

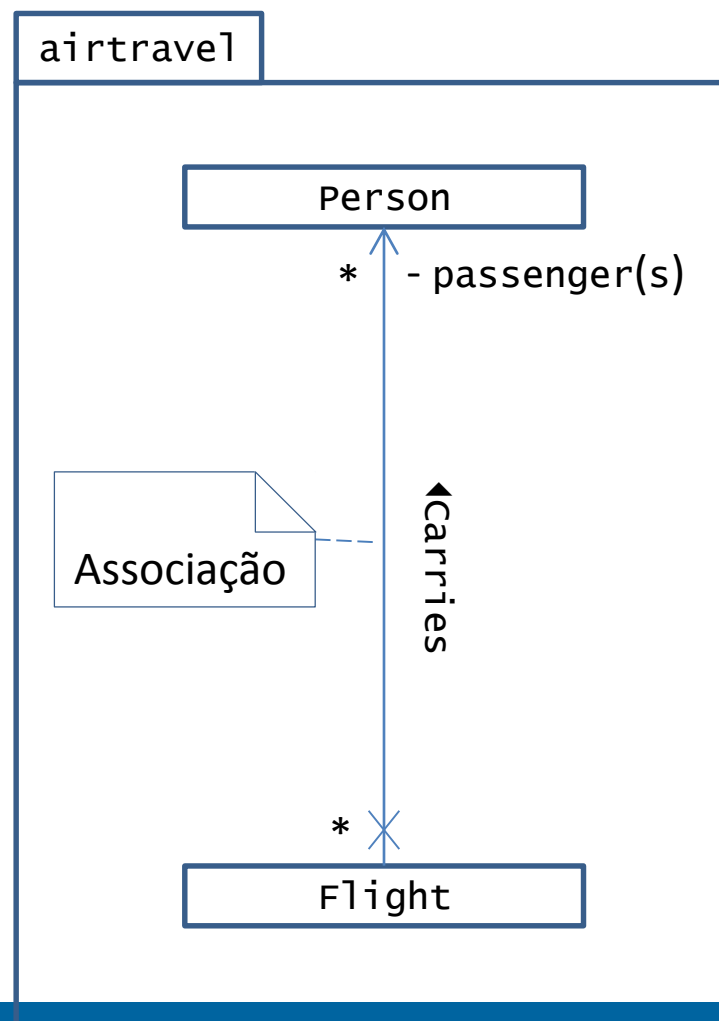
Introdução ao UML

Diagramas de classe

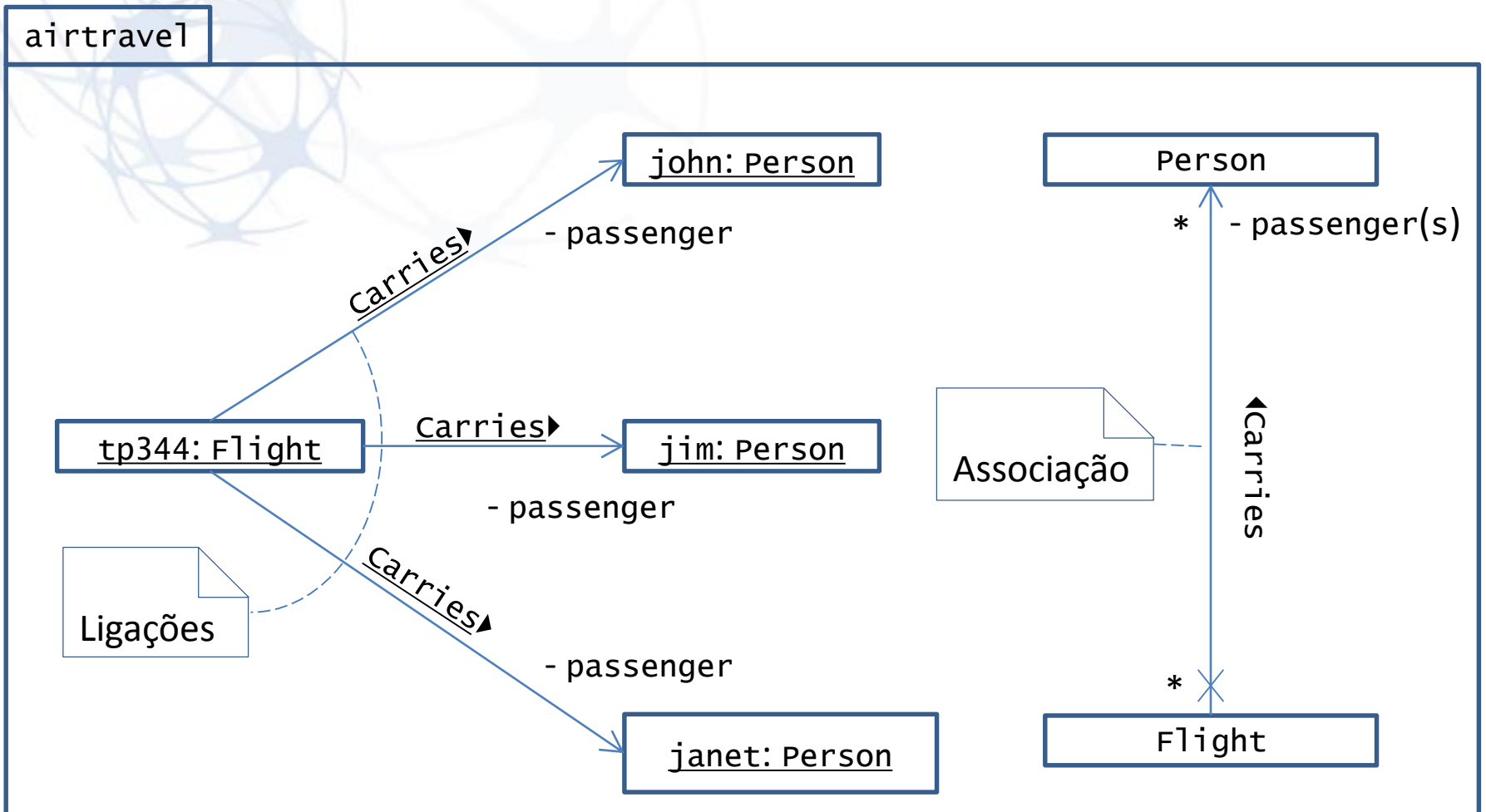
Diagramas de Sequência

Associação

```
package airtravel;  
public class Person {  
    ...  
}  
  
public class Flight {  
    private Set<Person>  
        passengers;  
  
    ...  
}
```

