

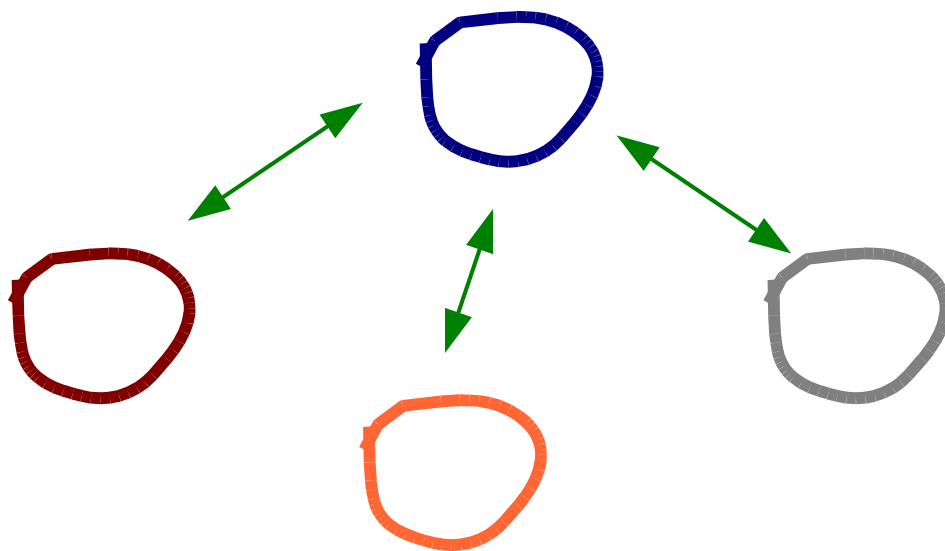
# Projeto Detalhado de Software

Padrões de Projetos  
(GoF Patterns)

Observer

# Padrão – Observer

- **Intenção:** Definir uma dependência um-para-muitos, tal que quando um objeto mudar de estado, os demais são notificados e atualizado automaticamente.

