

Adapter Pattern

Maxwell Street Polish



Adapter Pattern

pretty much what it sounds like

Maxwell Street Polish

In the real world...

we are very familiar with adapters and what they do.



Max

ish

What about object oriented adapters?

Intent:

Convert the interface of a class into another interface clients expect. Adapter lets classes work together that couldn't otherwise because of incompatible interfaces.

Classified as:

A Structural Pattern

(Structural patterns are concerned with how classes and objects are composed to form larger structures.)

Also Known As:

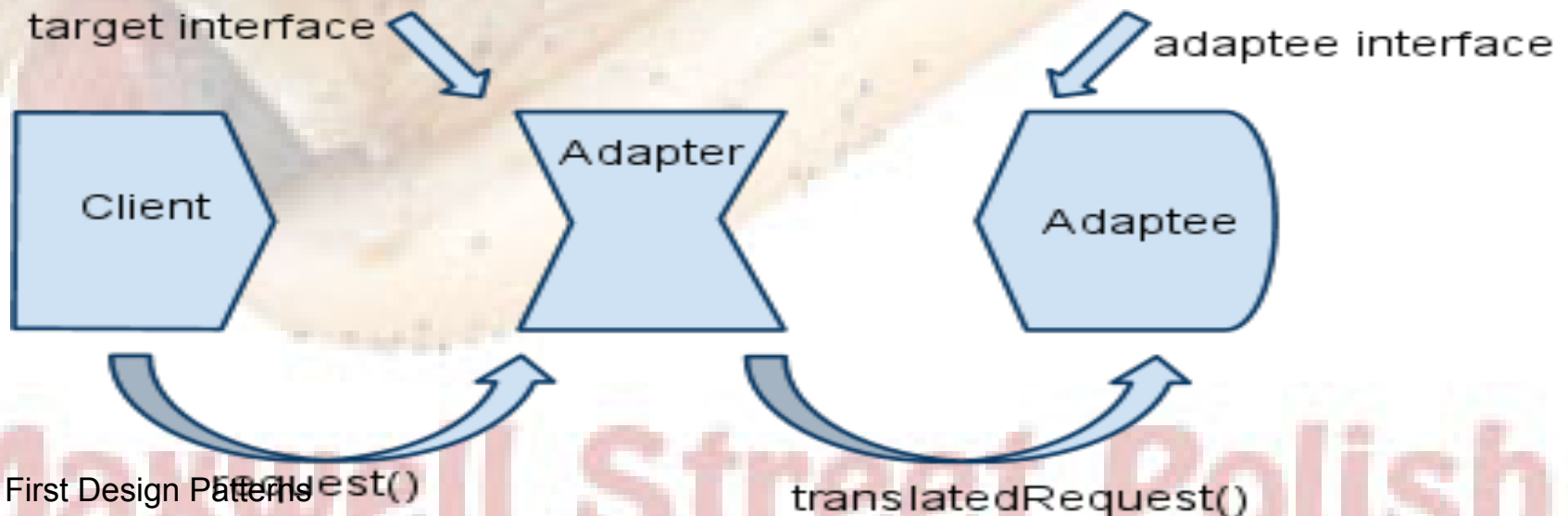
Wrapper

from: Design Patterns: Elements of Reusable Object-Oriented Software

Maxwell Street Polish

How does it work?

1. The client makes a request to the adapter by calling a method on it using the target interface.
2. The adapter translates the request into one or more calls on the adaptee using the adaptee interface.
3. The client receives the results of the call and never knows there is an adapter doing the translation.