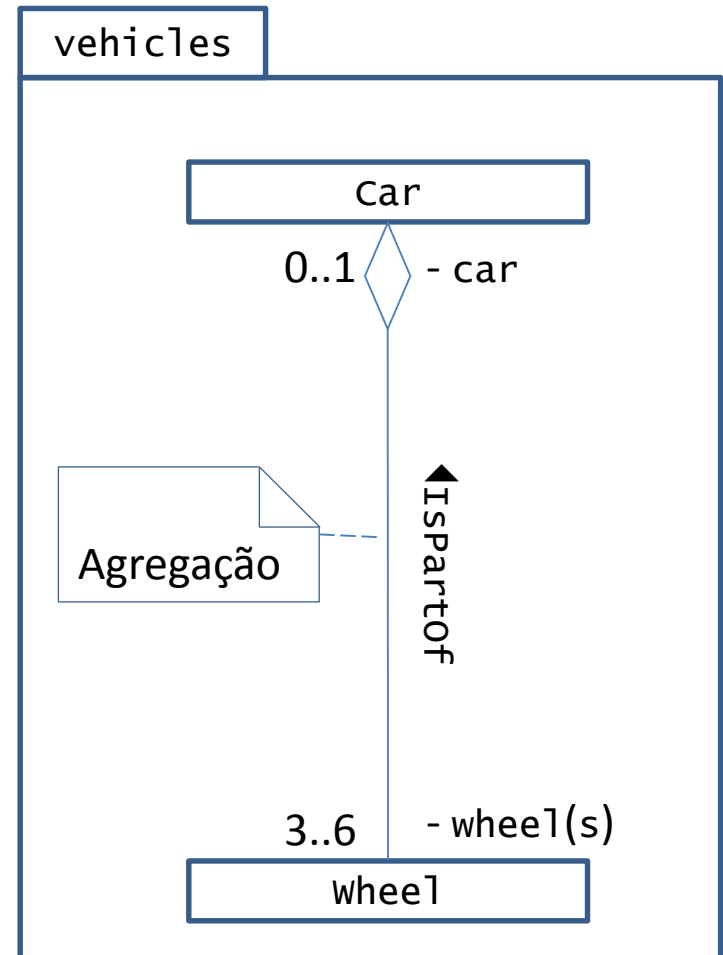


Ligação



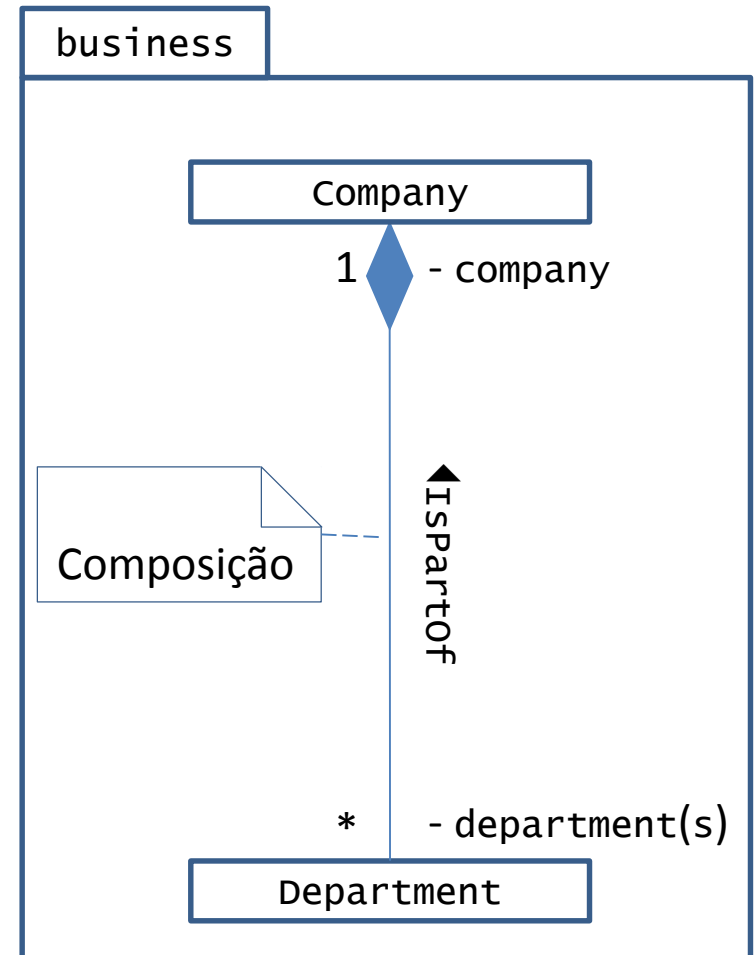
Associação: agregação

```
package vehicles;  
public class Car {  
    private Set<wheel> wheels;  
    ...  
}  
public class wheel {  
    private Car car;  
    ...  
}
```



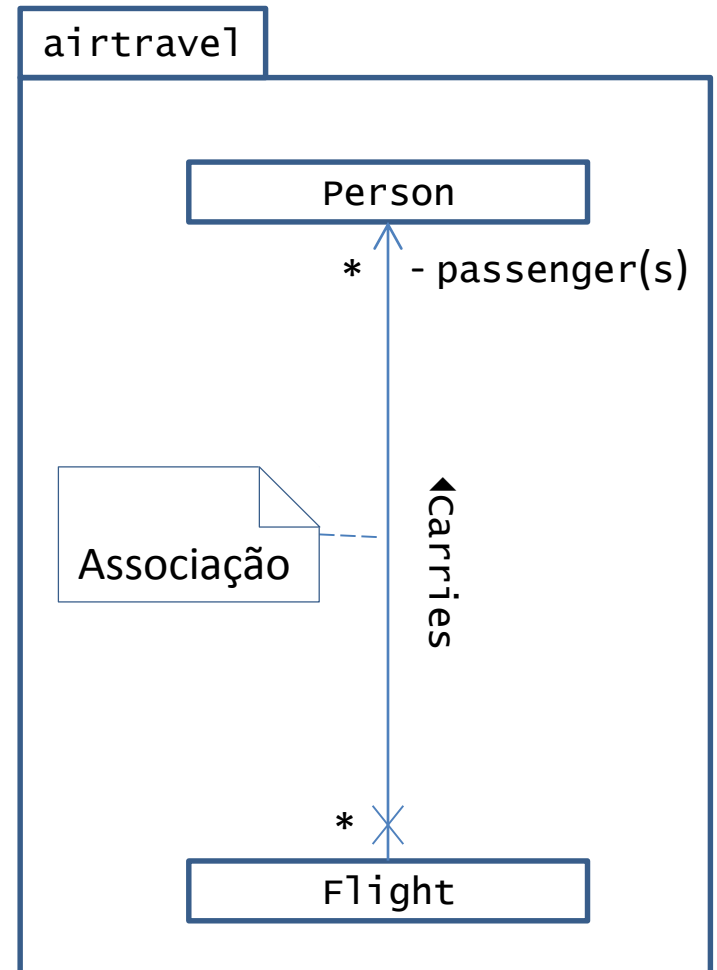
Associação: composição

```
package business;  
public class Company {  
    private Set<Department>  
        departments;  
    ...  
}  
public class Department {  
    private Company company;  
    ...  
}
```



