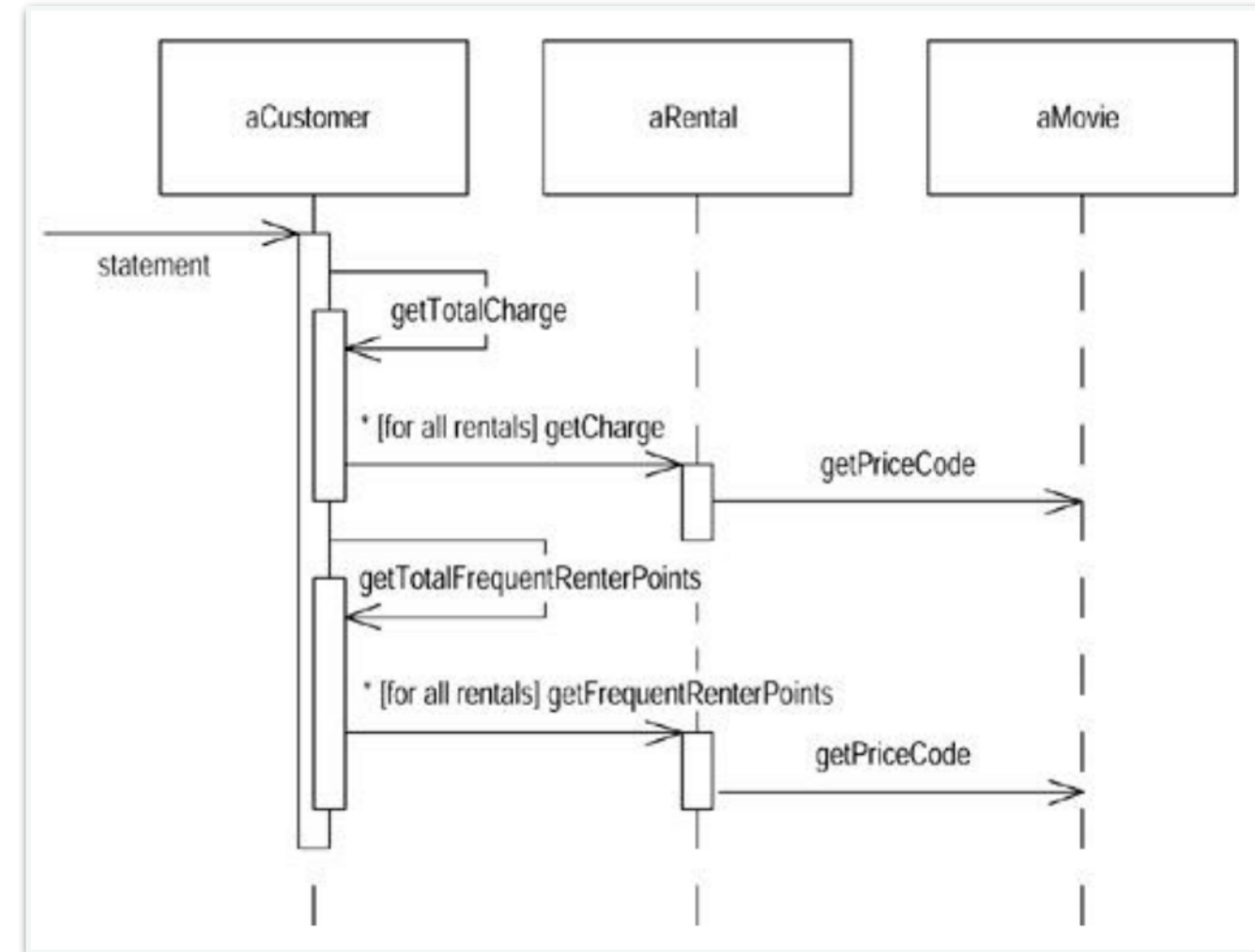


Diagramas de Sequência

primeiro



último



Diagramas de Sequência

último

```
public String statement() {  
    String result = "Rental record for " + getName() + "\n";  
    for (Rental rental : rentals) {  
        double amount = rental.getCharge();  
  
        // show figures for this rental  
        result += "\t" + rental.getMovie().getTitle() + "\t" +  
            String.valueOf(amount) + "\n";  
    }  
  
    result += "Amount owed is " + String.valueOf(getTotalCharge()) + "\n";  
    result += "You earned " + String.valueOf(getTotalFrequentRenterPoints()) +  
        " frequent renter points";  
    return result;  
}  
  
private double getTotalCharge() {  
    double total = 0;  
    for (Rental rental : rentals)  
        total += rental.getCharge();  
    return total;  
}  
  
private int getTotalFrequentRenterPoints() {  
    int total = 0;  
    for (Rental rental : rentals)  
        total += rental.getFrequentRenterPoints();  
    return total;  
}
```

