INTERFACES VERSUS ABSTRACT BASE CLASSES

AN INTERFACE IS A CLASS WITH ONLY METHOD SIGNATURES AND NO METHOD IMPLEMENTATIONS

AN ABSTRACT BASE CLASS IS A CLASS WITH AT LEAST ONE METHOD THAT IS ENTIRELY MISSING IMPLEMENTATION

INTERFACES AND ABSTRACT BASE CLASSES ARE PRETTY SIMILAR..

NEITHER CAN BE INSTANTIATED DIRECTLY, ONLY THROUGH THE OBJECTS OF SOME CLASS THAT INHERITS FROM THEM

BOTH OF THEM DRIVE BEHAVIOUR VIA "IS-A" INHERITANCE

BUT THEY ARE ALSO DIFFERENT IN IMPORTANT WAYS

JAVA SYNTAX: A BASE CLASS "IMPLEMENTS" AN INTERFACE, BUT "EXTENDS" A BASE CLASS

```
public class MySquare extends MyRectangle {

public class MyCircle implements IShape {
```

A JAVA CLASS CAN IMPLEMENT ANY NUMBER OF INTERFACES, BUT EXTEND ONLY ONE ABSTRACT BASE CLASS

INTERFACES VERSUS ABSTRACT BASE CLASSES

AN INTERFACE IS A CLASS WITH ONLY METHOD SIGNATURES AND NO METHOD IMPLEMENTATIONS

AN ABSTRACT BASE CLASS IS A CLASS WITH AT LEAST ONE METHOD THAT IS ENTIRELY MISSING IMPLEMENTATION

THERE ARE OTHER LESS IMPORTANT DIFFERENCES TOO

INTERFACES AND ABSTRACT BASE CLASSES ARE PRETTY SIMILAR..

NEITHER CAN BE INSTANTIATED DIRECTLY, ONLY THROUGH THE OBJECTS OF SOME CLASS THAT INHERITS FROM THEM

BOTH OF THEM DRIVE BEHAVIOUR VIA "IS-A" INHERITANCE

BASICALLY USE AN ABSTRACT CLASS TO PROVIDE DEFAULT FUNCTIONALITY

BUT THEY ARE ALSO DIFFERENT IN IMPORTANT WAYS

JAVA SYNTAX: A BASE CLASS "IMPLEMENTS" AN INTERFACE, BUT "EXTENDS" A BASE CLASS

```
public class MySquare extends MyRectangle {

public class MyCircle implements IShape {
```

(INTERFACE METHODS ARE IMPLICITLY PUBLIC BY DEFAULT, MEMBER VARIABLES IN AN INTERFACE ARE IMPLICITLY FINAL BY DEFAULT, ETC..)

A JAVA CLASS CAN IMPLEMENT ANY NUMBER OF INTERFACES, BUT EXTEND ONLY ONE ABSTRACT BASE CLASS