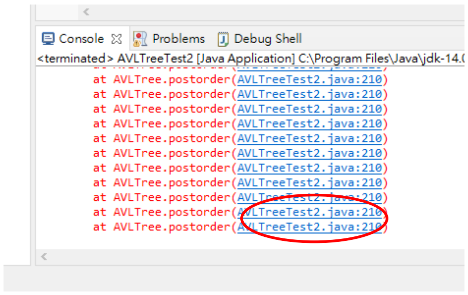
軟體工程實務 HW1

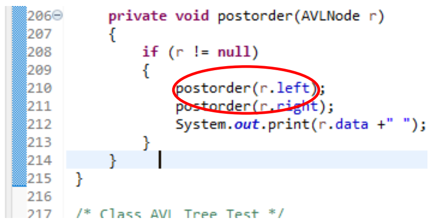
109525009 張祐綸

Bug1



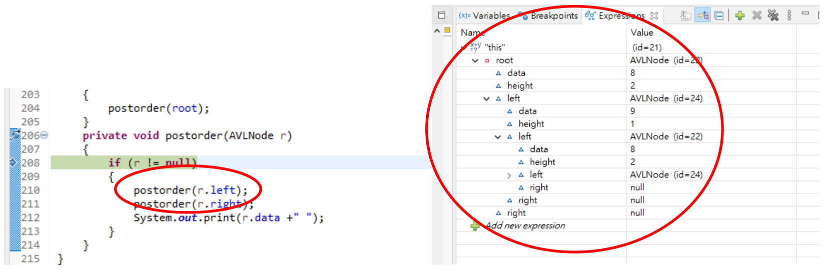
```
<terminated> AVLTreeTest2 [Java Application] C:\Program Files\Java\jdk-14.(
        at AVLTree.postorder(AVLTreeTest2.java:210)
        at AVLTree.postorder(AVLTreeTest2.java:210)
        at AVLTree.postorder(AVLTreeTest2.java:210)
        at AVLTree.postorder(AVLTreeTest2.java:210)
        at AVLTree.postorder(AVLTreeTest2.java:210)
        at AVLTree.postorder(AVLTreeTest2.java:210)
        at AVLTree.postorder(AVLTreeTest2.java:210)
        at AVLTree.postorder(AVLTreeTest2.java:210)
        at AVLTree.postorder(AVLTreeTest2.java:210)
        at AVLTree.postorder(AVLTreeTest2.java:210)
        at AVLTree.postorder(AVLTreeTest2.java:210)
```

Go to :210



```
206      private void postorder(AVLNode r)
207      {
208          if (r != null)
209          {
210              postorder(r.left);
211              postorder(r.right);
212              System.out.print(r.data +" ");
213          }
214      }
215  }
216
217  /* Class AVL Tree Test */
```

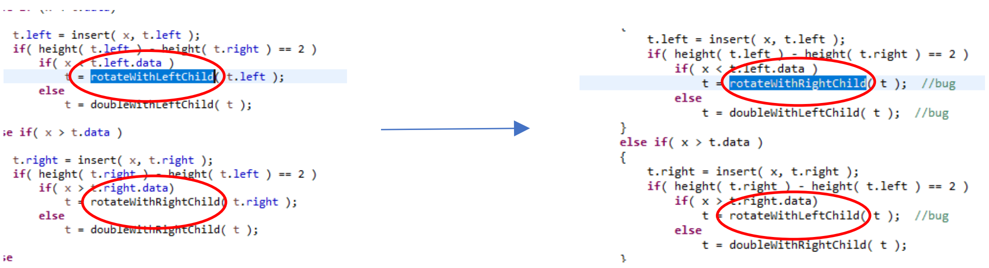
方法沒問題，因此考慮可能樹的結構有問題，在 order 之前下中斷點， 將變數設定 watch 確實發現樹不對



```
203      {
204          postorder(root);
205      }
206      private void postorder(AVLNode r)
207      {
208          if (r != null)
209          {
210              postorder(r.left);
211              postorder(r.right);
212              System.out.print(r.data +" ");
213          }
214      }
215  }
```

```
Name                      Value
  "this"                  (id=21)
    root                  AVLNode (id=22)
      data                8
      height              2
      left                AVLNode (id=24)
        data              9
        height            1
        left              AVLNode (id=22)
          data            8
          height          2
          left            AVLNode (id=24)
          right           null
        right             null
      right               null
  Add new expression
```