Customer

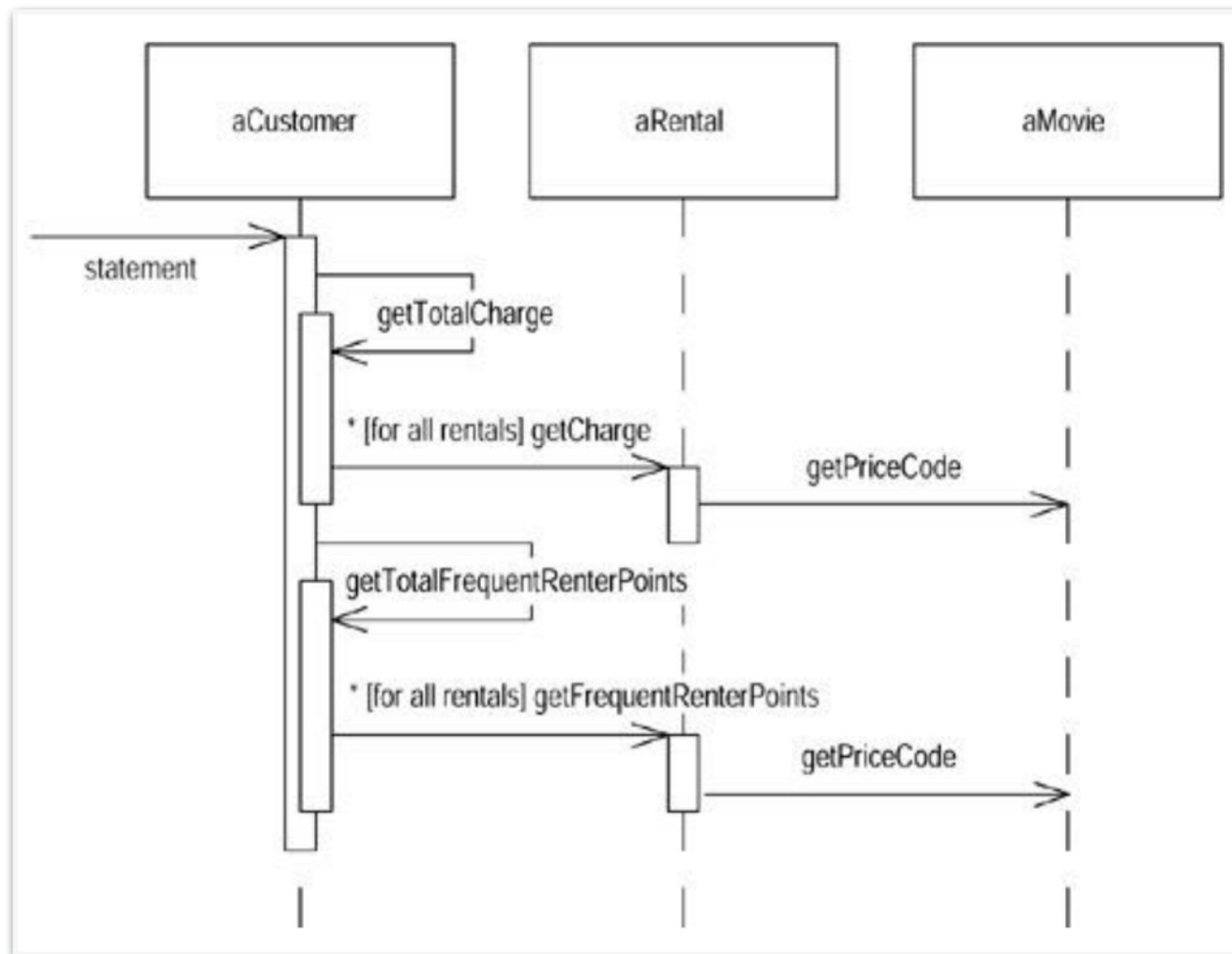


Diagrama de Sequência (lembrando)

