

Software Testing - HW02

ParameterizedTest

資訊三丙
D1009331 吳克廷

目錄

目錄	2
前言	3
BMI - 設計	4
BMI - 程式執行結果	6
BMI - Unit Test	7
MyCalendar2021 - 設計	11
MyCalendar2021 - Unit Test	12
Currency - 設計	15
Currency - Unit Test	16
Tomorrow - 設計	19
Tomorrow - Unit Test	20
心得	22

前言

本文件為解釋程式碼的運作原理，因此會擷取片段的程式碼，並非完整的程式碼。若想參閱完整的程式架構，可以根據以下所提供的網址，前往GitHub進行參閱，老師謝謝。

另外，因於作業上的要求與考量，只有第一題的BMI有包含使用者的使用輸入，其他皆為以個別的API去做測試。

- **BMI.java**: <https://github.com/alecwu44743/Software-Testing/blob/master/HW02/src/main/java/org/example/BMI.java>
- **BMITest.java**: <https://github.com/alecwu44743/Software-Testing/blob/master/HW02/src/test/java/org/example/BMITest.java>
-
- **MyCalendar2021.java**: <https://github.com/alecwu44743/Software-Testing/blob/master/HW02/src/main/java/org/example/MyCalendar2021.java>
- **MyCalendar2021Test.java**: <https://github.com/alecwu44743/Software-Testing/blob/master/HW02/src/test/java/org/example/MyCalendar2021Test.java>
- **Currency.java**: <https://github.com/alecwu44743/Software-Testing/blob/master/HW02/src/main/java/org/example/Currency.java>
- **CurrencyTest.java**: <https://github.com/alecwu44743/Software-Testing/blob/master/HW02/src/test/java/org/example/CurrencyTest.java>
- **Tomorrow.java**: <https://github.com/alecwu44743/Software-Testing/blob/master/HW02/src/main/java/org/example/Tomorrow.java>
- **TomorrowTest.java**: <https://github.com/alecwu44743/Software-Testing/blob/master/HW02/src/test/java/org/example/TomorrowTest.java>

BMI - 設計

在BMI的程式中，我這裡多設計了try-exception，讓使用者在輸入 weight、height時，可以避免輸入錯誤而程式終止，讓使用者可以一直輸入，直到輸入正確才可以進入下一個狀態。

```
1  public static void main(String[] args) {
2      Scanner scanner = new Scanner(System.in);
3      double height;
4      double weight;
5
6      System.out.print("BMI Calculator\n");
7
8      while (true){
9          System.out.print("Enter your height in cm: ");
10         try {
11             height = Double.parseDouble(scanner.next()); // read the double
12             heightCheck(height);
13             break; // will only get to here if input was a double
14         } catch (NumberFormatException ignore) { // if input was not a double
15             System.out.println("Invalid input");
16         }catch (Exception e){ // prints the error message
17             System.out.println(e.getMessage());
18         }
19     }
20
21     while (true){
22         System.out.print("Enter your weight in kg: ");
23         try {
24             weight = Double.parseDouble(scanner.next()); // read the double
25             weightCheck(weight);
26             break; // will only get to here if input was a double
27         } catch (NumberFormatException ignore) { // if input was not a double
28             System.out.println("Invalid input");
29         }catch (Exception e){ // prints the error message
30             System.out.println(e.getMessage());
31         }
32     }
33
34     BMI myBMI = new BMI(height, weight);
35     System.out.println(myBMI.showMyInfo());
36 }
```

接著將討論checker的部分，包括了身高與體重的部分，以判斷使用者所輸入的資料是否正確，這裡將設定一個範圍，若失誤時，將丟出一個exception。

```
● ● ●  
1 public static void heightCheck(double height) throws Exception {  
2     if (height < 10 || height > 300) {  
3         throw new Exception("Height cannot be less than 10 or greater than 300\n" + "Please use cm");  
4     }  
5 }  
6  
7 public static void weightCheck(double weight) throws Exception {  
8     if (weight <= 0 || weight > 500) {  
9         throw new Exception("Weight cannot be less than 0 or greater than 500\n" + "Please use kg");  
10    }  
11 }
```

