1. 理解：為了解決java中的空指針的

Optional<T > 類 (java.util.Optional)是一個容器類，它可以保存類型 T 的值，代表這個值存在。或者僅僅保存null，表示這個值不存在。原來用 null 表示一個值不存在，現在 Optional可以更好的表達這個概念。並且可以避免空指針異常。

1. 常用方法：
2. public void test1*(){  
    //empty():創建的Optional對象內部的value = null* Optional*<*Object*>* op1 = Optional.*empty()*;  
    if*(*!op1.isPresent*()){//Optional封裝的數據是否包含數據* System.*out*.println*(*"數據為空"*)*;  
    *}* System.*out*.println*(*op1*)*;  
    System.*out*.println*(*op1.isPresent*())*;  
    *//如果Optional封裝的數據value為空，則get()報錯。否則，value不為空時，返回value.  
   // System.out.println(op1.get());  
    }*  
    public void test2*(){* String str = "hello";  
   *// str = null;  
    //of(T t):封裝數據t生成Optional對象。要求t非空，否則報錯。* Optional*<*String*>* op1 = Optional.*of(*str*)*;  
    *//get()通常與of()方法搭配使用。用於獲取內部的封裝的數據value* String str1 = op1.get*()*;  
    System.*out*.println*(*str1*)*;  
    *}* public void test3*(){* String str = "beijing";  
    str = null;  
    *//ofNullable(T t) ：封裝數據t賦給Optional內部的value。不要求t非空* Optional*<*String*>* op1 = Optional.*ofNullable(*str*)*;  
    *//orElse(T t1):如果Optional內部的value非空，則返回此value值。如果  
    //value為空，則返回t1.* String str2 = op1.orElse*(*"shanghai"*)*;  
    System.*out*.println*(*str2*)*;*//  
    }*
3. 典型練習：(能保證如下的方法執行中不會出現空指針的異常)

public String getGirlName2*(*Boy boy*){* Optional*<*Boy*>* boyOptional = Optional.*ofNullable(*boy*)*;  
 *//此時的boy1一定非空* Boy boy1 = boyOptional.orElse*(*new Boy*(*new Girl*(*"迪麗熱巴"*)))*;  
  
 Girl girl = boy1.getGirl*()*;  
  
 Optional*<*Girl*>* girlOptional = Optional.*ofNullable(*girl*)*;  
 *//girl1一定非空* Girl girl1 = girlOptional.orElse*(*new Girl*(*"古力娜紮"*))*;  
  
 return girl1.getName*()*;  
*}*public void test5*(){* Boy boy = null;  
 boy = new Boy*()*;  
 boy = new Boy*(*new Girl*(*"蒼老師"*))*;  
 String girlName = getGirlName2*(*boy*)*;  
 System.*out*.println*(*girlName*)*;  
  
*}*