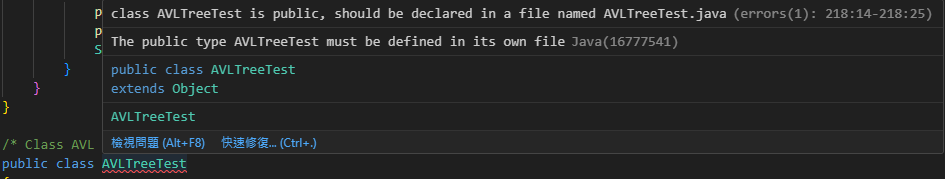
軟體工程實務HW1

學號：113522118

姓名：韓志鴻

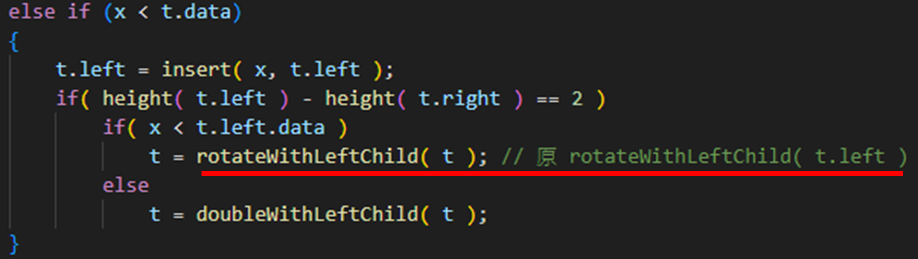
1. **Bug：**Java檔名和public class不同。

**Solution：**因為Java檔名和public class的名稱相同才可以順利執行，所以要先把原始檔名AVLtree-incorrect.java改成AVLTreeTest.java才可以執行。

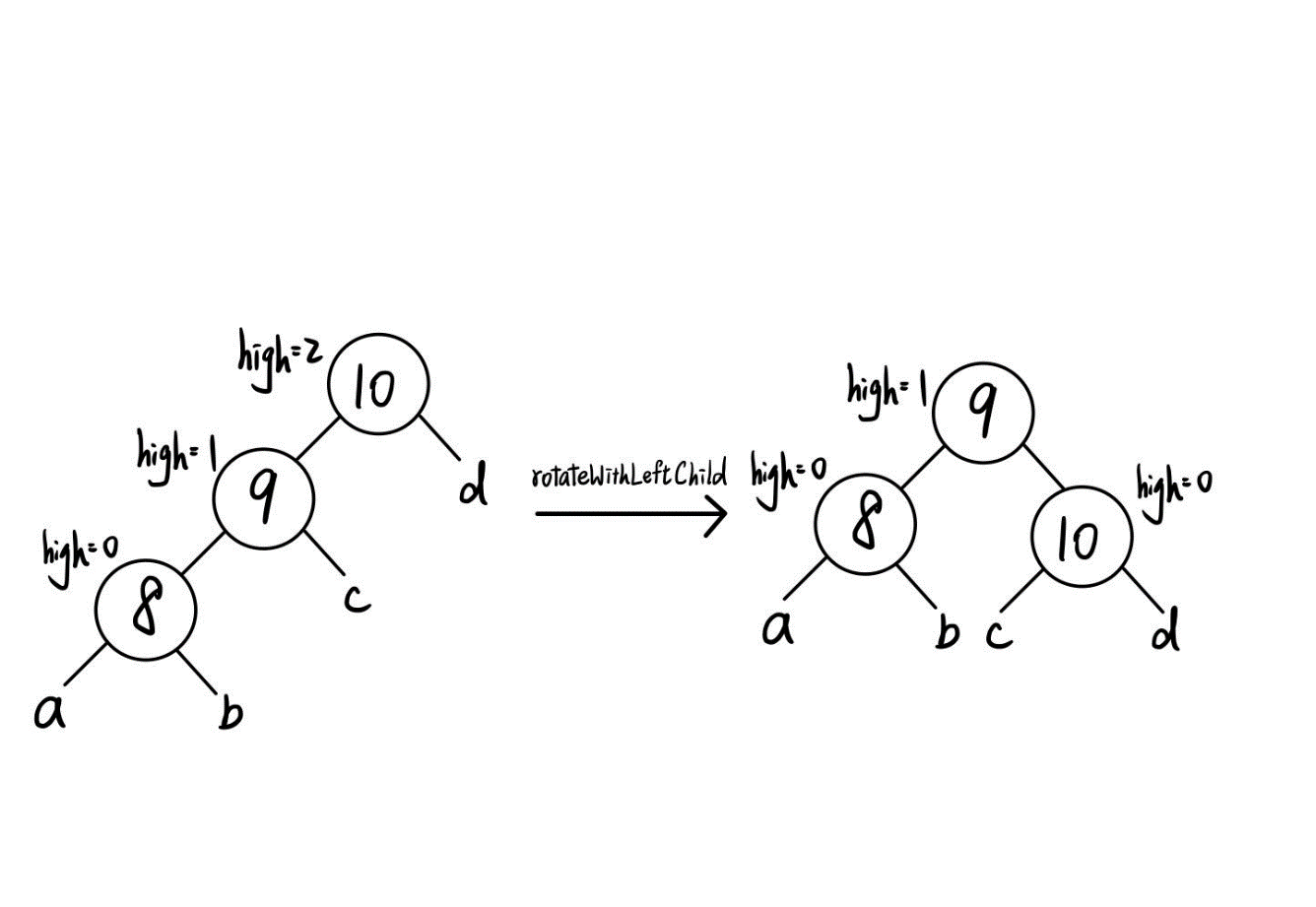


* 圖1：在public class顯示的錯誤訊息

1. **Bug：**rotateWithLeftChild()**參數傳遞錯誤**，以及**旋轉寫法錯誤**。
2. **Solution：**如下圖2和圖3，當t = 10時，若按原先寫法rotateWithLeftChild( t.left )，會把9傳入rotateWithLeftChild函式內部進行向右旋轉。