這裡將介紹兩個function，一個是calculateBMI()，這是用來計算使用者BMI，並且存入於BMI的attribute，而另一個是showMyInfo()，用於輸出使用者的身高、體重、BMI，以提升使用者的使用體驗。

```
● ● ●  
1 public double calculateBMI() {  
2     return this.weight / ((this.height/100) * (this.height/100)); // height is in cm, so need to divide by 100  
3 }
```

```
● ● ●  
1 public String showMyInfo() {  
2     DecimalFormat df = new DecimalFormat("0.00"); // to show only 2 decimal places  
3     String correctBmi = df.format(this.bmi);  
4     String correctHeight = df.format(this.height);  
5     String correctWeight = df.format(this.weight);  
6  
7     return "Height: " + correctHeight + "cm | " + "Weight: " + correctWeight + "kg | " + "BMI: " + correctBmi;  
8 }
```

BMI - 程式執行結果

▸ 正常輸入

```
/Library/Java/JavaVirtualMachines/adoptopenjdk-16.jdk/Contents/Home/bin/java -javaagent:/Applications/IntelliJ IDEA  
BMI Calculator  
Enter your height in cm: 180  
Enter your weight in kg: 60  
Height: 180.00cm | Weight: 60.00kg | BMI: 18.52  
  
Process finished with exit code 0
```

▸ 發生錯誤的輸入

```
/Library/Java/JavaVirtualMachines/adoptopenjdk-16.jdk/Contents/Home/bin/java -javaagent:/Applications/IntelliJ IDEA  
BMI Calculator  
Enter your height in cm: Cathay  
Invalid input  
Enter your height in cm: 180  
Enter your weight in kg: Pacific  
Invalid input  
Enter your weight in kg: 60  
Height: 180.00cm | Weight: 60.00kg | BMI: 18.52  
  
Process finished with exit code 0
```

在輸入字串時，會提示使用者錯誤的輸入，由於有exception的處理，因此程式可以繼續執行，直到使用者輸入正確與完畢。

BMI - Unit Test

▶ 測試報告書

BMITest: 36 total, 36 passed		106 ms
		Collapse Expand
/Library/Java/JavaVirtualMachines/adoptopenjdk-16.jdk/Contents/Home/bin/java -ea -Didea.test.cyclic.buffer.size=1048576 -javaagent:/Applications/IntelliJ IDEA CE.app/Contents/lib/idea_rt.jar=51246:/Applications/IntelliJ IDEA CE.app/Contents/bin -Dfile.encoding=UTF-8 -classpath /Users/alecwu/.m2/repository/org/junit/platform/junit-platform-launcher/1.9.1/junit-platform-launcher-1.9.1.jar:/Applications/IntelliJ IDEA CE.app/Contents/lib/idea_rt.jar:/Applications/IntelliJ IDEA CE.app/Contents/plugins/unit/lib/junit5-r.jar:/Applications/IntelliJ IDEA CE.app/Contents/plugins/unit/lib/junit-r.jar:/Users/alecwu/Desktop/軟體測試/HW02/target/test-classes:/Users/alecwu/Desktop/軟體測試/HW02/target/classes:/Users/alecwu/.m2/repository/org/junit/jupiter/junit-jupiter-engine/5.9.1/junit-jupiter-engine-5.9.1.jar:/Users/alecwu/.m2/repository/org/junit/platform/junit-platform-engine/1.9.1/junit-platform-engine-1.9.1.jar:/Users/alecwu/.m2/repository/org/openetc/junit/openteast4j/1.2.0/openteast4j-1.2.0.jar:/Users/alecwu/.m2/repository/org/junit/platform/junit-platform-commons/1.9.1/junit-platform-commons-1.9.1.jar:/Users/alecwu/.m2/repository/org/junit/jupiter-api/5.9.1/junit-jupiter-api-5.9.1.jar:/Users/alecwu/.m2/repository/org/apiguardian/apiguardian-api/1.1.2/apiguardian-api-1.1.2.jar:/Users/alecwu/.m2/repository/org/junit/jupiter/junit-jupiter-params/5.9.3/junit-jupiter-params-5.9.3.jar		
Test for weightCheck		56 ms
Test for getBMI		8 ms
Test for BMI Calculation		4 ms
Test for weightCheck		5 ms
Test for heightCheck		2 ms
Test for getHeight		5 ms
Test for showMyInfo		17 ms
Test for getWeight		4 ms
Test for heightCheck		5 ms

Generated by IntelliJ IDEA on 2023/11/1 上午1:28

這裡進行了36種測試，而測試結果為全部通過，下個階段將介紹每個測試的詳細過程與設計。

▶ 訊息提示

```
@DisplayName("Test for BMI Calculation")
@ParameterizedTest(name = "{index} => height={0}, weight={1}, res={2}")
```