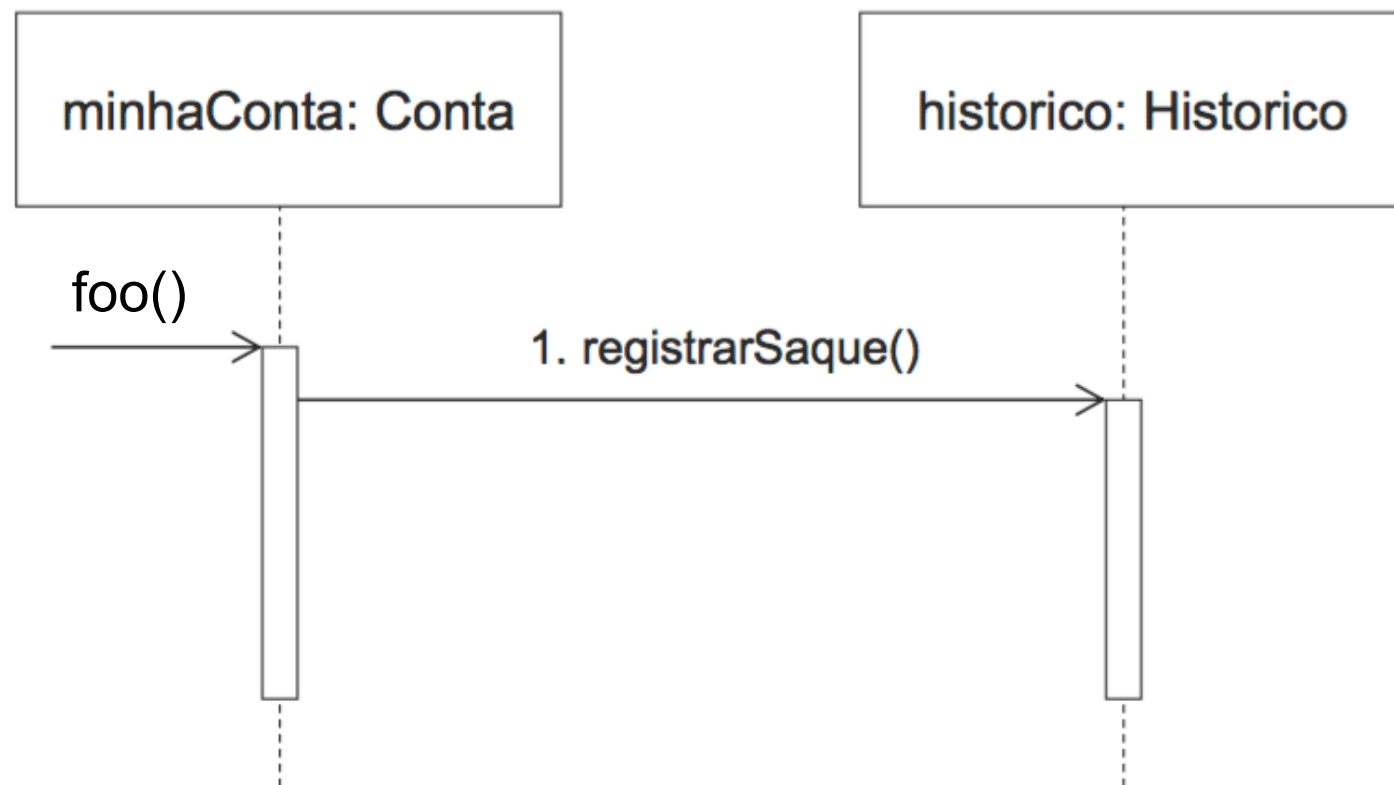