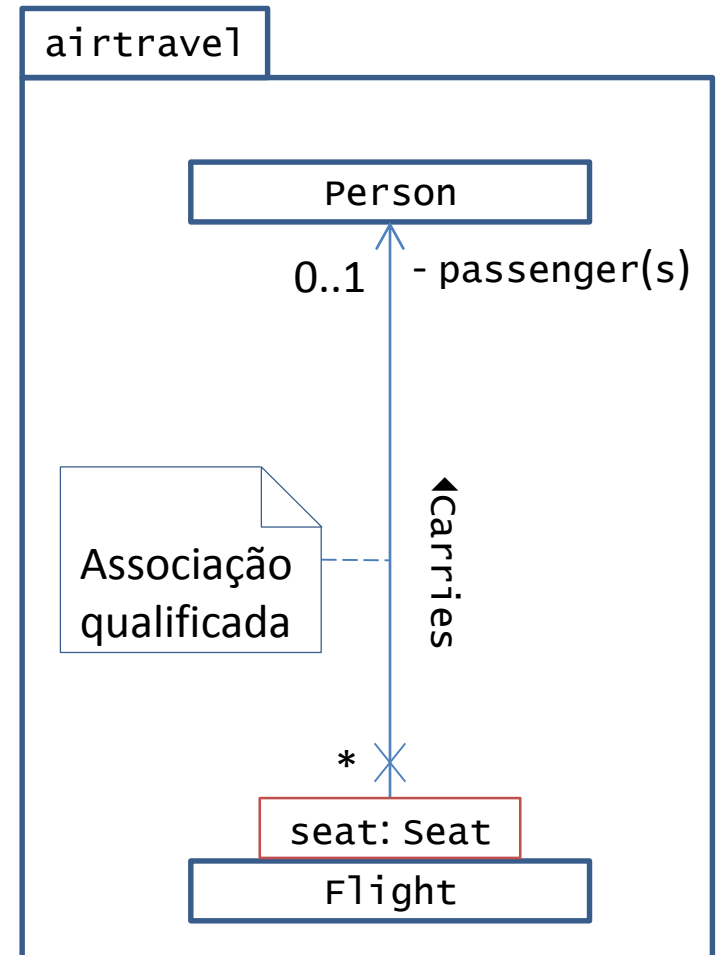
Associação

- E os lugares?



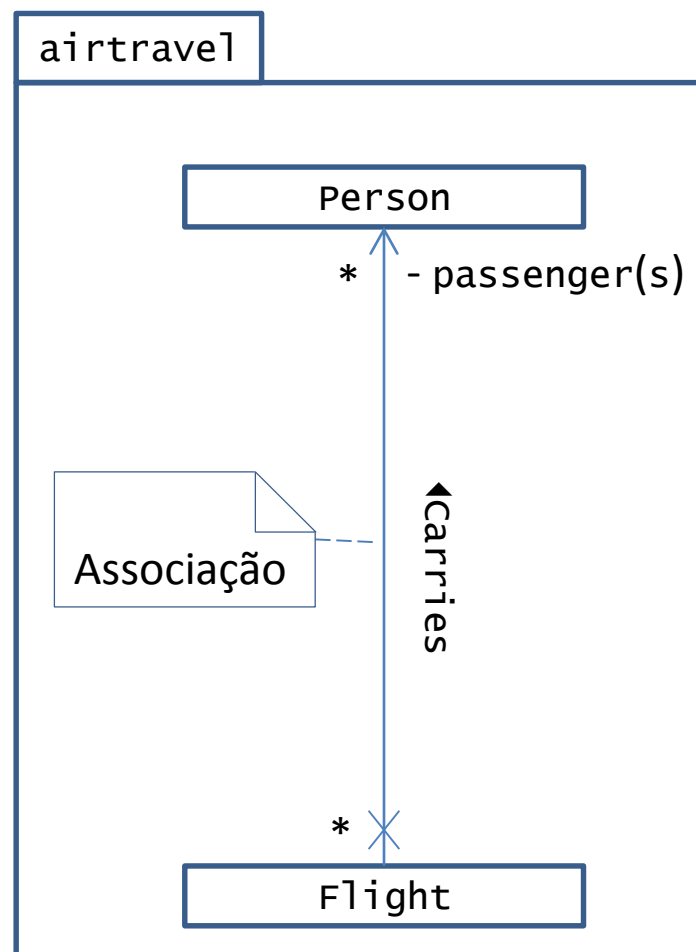
Associação qualificada

```
package airtravel;  
public class Person {  
    ...  
}  
  
public class Flight {  
    private Map<Seat, Person>  
        passengers;  
    ...  
}
```



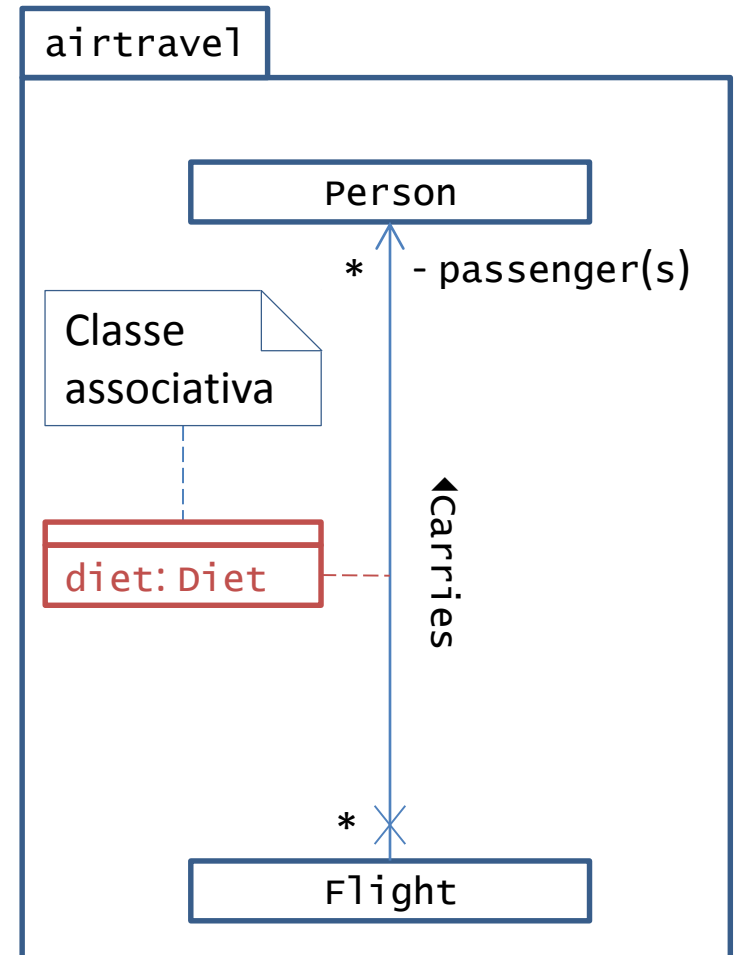
Associação

- E características especiais, como a dieta pretendida?



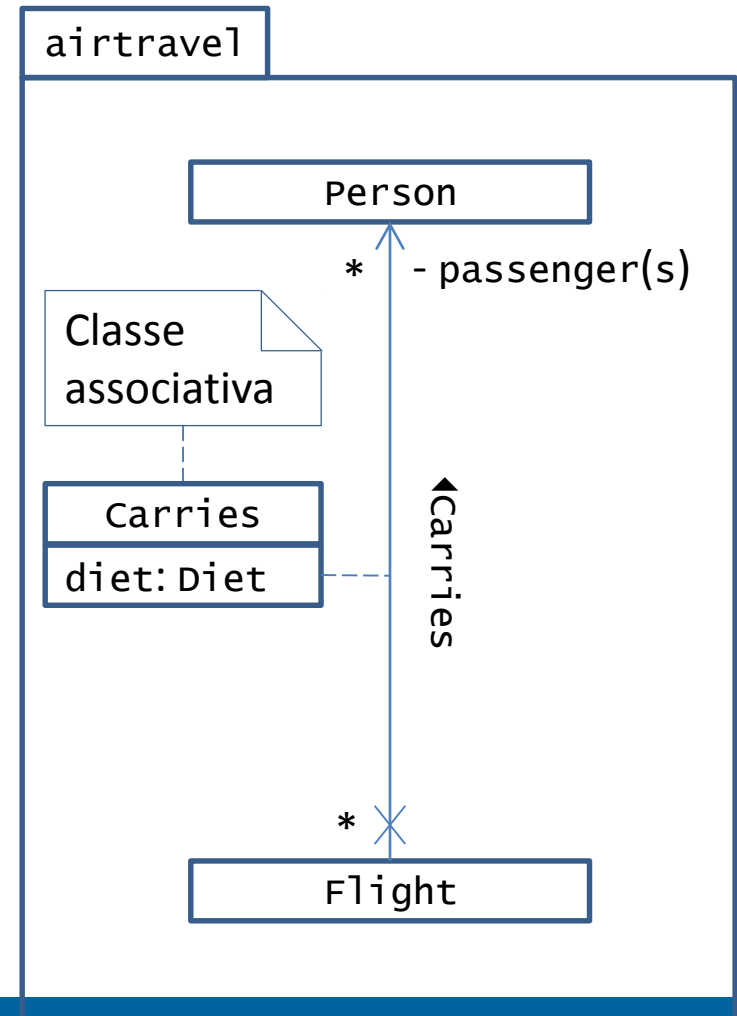
Classe associativa

```
package airtravel;  
public class Person { ... }  
public class Carries {  
    private Person passenger;  
    private Diet diet; ...  
}  
public class Flight {  
    private Set<Carries>  
        passengers;  
    ...  
}
```