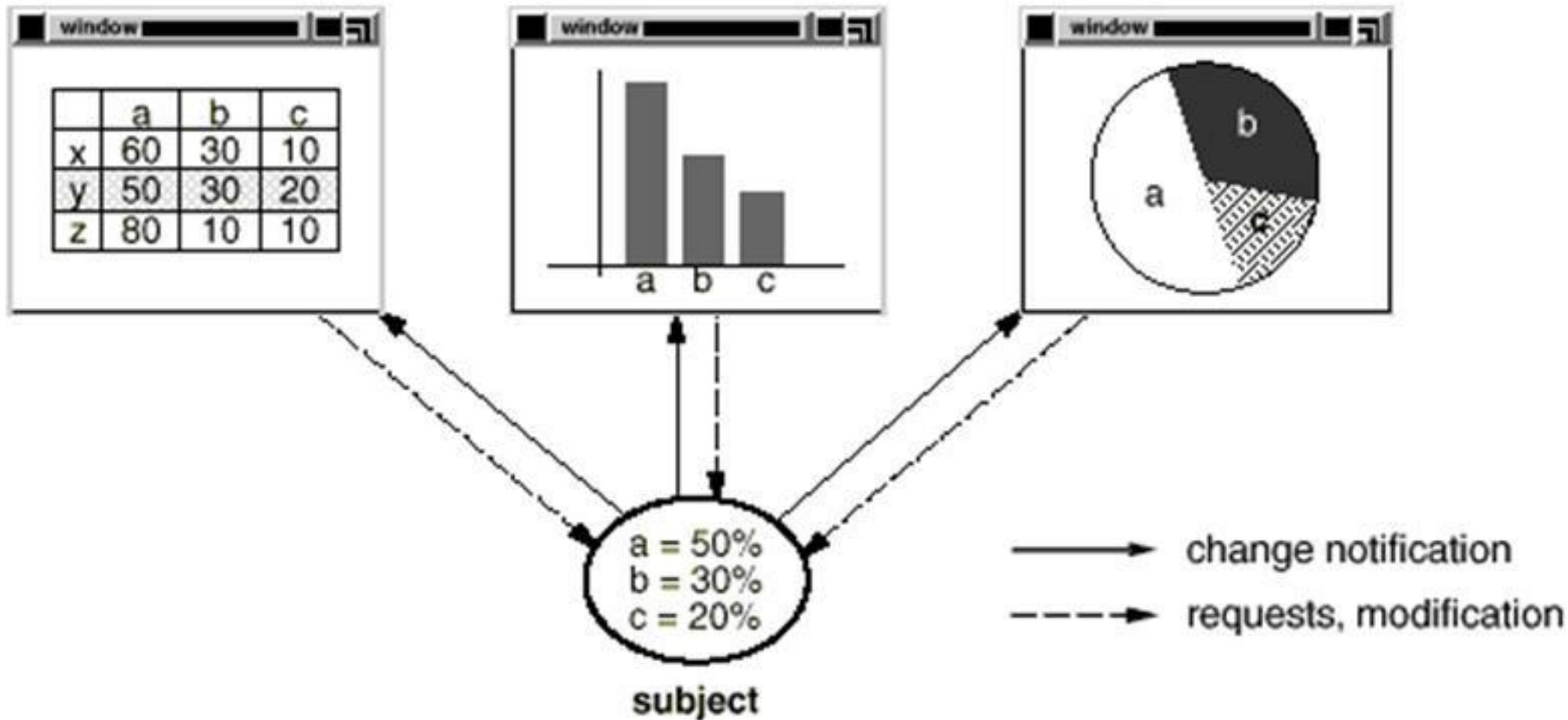


## MOTIVAÇÃO

Ao desacoplar módulos, é necessário manter a consistência de informações

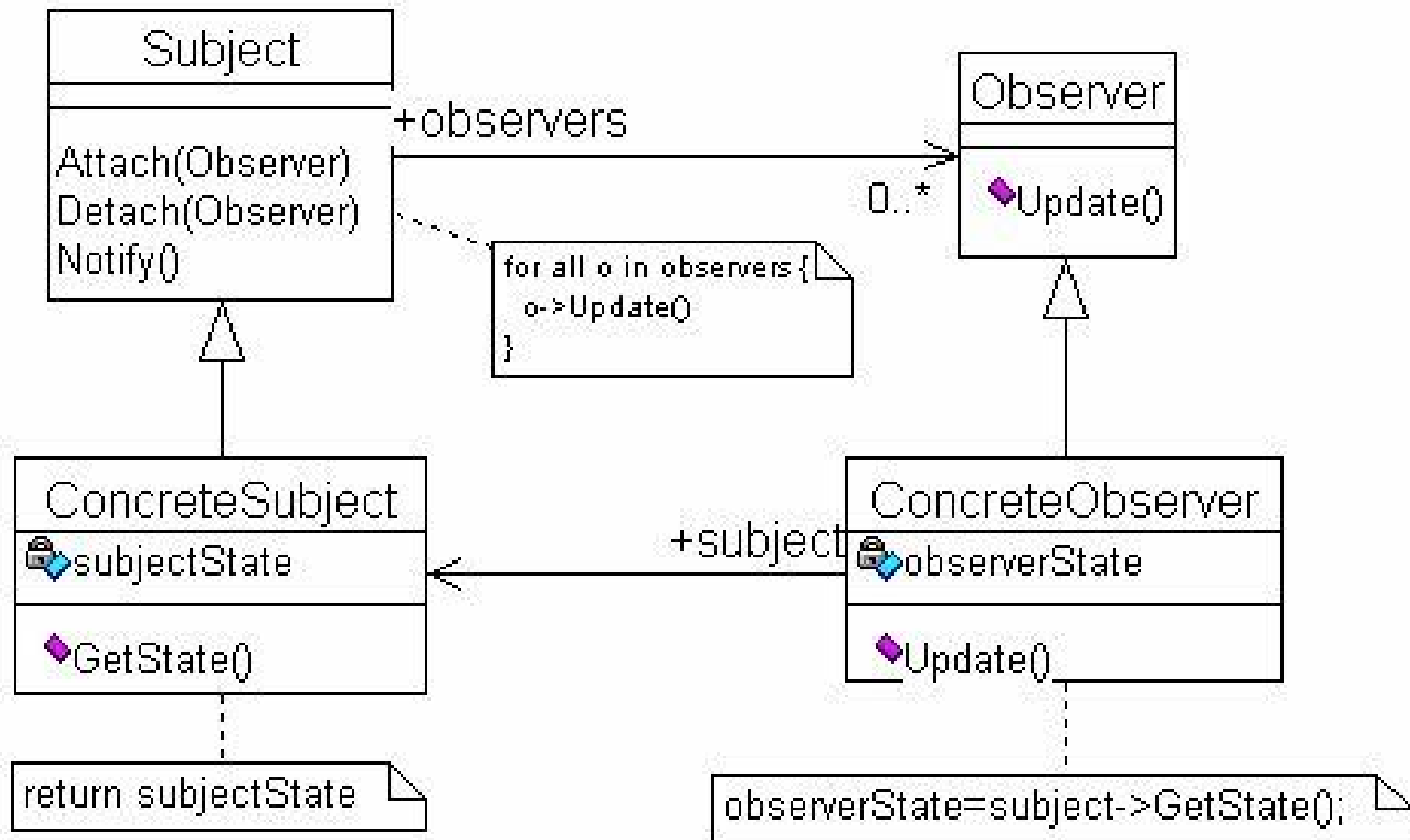
