THE ITERATOR PATTERN

ITS IMPORTANT TO REALIZE THAT IN JAVA, THE ITERATOR IS A SEPARATE OBJECT FROM THE COLLECTION

BUT IN GENERAL, THE COLLECTION ITSELF MIGHT BE AN ITERATOR - THAT'S QUITE OK AS WELL

THIS LEADS TO THE DIFFERENCE BETWEEN "INTERNAL" AND "EXTERNAL" ITERATORS"

THE COLLECTION IS `Iterable<T>`

AND PROVIDES A WAY TO GET AN `Iterator<T>`

EXTERNAL ITERATORS HAVE THE ADVANTAGE THAT BECAUSE THE ITERATOR SITS OUTSIDE THE COLLECTION, ITS EASY ENOUGH TO DEFINE DIFFERENT ITERATORS FOR A COLLECTION

FORWARD AND REVERSE ITERATORS ARE QUITE STANDARD IN LIBRARIES SUCH AS THE STANDARD TEMPLATE LIBRARY IN C++

THE JAVA ITERATOR
INTERFACE ALLOWS REMOVAL -

REMOVAL

SO ITS POSSIBLE FOR ONE ITERATOR TO
BE WALKING OVER A COLLECTION,
WHILE ANOTHER ITERATOR ON
ANOTHER THREAD REMOVES
AN ELEMENT..

IN GENERAL, THREADING AND
CONCURRENCY ISSUES RELATED
TO CONTAINERS ARE QUITE
COMPLICATED

ITERATORS IN PYTHON

AS USUAL IN THIS CLASS, WE HAVE FOCUSED ON JAVA'S ITERATORS -

BUT THE USE OF ITERATORS IN PYTHON IS WORTH A SPECIAL MENTION

PYTHON IS SOMEWHAT FUNCTIONAL, WHILE JAVA IS CLEARLY OBJECT-ORIENTED

THIS MEANS THAT PYTHON CODE TENDS TO HAVE FAR FEWER FOR LOOPS, AND FAR MORE LAMBDA FUNCTIONS APPLIED TO LISTS

IT IS THE RICH ITERATOR SUPPORT IN THE PYTHON LANGUAGE THAT MAKES THIS POSSIBLE