Class Adapter Pattern

For when you'd like to reuse a particular class but can't due to incompatible interfaces

Lucas Moser

What is a Class Adapter?

- Convert the interface of a class into another interface clients expect.

- Adapter lets classes work together that couldn't otherwise because of incompatible interfaces.

Appropriately Named

"There are only two hard problems in Computer Science: cache invalidation and naming things."

Phil Karlton

General Idea of an Adapter

- AppleTV Adapter Example

- Caveat: This example adapts data. The typical application is to adapt operations of a class.

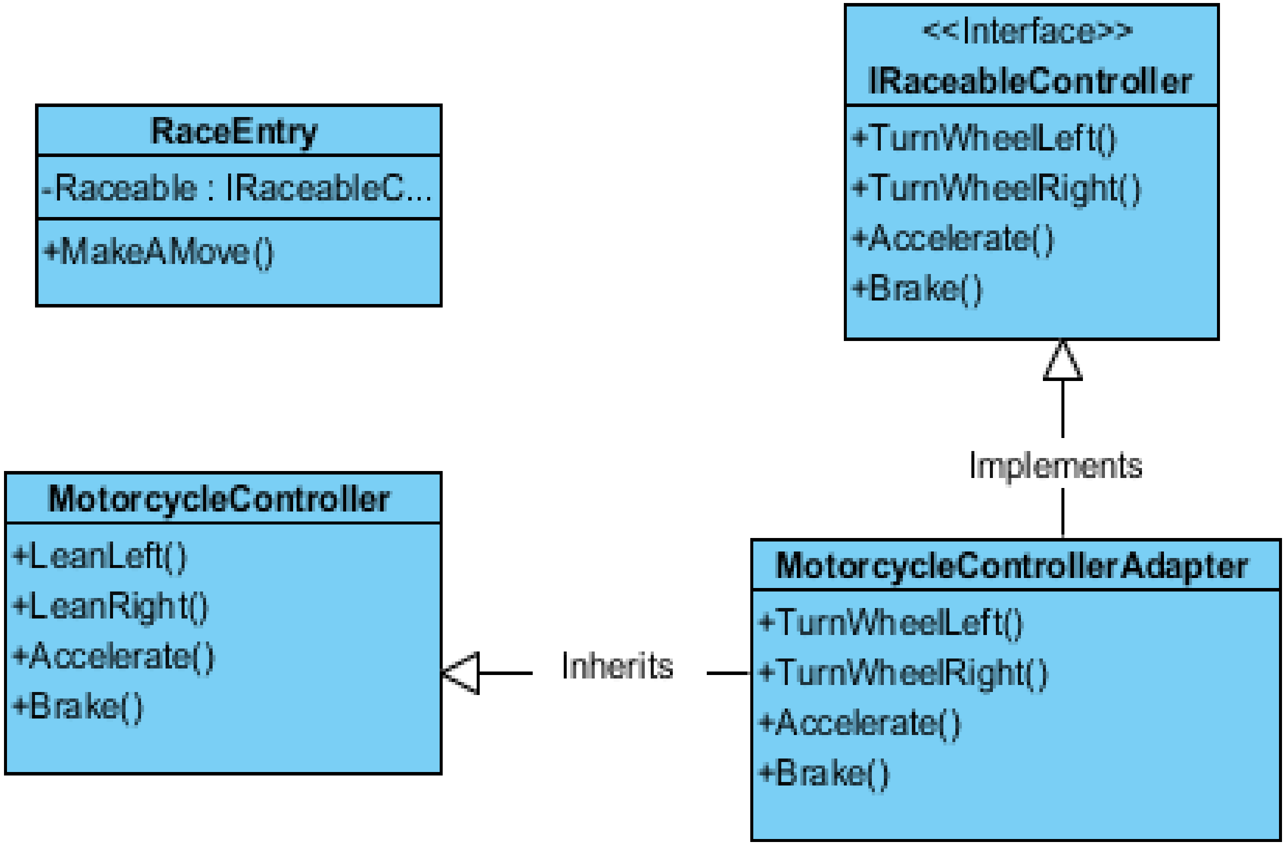
Arcade Example

- You've already created a very successful motorcycle racing arcade game.
- Based on your success your colleague just created a new car racing game.
- You decide the car racing game could be better if you put motorcycles into it.

Arcade Example cont.

Problem: The car racing game's controller configuration is designed for cars, not motorcycles (i.e. you turn left and right by steering wheel rather than leaning).

Arcade Example Solution



```
public class RaceEntry{
    IRaceableController Controller;
    public void MakeMove() {
        Controller.TurnWheelLeft();
        Controller.Accelerate();
        Controller.TurnWheelRight();
        Controller.Brake();
    }
}
public class MotorcycleController {
    public void LeanLeft() { ⋯ }
    public void LeanRight() { ⋯ }
    public void Accelerate() { ⋯ }
    public void Brake() { ... }
}
```

```
public interface IRaceableController {
    public void TurnWheelLeft();
    public void TurnWheelRight();
    public void Accelerate();
    public void Brake();
}
public class
     MotorcycleControllerAdapter
extends MotorcycleController
implements IRaceableController{
    public void TurnWheelLeft(){
        this.LeanLeft();
    }
    public void TurnWheelRight() {
        this.LeanRight();
    }
}
```