# Padrão – Observer

observers



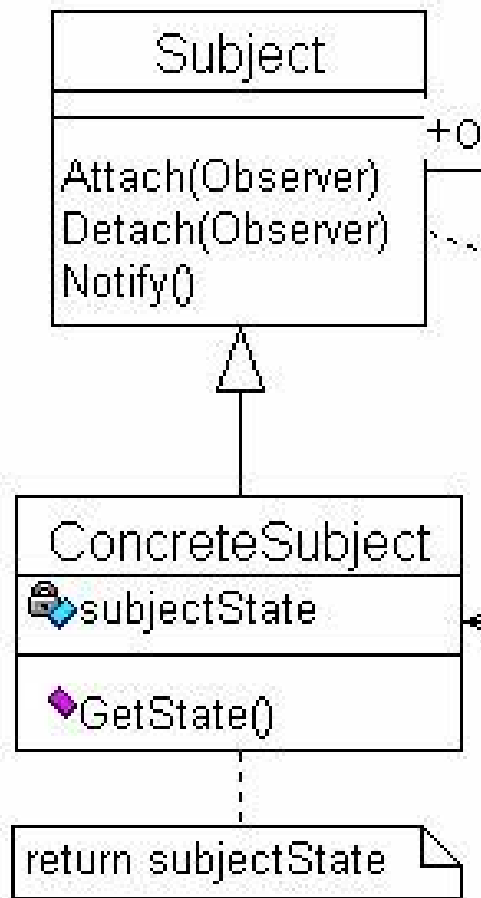
# Padrão – Observer

- Diagrama conceitual



# Padrão – Mediator

- **Participantes**



