

Assignment 2 : Unit Testing and Black-Box Testing

01518573 Software Testing and Verification

ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยเกษตรศาสตร์

สำหรับงานแต่ละข้อด้านล่าง ให้ส่งงานโดยให้บีบอัดไฟล์และส่งงานตามที่ระบุในแต่ละข้อ

ข้อ 1 : ทดสอบเมธอด power() ด้วยเทคนิค EP

ให้ทดสอบเมธอด power() ด้านล่างที่คำนวณการยกกำลัง ด้วยเทคนิค equivalence partitioning (EP) โดยส่งดังนี้

- เขียนอธิบายการออกแบบกรณีทดสอบด้วยเทคนิค equivalence partitioning (EP) คล้ายในสไลด์ โดยระบุ (1) การจัดกลุ่ม (2) อินพุต (3) expected output ในการออกแบบกรณีทดสอบ ให้ส่งเป็นไฟล์รูป เอกสาร หรือ pdf
- ส่งผลการทดสอบด้วย unit testing framework ที่นิสิตถนัด เพื่อทำ unit testing ของเมธอด power() โดยใช้ อินพุตและ expected output จากข้อมูลในข้อ a)

```
public int power(int base, int exp) {  
    // Ex: power(3,2) = 9  
    int result = 1;  
    if (exp < 0)  
        return -1;  
    for (int i = 1; i <= exp; i++)  
        result = result * base;  
    return result;  
}
```

```
def power(self, base, exp):  
    # Ex: power(3,2) = 9  
    result = 1  
    if exp < 0:  
        return -1  
    for i in range(exp):  
        result = result * base  
    return result
```

ข้อ 2 : ทดสอบเมธอดเช็คประเภทของสามเหลี่ยม ด้วยเทคนิค EP

ให้ทดสอบเมธอด getTriangleType() ในภาษาจาวา หรือ get_triangle_type ในภาษาไพธอน ด้านล่าง เพื่อเช็คประเภทสามเหลี่ยมด้วยเทคนิค equivalence partitioning (EP) โดยส่งดังนี้

- เขียนอธิบายการออกแบบกรณีทดสอบด้วยเทคนิค equivalence partitioning (EP) คล้ายในสไลด์ โดยระบุ (1) การจัดกลุ่ม (2) อินพุต (3) expected output ในการออกแบบกรณีทดสอบ ให้ส่งเป็นไฟล์รูป เอกสาร หรือ pdf
- ส่งผลการทดสอบด้วย unit testing framework ที่นิสิตถนัด เพื่อทำ unit testing ของเมธอด getTriangleType / get_triangle_type โดยใช้ อินพุตและ expected output จากข้อมูลในข้อ a)

```
public class Geometry {  
  
    public static String getTriangleType(int a, int b, int c) {  
        ... code ... (เนื่องจากเป็น black-box testing ให้หากรณีทดสอบโดยไม่ดูโค้ดภายใน) โดย  
        เมธอดนี้จะรับค่าเป็นด้าน 3 ด้านของสามเหลี่ยม ทำการตรวจสอบประเภทของสามเหลี่ยม และคืนค่าเป็นสตริงที่ตรงกับ  
        ประเภทของสามเหลี่ยมที่รับเข้ามา โดยจะเป็นสตริงจากรายการต่อไปนี้ "Equilateral", "Isosceles",  
        "Scalene", "Invalid"  
    }  
}
```

```
def get_triangle_type(a, b, c):  
  
    ... code ... (เนื่องจากเป็น black-box testing ให้หากรณีทดสอบโดยไม่ดูโค้ดภายใน) โดย  
    เมธอดนี้จะรับค่าเป็นด้าน 3 ด้านของสามเหลี่ยม ทำการตรวจสอบประเภทของสามเหลี่ยม และคืนค่าเป็นสตริงที่ตรงกับ  
    ประเภทของสามเหลี่ยมที่รับเข้ามา โดยจะเป็นสตริงจากรายการต่อไปนี้ "Equilateral", "Isosceles",  
    "Scalene", "Invalid"
```

