Object-Oriented Programming (CS F213)

Module III: Inheritance and Polymorphism in Java

CS F213 RL 11.1: Java Type System

BITS Pilani

Dr. Pankaj Vyas
Department of Computer Science, BITS-Pilani, Pilani Campus

CS F213 RL 11.1 : Topics

- Java Type System

Java Type System

- **What is type in Java?**

    Set of Values with a set of Operations that can be applied to the values.

Example :
int type   -→ **Set of values { [-2^31] to [+2^31-1])}**
         Operations : All Arithmetic Operations

BOX type   -→ **Set of all BOX object references**
         Operations : area() , volume() , toString()

❑ **Java is Strongly Typed Language.**

       Type checking is done both at Compile and Run Time.

- **If a Type Check Fails at Compile Time → Compile Time Error**

- **If a Type Check Fails at Run Time → *RunTime Exception***

Various Types in Java

1. *Primitive Type(int, float, double, byte, char, boolean, short, long)*

2. *A class Type [BOX, String, Arrays , Student etc]*

3. *An interface Type [Either User Defined OR Libarary interfaces such as Comparable]*

4. *An array type*

5. *null type*

Note :

1. *Arrays have a component type (Type of array elements) e.g String[] array has component type as String int[] array has Component type as int*

2. *void is not a type in java. [only a keyword to denote that a method does not return anything]*

3. *Object[] names = new String[10]; What's Array Type and its component type.*

Various Types of Values in Java

1. **A Value of Primitive type**
2. **A reference to an object**
3. **reference to an array**
4. **Null**
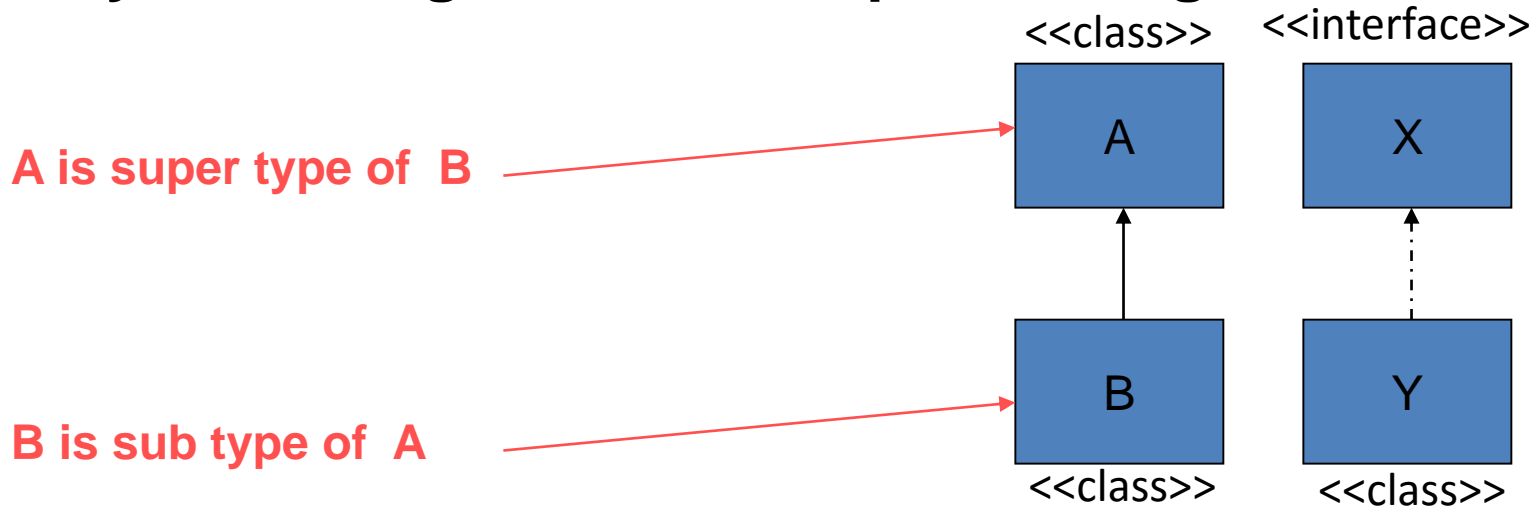
Examples :

    10 , 10.35 , true **[Primitive Type Values]**

      new BOX(10,6,8); **[Class Type Value]**

      new int[] { 10,6,8} **[Array Type value]**

            null **[null Type Value]**

   Note : You can not have a value of type interface.

What are Sub-Types in Java?

Sub type specifies the inheritance relationship either by extending a class or implementing an interface

<<class>>   <<interface>>

A is super type of B

A   X

B is sub type of A

B   Y

<<class>>   <<class>>

Similarly X is super type for Y and Y is sub type for X.

You can substitute a value of subtype whenever supertype value is expected

Java Type System : Example

What's Expected

X   x1   = new   ?

x1 is reference variable of type X and 'X' is not primitive type?

If X is an interface Then

      RHS can be an instance of any class implementing the interface X.