**Define uma interface para anexar e desanexar objetos Observadores.**

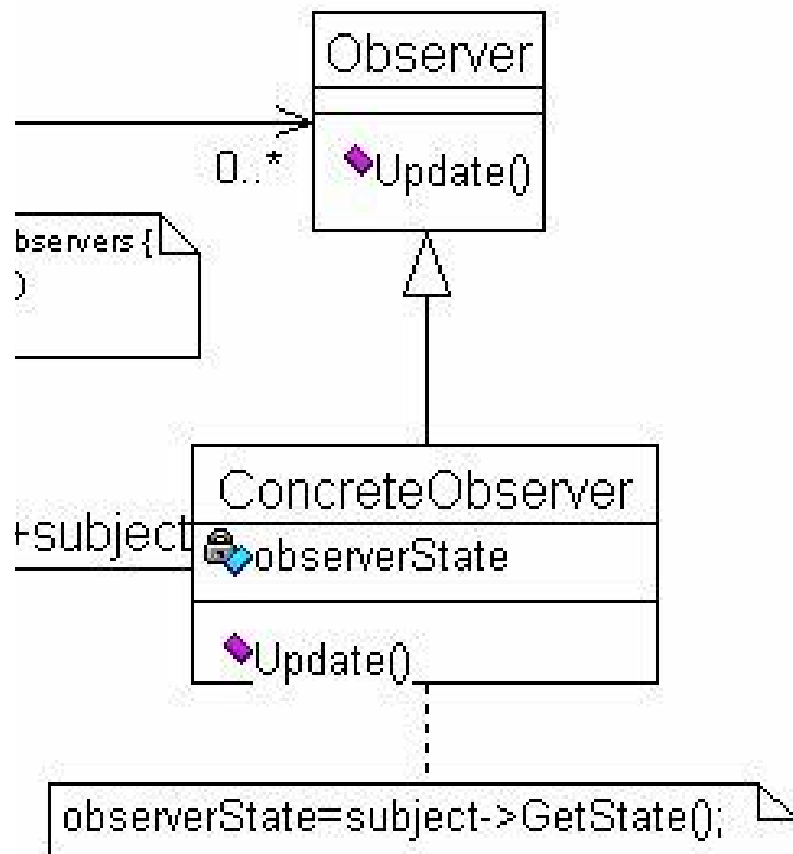
**Conhece seus múltiplos Observadores.**