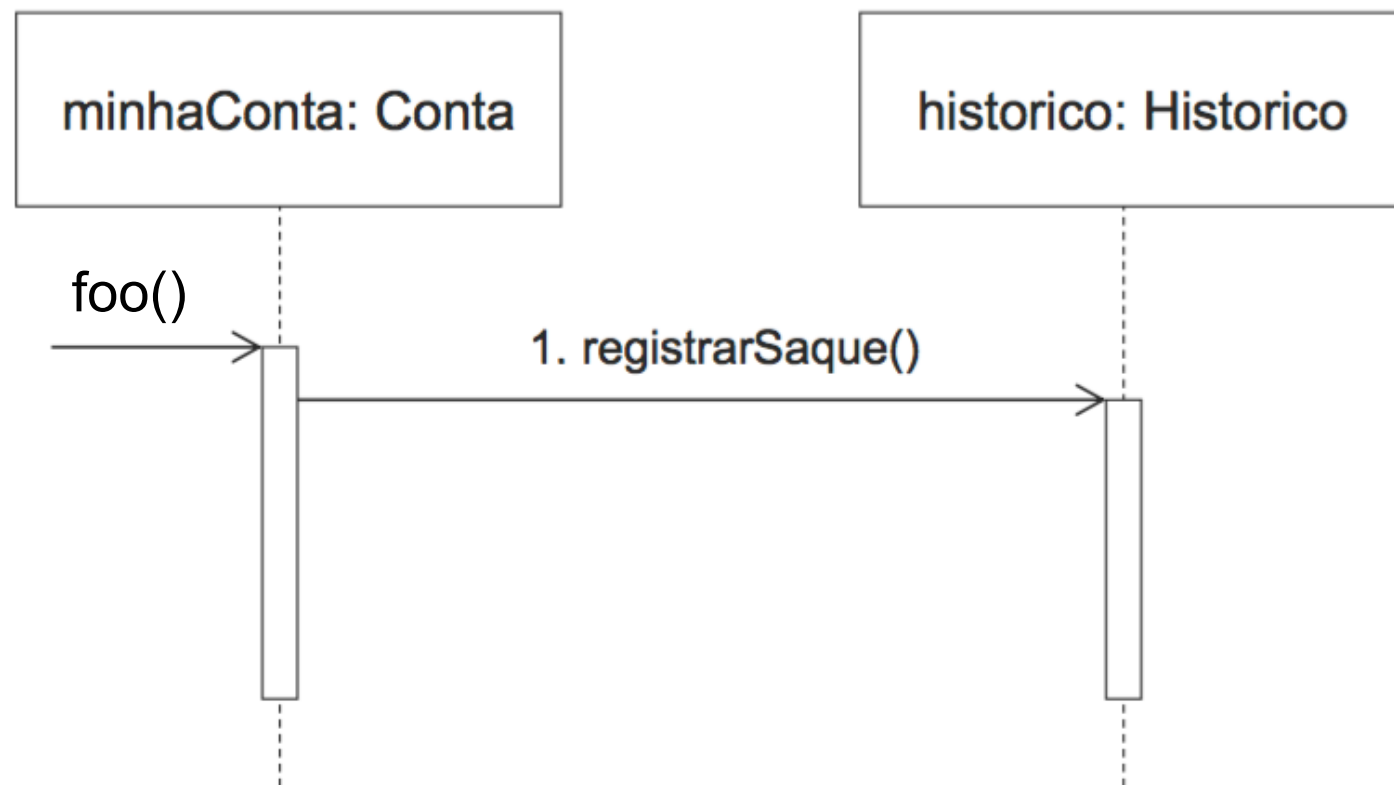
