

Seminário TP

Grupo 2 - Gustavo, Ícaro, Mateus, Reinaldo.

Tell, Don't Ask | Builder

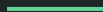
Tell, Don't Ask

O que é o padrão/princípio ?

O princípio é fazer com que dado objeto anuncie um evento, ao invés de monitorar o objeto continuamente até que o determinado evento ocorra.

Para que ele serve?

- Maximizar desempenho
- Evitar desperdício de tempo programacional



Em quais situações devo aplicá-lo?

Quando o seu código estiver esperando um evento que será realizado em um tempo indefinido e dependente de uma situação específica

Exemplo de código

```
public void Example () {  
    button botao = findViewById(R.id.botao) ;  
    botao.setOnClickListener(  
  
        // Manipulação Aqui...  
    ) ;  
}
```

Tell, Don't Ask seria útil para otimizar o código, de modo que o botão anuncie quando o evento `setOnClickListener()`; ao programa. Isso evita que o programa verifique continuamente sobre a ocorrência do evento.

Builder

O que é o
padrão/princípio
?

No padrão de design *Builder* o ato de criação do objeto é removido do objeto propriamente dito. É criado um objeto *builder* para essa função.

Para que ele serve?

- Separar a construção de um objeto de sua representação.
- Simplificar a criação de objetos complexos.



Vantagens

- Ele evita a necessidade de várias subclasses para cada variação do objeto
 - Ele evita que o *constructor* seja muito complexo.
-

Exemplo de código

```
AlertDialog.Builder builder = new AlertDialog.Builder(getActivity());
builder.setTitle("title")
    .setMessage("texto")
    .setPositiveButton(R.string.ok, new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int id) {
            //ação
        }
    })
    .setNegativeButton(R.string.cancel, new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int id) {
            //ação
        }
    });
```

Nesse exemplo se usa a representação “Um diálogo com um botão positivo e um negativo” para, através do builder, criar um objeto complexo

Em quais situações
devo aplicá-lo?

Sempre que a
complexidade do objeto
justifique o uso de um
objeto separado para a sua
construção.

Referências

<https://refactoring.guru/design-patterns/builder>

https://sourcemaking.com/design_patterns/builder

<https://martinfowler.com/bliki/TellDontAsk.html>

<https://pragprog.com/articles/tell-dont-ask>