**Mantém o estado de interesse dos Observadores.**

**Notifica os Observadores quando há mudança de estado.**

# Padrão – Observer

- **Participantes**

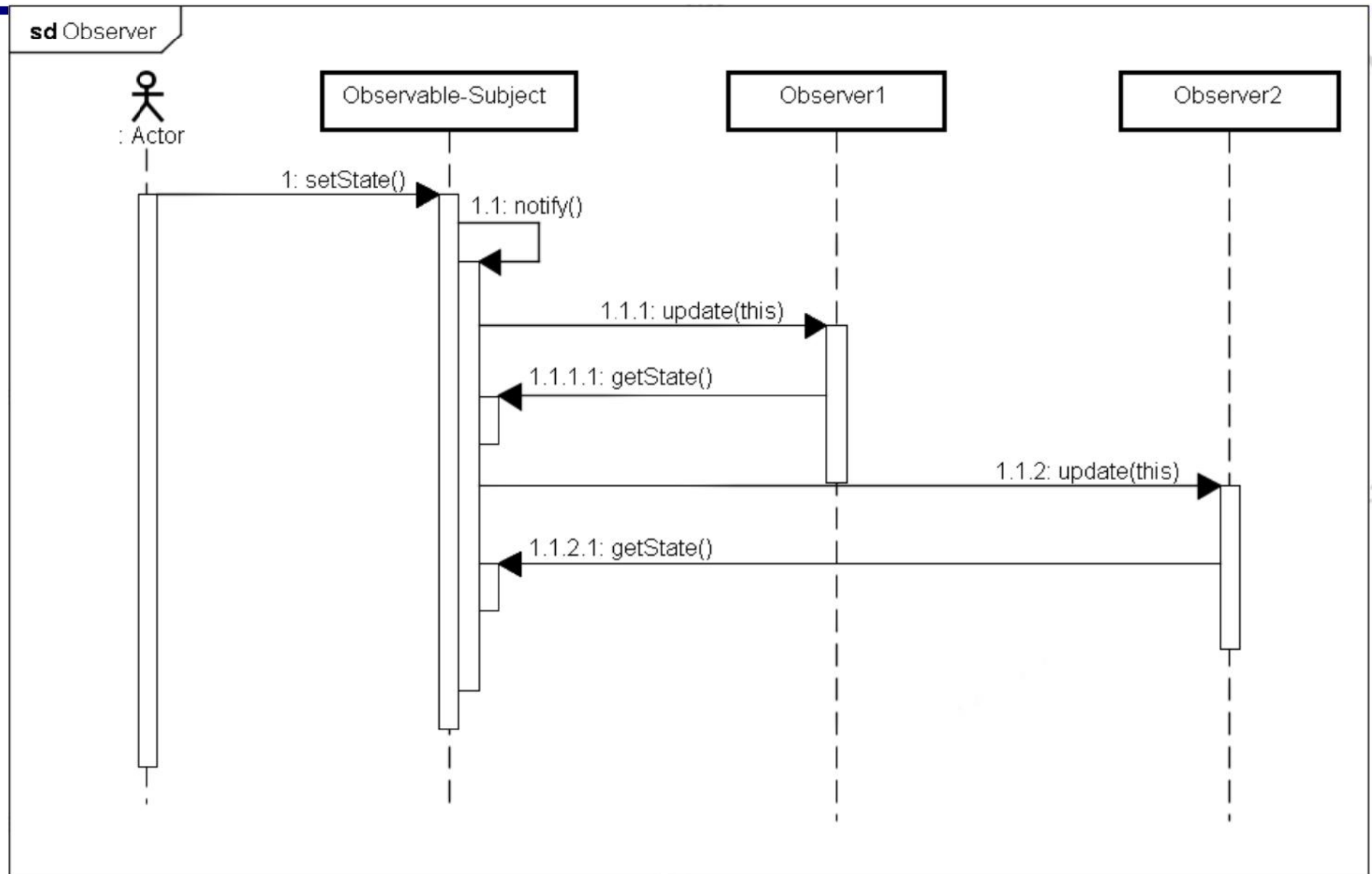