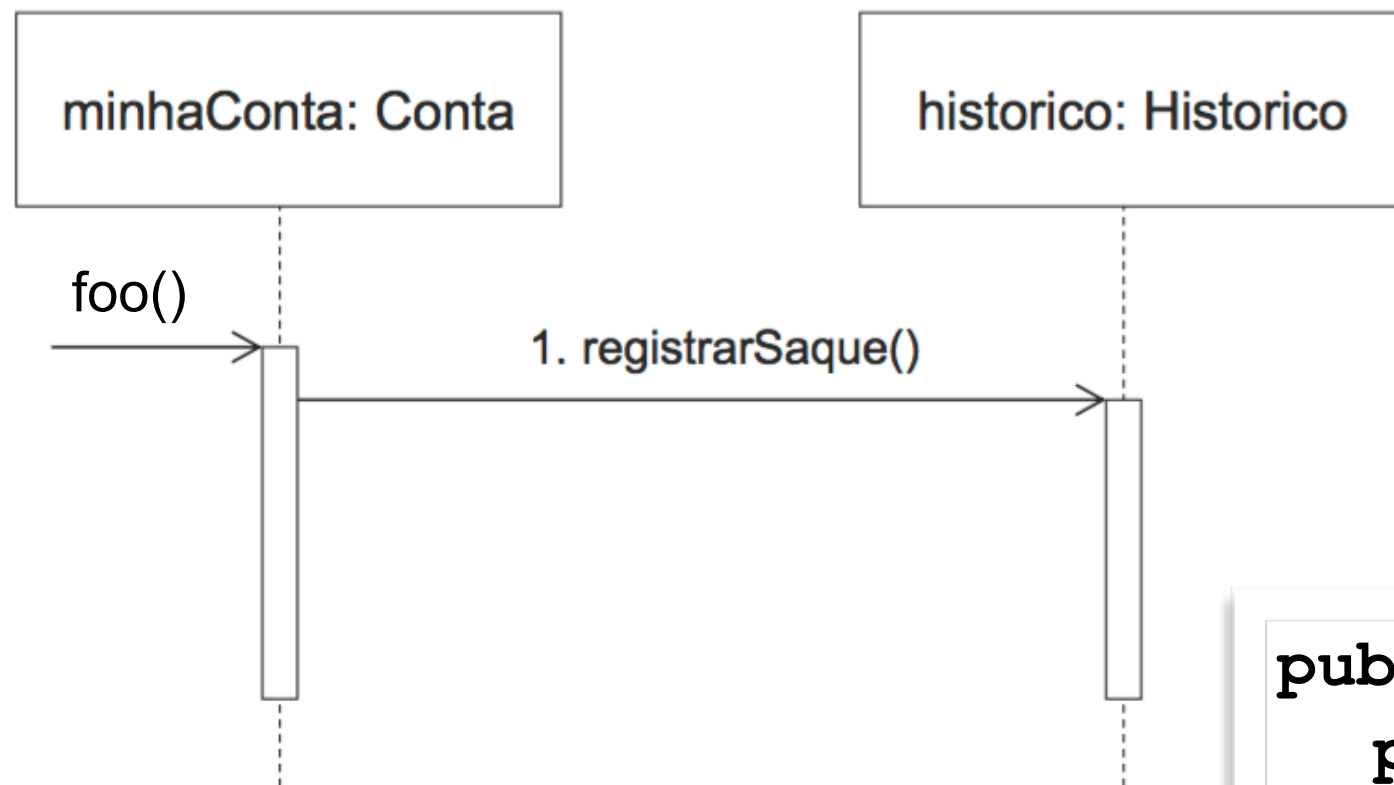