在每個測試，都會為測試命名，以及輸出該測試的參數配置，來提醒開發人員。

▶ Test for Checker

Test for weightCheck			56 ms
1 => weight=10.0		passed	53 ms
2 => weight=15.0		passed	1 ms
3 => weight=499.9		passed	1 ms
4 => weight=500.0		passed	1 ms

Test for weightCheck			5 ms
1 => weight=-1.0		passed	2 ms
2 => weight=0.0		passed	1 ms
3 => weight=500.1		passed	1 ms
4 => weight=600.0		passed	1 ms

Test for heightCheck			2 ms
1 => height=10.0		passed	1 ms
2 => height=150.0		passed	1 ms
3 => height=300.0		passed	1 ms

Test for heightCheck			5 ms
1 => height=5.0		passed	1 ms
2 => height=9.9		passed	1 ms
3 => height=300.1		passed	1 ms
4 => height=350.0		passed	2 ms

經由上述報告可以發現，在測試的設計上會很貼心的提醒開發人員，每個測試的內容以及號碼，方便開發人員的開發。(以heightChecker為例)

```
└ alecwu44743
  @DisplayName("Test for heightCheck")
  @ParameterizedTest(name = "{index} => height={0}")
  @ValueSource(doubles = { 5.0, 9.9, 300.1, 350.0 })
  void test_invalid_heightCheck(double height) {
    assertThrows(Exception.class, () -> {
      BMI.heightCheck(height);
    });
  }

└ alecwu44743
  @DisplayName("Test for heightCheck")
  @ParameterizedTest(name = "{index} => height={0}")
  @ValueSource(doubles = { 10.0, 150.0, 300.0 })
  void test_valid_heightCheck(double height) {
    assertDoesNotThrow(() -> {
      BMI.heightCheck(height);
    });
  }
```

▶ Test for Getter

Test for getBMI	8 ms
1 => height=170, weight=60, res=20.761245674740486	passed 4 ms
2 => height=180, weight=70, res=21.604938271604937	passed 1 ms
3 => height=190, weight=80, res=22.1606648199446	passed 2 ms
4 => height=160, weight=50, res=19.531249999999996	passed 1 ms

Test for getHeight	5 ms
1 => height=170, res=170	passed 1 ms
2 => height=180, res=180	passed 1 ms
3 => height=169.98, res=169.98	passed 2 ms
4 => height=160, res=160	passed 1 ms

Test for getWeight	4 ms
1 => weight=60, res=60	passed 1 ms
2 => weight=70, res=70	passed 1 ms
3 => weight=80, res=80	passed 1 ms
4 => weight=50, res=50	passed 1 ms
5 => weight=49.99, res=49.99	passed

經上述的報告，可以確定每個getter都可以順利的取得attribute。

```
@DisplayName("Test for getBMI")
@ParameterizedTest(name = "{index} => height={0}, weight={1}, res={2}")
@CsvSource({
    "170, 60, 20.761245674740486",
    "180, 70, 21.604938271604937",
    "190, 80, 22.1606648199446",
    "160, 50, 19.531249999999996",
})
void test_getBMI(double height, double weight, double res) {
    BMI bmi = new BMI(height, weight);
    assertEquals(res, bmi.getBMI());
}
```

```
@DisplayName("Test for getWeight")
@ParameterizedTest(name = "{index} => weight={0}, res={1}")
@CsvSource({
    "60, 60",
    "70, 70",
    "80, 80",
    "50, 50",
    "49.99, 49.99",
})
void test_getWeight(double weight, double res) {
    BMI bmi = new BMI(_height: 170, weight);
    assertEquals(res, bmi.getWeight());
}
```

▶ Test for Other Function

Test for BMI Calculation		4 ms
1 => height=170, weight=60, res=20.761245674740486	passed	1 ms
2 => height=180, weight=70, res=21.604938271604937	passed	1 ms
3 => height=190, weight=80, res=22.1606648199446	passed	1 ms
4 => height=160, weight=50, res=19.531249999999996	passed	1 ms

這裡用於測試BMI的轉換是否可以正常轉換，而這裡可以確保城市都可以正常運作。

```
@DisplayName("Test for BMI Calculation")
@ParameterizedTest(name = "{index} => height={0}, weight={1}, res={2}")
@CsvSource({
    "170, 60, 20.761245674740486",
    "180, 70, 21.604938271604937",
    "190, 80, 22.1606648199446",
    "160, 50, 19.531249999999996",
})
void test_calculateBMI(double height, double weight, double res) {
    BMI bmi = new BMI(height, weight);
    assertEquals(res, bmi.getBMI());
}
```