然而在函式內部無法從9取得父子點10的位置，因此無法把10改成9的右子點。故必須改成rotateWithLeftChild( t )，把旋轉目標當作參數傳入，後續才能順利執行。

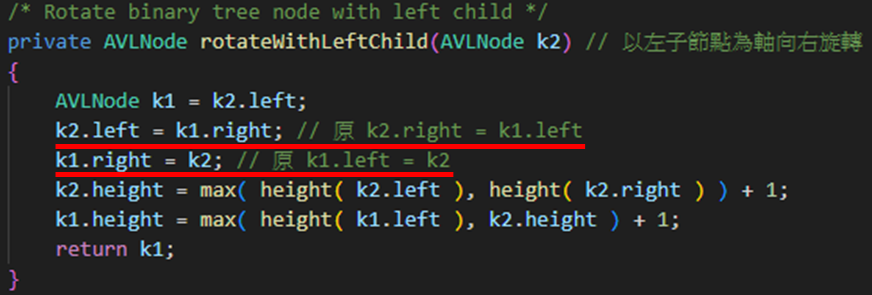


* 圖2：程式碼修改



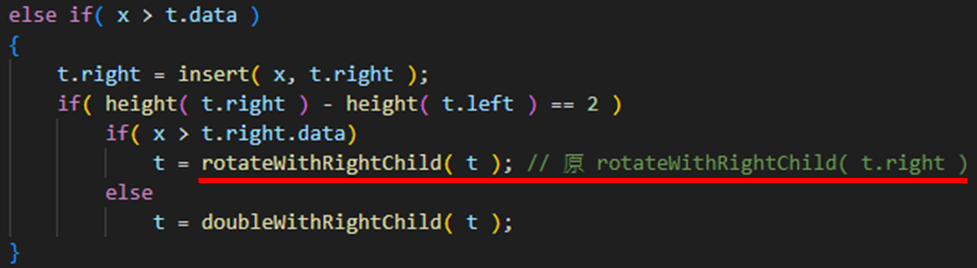
* 圖3：rotateWithLeftChild執行示意圖

1. **Solution：**如上圖3，若要從左邊的樹旋轉成右邊的樹，則需要以9為軸，將c節點變成10的左節點，接著把10當作9的右子點。同理對應下圖4的code，其中k1 < k2(因為k1是k2的左子點)，以k1為軸，要先把k1的右子點變成k2的左子點，接著把k2當成k1的右子點，才能正確完成AVL樹的平衡。故原code中的left和right部分全部都寫相反的寫法是錯誤的。