Diagrama de Sequência (lembrando)



```
public class Historico {  
    //...  
    public void registrarSaque() {  
        //...  
    }  
}
```

Diagrama de Sequência (lembrando)



```
public class Historico {  
    //...  
    public void registrarSaque() {  
        //...  
    }  
}
```

```
public class Conta {  
    private Historico historico;  
    public void foo() {  
        historico.registrarSaque();  
        //...  
    }  
}
```

Diagrama de Sequência (lembrando)

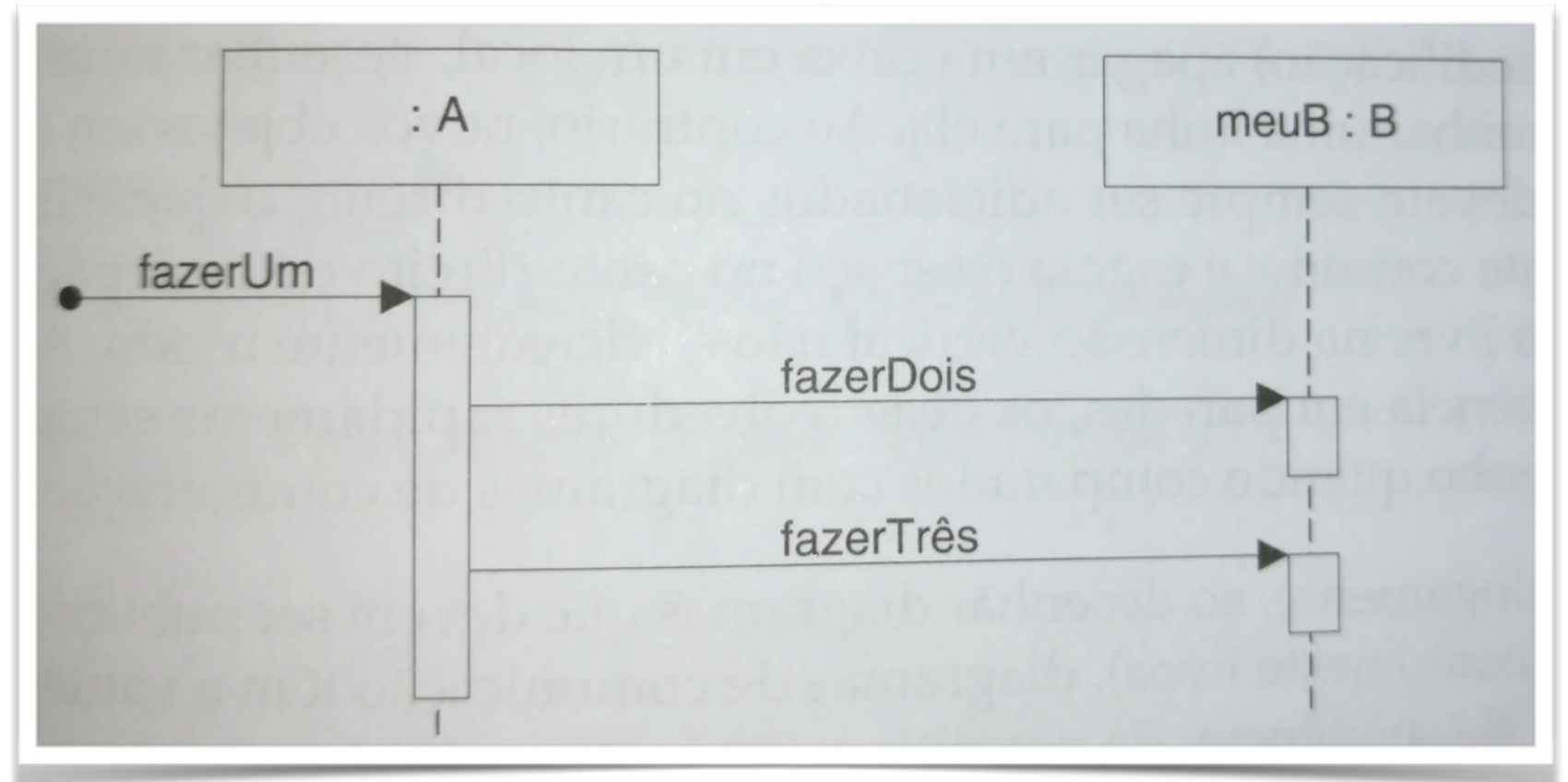
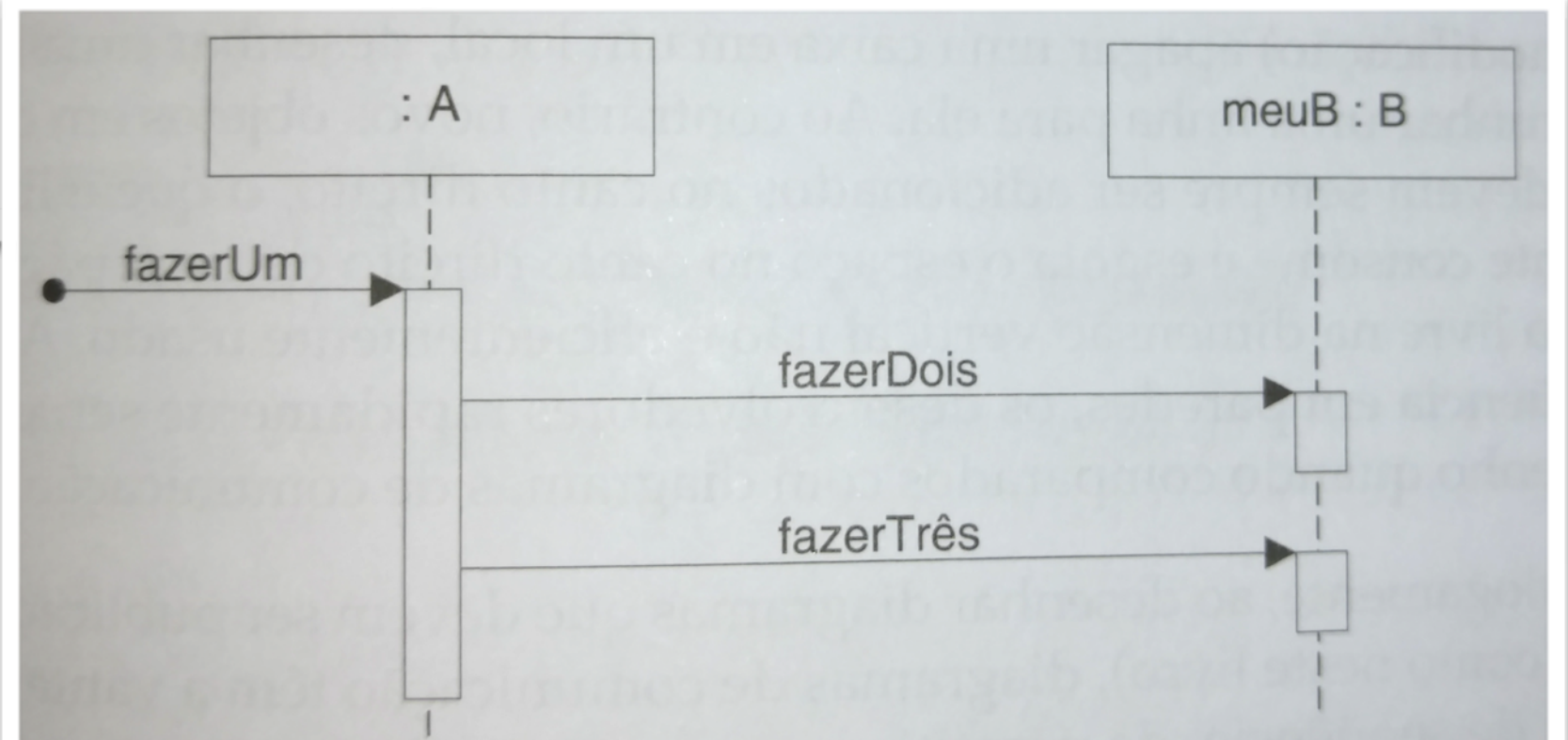


Diagrama de Sequência (lembrando)

```
public class A {  
    private B meuB;  
    public void fazerUm() {  
        meuB.fazerDois();  
        meuB.fazerTres();  
    }  
    // ...  
}
```



Refatorações vistas hoje

- Extract Method
- Move Method
- Replace Temp with Query
- ...