Test for showMyInfo		17 ms
1 => height=170.68, weight=68.04, res=Height: 170.68cm Weight: 68.04kg BMI: 23.36	passed	14 ms
2 => height=180.00, weight=70.00, res=Height: 180.00cm Weight: 70.00kg BMI: 21.60	passed	2 ms
3 => height=190.00, weight=80.00, res=Height: 190.00cm Weight: 80.00kg BMI: 22.16	passed	1 ms
4 => height=160.00, weight=50.00, res=Height: 160.00cm Weight: 50.00kg BMI: 19.53	passed	

這裡用於測試使用者的資訊輸出是否可以正常輸出，而這裡可以確保城市都可以正常運作。

```
@DisplayName("Test for showMyInfo")
@ParameterizedTest(name = "{index} => height={0}, weight={1}, res={2}")
@CsvSource({
    "170.68, 68.04, Height: 170.68cm | Weight: 68.04kg | BMI: 23.36",
    "180.00, 70.00, Height: 180.00cm | Weight: 70.00kg | BMI: 21.60",
    "190.00, 80.00, Height: 190.00cm | Weight: 80.00kg | BMI: 22.16",
    "160.00, 50.00, Height: 160.00cm | Weight: 50.00kg | BMI: 19.53"
})
void showMyInfo(double height, double weight, String res) {
    BMI bmi = new BMI(height, weight);
    assertEquals(res, bmi.showMyInfo());
}
```

MyCalendar2021 - 設計

這裡是未來讓使用者可以輸入僅限於2021年的日期，並且輸出該日期的星期，因此這裡的開發我採用Java內建的package，Calendar、Date，以提升開發的速度以及正確性。

```
● ● ●
1 package org.example;
2
3 import java.text.SimpleDateFormat; // use SimpleDateFormat to format date
4 import java.util.Calendar; // use Calendar to get day of week
5 import java.util.Date; // use Date to format date
6
7 public class MyCalendar2021 {
8
9     public String myDays(int month, int day){
10         Calendar calendar = Calendar.getInstance(); // construct a Calendar instance
11         // int currentYear = calendar.get(Calendar.YEAR);
12
13         Calendar selectedDate = Calendar.getInstance();
14         selectedDate.set(2021, month - 1, day); // set year, month, day, month - 1 because month starts from 0
15
16         SimpleDateFormat dayFormat = new SimpleDateFormat("EEEE"); // format day of week to "EEEE" -> "Monday"
17         Date date = selectedDate.getTime(); // convert Calendar to Date
18         String dayOfWeek = dayFormat.format(date); // format date to day of week
19
20         return dayOfWeek;
21     }
22 }
```

MyCalendar2021 - Unit Test

► 測試報告書

```
MyCalendar2021Test: 14 total, 14 passed  
109 ms  
Collapse | Expand  
/Library/Java/JavaVirtualMachines/adoptopenjdk-16.jdk/Contents/Home/bin/java -ea -Didea.test.cyclic.buffer.size=1048576 -javaagent:/Applications/IntelliJ IDEA CE.app/Contents/lib/idea_rt.jar=51677:/Applications/IntelliJ IDEA CE.app/Contents/bin -Dfile.encoding=UTF-8 -classpath /Users/alecceu/.m2/repository/org/junit/platform/junit-platform-launcher/1.9.1/junit-platform-launcher-1.9.1.jar:/Applications/IntelliJ IDEA CE.app/Contents/lib/idea_rt.jar:/Applications/IntelliJ IDEA CE.app/Contents/plugins/unit/lib/junit5-r.jar:/Applications/IntelliJ IDEA CE.app/Contents/plugins/unit/lib/junit_r.jar:/Users/alecceu/Desktop/軟體測試/HW02/target/test-classes:/Users/alecceu/Desktop/軟體測試/HW02/target/classes:/Users/alecceu/.m2/repository/org/junit/jupiter/junit-jupiter-engine/5.9.1/junit-jupiter-engine-5.9.1.jar:/Users/alecceu/.m2/repository/org/junit/jupiter/junit-jupiter-engine/5.9.1/junit-jupiter-engine-5.9.1.jar:/Users/alecceu/.m2/repository/org/junit/jupiter/junit-jupiter-api/5.9.1/junit-jupiter-api-5.9.1.jar:/Users/alecceu/.m2/repository/org/junit/jupiter/junit-jupiter-api/5.9.1/junit-jupiter-api-5.9.1.jar:/Users/alecceu/.m2/repository/org/apiguardian/apiguardian-api/1.2/apiguardian-api-1.2.jar:/Users/alecceu/.m2/repository/org/junit/jupiter/junit-jupiter-params/5.9.3/junit-jupiter-params-5.9.3.jar  
com.intellij.rt.junit.UnitStarter -ideVersion5 -junit5 org.example.MyCalendar2021Test Process finished with exit code 0  
Test for myDays using csvFileSource  
91 ms  
Test for myDays using csvSource  
18 ms  
Generated by IntelliJ IDEA on 2023/11/1 上午1:56
```