If X is an abstract class Then

      RHS can be an instance of any concrete (complete) subclass of X

If X is a concrete (complete) class Then

      RHS can be either an instance of X or any of its concrete subclass

Rules for Subtype Relationships
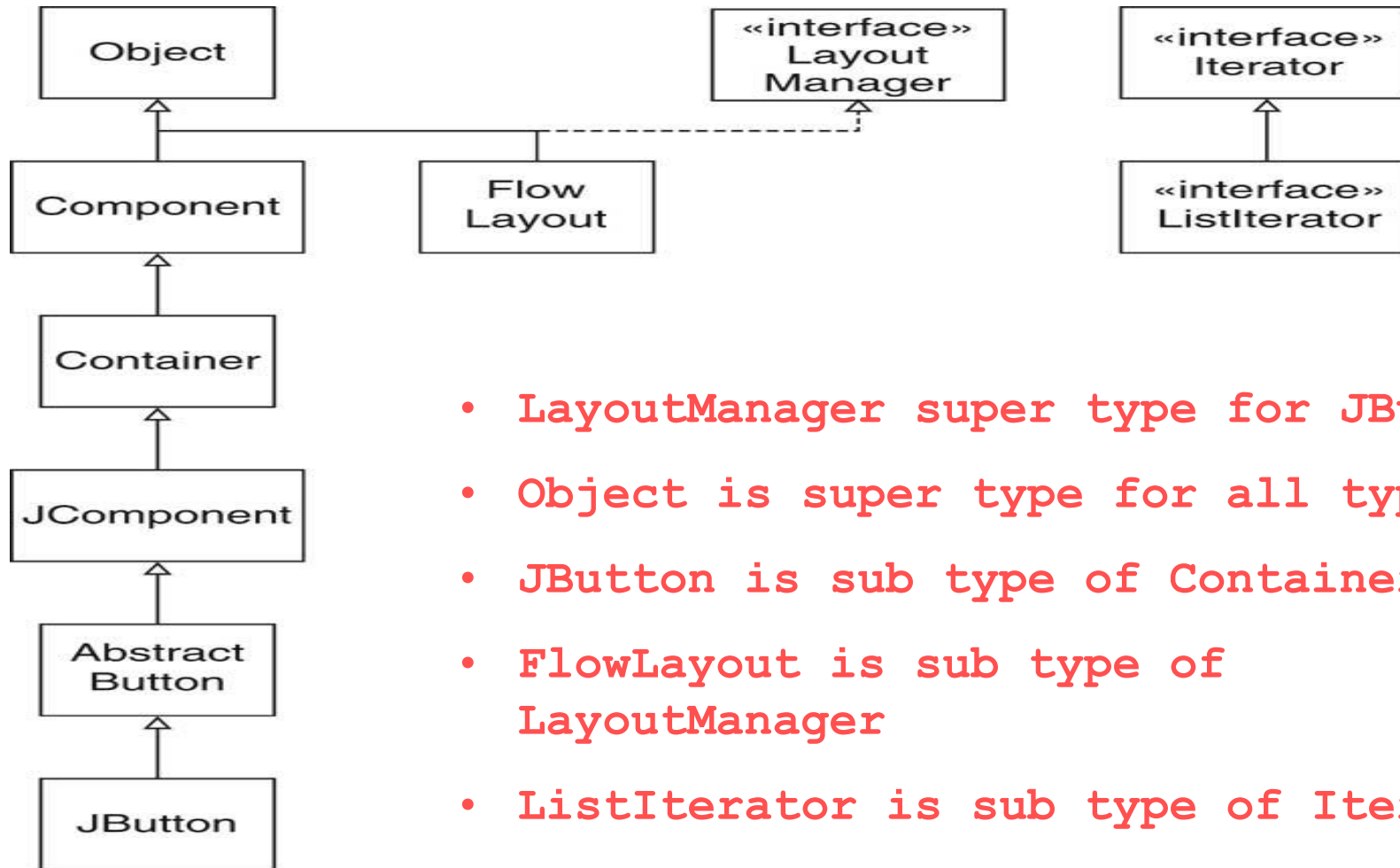
S is a subtype of T if

1. S and T are the same type
2. S and T are both class types, and T is a direct or indirect superclass of S
3. S is a class type, T is an interface type, and S or one of its superclasses implements T
4. S and T are both interface types,and T is a direct or indirect superinterface of S
5. S and T are both array types,and the component type of S is a subtype of the component type of T
6. S is not a primitive type and T is the type Object
7. S is an array type and T is Cloneable or Serializable
8. S is the null type and T is not a primitive type

