

Tracking

Objetivo

Receber as requisições dos dispositivos, de forma padronizada que o aplicativo acesse os dados através de um único ponto que utiliza os comandos dos diferentes tipos de sensores para transcrever os dados de forma que o aplicativo entenda.

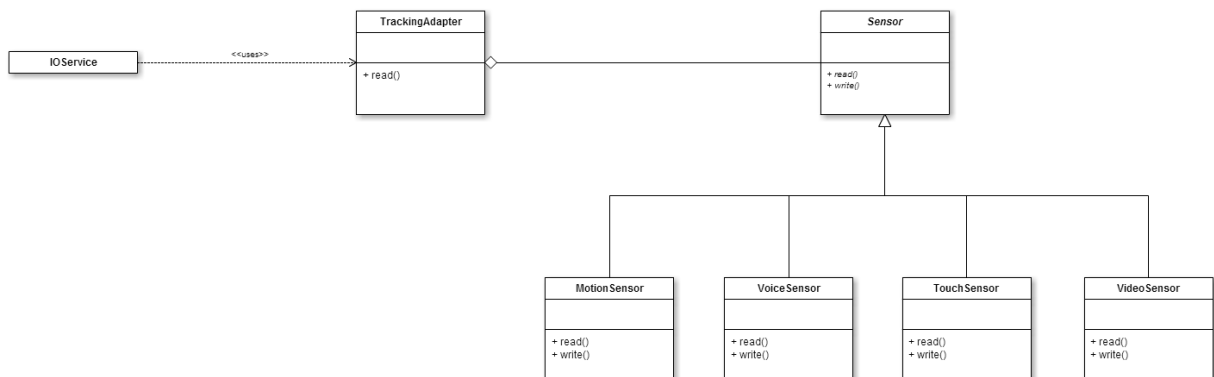
Motivação

Nos dispositivos vestíveis há tipos diferentes de sensores. Para cada tipo de sensor há um tipo de recebimento dos dados. Para manter um controle e organização no recebimento dos dados do dispositivo físico, a proposta deste pattern é o controle dos diferentes tipos de sensores através de uma única classe.

Aplicabilidade

Usar este padrão para receber os dados do sensor físico por um único ponto, que utilizará as diferentes classes de sensores para transcrever a mensagem para que o aplicativo possa utilizá-lo.

Estrutura



Participantes

- **TrackingAdapter**: Classe responsável pelo controle do recebimento dos dados dos sensores.
- **Sensor**: Classe para a comunicação com os sensores. Mais detalhes no pattern Communication.
- **VideoSensor/TouchSensor/VoiceSensor/MotionSensor**: Classes derivadas da **Sensor** para comunicação com cada tipo de sensor físico.
- **IOService**: Classe responsável por controlar o serviço de comunicação de entrada e saída. Mais detalhes no pattern Processing.

Colaborações

O recebimento dos dados do dispositivo físico por uma classe adaptiva, para que o aplicativo tenha a disposição os dados de forma transparente.

Consequência

- Melhor coordenação para a comunicação entre a parte física do dispositivo vestível e a aplicação.
- Transparência no recebimento dos dados do dispositivo físico.
- Os diferentes tipos de sensores são implementados na forma de componentes.
- Refatorável, ajudando a refatorar o código para utilização em outros aplicativos.

Implementação**Exemplo de Código****Usos conhecidos****Patterns Relacionados**

Adapter, Processing.