這裡進行了14種測試，而測試結果為全部通過，下個階段將介紹每個測試的詳細過程與設計。

下頁會開始仔細說明。

▶ Using CsvSource

這裡使用CsvSource，直接在程式內初始資料，以供測試。

Test for myDays using csvSource

```
1 => month=1, day=1, res=星期五
2 => month=2, day=4, res=星期四
3 => month=3, day=1, res=星期一
4 => month=12, day=20, res=星期一
5 => month=12, day=21, res=星期二
6 => month=12, day=22, res=星期三
7 => month=12, day=25, res=星期六
```

18 ms

```
@DisplayName("Test for myDays using csvSource")
@ParameterizedTest(name = "{index} => month={0}, day={1}, res={2}")
@CsvSource({
    "1, 1, 星期五",
    "2, 4, 星期四",
    "3, 1, 星期一",
    "12, 20, 星期一",
    "12, 21, 星期二",
    "12, 22, 星期三",
    "12, 25, 星期六",
})
void test_myDays(int mon, int day, String res) { // test for myDays using csvSource
    MyCalendar2021 myCalendar2021 = new MyCalendar2021();
    assertEquals(res, myCalendar2021.myDays(mon, day));
}
```

▶ Using CsvFileSource

這裡使用CsvFileSource，而是改在csv裡面初始資料。

Test for myDays using csvFileSource		91 ms
1 => month=1, day=1, res=星期五		passed 81 ms
2 => month=2, day=4, res=星期四		passed 2 ms
3 => month=3, day=1, res=星期一		passed 2 ms
4 => month=12, day=20, res=星期一		passed 1 ms
5 => month=12, day=21, res=星期二		passed 2 ms
6 => month=12, day=22, res=星期三		passed 2 ms
7 => month=12, day=25, res=星期六		passed 1 ms

```
mon, day, res
1, 1, 星期五
2, 4, 星期四
3, 1, 星期一
12, 20, 星期一
12, 21, 星期二
12, 22, 星期三
12, 25, 星期六
```

- test_myDays.csv，內含之後要測試的測資

```
@DisplayName("Test for myDays using csvFileSource")
@ParameterizedTest(name = "{index} => month={0}, day={1}, res={2}")
@CsvFileSource(resources = "/test_myDays.csv", numLinesToSkip = 1)
void test_myDays_withfile(int mon, int day, String res) { // test for myDays using csvFileSource
    MyCalendar2021 myCalendar2021 = new MyCalendar2021();
    assertEquals(res, myCalendar2021.myDays(mon, day));
}
```

Currency - 設計

Currency的部分是可為使用者提供貨幣加總的服務，可以提供美金與台幣的換匯，在這裡使用者可以先創一個美金或台幣的物件，接著進行add時，系統會自動轉換成該物件的幣值再進行加總，提供使用者順暢的換匯服務。

```
● ● ●
1 package org.example;
2
3 import java.text.DecimalFormat;
4 public class Currency {
5     double amount;
6     String currency;
7
8     public Currency(double _amount, String _currency) {
9         this.amount = _amount;
10        this.currency = _currency;
11    }
12
13    public Currency add(Currency other) {
14        // this.currency = "USD"; -> should change to "USD" then sum up, NTD / 30 -> USD
15        // other.currency = "NTD"; -> should change to "NTD" then sum up, USD * 30 -> NTD
16
17        if (this.currency.equals("USD") && other.currency.equals("NTD")) { // this.currency == "USD" && other.currency == "NTD"
18            return new Currency(this.amount + other.amount / 30, "USD");
19        } else if (this.currency.equals("NTD") && other.currency.equals("USD")) { // this.currency == "NTD" && other.currency == "USD"
20            return new Currency(this.amount + other.amount * 30, "NTD");
21        } else { // this.currency == other.currency
22            return new Currency(this.amount + other.amount, this.currency);
23        }
24    }
25
26    public String myCurrency() {
27        DecimalFormat df = new DecimalFormat("0.00");
28        String correctAmount = df.format(this.amount);
29
30        return "Amount: " + correctAmount + " " + this.currency;
31    }
32 }
33 }
```