Sub-Types : Examples

- **LayoutManager super type for JButton**

- **Object is super type for all types**

- **JButton is sub type of Container**

- **FlowLayout is sub type of LayoutManager**

- **ListIterator is sub type of Iterator**

Sub-Types : Examples …

1. Is Container is a subtype of Component ?.   *TRUE*
2. Is JButton is a subtype of Component ?   *TRUE*
3. Is FlowLayout is a subtype of LayoutManager?   *TRUE*
4. Is ListIterator is a subtype of Iterator ?   *TRUE*
5. Is Rectangle[ ] is a subtype of Shape[ ] ?   *TRUE*
6. Is int[ ] is a subtype of Object ?   *TRUE*
7. Is int is subtype of long ?   *NO*
8. Is long is a subtype of int ?   *NO*
9. Is int[ ] is a subtype of Object[ ] ?   *NO*
10. Is int[] is a subtype of Object ?   *YES*

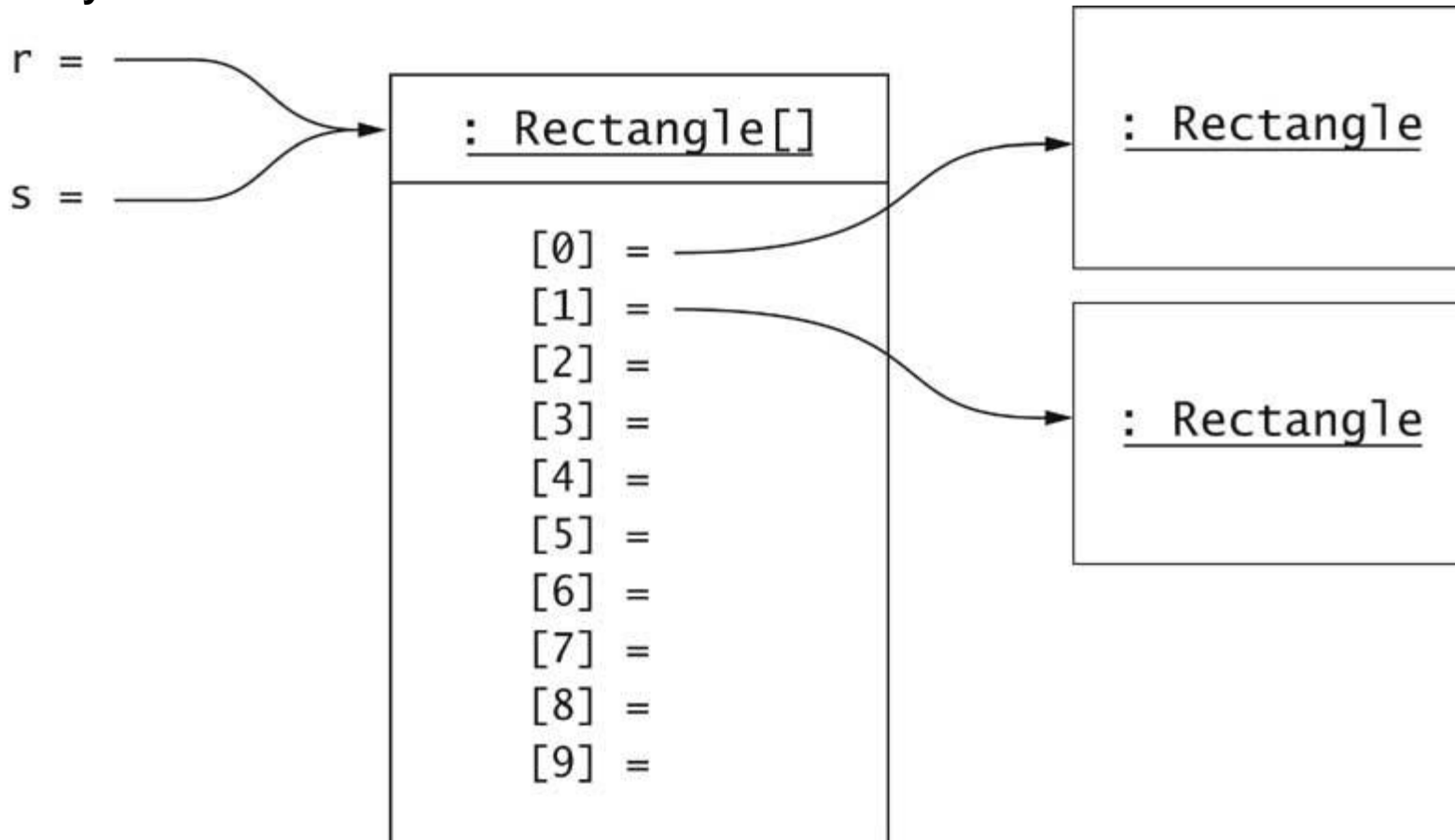## Primitive Types are not implemented as Objects

ArrayStoreException

- **Rectangle[] is a subtype of Shape[] ?.**   **TRUE**

- Can assign Rectangle[] value to Shape[] variable:
- Rectangle[] r = new Rectangle[10];
  Shape[] s = r;

- Both r and s are references to the same array

- The assignment
  s[0] = new Polygon(); compiles
  But Throws an ArrayStoreException at runtime

- Each array remembers its component type

ArrayStoreException …

Array References

Thank You