

# THE COMMAND PATTERN

# PROGRAMMING PARADIGMS

OLD-SCHOOL LANGUAGES LIKE C

## IMPERATIVE PROGRAMMING

VARIABLES, LISTS, DICTIONARIES, FOR-LOOPS

## FUNCTIONAL PROGRAMMING

FUNCTIONS CALLING FUNCTIONS CALLING FUNCTIONS

## OBJECT-ORIENTED PROGRAMMING

CLASSES DEFINE INTERFACES; OBJECTS  
INSTANTIATE CLASSES; OBJECTS HAVE  
STATE (MEMBERS) AND BEHAVIOUR (METHODS)

EXCEL (YEP IN MANY WAYS  
EXCEL IS A PROGRAMMING LANGUAGE  
TOO! JUST NO FOR LOOPS)

JAVA, C++, C#

THE LINES BETWEEN THESE PROGRAMMING PARADIGMS  
CAN SOMETIMES GET A BIT BLURRY

CONSIDER A CLASS  
THAT HAS JUST 1  
METHOD

IT ENCAPSULATES ALL  
THE STATE NEEDED  
FOR THAT ONE METHOD  
TO DO ITS THING

# PROGRAMMING PARADIGMS

OLD-SCHOOL LANGUAGES LIKE C

## IMPERATIVE PROGRAMMING

VARIABLES, LISTS, DICTIONARIES, FOR-LOOPS

## FUNCTIONAL PROGRAMMING

FUNCTIONS CALLING FUNCTIONS CALLING FUNCTIONS

## OBJECT-ORIENTED PROGRAMMING

CLASSES DEFINE INTERFACES; OBJECTS  
INSTANTIATE CLASSES; OBJECTS HAVE  
STATE (MEMBERS) AND BEHAVIOUR (METHODS)

EXCEL (YEP IN MANY WAYS  
EXCEL IS A PROGRAMMING LANGUAGE  
TOO! JUST NO FOR LOOPS)

JAVA, C++, C#

THE LINES BETWEEN THESE PROGRAMMING PARADIGMS  
CAN SOMETIMES GET A BIT BLURRY

CONSIDER A CLASS  
THAT HAS JUST 1  
METHOD

IT ENCAPSULATES ALL  
THE STATE NEEDED  
FOR THAT ONE METHOD  
TO DO ITS THING

IS SUCH A CLASS REALLY A CLASS,  
OR IS IT REALLY A FUNCTION?

# COMMAND OBJECTS

LIE AT THE HEART OF THE COMMAND PATTERN

A COMMAND OBJECT HAS A SINGLE METHOD, AND WHATEVER STATE IS NEEDED FOR THAT METHOD TO DO ITS THING

YOU CAN THINK OF THE COMMAND OBJECT AS A METHOD THAT CAN BE PASSED AROUND LIKE AN OBJECT



A COMMAND OBJECT SEPARATES AN ACTION FROM THE EXECUTION OF THAT ACTION

FOR THAT REASON, COMMAND OBJECTS ARE ALSO OFTEN CALLED ACTION OBJECTS



# A COMMAND OBJECT SEPARATES THE EXECUTION OF AN ACTION FROM THE ACTION ITSELF

THIS IDEA IS A CRUCIAL ONE

CONSIDER A SIMPLE USER INTERFACE

WHEN A USER CLICKS  
THE BUTTON..

THE DISPLAY SHOWS  
A MESSAGE



WE DON'T KNOW WHEN THE USER  
WILL CLICK THE BUTTON, BUT WE DO  
KNOW THE ACTION THAT WE WOULD  
LIKE TO TAKE WHEN THE BUTTON IS  
CLICKED

## A COMMAND OBJECT IS JUST WHAT WE NEED

# THE CLIENT

THE USER WHO  
TRIGGERS THE  
EXECUTION OF  
THE ACTION

HAS AN OFFICIAL NAME IN THE  
COMMAND PATTERN

# THE RECEIVER

THE DISPLAY AREA  
WHICH IS ACTED  
UPON BY THE  
ACTION

# THE INVOKER

THE BUTTON WHICH  
IS CLICKED BY THE  
CLIENT TO TRIGGER  
EXECUTION OF THE  
ACTION

THE TRIGGER THAT CAUSES THE ACTION  
TO BE EXECUTED IS CALLED

# AN EVENT

UI ELEMENTS "PUBLISH" A WIDE  
RANGE OF EVENTS, AND OUR CODE  
CAN "SUBSCRIBE" TO THOSE EVENTS

THIS IS PRECISELY THE  
OBSERVER DESIGN PATTERN

THE TRIGGER THAT CAUSES THE ACTION  
TO BE EXECUTED IS CALLED

**AN EVENT**

UI ELEMENTS "PUBLISH" A WIDE  
RANGE OF EVENTS, AND OUR CODE  
CAN "SUBSCRIBE" TO THOSE EVENTS

THIS IS PRECISELY THE  
OBSERVER DESIGN PATTERN

**THIS INTERPLAY OF THE OBSERVER  
AND COMMAND PATTERNS IS THE  
BEDROCK OF MOST UI PROGRAMMING**