在add()的API中，系統會依照三種不同的情形，進行轉換與加總，讓使用者無需再換匯而即可加總，另外提供myCurrency的API讓使用者可以輸出該物件的幣值與總合資訊。

Currency - Unit Test

► 測試報告書

```
CurrencyTest: 8 total, 8 passed  
71 ms  
Collapse | Expand  
/Library/Java/JavaVirtualMachines/adoptopenjdk-16.jdk/Contents/Home/bin/java -ea -Didea.test.cyclic.buffer.size=1048576 -javaagent:/Applications/IntelliJ IDEA CE.app/Contents/lib/idea_rt.jar=52430:/Applications/IntelliJ IDEA CE.app/Contents/bin -Dfile.encoding=UTF-8 -classpath /Users/alecwu/.m2/repository/org/junit/platform/junit-platform-launcher/1.9.1/junit-platform-launcher-1.9.1.jar:/Applications/IntelliJ IDEA CE.app/Contents/lib/idea_rt.jar:/Applications/IntelliJ IDEA CE.app/Contents/plugins/junit/lib/junit5-rt.jar:/Applications/IntelliJ IDEA CE.app/Contents/plugins/junit/lib/junit-rt.jar:/Users/alecwu/Desktop/軟體測試/HW02/target/test-classes:/Users/alecwu/Desktop/軟體測試/HW02/target/classes:/Users/alecwu/.m2/repository/org/junit/jupiter/junit-jupiter-engine/5.9.1/junit-jupiter-engine-5.9.1.jar:/Users/alecwu/.m2/repository/org/junit/platform/junit-platform-engine/1.9.1/junit-platform-engine-1.9.1.jar:/Users/alecwu/.m2/repository/org/open-test4j/open-test4j/1.2.0/open-test4j-1.2.0.jar:/Users/alecwu/.m2/repository/org/junit/platform/junit-platform-commons/1.9.1/junit-platform-commons-1.9.1.jar:/Users/alecwu/.m2/repository/org/junit/jupiter/junit-jupiter-api/5.9.1/junit-jupiter-api-5.9.1.jar:/Users/alecwu/.m2/repository/org/apiguardian/apiguardian-api/1.1.2/apiguardian-api-1.1.2.jar:/Users/alecwu/.m2/repository/org/junit/jupiter/junit-jupiter-params/5.9.3/junit-jupiter-params-5.9.3.jar  
com.intellij.rt.junit.JUnitStarter -ideVersion5 -junit5 org.example.CurrencyTest Process finished with exit code 0  
Test for Currency Calculation 58 ms  
Test for getCurrency 13 ms  
Generated by IntelliJ IDEA on 2023/11/1 上午3:07
```

這裡進行了8種測試，而測試結果為全部通過，下個階段將介紹每個測試的詳細過程與設計。

下頁會開始仔細說明。

► Test for add()

一開始我其實以為我的程式碼是正常運作的，但後來多加個測試時就出現了error，後來才發現是台幣的部分加總有問題，將除改成乘。

The screenshot shows the IntelliJ IDEA interface with the following details:

- Project:** HW02
- File:** CurrencyTest.add (1)
- Code Snippet:** (from line 19)

```
public Currency add(Currency other) {
    if (this.currency.equals("USD") && other.currency.equals("NTD")) {
        return new Currency(_amount: this.amount + other.amount / 30, _currency: "USD");
    } else if (this.currency.equals("NTD") && other.currency.equals("USD")) {
        return new Currency(_amount: this.amount / 30 + other.amount, _currency: "NTD");
    } else {
        return new Currency(_amount: this.amount + other.amount, this.currency);
    }
}
```
- Run Results:** Tests failed: 1, passed: 4 of 5 tests - 70 ms
- Test Cases (Failed):**
 - [1] USD, 100, NTD, 3000 54ms: org.opentest4j.AssertionFailedError: Expected :6000.0 Actual :200.0
 - [2] NTD, 3000, USD, 100, 8ms
 - [3] NTD, 2650, NTD, 2650 2ms
 - [4] USD, 100, USD, 100, 0 2ms
 - [5] USD, 100, NTD, 3000 4ms

