THE ADAPTER PATTERN

LET'S CONSIDER A MORE SUBSTANTIVE EXAMPLE

MANY LANGUAGES HAVE POWERFUL SUPPORT FOR VISUALIZING DATA IN THE FORM OF CHARTS AND TABLES

IN JAVA, THE NEW JAVAFX FRAMEWORK HAS EXCELLENT TABLE AND CHART CLASSES

ON THE ONE HAND

BOTH CHARTS AND TABLES TEND TO NEED DATA IN A SPECIFIC FORMAT – EITHER IN ROWS AND COLUMNS, OR AS A SERIES OF POINTS

ON THE OTHER HAND

DATA IS ALMOST NEVER READILY AVAILABLE IN THE FORM OF ROWS OR COLUMNS OR POINTS – IT IS USUALLY IN LISTS OR MAPS OF OBJECTS

IN JAVA, THE NEW JAVAFX FRAMEWORK HAS EXCELLENT TABLE AND CHART CLASSES

ON THE ONE HAND

BOTH CHARTS AND TABLES TEND TO NEED DATA IN A SPECIFIC FORMAT - EITHER IN ROWS AND COLUMNS, OR AS A SERIES OF POINTS

ON THE OTHER HAND

DATA IS ALMOST NEVER READILY AVAILABLE IN THE FORM OF ROWS OR COLUMNS OR POINTS - IT IS USUALLY IN LISTS OR MAPS OF OBJECTS

THIS CONVERSION FROM LISTS OR MAPS OF OBJECTS INTO ROWS AND COLUMNS OR POINTS INVARIABLY REQUIRES THE USE OF

ADAPTERS

ADAPTERS IN CODE ARE PRETTY SIMILAR -

AN ADAPTER TAKES IN AN OBJECT
THAT IMPLEMENTS AN INTERFACE,
AND GIVES OUT AN OBJECT THAT
IMPLEMENTS A DIFFERENT INTERFACE

WHY IS THIS ADAPTER USEFUL?

BECAUSE IN JAVA THERE ALL KINDS
OF COOL UTILITIES FOR WORKING WITH
LISTS -

WAY MORE POWERFUL THAN THOSE
AVAILABLE FOR WORKING WITH ARRAYS