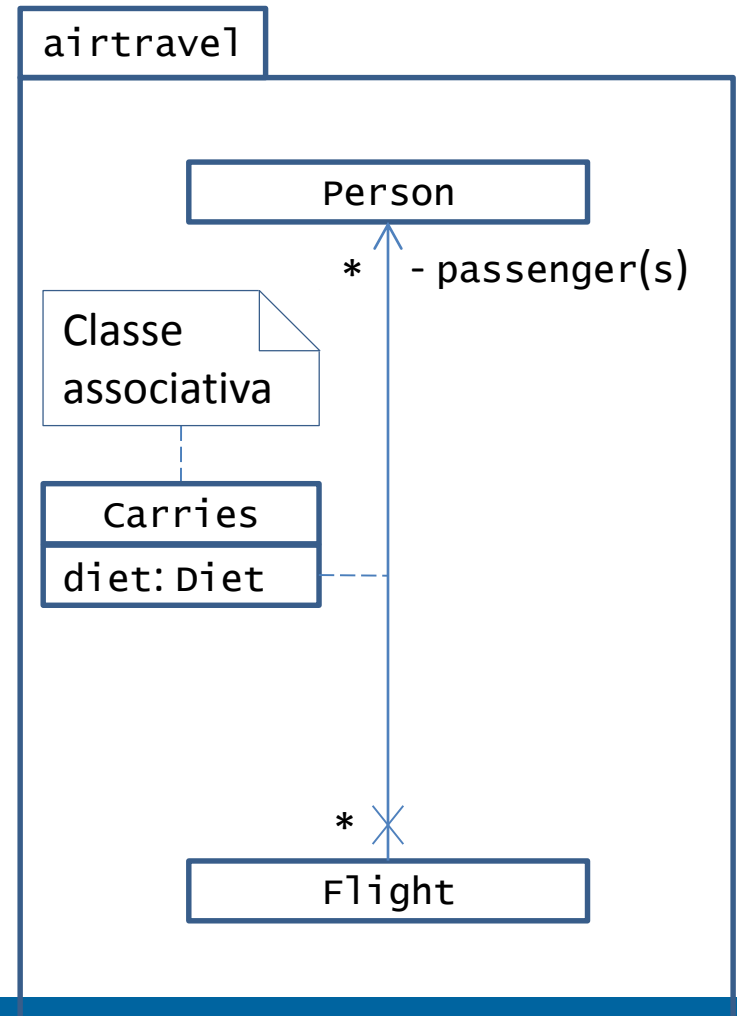
Classe associativa

```
package airtravel;  
public class Person {...}  
public class Carries {  
    private Person passenger;  
    private Diet diet;    ...  
}  
public class Flight {  
    private Set<Carries>  
        passengers;  
    ...  
}
```



Classe associativa

```
package airtravel;  
public class Person { ... }  
public class Carries {  
    private Person passenger;  
    private Diet diet;    ...  
}  
public class Flight {  
    private Set<Carries>  
        passengers;  
    ...  
}
```



Multiplicidade de classes

MyClass *n..m*

- Pode restringir-se número de instâncias simultâneas de classe no sistema
- Implementação em Java recorre a truques

Solitão (*singleton*)

```
package mypackage;
public final class MySingleton {
    private static final MySingleton INSTANCE =
        new MySingleton();
    private MySingleton() {
        assert INSTANCE == null : ...;
    }
    public static MySingleton getInstance() {
        return INSTANCE;
    } ...
}
```

MySingleton ¹

Classe-pacote

```
package java.lang;
```

```
public final class Math {  
    private Math() {  
        assert false : ...;  
    }  
}
```

```
    public static final double PI = 3.14...;
```

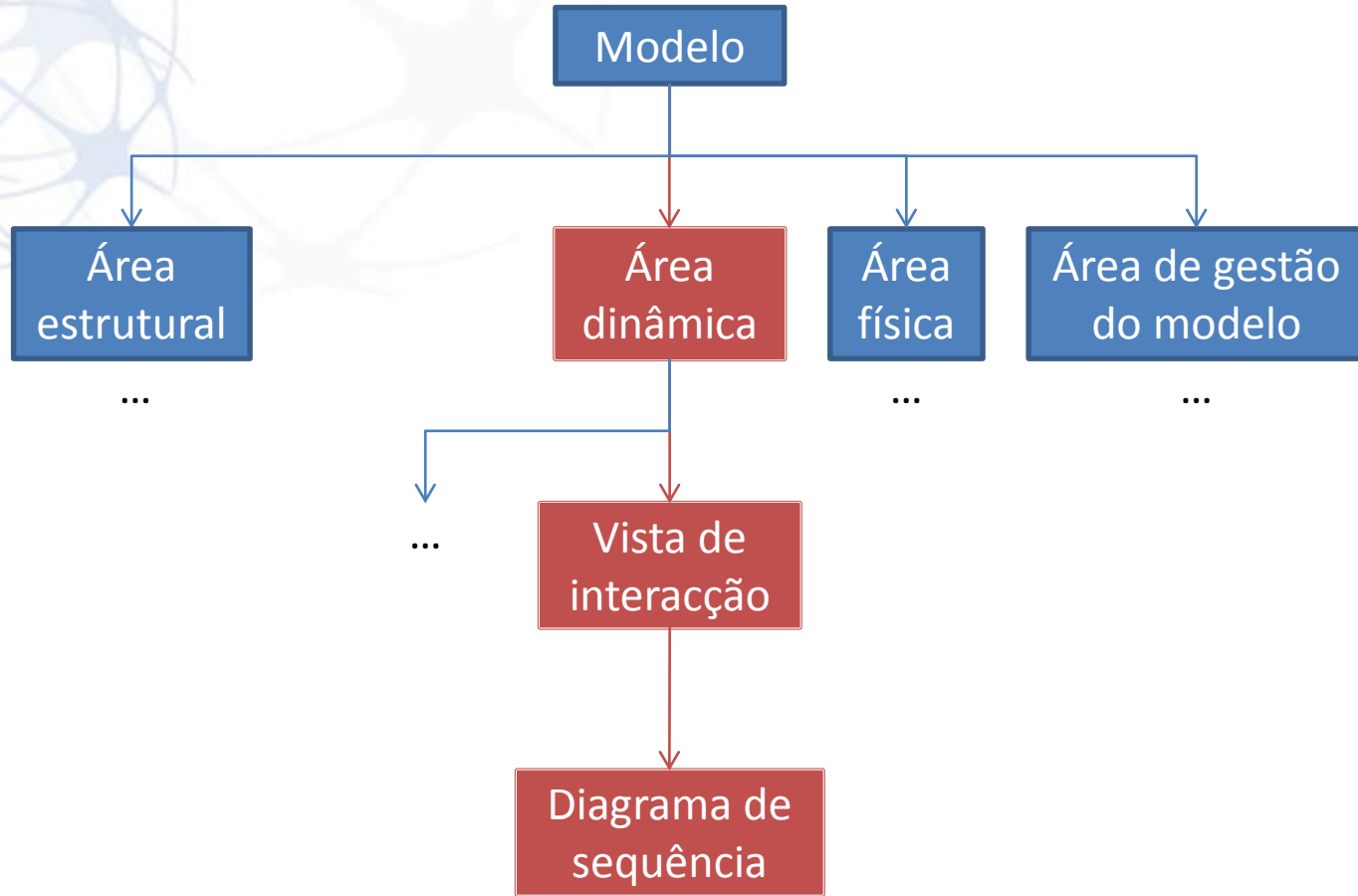
```
    public static double sin(final double angle) {...}  
    public static double cos(final double angle) {...}  
    ...
```

```
}
```