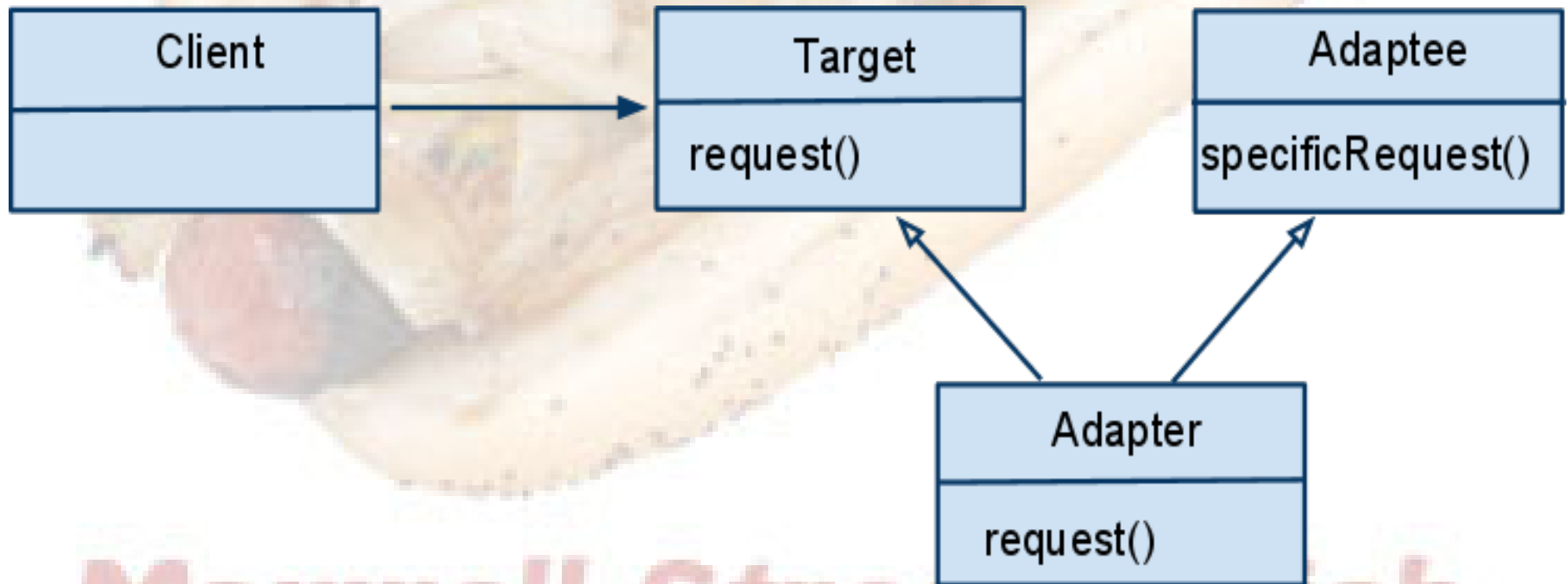
Class Adapter

- adapts Adaptee to Target by committing to a concrete Adapter class. As a consequence, a class adapter won't work when we want to adapt a class and all its subclasses.
- lets Adapter override some of Adaptee's behavior, since Adapter is a subclass of Adaptee.
- introduces only one object, and no additional pointer indirection is needed to get to the adaptee.
- requires multiple inheritance

Maxwell Street Polish

Class Adapter

A class adapter uses multiple inheritance to adapt one interface to another:



