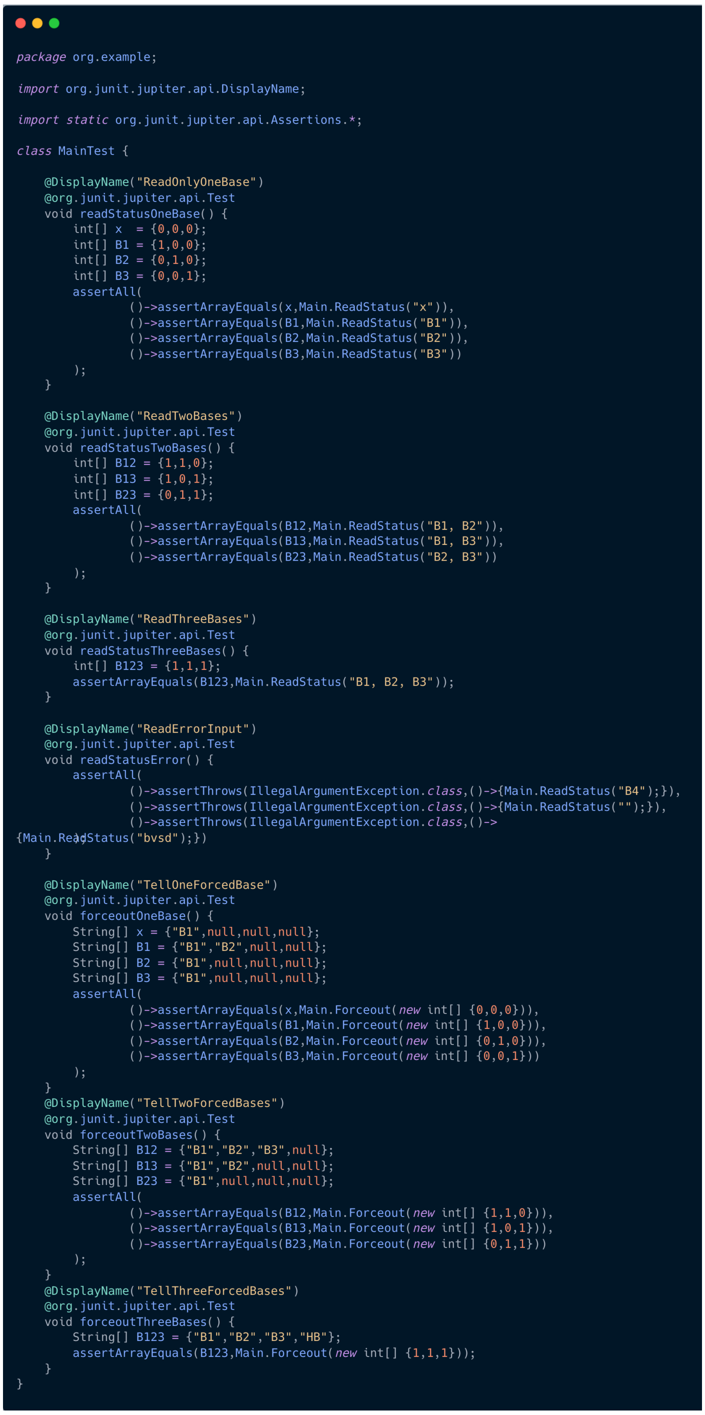
軟體測試 HW3 報告

D1149576 曹世杰

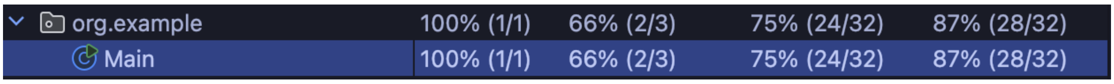
HWA：

Junit 測試程式碼：



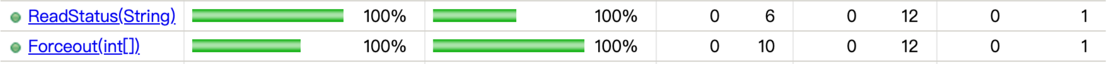
```java
package org.example;

import org.junit.jupiter.api.DisplayName;

import static org.junit.jupiter.api.Assertions.*;

class MainTest {

    @DisplayName("ReadOnlyOneBase")
    @org.junit.jupiter.api.Test
    void readStatusOneBase() {
        int[] x  = {0,0,0};
        int[] B1 = {1,0,0};
        int[] B2 = {0,1,0};
        int[] B3 = {0,0,1};
        assertAll(
                ()->assertArrayEquals(x,Main.ReadStatus("x")),
                ()->assertArrayEquals(B1,Main.ReadStatus("B1")),
                ()->assertArrayEquals(B2,Main.ReadStatus("B2")),
                ()->assertArrayEquals(B3,Main.ReadStatus("B3"))
        );
    }

    @DisplayName("ReadTwoBases")
    @org.junit.jupiter.api.Test
    void readStatusTwoBases() {
        int[] B12 = {1,1,0};
        int[] B13 = {1,0,1};
        int[] B23 = {0,1,1};
        assertAll(
                ()->assertArrayEquals(B12,Main.ReadStatus("B1, B2")),
                ()->assertArrayEquals(B13,Main.ReadStatus("B1, B3")),
                ()->assertArrayEquals(B23,Main.ReadStatus("B2, B3"))
        );
    }

    @DisplayName("ReadThreeBases")
    @org.junit.jupiter.api.Test
    void readStatusThreeBases() {
        int[] B123 = {1,1,1};
        assertArrayEquals(B123,Main.ReadStatus("B1, B2, B3"));
    }

    @DisplayName("ReadErrorInput")
    @org.junit.jupiter.api.Test
    void readStatusError() {
        assertAll(
                ()->assertThrows(IllegalArgumentException.class,()->{Main.ReadStatus("B4");}),