**Define uma interface de atualização.**

**Mantém referência ou recebe referência para o objeto em observação.**

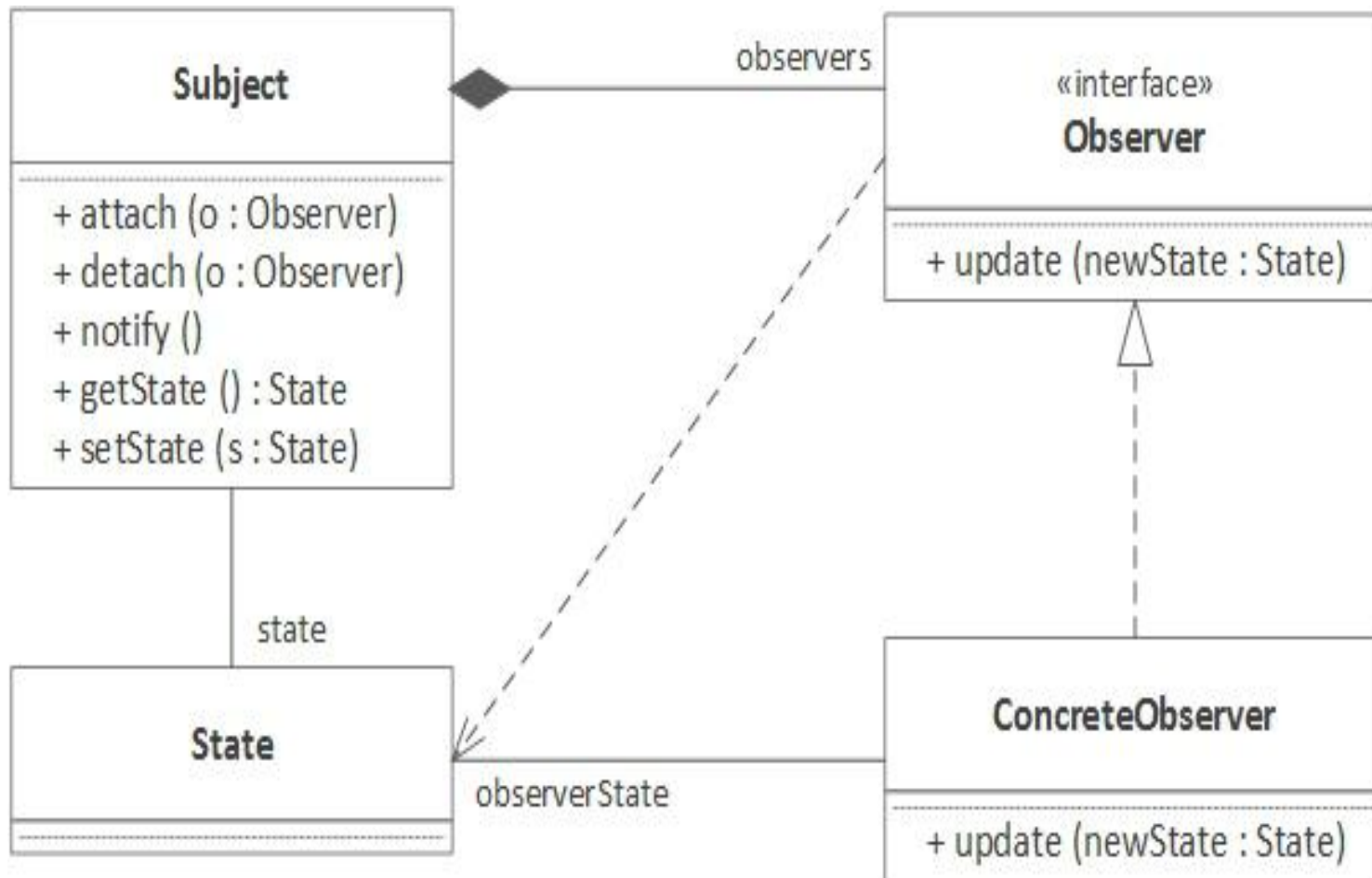
**Implementa a interface de atualização.**