後來進行更改後，程式就可以正常的運作了。

The screenshot shows the IntelliJ IDEA interface with the following details:

- Project:** HW02
- File:** CurrencyTest.add (1)
- Code Snippet:** (from line 19)

```
public Currency add(Currency other) {
    if (this.currency.equals("USD") && other.currency.equals("NTD")) {
        return new Currency(_amount: this.amount + other.amount / 30, _currency: "USD");
    } else if (this.currency.equals("NTD") && other.currency.equals("USD")) {
        return new Currency(_amount: this.amount / 30 + other.amount, _currency: "NTD");
    } else {
        return new Currency(_amount: this.amount + other.amount, this.currency);
    }
}
```
- Run Results:** Tests passed: 5 of 5 tests - 61ms
- Test Cases (Passed):**
 - [1] USD, 100, NTD, 3000, 51ms
 - [2] NTD, 3000, USD, 100, 2ms
 - [3] NTD, 2650, NTD, 2650, 3ms
 - [4] USD, 100, USD, 100, 0 3ms
 - [5] USD, 100, NTD, 3000, 2ms

後來程式進行最終unit test，也確認了成是的正確性。

Test for Currency Calculation			58 ms
1 => currency1=USD, amount1=100, currency2=NTD, amount2=3000, resCurrency=USD, resAmount=200	passed	50 ms	
2 => currency1=NTD, amount1=3000, currency2=USD, amount2=100, resCurrency=NTD, resAmount=6000	passed	2 ms	
3 => currency1=NTD, amount1=2650, currency2=NTD, amount2=2650, resCurrency=NTD, resAmount=5300	passed	2 ms	
4 => currency1=USD, amount1=100, currency2=USD, amount2=100, resCurrency=USD, resAmount=200	passed	2 ms	
5 => currency1=USD, amount1=100, currency2=NTD, amount2=3000, resCurrency=USD, resAmount=200	passed	2 ms	

▶ Test for `add()`

這裡是用於測試當使用者在取得資訊時，是否可以正常運作，以取的總數以及幣值。

Test for getCurrency			13 ms
1 => currency=USD, amount=100 -> res=Amount: 100.00 USD	passed	10 ms	
2 => currency=NTD, amount=3000 -> res=Amount: 3000.00 NTD	passed	2 ms	
3 => currency=NTD, amount=2650.79 -> res=Amount: 2650.79 NTD	passed	1 ms	

```
● ● ●

1  @DisplayName("Test for getCurrency")
2  @ParameterizedTest(name = "{index} => currency={0}, amount={1} -> res={2}")
3  @CsvSource({
4      "USD, 100, Amount: 100.00 USD",
5      "NTD, 3000, Amount: 3000.00 NTD",
6      "NTD, 2650.79, Amount: 2650.79 NTD",
7  })
8  void myCurrency(String currency, double amount, String res) {
9      Currency currency1 = new Currency(amount, currency);
10     assertEquals(res, currency1.myCurrency());
11 }
```

Tomorrow - 設計

這裡的將提供使用者輸入當日日期，即可得到隔天的日期，而這裡也使用了與MyCalendar相同的package，並提供plusDay()的API，以協助做開發自己所定義的API — nextDay()，使程式的開發可以更快及更正確。

```
● ● ●
1 package org.example;
2
3 import java.time.LocalDate; // use LocalDate to parse inputDate
4 import java.time.format.DateTimeFormatter; // use DateTimeFormatter to format date
5
6 public class Tomorrow {
7     public String nextDay(String inputDate) {
8         // use LocalDate to parse inputDate
9         DateTimeFormatter formatter = DateTimeFormatter.ofPattern("yyyy-MM-dd"); // format inputDate to yyyy-MM-dd
10        LocalDate date = LocalDate.parse(inputDate, formatter); // parse inputDate to LocalDate
11
12        // add one day to inputDate
13        LocalDate nextDay = date.plusDays(1); // add one day to inputDate
14
15        // format outputDate to yyyy-MM-dd, and return
16        String outputDate = nextDay.format(formatter); // format nextDay to yyyy-MM-dd
17        return outputDate;
18    }
19 }
20
```

