

Padrão de Projeto – Observer – **Alternativo**

====[Classe SUBJECT (Observable)]=====

```
package principal;

import java.beans.PropertyChangeListener;
import java.beans.PropertyChangeSupport;

public class RevistaInformatica {

    private String edicao;

    /* Cria-se um objeto PropertyChangeSupport para apoio ao
     * invés de fazer a herança da classe "Observable" (deprecated).
     */
    PropertyChangeSupport pcs = new PropertyChangeSupport(this);

    public void addObserver(PropertyChangeListener l) {
        pcs.addPropertyChangeListener("theProperty", l);
    }

    public void setProperty(String val) {
        String old = edicao;
        edicao = val;
        pcs.firePropertyChange("theProperty", old, val);
    }
}
```

====[Classe OBSERVER]=====

```
package principal;

import java.beans.PropertyChangeEvent;
import java.beans.PropertyChangeListener;

public class Assinante implements PropertyChangeListener {

    String edicaoNovaRevista;

    @Override
    public void propertyChange(PropertyChangeEvent evt) {
        edicaoNovaRevista = (String) evt.getNewValue();
    }
}
```

====[Classe Main]=====

```
package principal;

public class Main {

    public static void main(String[] args) {
        RevistaInformatica revista = new RevistaInformatica();

        Assinante assinante = new Assinante();
        revista.addObserver(assinante);
    }
}
```

Padrão de Projeto – Observer – **Alternativo**

```
        revista.setProperty("Primeira");  
        System.out.println("Edição: " + assinante.edicaoNovaRevista);  
        revista.setProperty("Segunda");  
        System.out.println("Edição: " + assinante.edicaoNovaRevista);  
        revista.setProperty("Terceira");  
        System.out.println("Edição: " + assinante.edicaoNovaRevista);  
    }  
}
```

Adaptado de: TORCHIANO, Marco. GitHub. Disponível em:
<<https://gist.github.com/mtorchiano/e69ac7e309fee81bd17f4f0740b9ffa9>>. Acesso
em: 23 jan. 2020.