# Padrão – Observer



# Padrão – Observer

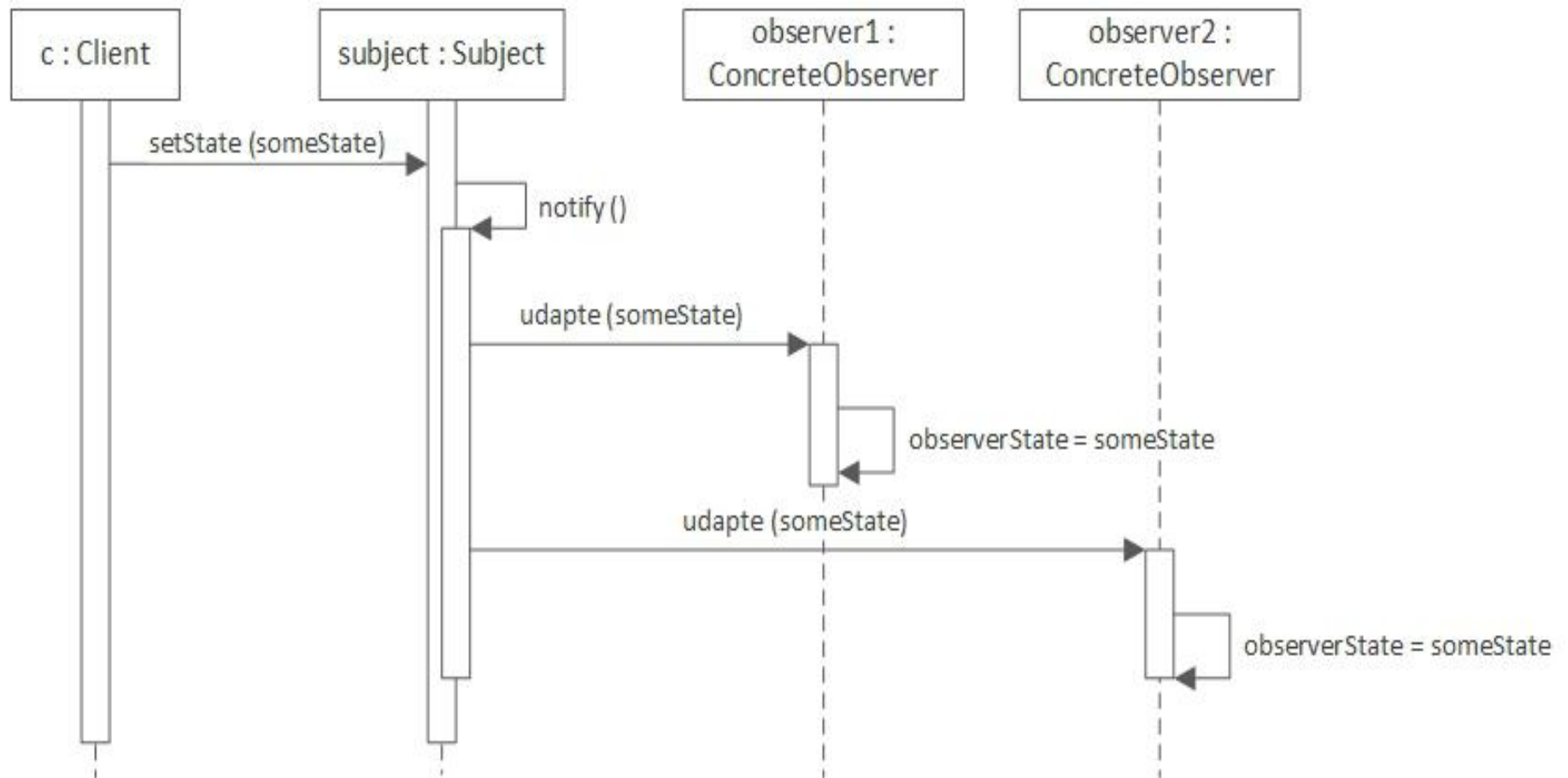
- Outra opção





# Padrão – Observer

- Outra opção



# Padrão – Observer

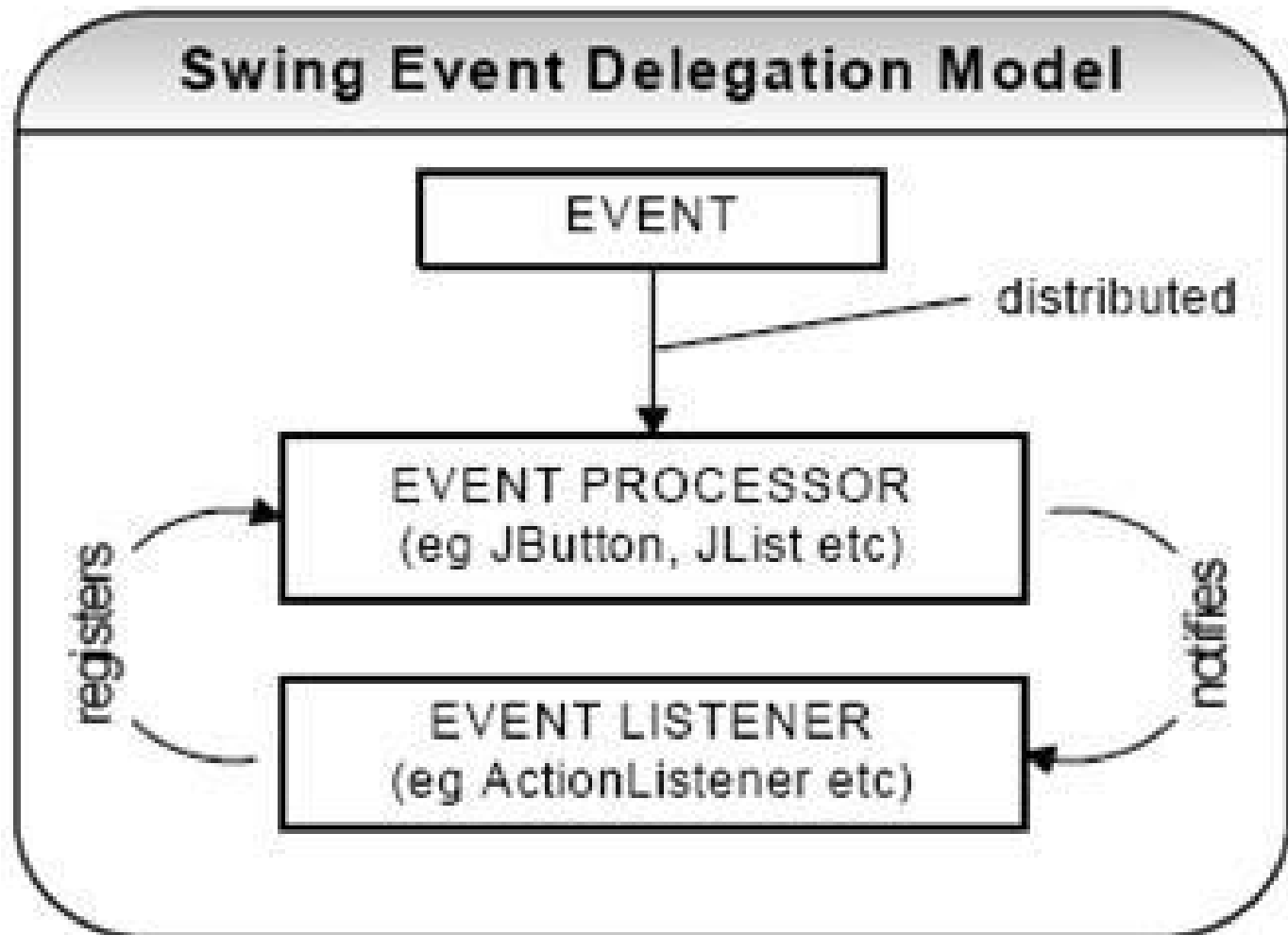
---

- **Aplicabilidade**

- Quando uma mudança em um objeto dispara mudanças em outros objetos, e não se sabe quantos são esses outros objetos
- Quando um objeto necessita notificar outros, sem precisar assumir nada sobre a natureza desses objetos

# Padrão – Observer

- **Exemplo: ActionListener no Java Swing**



# Padrão – Observer

---

- **Outros Exemplos de Uso Comum:**
  - **OnClick, OnMouseOver, OnEnter, OnExit**
  - **GetKeyDown, GetKeyUp, GetKey**
  - **Na Unity**
    - **OnCollisionEnter, OnCollisionExit,**  
**OnTriggerEnter, OnTriggerExit**

# Padrão – Observer

---

- Exercício:
  - O ônibus fornecido pela prefeitura tem horário de chegada variáveis
  - Alunos desejam saber se o ônibus chegou no campus para saber que horário precisam se deslocar para a parada
  - A Prefeitura de Quixadá deseja saber que horário e quantas vezes o ônibus chegou no campus para realizar os pagamentos
  - Faça o modelo usando Observadores

# Padrão – Observer

```
public class Onibus{  
    ...  
    ArrayList observers = new  
        ArrayList<OnibusObserver>();  
  
    public void addObserver(OnibusObserver  
        observer) {  
        this.observers.add(observer);  
    }  
  
    public void notify(){  
        for(OnibusObserver observer: observers){  
            observer.onibusChegou(this);  
        }  
    }  
}
```

```
public interface OnibusObserver{  