                ()->assertThrows(IllegalArgumentException.class,()->{Main.ReadStatus("");}),
                ()->assertThrows(IllegalArgumentException.class,()->
{Main.ReadStatus("bvsd");})
    }

    @DisplayName("TellOneForcedBase")
    @org.junit.jupiter.api.Test
    void forceoutOneBase() {
        String[] x = {"B1",null,null,null};
        String[] B1 = {"B1","B2",null,null};
        String[] B2 = {"B1",null,null,null};
        String[] B3 = {"B1",null,null,null};
        assertAll(
                ()->assertArrayEquals(x,Main.Forceout(new int[] {0,0,0})),
                ()->assertArrayEquals(B1,Main.Forceout(new int[] {1,0,0})),
                ()->assertArrayEquals(B2,Main.Forceout(new int[] {0,1,0})),
                ()->assertArrayEquals(B3,Main.Forceout(new int[] {0,0,1}))
        );
    }
    @DisplayName("TellTwoForcedBases")
    @org.junit.jupiter.api.Test
    void forceoutTwoBases() {
        String[] B12 = {"B1","B2","B3",null};
        String[] B13 = {"B1","B2",null,null};
        String[] B23 = {"B1",null,null,null};
        assertAll(
                ()->assertArrayEquals(B12,Main.Forceout(new int[] {1,1,0})),
                ()->assertArrayEquals(B13,Main.Forceout(new int[] {1,0,1})),
                ()->assertArrayEquals(B23,Main.Forceout(new int[] {0,1,1}))
        );
    }
    @DisplayName("TellThreeForcedBases")
    @org.junit.jupiter.api.Test
    void forceoutThreeBases() {
        String[] B123 = {"B1","B2","B3","HB"};
        assertArrayEquals(B123,Main.Forceout(new int[] {1,1,1}));
    }
}
```

Coverage：



org.example	100% (1/1)	66% (2/3)	75% (24/32)	87% (28/32)
Main	100% (1/1)	66% (2/3)	75% (24/32)	87% (28/32)