最後，API會回傳一個設定好的格式y-m-d的字串給使用者，讓使用者可以取得資訊。

Tomorrow - Unit Test

► 測試報告書

TomorrowTest: 4 total, 4 passed

78 ms

[Collapse](#) | [Expand](#)

```
/Library/Java/JavaVirtualMachines/adoptopenjdk-16.jdk/Contents/Home/bin/java -ea -Didea.test.cyclic.buffer.size=1048576 -javaagent:/Applications/IntelliJ IDEA CE.app/Contents/lib/idea_rt.jar=52740:/Applications/IntelliJ IDEA CE.app/Contents/bin -Dfile.encoding=UTF-8 -classpath /Users/alecceu/.m2/repository/org/junit/platform/junit-platform-launcher/1.9.1/junit-platform-launcher-1.9.1.jar:/Applications/IntelliJ IDEA CE.app/Contents/lib/idea_rt.jar:/Applications/IntelliJ IDEA CE.app/Contents/plugins/unit/lib/junit5-rt.jar:/Applications/IntelliJ IDEA CE.app/Contents/plugins/junit/lib/junit-rt.jar:/Users/alecceu/Desktop/軟體測試/HW02/target/test-classes:/Users/alecceu/Desktop/軟體測試/HW02/target/classes:/Users/alecceu/.m2/repository/org/junit/jupiter/junit-jupiter-engine/5.9.1/junit-jupiter-engine-5.9.1.jar:/Users/alecceu/.m2/repository/org/junit/platform/junit-platform-engine/1.9.1/junit-platform-engine-1.9.1.jar:/Users/alecceu/.m2/repository/org/open-test4j/open-test4j/1.2.0/open-test4j-1.2.0.jar:/Users/alecceu/.m2/repository/org/junit/platform/junit-platform-commons/1.9.1/junit-platform-commons-1.9.1.jar:/Users/alecceu/.m2/repository/org/junit/jupiter/junit-jupiter-api/5.9.1/junit-jupiter-api-5.9.1.jar:/Users/alecceu/.m2/repository/org/apiguardian/apiguardian-api/1.1.2/apiguardian-api-1.1.2.jar:/Users/alecceu/.m2/repository/org/junit/jupiter/junit-jupiter-params/5.9.3/junit-jupiter-params-5.9.3.jar
```

com.intellij.rt.junit.JUnitStarter -ideVersion5 -junit5 org.example.TomorrowTest Process finished with exit code 0

Test for nextDay

78 ms

Generated by IntelliJ IDEA on 2023/11/1 上午3:25

這裡進行了4種測試，而測試結果為全部通過，下個階段將介紹每個測試的詳細過程與設計。

下頁會開始仔細說明。

▶ Test for `nextDay()`

這裡的測資包括了跨月、跨年等測資，以確保程式碼在運作時，可以正確無誤，即便遇到特殊案例時，也可以輸出正確的答案。

Test for nextDay		78 ms
1 => inputDate=2021-01-01, nextDate=2021-01-02	passed	71 ms
2 => inputDate=2021-01-31, nextDate=2021-02-01	passed	2 ms
3 => inputDate=2023-12-31, nextDate=2024-01-01	passed	3 ms
4 => inputDate=2021-02-28, nextDate=2021-03-01	passed	2 ms

```
● ● ●

1  @DisplayName("Test for nextDay")
2  @ParameterizedTest(name = "{index} => inputDate={0}, nextDate={1}")
3  @CsvSource({
4      "2021-01-01, 2021-01-02",
5      "2021-01-31, 2021-02-01",
6      "2023-12-31, 2024-01-01",
7      "2021-02-28, 2021-03-01",
8  })
9  void nextDay(String inputDate, String res) {
10     Tomorrow tomorrow = new Tomorrow();
11     assertEquals(res, tomorrow.nextDay(inputDate));
12 }
```

心得

這次的軟體測試作業二，老師教導我們使用參數化的測試，以增加我們的測試效率，並結合之前所教的內容，其中讓我比較印象深刻的是currency那個部分，因為本來我以為我的程式是對的，結果在作業的繳交日前幾天，多放了個測資時，才發現原來我寫的code是錯的，後來經過修正，重跑一次test，才確定程式是正確的，實在讓我有許多的體會。

還記得在公司時，每次發Pull Request時，學長總會叮嚀要把code與test寫好才可以發request，經由這次的經驗也讓我明白，一行程式碼就可以關係到公司的財報與存亡，我相信如果我那個換匯程式用在現在的台新、國泰、中信等銀行，那這個後果可讓公司損失好幾個百萬，因此單元測試的重要性也在這裡彰顯出來，也因此開發人員可以guarantee自己的code可以運作正確無誤，也是日後漸漸要在職場上所培養的。

October 31, 2023