ข้อ 3 : ทดสอบเมธอด withdraw() ด้วยเทคนิค EP และ BVA

ให้ทดสอบเมธอด withdraw() ในคลาส BankAccount ด้วยเทคนิค equivalence partitioning (EP) และ boundary value analysis (BVA) โดยใช้คลาส BankAccount ตามตัวอย่างในสไลด์ หรือดาวน์โหลดโค้ดคลาส BankAccount ได้ที่ <https://github.com/ladyusa/bankaccount> และให้ส่งงานดังนี้

- เขียนอธิบายการออกแบบกรณีทดสอบด้วยเทคนิค EP และ BVA คล้ายในสไลด์ โดยระบุ (1) การจัดกลุ่ม (2) อินพุต (3) expected output ในการออกแบบกรณีทดสอบ ให้ส่งเป็นไฟล์รูป เอกสาร หรือ pdf
- ส่งผลการทดสอบด้วย unit testing framework ที่นิสิตถนัด เพื่อทำ unit testing ของเมธอด withdraw() โดยใช้อินพุตและ expected output จากข้อมูลในข้อ a)

ข้อ 4 : ทดสอบการคำนวณค่าไฟฟ้าด้วยเทคนิค EP และ BVA

พิจารณาการคำนวณค่าไฟฟ้าสำหรับบ้านที่อยู่อาศัย โดยอัตราการคิดค่าไฟฟ้ารายเดือนขึ้นอยู่กับปริมาณการใช้พลังงานไฟฟ้า (unit) บวกกับค่าบริการรายเดือน 38.22 บาท โดยใช้อัตราตามตารางดังนี้

ค่าพลังงานไฟฟ้า (Unit)	หน่วยละ (บาท)
ไม่เกิน 150 หน่วย	3.25
มากกว่า 150 หน่วย แต่ไม่เกิน 400 หน่วย	4.22
เกินกว่า 400 หน่วย	4.42

ตัวอย่าง: ถ้าใช้ไฟฟ้า 200 หน่วย ค่าไฟฟ้าจะเป็น $(150 * 3.25) + (50 * 4.22) + 38.22 = 736.72$ บาท

สมมติฐานว่า เรากำลังทดสอบในระดับ system test เพื่อคำนวณค่าไฟ ดังนั้น ให้เขียนอธิบายการออกแบบกรณีทดสอบด้วยเทคนิค EP และ BVA คล้ายในสไลด์ โดยระบุ (1) การจัดกลุ่ม (2) อินพุต (3) expected output ในการออกแบบกรณีทดสอบ ให้ส่งเป็นไฟล์รูป เอกสาร หรือ pdf โดยไม่ต้องส่งส่วนของโค้ด

ข้อ 5 : ทากรณีทดสอบด้วย pairwise testing

สมมติว่าเราได้สร้าง mobile application ที่ทำงานบนระบบปฏิบัติการแอนดรอยด์ขึ้นมาซึ่งต้องเชื่อมต่อกับเครื่องแม่ข่าย และต้องการทดสอบว่า app ของเราสามารถรันได้อย่างมีประสิทธิภาพสำหรับการใช้งานที่หลากหลายได้หรือไม่ ซึ่งสามารถแบ่งกลุ่มการใช้งานได้ดังนี้

ฮาร์ดแวร์: [Phone, Tablet]

เวอร์ชันแอนดรอยด์: [Upside Down Cake (14.0), Tiramisu (13.0), Snow Cone (12.0)]

ความเร็วของการเชื่อมต่อ: [4G, 5G, Wifi]

จงหา input ที่เป็น configuration ต่าง ๆ เพื่อทดสอบการทำงานของ app แบบ pair-wise testing (จงแสดงการจับคู่ด้วยไม่ต้องตอบ expected output) ให้ส่งเป็นไฟล์รูป เอกสาร หรือ pdf