Maxwell Street Polish

Applicability

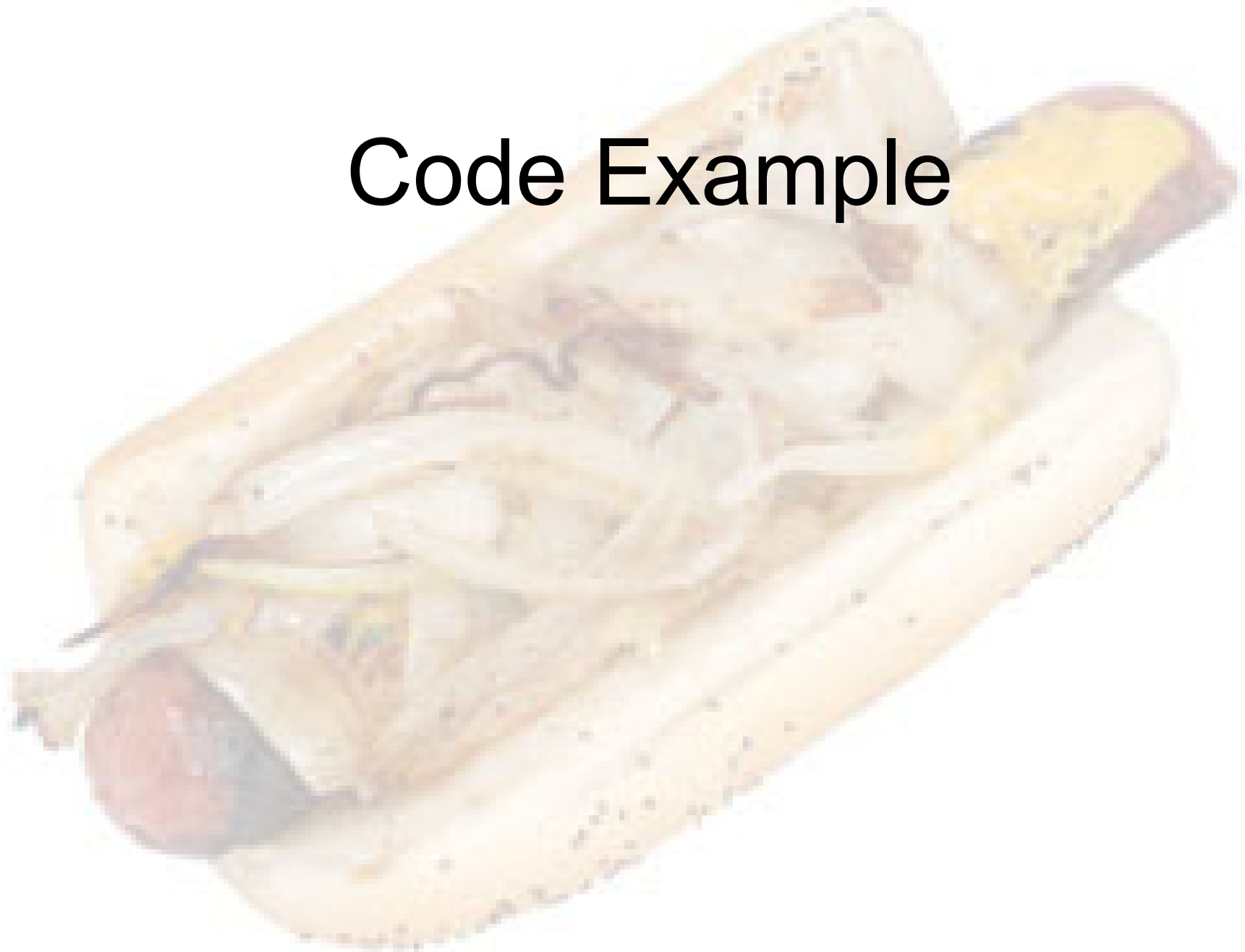


Use the Adapter Pattern when:

- you want to use an existing class, and its interface does not match the one you need.
- you want to create a reusable class that cooperates with unrelated or unforeseen classes, that is, classes that don't necessarily have compatible interfaces.