Alteração 1:

Fatura em HTML

- Fatura deve ser impressa em HTML para que o sistema possa rodar na web

Alteração 1:

Fatura em HTML

```
public String htmlStatement() {  
    String result = "<h1>Rental record for <b>" + getName() + "</b></h1>\n";  
    for (Rental rental : rentals)  
        result += "<p>" + rental.getMovie().getTitle() + "\t" + String.valueOf(rental.getCharge()) + "</p>\n";  
    result += "<p>Amount owed is <b>" + String.valueOf(getTotalCharge()) + "</b></p>\n";  
    result += "<p>You earned <b>" + String.valueOf(getTotalFrequentRenterPoints()) + " frequent renter points</b></p>";  
    return result;  
}
```

Alteração 1:

Fatura em HTML

```
public String htmlStatement() {  
    String result = "<h1>Rental record for <b>" + getName() + "</b></h1>\n";  
    for (Rental rental : rentals)  
        result += "<p>" + rental.getMovie().getTitle() + "\t" + String.valueOf(rental.getCharge()) + "</p>\n";  
    result += "<p>Amount owed is <b>" + String.valueOf(getTotalCharge()) + "</b></p>\n";  
    result += "<p>You earned <b>" + String.valueOf(getTotalFrequentRenterPoints()) + " frequent renter points</b></p>";  
    return result;  
}
```

- `htmlStatement()` reusa todos os métodos de cálculo
- Se as regras de cálculo mudam, só existe um lugar para alterar
- Qualquer outro formato (CSV, XML, JSON, FOO, BAR), será muito fácil de adicionar, sem duplicação de código
- Claro, o código de `statement()` pode ainda ser refatorado (extrair header, footer, detail)

antes

```
public String statement() {
    double totalAmount = 0;
    int frequentRenterPoints = 0;

    String result = "Rental record for " + getName() + "\n";
    for (Rental rental : rentals) {
        double amount = 0;
        switch (rental.getMovie().getPriceCode()) {
            case Movie.REGULAR:
                amount += 2;
                if (rental.getDaysRented() > 2)
                    amount += (rental.getDaysRented() - 2) * 1.5;
                break;
            case Movie.NEW_RELEASE:
                amount += rental.getDaysRented() * 3;
                break;
            case Movie.CHILDREN:
                amount += 1.5;
                if (rental.getDaysRented() > 3)
                    amount += (rental.getDaysRented() - 3) * 1.5;
                break;
        }

        // add frequent renter points
        frequentRenterPoints++;
        // add bonus for a two day new release rental
        if (rental.getMovie().getPriceCode() == Movie.NEW_RELEASE && rental.getDaysRented() > 1)
            frequentRenterPoints++;

        // show figures for this rental
        result += "\t" + rental.getMovie().getTitle() + "\t" + String.valueOf(amount) + "\n";

        totalAmount += amount;
    }

    result += "Amount owed is " + String.valueOf(totalAmount) + "\n";
    result += "You earned " + String.valueOf(frequentRenterPoints) + " frequent renter points";

    return result;
}
```

depois

```
public String statement() {
    String result = "Rental record for " + getName() + "\n";
    for (Rental rental : rentals)
        result += "\t" + rental.getMovie().getTitle() + "\t" + String.valueOf(rental.getCharge()) + "\n";
    result += "Amount owed is " + String.valueOf(getTotalCharge()) + "\n";
    result += "You earned " + String.valueOf(getTotalFrequentRenterPoints()) + " frequent renter points";
    return result;
}
```

Alteração 2:

Novos Tipos de Filmes

- Os tipos dos filmes vão mudar constantemente
 - Tipos hoje: Regular, Children e New Releases
 - Tipos amanhã: ?
- Importante: essa alteração afeta diretamente a forma de cobrança e o cálculo das pontuações