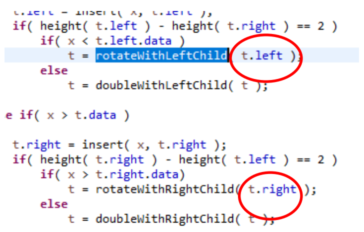
便開始看 8 如何被 insert，因此在 insert 下中斷點



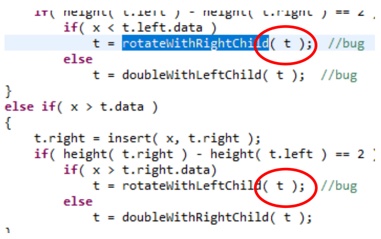
```
  t.left = insert( x, t.left );
  if( height( t.left ) - height( t.right ) == 2 )
      if( x < t.left.data )
          t = rotateWithLeftChild( t.left );
      else
          t = doubleWithLeftChild( t );

se if( x > t.data )

  t.right = insert( x, t.right );
  if( height( t.right ) - height( t.left ) == 2 )
      if( x > t.right.data)
          t = rotateWithRightChild( t.right );
      else
          t = doubleWithRightChild( t );

se
```

```
        t.left = insert( x, t.left );
        if( height( t.left ) - height( t.right ) == 2 )
            if( x < t.left.data )
                t = rotateWithRightChild( t );  //bug
            else
                t = doubleWithLeftChild( t );  //bug
    }
    else if( x > t.data )
    {
        t.right = insert( x, t.right );
        if( height( t.right ) - height( t.left ) == 2 )
            if( x > t.right.data)
                t = rotateWithLeftChild( t );  //bug
            else
                t = doubleWithRightChild( t );
    }
```

Bug2

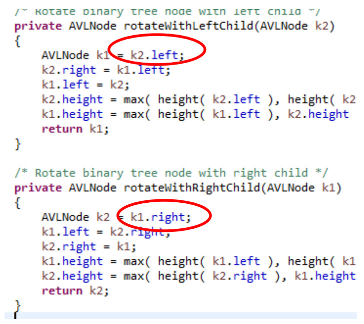


```
t.left = insert( x, t.left );
if( height( t.left ) - height( t.right ) == 2 )
    if( x < t.left.data )
        t = rotateWithLeftChild( t.left );
    else
        t = doubleWithLeftChild( t );

e if( x > t.data )

t.right = insert( x, t.right );
if( height( t.right ) - height( t.left ) == 2 )
    if( x > t.right.data)
        t = rotateWithRightChild( t.right );
    else
        t = doubleWithRightChild( t );
```

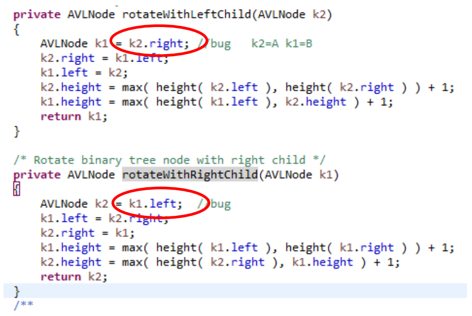


```
    if( height( t.left ) - height( t.right ) == 2
        if( x < t.left.data )
            t = rotateWithRightChild( t );  //bug
        else
            t = doubleWithLeftChild( t );  //bug
}
else if( x > t.data )
{
    t.right = insert( x, t.right );
    if( height( t.right ) - height( t.left ) == 2 
        if( x > t.right.data)
            t = rotateWithLeftChild( t ); //bug
        else
            t = doubleWithRightChild( t );
}
```

Bug3



```
/* Rotate binary tree node with left child */
private AVLNode rotateWithLeftChild(AVLNode k2)
{
    AVLNode k1 = k2.left;
    k2.right = k1.left;
    k1.left = k2;
    k2.height = max( height( k2.left ), height( k2
    k1.height = max( height( k1.left ), k2.height
    return k1;
}

/* Rotate binary tree node with right child */
private AVLNode rotateWithRightChild(AVLNode k1)
{
    AVLNode k2 = k1.right;
    k1.left = k2.right;
    k2.right = k1;
    k1.height = max( height( k1.left ), height( k1
    k2.height = max( height( k2.right ), k1.height
    return k2;
}
```



```
private AVLNode rotateWithLeftChild(AVLNode k2)
{
    AVLNode k1 = k2.right; //bug   k2=A k1=B
    k2.right = k1.left;
    k1.left = k2;
    k2.height = max( height( k2.left ), height( k2.right ) ) + 1;
    k1.height = max( height( k1.left ), k2.height ) + 1;
    return k1;
}

/* Rotate binary tree node with right child */
private AVLNode rotateWithRightChild(AVLNode k1)
{
    AVLNode k2 = k1.left;  //bug
    k1.left = k2.right;
    k2.right = k1;
    k1.height = max( height( k1.left ), height( k1.right ) ) + 1;
    k2.height = max( height( k2.right ), k1.height ) + 1;
    return k2;
}
/**
```

Bug 之心路歷程：在尋找 Bug 的開始，只知道 insert 到 8 時會產生錯誤，接著開始在 order 方法下中斷點，而後發現樹的結構怪怪的，再到 insert 裡面看並下中斷點，並觀察樹的樣子發現不對，在進行 rotate 時應該要往右轉(此時是 LL 型)，開始發現名稱不太對，接著看到變數是 t.left 便想到是要改變 t，因此將第 77 行的 t.left 更改為 t，使用的方法也改成 rotateWithRightChild，再來就是 rotateWithRightChild 裡面的實作，我參考了網路上的資料，因此將 109 行的 k1.right 改成 k1.left，以上是找到 Bug 的過程，剩下就是更改相應的程式碼。

參考資料：

https://josephjsf2.github.io/data/structure/and/algorithm/2019/06/22/avl-tree.html