Maxwell Street Polish

Code Example



Maxwell Street Polish

Code Example

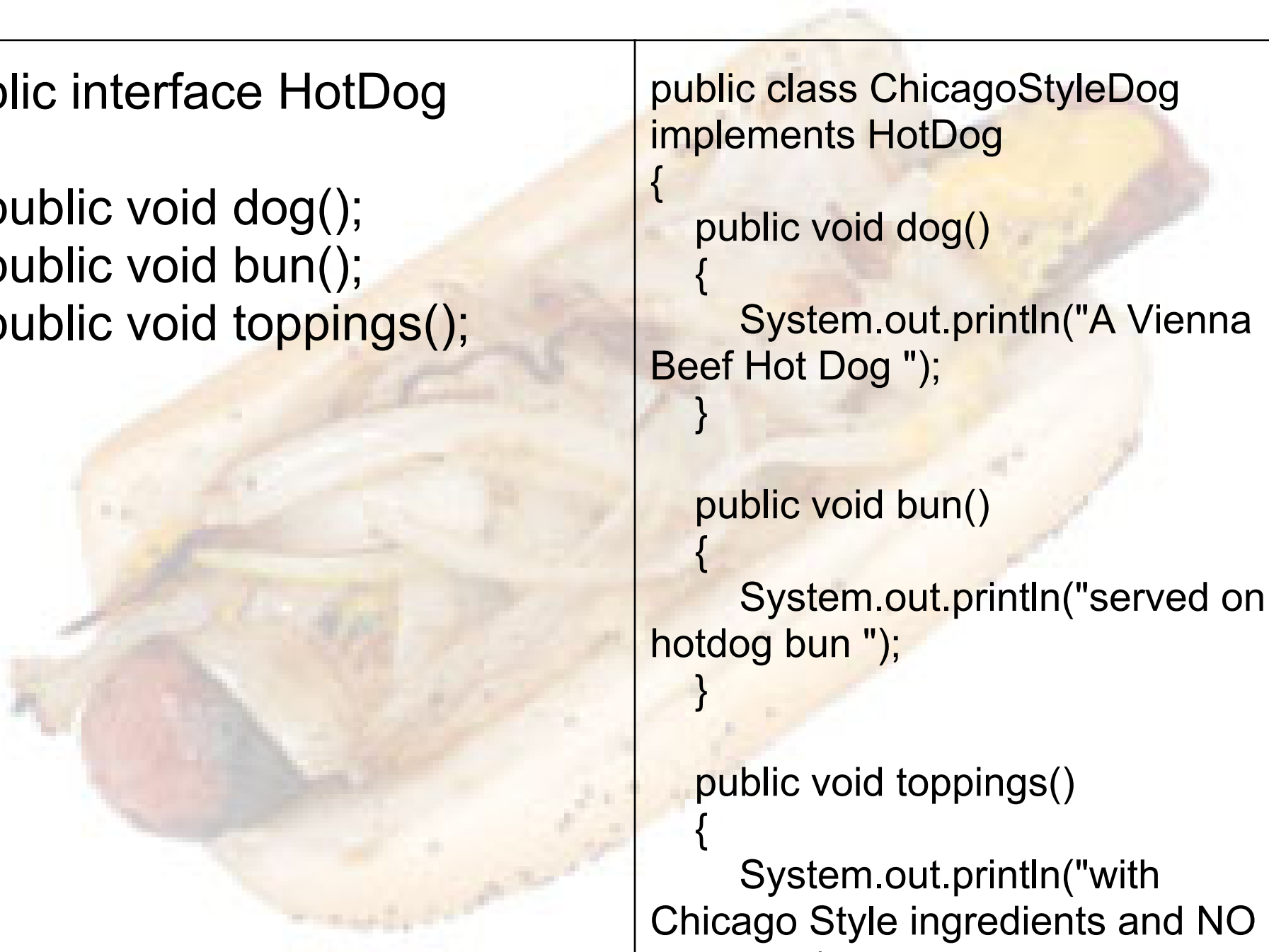


I'm a Chicago hot dog vendor and I buy off the shelf hot dog description software. It's pretty good for describing my world famous Chicago style hot dogs. But I also sell Maxwell Street Polish Sausages.

Maxwell Street Polish



Maxwell Street Polish



```
public interface HotDog
{
    public void dog();
    public void bun();
    public void toppings();
}
```

```
public class ChicagoStyleDog
implements HotDog
{
    public void dog()
    {
        System.out.println("A Vienna
Beef Hot Dog ");
    }

    public void bun()
    {
        System.out.println("served on a
hotdog bun ");
    }

    public void toppings()
    {
        System.out.println("with
Chicago Style ingredients and NO
ketchup!");
    }
}
```

Maxwell Street Polish

```
public class
MaxwellStreetPolish
{
    public void sausage()
    {
        System.out.println("A
Vienna Beef Polish Sausage ");
    }

    public void toppings()
    {
        System.out.println("with
grilled onions and
mustard. Peppers are
optional.");
    }
}
```

```
public class PolishAdapter extends
MaxwellStreetPolish implements
HotDog
{
    public void bun()
    {
        System.out.println("served on a
poppyseed hotdog bun");
    }

    public void dog()
    {
        this.sausage();
    }
}
```

Maxwell Street Polish

```
public class HotDogTest
{
    public static void main(String[]
args)
    {
        // Create a HotDog
        ChicagoStyleDog cshotdog =
new ChicagoStyleDog();

        //Create a Polish
        MaxwellStreetPolish mspolish =
new MaxwellStreetPolish();
        //Test the polish
        System.out.println("If you get a
polish you get:");
        mspolish.sausage();
        mspolish.toppings();

        //Create an adapted polish
        PolishAdapter adaptedPolish =
new PolishAdapter();
        //Test the adapted polish as a
polish
        System.out.println("\nIf you get a
polish you get:");
```

```
        adaptedPolish.sausage();
        adaptedPolish.toppings();

        //test the hotdog
        System.out.println("\nIf you get a
hotdog you get:");
        testHotDog(cshotdog);

        //Now use the polish as a hotdog
        System.out.println("\nIf you get
an adaptedPolish you get:");
        testHotDog(adaptedPolish);
    }
    //Here is the testHotDog method --
it gets a hotdog and calls
    //its methods.
    static void testHotDog(HotDog
hotdog)
    {
        hotdog.dog();
        hotdog.bun();
        hotdog.toppings();
    }
}
```

Output:

If you get a polish you get:

A Vienna Beef Polish Sausage
with grilled onions and mustard. Peppers are optional.

If you get a polish you get:

A Vienna Beef Polish Sausage
with grilled onions and mustard. Peppers are optional.

If you get a hotdog you get:

A Vienna Beef Hot Dog
served on a hotdog bun
with Chicago Style ingredients and NO ketchup!

If you get an adapted Polish you get:

A Vienna Beef Polish Sausage
served on a poppyseed hotdog bun
with grilled onions and mustard. Peppers are optional.

Maxwell Street Polish



Maxwell Street Polish