Math

0

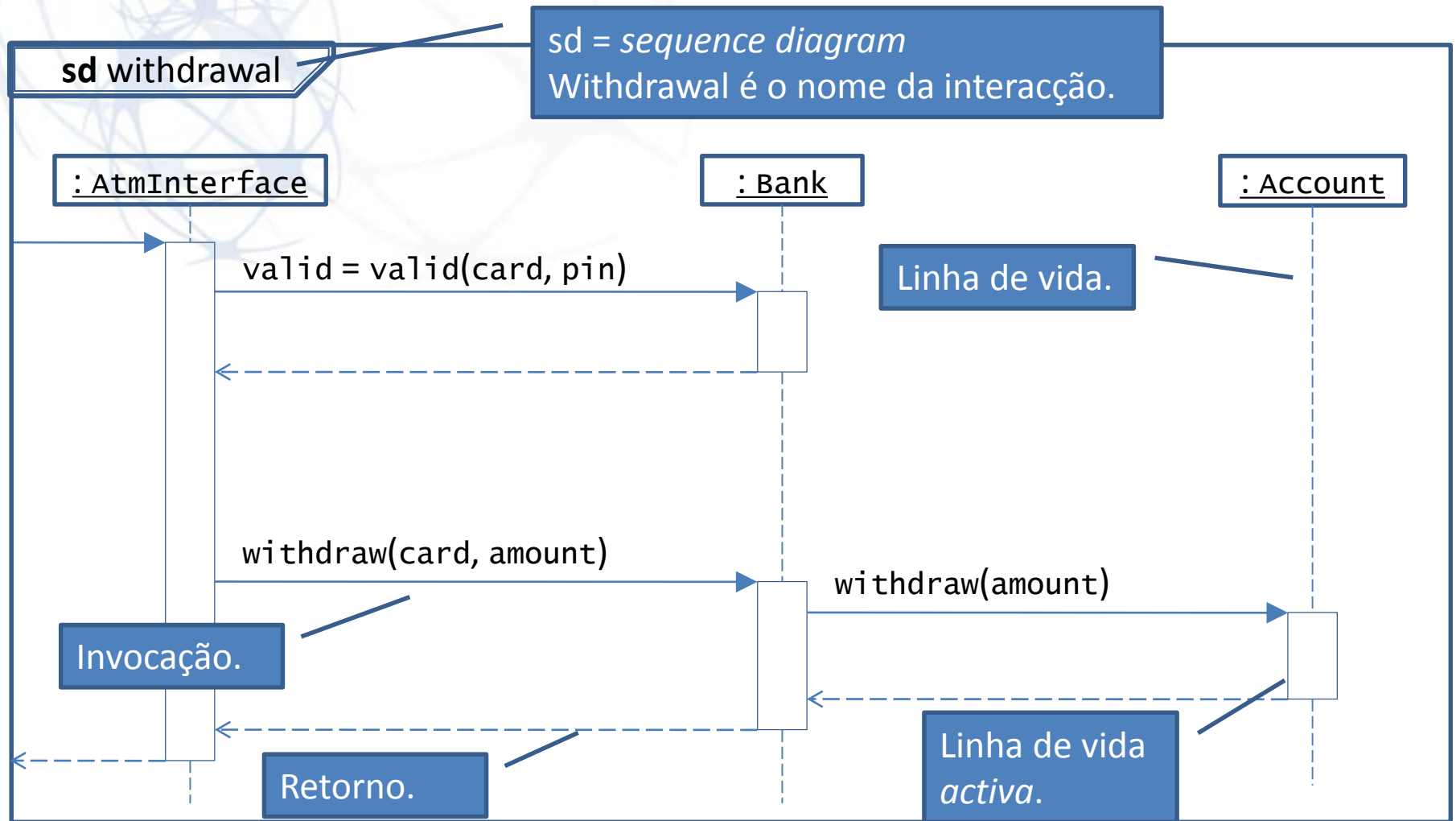
Diagramas



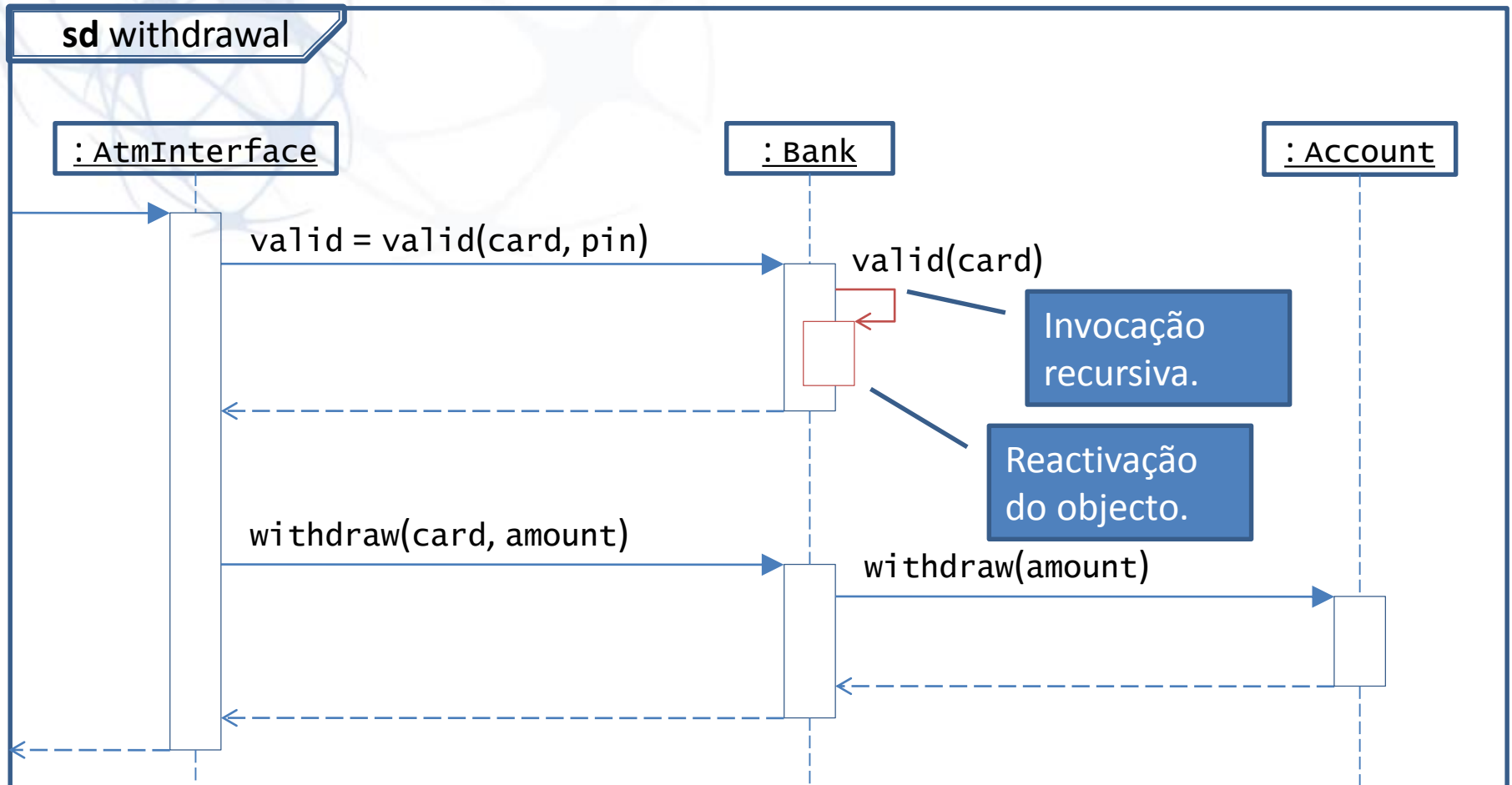
Diagramas de sequência

- Mostram **interacções** entre **entidades** numa **sequência temporal**
- Mostram
 - entidades envolvidas numa interacção
 - sequências de **mensagens** trocadas entre entidades
- Entidades podem ser
 - actores e sistema trocando mensagens (análise; domínio do problema)
 - objectos invocando operações (desenho; domínio da solução)