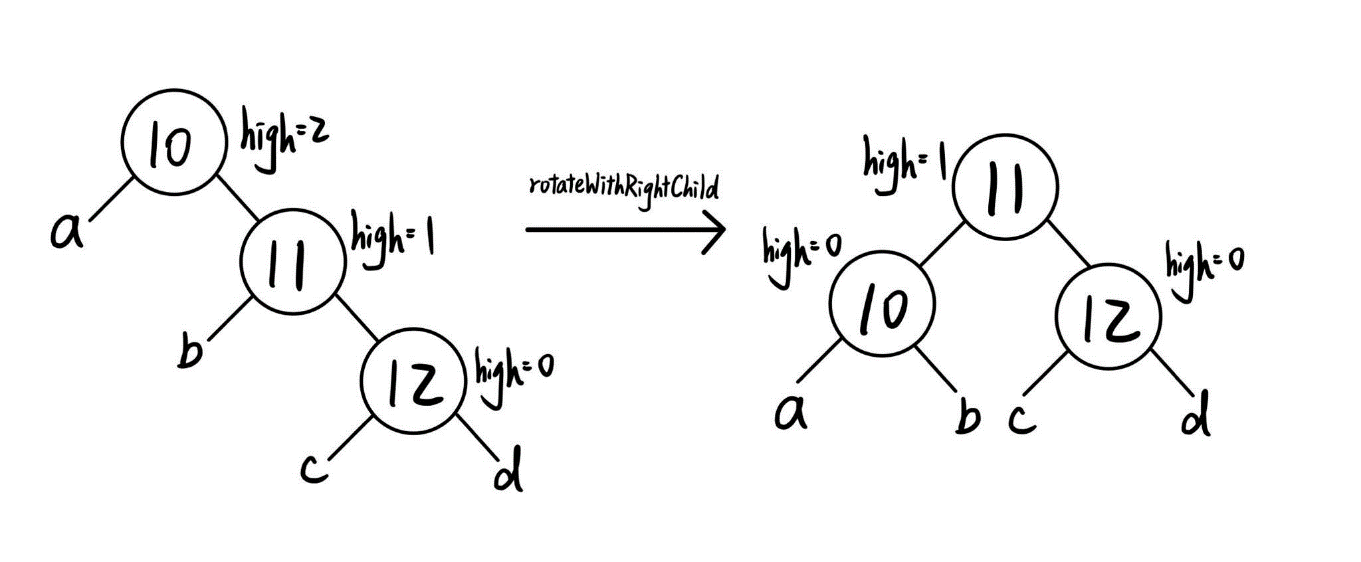
****

* 圖4：rotateWithLeftChild內部旋轉程式碼修改

1. **Bug：**rotateWithRightChild ()**參數傳遞錯誤**，以及**旋轉寫法錯誤。**
2. **Solution：**如下圖5和圖6，與先前的rotateWithLeftChild同理。當t = 10時，若按原先寫法rotateWithRightChild ( t.right )，會把11傳入rotateWithRightChild函式內部進行向左旋轉。然而在函式內部無法從11取得父子點10的位置，因此無法把10改成11的左子點。故必須改成rotateWithRightChild ( t )，把旋轉目標當作參數傳入，後續才能順利執行。

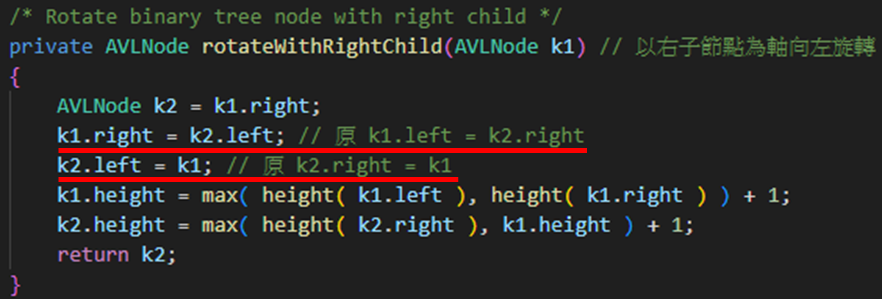
****

* 圖5：程式碼修改

****

* 圖6：rotateWithRightChild執行示意圖

1. **Solution：**如上圖6，若要從左邊的樹旋轉成右邊的樹，則需要以11為軸，將b節點變成10的右節點，接著把10當作11的左子點。同理對應下圖7的code，其中k1 < k2(因為k2是k1的右子點)，以k2為軸，要先把k2的左子點變成k1的右子點，接著把k1當成k2的左子點，才能正確完成AVL樹的平衡。故原code中的left和right部分全部都寫相反的寫法是錯誤的。



* 圖7：rotateWithRightChild內部旋轉程式碼修改