    public void onibusChegou(Onibus onibus);  
}  
  
public class Prefeitura implements  
    OnibusObserver{  
    int viagensMes;  
    ...  
    public void onibusChegou(Onibus onibus){  
        this.viagensMes++;  
    }  
}
```

# Padrão – Observer

```
public class Onibus{
    ...
    ArrayList observers = new
        ArrayList<OnibusObserver>();

    public void addObserver(OnibusObserver
        observer) {
        this.observers.add(observer);
    }

    public void notify(){
        for(OnibusObserver observer: observers){
            observer.onibusChegou(this);
        }
    }
}
```

```
public interface OnibusObserver{
    public void onibusChegou(Onibus onibus);
}

public class Prefeitura implements
    OnibusObserver{
    int viagensMes;
    ...
    public void onibusChegou(Onibus onibus){
        this.viagensMes++;
    }
}
```

# Padrão – Observer

---

```
public class Aluno implements OnibusObserver{  
    int viagensMes;  
  
    ...  
  
    public void onibusChegou(Onibus onibus){  
        system.out.println("Onibus Chegou!! Corre  
        pessoa!!");  
    }  
}
```



# Conclusão

---

- Observer

**FUNCIONA COMO UMA INVENÇÃO PURA PARA  
AJUDAR A DESACOPLAR RELAÇÕES 1:N**

# Referências

- **“Padrões de Projeto: Soluções Reutilizáveis...”** Erich Gamma et al. Padrão Observer
- **“Padrão de Projeto em Jogos - Observer”** -  
<http://abrindoojogo.com.br/padroes-de-projeto-em-games-observer>
- <https://www.thiengo.com.br/padrao-de-projeto-observer>
- <https://dzone.com/articles/the-observer-pattern-using-modern-java>