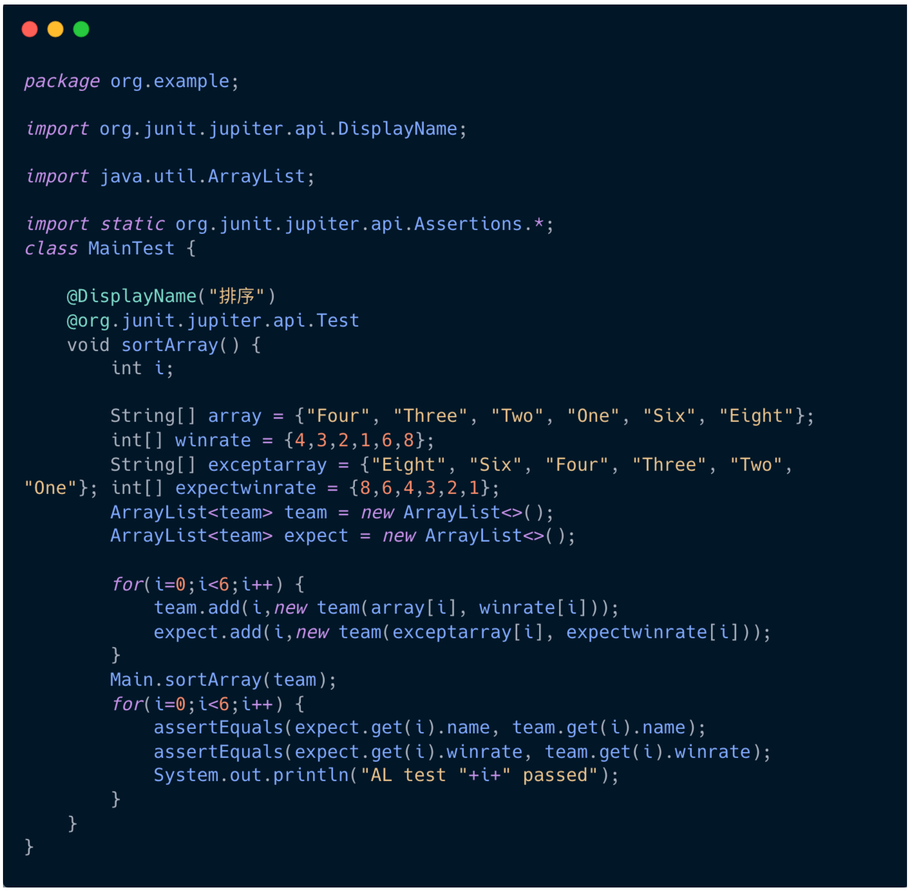
Jacoco：



ReadStatus(String)	🟩	100%	🟩	100%	0	6	0	12	0	1
Forceout(int[])	🟩	100%	🟩	100%	0	10	0	12	0	1

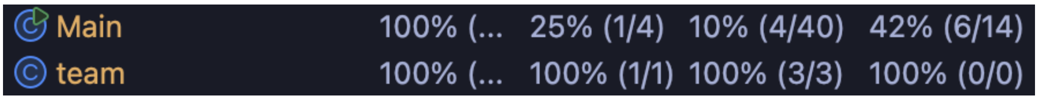
HWB：

Junit 測試程式碼：



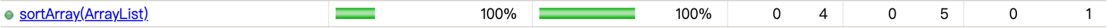
```java
package org.example;

import org.junit.jupiter.api.DisplayName;

import java.util.ArrayList;

import static org.junit.jupiter.api.Assertions.*;
class MainTest {

    @DisplayName("排序")
    @org.junit.jupiter.api.Test
    void sortArray() {
        int i;

        String[] array = {"Four", "Three", "Two", "One", "Six", "Eight"};
        int[] winrate = {4,3,2,1,6,8};
        String[] exceptarray = {"Eight", "Six", "Four", "Three", "Two",
"One"}; int[] expectwinrate = {8,6,4,3,2,1};
        ArrayList<team> team = new ArrayList<>();
        ArrayList<team> expect = new ArrayList<>();

        for(i=0;i<6;i++) {
            team.add(i,new team(array[i], winrate[i]));
            expect.add(i,new team(exceptarray[i], expectwinrate[i]));
        }
        Main.sortArray(team);
        for(i=0;i<6;i++) {
            assertEquals(expect.get(i).name, team.get(i).name);
            assertEquals(expect.get(i).winrate, team.get(i).winrate);
            System.out.println("AL test "+i+" passed");
        }
    }
}
```

Coverage：



Main	100% (...	25% (1/4)	10% (4/40)	42% (6/14)
team	100% (...	100% (1/1)	100% (3/3)	100% (0/0)

Jacoco：



sortArray(ArrayList)	🟩	100%	🟩	100%	0	4	0	5	0	1