

THE OBSERVER PATTERN IN ACTION: FRAMEWORKS

IF YOU WANT TO TRAVEL A SHORT DISTANCE, YOU **WALK**

IF YOU WANT TO TRAVEL REALLY FAR, YOU **FLY**

IF YOU WANT TO CODE UP SOMETHING SMALL,
YOU CODE IT UP FROM **SCRATCH**

IF YOU WANT TO CODE UP SOMETHING BIG, YOU

USE A FRAMEWORK

USE A FRAMEWORK

FRAMEWORKS ARE COMPLICATED COLLECTIONS OF
INTERCONNECTED CLASSES

TO FLY A PLANE..

DRESS THE PART

GET THE LITTLE DETAILS RIGHT
AND TRUST THE PLANE TO DO
THE REST

"LISTEN" TO THE PLANE - WATCH THE CONTROLS
PAY ATTENTION TO THE ENVIRONMENT, AND
RESPOND ACCORDINGLY

TO USE A FRAMEWORK..

DRESS THE PART..IMPLEMENT INTERFACES AND
OVERRIDE CLASS METHODS AS YOU NEED TO

GET THE LITTLE DETAILS RIGHT - PUT IN SOME BITS
OF BOILERPLATE CODE, AND TRUST THE FRAMEWORK
TO DO THE REST

"LISTEN" TO THE FRAMEWORK BY SETING UP
LISTENERS ON EVENTS IN THE FRAMEWORK
AND RESPOND ACCORDINGLY

WE HAVE USED THE TERMS "EVENT" AND
"LISTEN" A FEW TIMES, AND IT BEARS
REPEATING -

LISTENING ON EVENTS IS AN IMPORTANT PART OF USING FRAMEWORKS CORRECTLY

A FRAMEWORK IS A COMPLICATED COLLECTION OF INTERCONNECTED CLASSES

YOU DO LITTLE BITS OF BOILERPLATE STUFF TO GET THE FRAMEWORK DOING ITS THING..

..AND THEN WAIT FOR THE FRAMEWORK TO SAY TO YOU - "THERE IS THIS SPECIFIC BIT THAT ONLY YOU CAN TAKE CARE OF - HERE IT IS FOR YOUR ATTENTION"

THE FRAMEWORK DOING ITS THING..

THAT SPECIFIC BIT IS CALLED AN

EVENT

AND THE PLACE WHERE YOU, THE PROGRAMMER,
STAND BY WAITING FOR THE EVENT AND DECIDE HOW TO
REACT TO THE EVENT IS CALLED

..AND THEN WAIT FOR THE FRAMEWORK
TO SAY TO YOU - "THERE IS THIS SPECIFIC
BIT THAT ONLY YOU CAN TAKE CARE OF -
HERE IT IS FOR YOUR ATTENTION"

THE LISTENER

NOW, THE FRAMEWORK DOES NOT KNOW
WHAT SPECIFIC EVENTS ARE IMPORTANT
AND NEED YOUR INTERVENTION, SO IT
PROVIDES A WAY FOR YOU TO LISTEN TO
WHATEVER EVENTS MATTER TO YOU

"REGISTER TO LISTEN
ON AN EVENT"

IS HOW THIS IS PROCESS
OF SIGNING UP FOR UPDATES
IS DESCRIBED

THIS ANONYMOUS INNER CLASS
IMPLEMENTS SOME INTERFACE
SO THAT THE FRAMEWORK KNOWS
HOW TO PASS THE EVENT OBJECT
TO THE EVENT LISTENER

WHEN THE EVENT ACTUALLY OCCURS,
THE FRAMEWORK WILL CALL THE LISTENER
CODE AND PASS IT AN OBJECT WITH THE
DETAILS OF WHAT JUST HAPPENED -

AN EVENT
OBJECT

THIS IS A PRECURSOR TO

THE OBSERVER PATTERN

**THE OBSERVER PATTERN IN ACTION:
OBSERVER + MVC = UI MAGIC**

ARCHITECTURAL PATTERNS, LIKE MVC, ARE POWERFUL TOOLS IN DESIGNING SOFTWARE - THEY ARE IN EFFECT 'SET PIECES' THAT ALLOW PROGRAMMERS TO AVOID STRUGGLING WITH SOLVED PROBLEMS

