A CLASS ADHERES TO THE PROTOTYPE DESIGN PATTERN WHEN –

AN OBJECT OF THAT CLASS CAN BE CREATED AS A CLONE OF ANOTHER OBJECT OF THAT CLASS

PRACTICALLY, THAT MEANS THAT THE CLASS HAS A CONSTRUCTOR THAT TAKES IN ANOTHER OBJECT OF THE SAME CLASS

(SUCH A CONSTRUCTOR IS CALLED A COPY CONSTRUCTOR)

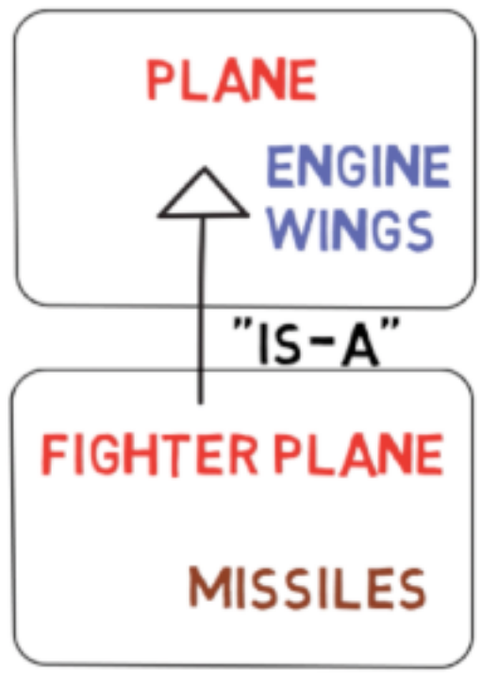
IN JAVA, THIS FUNCTIONALITY IS EFFECTED NOT VIA A CONSTRUCTOR, BUT RATHER VIA THE CLONE METHOD OF THE INTERFACE Cloneable

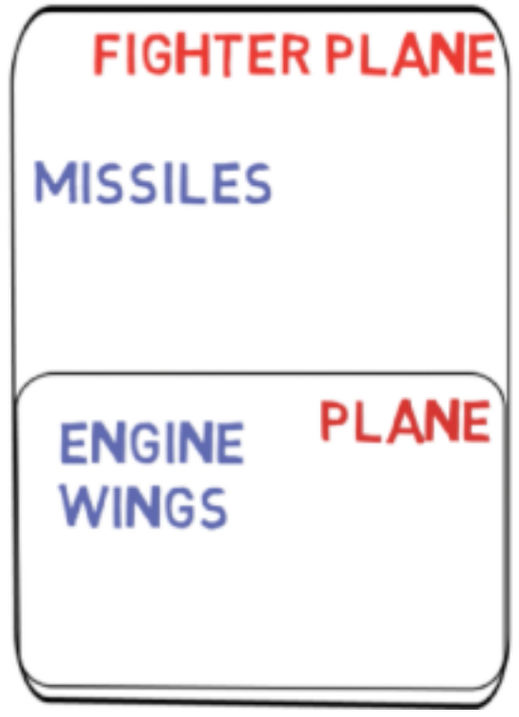
IN ORDER TO CLONE AN OBJECT, YOU NEED TO COMPLETE 2 STEPS

1. CLONE THE PARENT PORTION OF THE OBJECT
2. CLONE EACH MEMBER VARIABLE OF THIS OBJECT

BOTH THESE STEPS HAVE SOME SUBTLETIES, SO ITS WORTH DISCUSSING THEM IN SOME DETAIL

WHEN A CLASS INHERITS FROM ANOTHER CLASS, EVERY OBJECT OF THE CHILD CLASS HAS AN ACTUAL COPY OF THE PARENT INSIDE IT

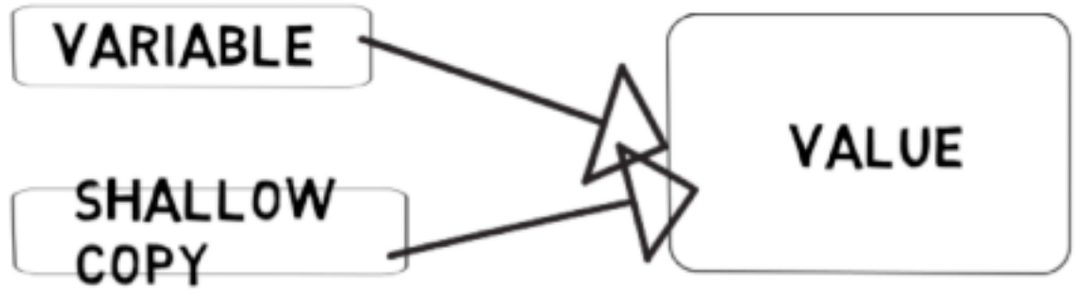




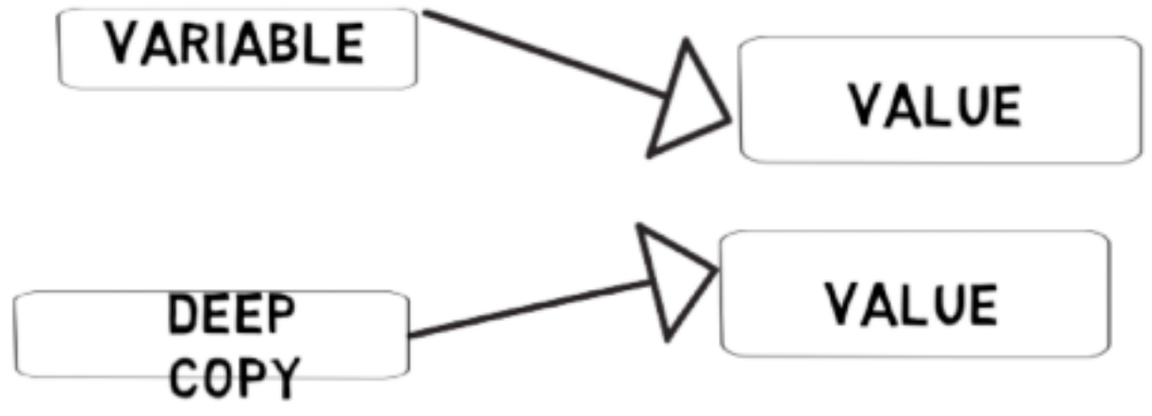
CLONING EACH MEMBER VARIABLE OF AN OBJECT IS NOT TRIVIAL EITHER

THERE ARE TWO TYPES OF COPY POSSIBLE IN A LANGUAGE LIKE JAVA

SHALLOW COPY



DEEP COPY



IN GENERAL, THE JAVA CLONEABLE INTERFACE IS NOT VERY HIGHLY REGARDED

1. THE .CLONE METHOD BELONGS IN THE OBJECT CLASS (I.E. EVERY OBJECT HAS THIS METHOD), WHEN IT OUGHT TO HAVE BELONGED IN THE CLONEABLE INTERFACE

2. THE IMPLICATION OF (1) IS THAT ALL OBJECTS HAVE A CLONE METHOD, BUT IF YOU TRY TO CLONE AN OBJECT THAT DOES NOT IMPLEMENT CLONEABLE, A NOTCLONEABLE EXCEPTION IS THROWN

3. CLONE() IS EFFECTIVELY A COPY CONSTRUCTOR, EXCEPT THAT ITS NOT SET UP AS A CONSTRUCTOR (AS IT IS IN C++ AND OTHER LANGUAGES)

THE CLONE METHOD SHOULD BE IN THE CLONEABLE INTERFACE!