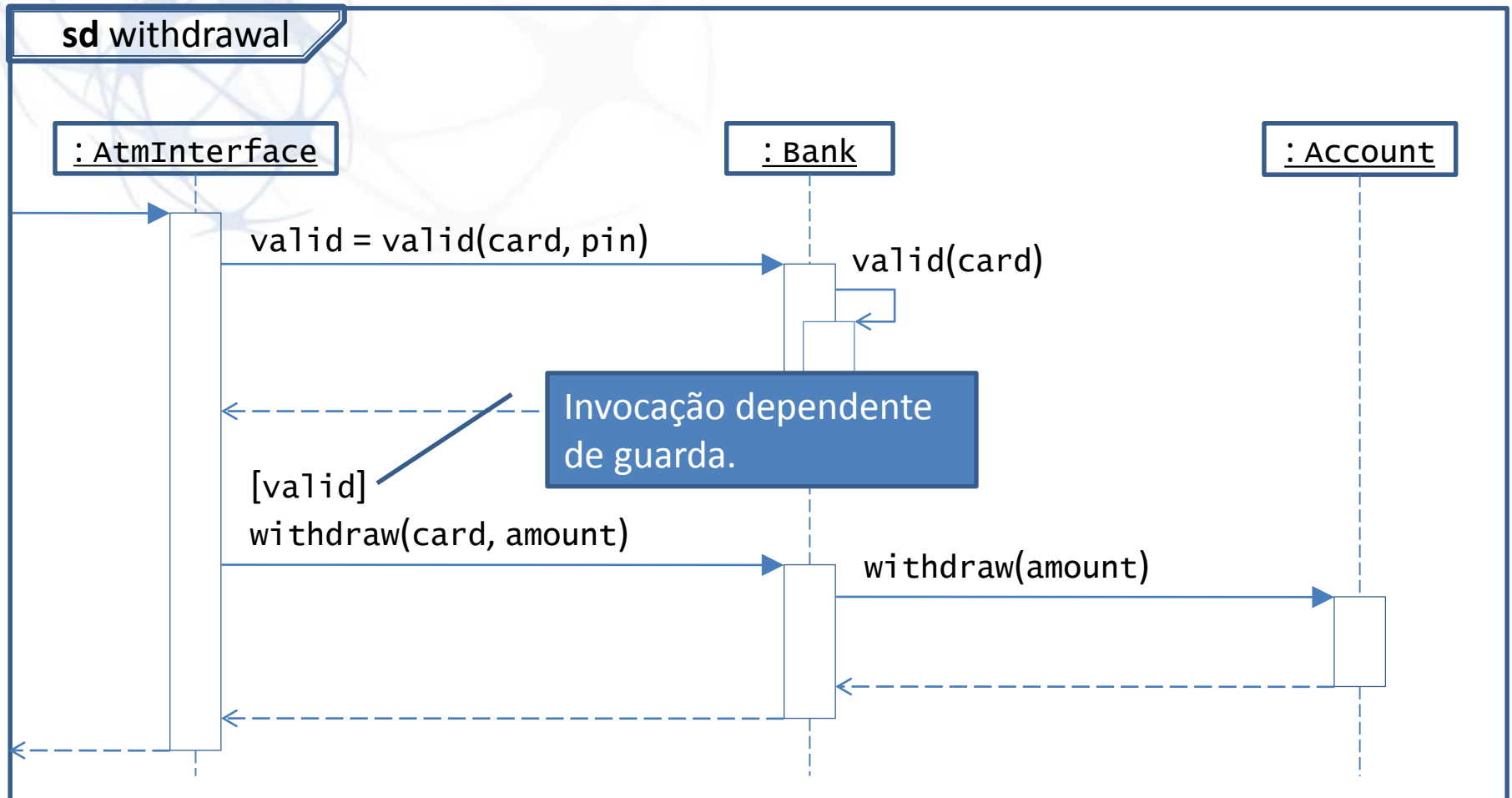
Exemplo 1



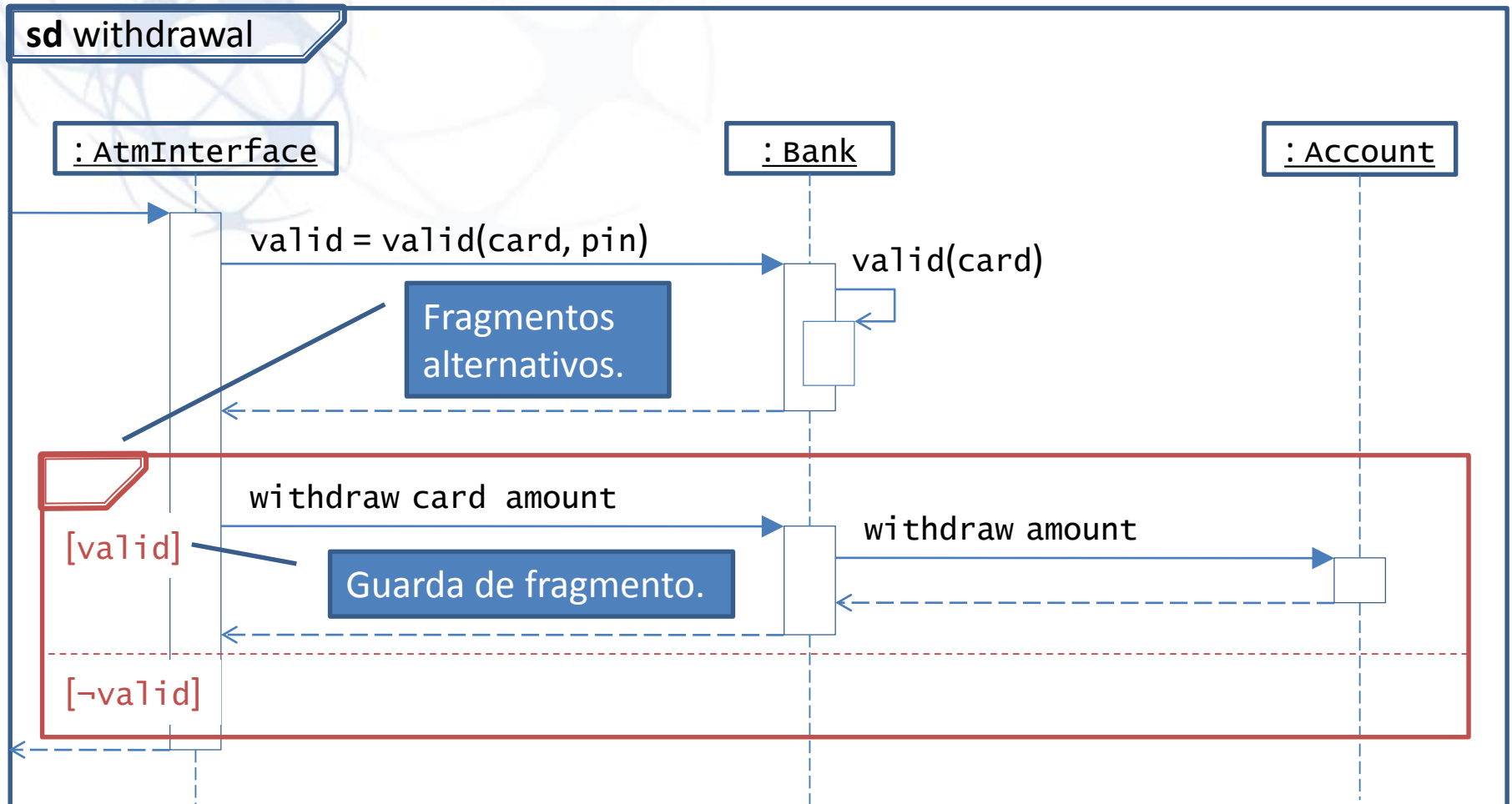
Exemplo 2



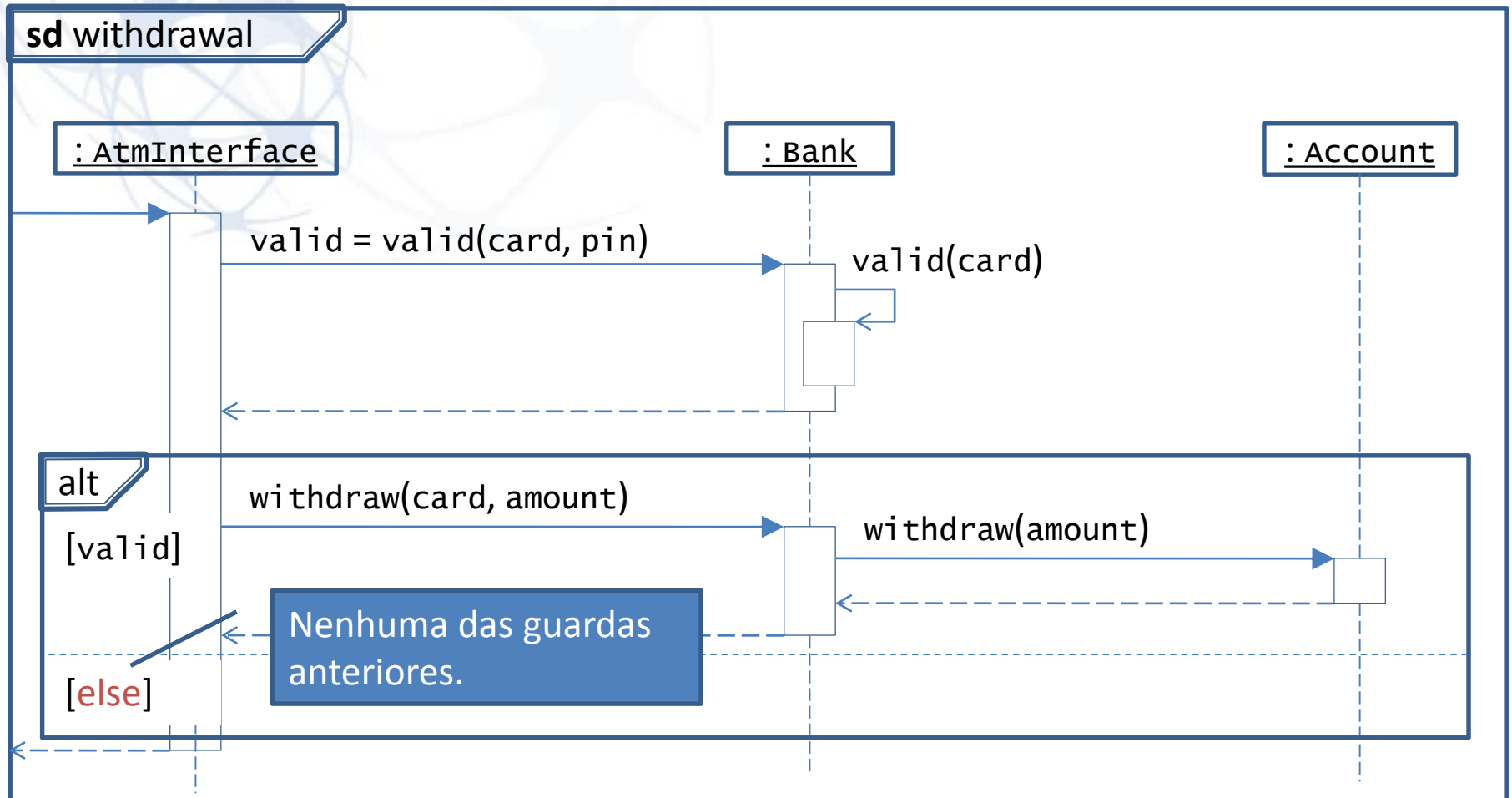
Exemplo 3



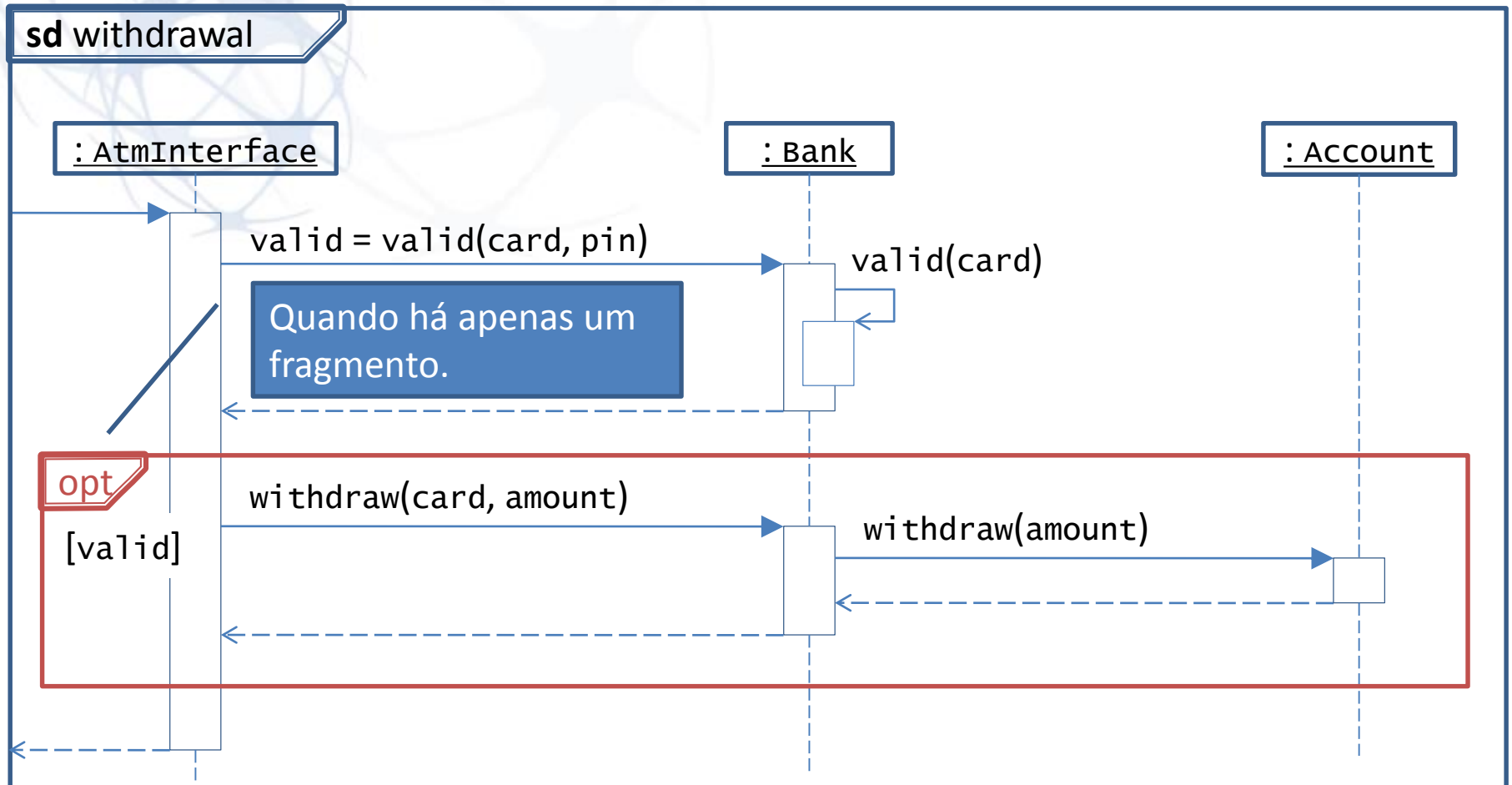
Exemplo 4



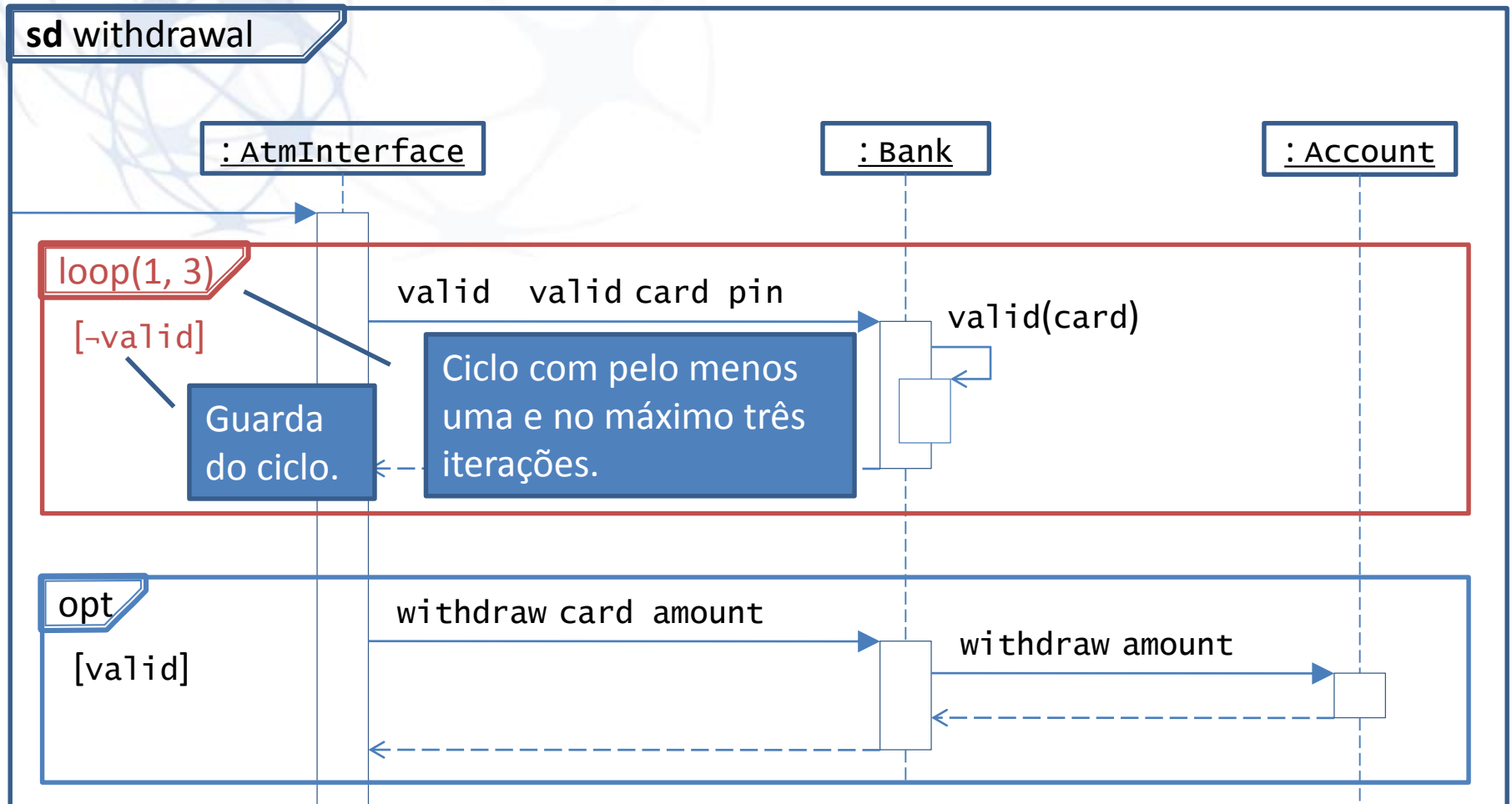
Exemplo 5



Exemplo 6



Exemplo 7



Referências

- UML® Resource Page (<http://www.uml.org/>)
- Martin Fowler, *UML Distilled: A Brief Guide to the Standard Object Modeling Language*, 3.ª edição, Addison-Wesley, 2003.
ISBN: 0-321-19368-7
(1.ª e 2.ª edições na biblioteca)
- James Rumbaugh et al., *The Unified Modeling Language Reference Manual*, 2.ª edição, Addison-Wesley, 2005.
ISBN: 0-321-24562-8
(1.ª edição do guia do utilizador na biblioteca)

Sumário

- Introdução ao UML
 - Ligações como instâncias de associações
 - Multiplicidade de classes (solitões e classes-pacote)
 - Diagramas de sequência