## **แบบฝึกหัดเรื่อง Equivalent Class Partitioning และ Boundary Value Analysis**

## แบบฝึกหัดนี้เป็นงานเดี่ยว ให้นักศึกษาดาวน์โหลดไฟล์นี้เพื่อเติมคำตอบ เมื่อนักศึกษาทำเสร็จแล้วให้เซฟไฟล์เป็น PDF แล้วส่งงานบน MyCourses พร้อมกับ Source Code ที่มี Unit Tests ครบถ้วน

**Zoo Project**

**นักศึกษากำลังทดสอบโปรแกรมสวนสัตว์แห่งหนึ่ง ซึ่งรับค่าเป็นอายุของผู้เข้าชม และกำหนดราคาค่าตั๋วตามช่วงอายุดังต่อไปนี้**

|  |  |
| --- | --- |
| **อายุ (age)** | **ราคาตั๋ว** |
| 0-12 | 50 |
| 13-20 | 100 |
| 21-60 | 150 |
| 60 ขึ้นไป | 100 |

## ข้อที่ 1: จากหลักการของ Equivalent Class Partitioning ให้นักศึกษาออกแบบคลาสของค่าอายุ (age) และเลือกค่าตัวแทน (representative)

|  |  |  |
| --- | --- | --- |
| **Class** | **Representative** | **Expected Result** |
| Age < 0 | -1 | Err |
| 0 <= age < 13 | 1 | 50 |
| 13 <= age < 21 | 14 | 100 |
| 21 <= age < 61 | 22 | 150 |
| Age >= 61 | 62 | 100 |

## ข้อที่ 2: จากหลักการของ Boundary Value Analysis ให้นักศึกษาออกแบบคลาสของค่า amount และเลือกค่าตัวแทน (representative) โดยแบ่งออกเป็น All Cases, Belonging Cases และ Reduced Class

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** | **All Cases** | **Belonging Cases** | **Reduced Class** | **Expected Result** |
| Low | -1,0,1 | -1 | -1 | Err |
| 0-12 | -1,0,1 / 11,12,13 | 0,1,11,12 | 0,12 | 50 |
| 13-20 | 12,13,14 / 19,20,21 | 13,14,19,20 | 13,20 | 100 |
| 21-60 | 20,21,22 / 59,60,61 | 21,22,59,60 | 21,60 | 150 |
| High | 59,60,61 | 61 | 61 | 100 |

ข้อที่ 3: Source code ของโปรเจคท์นี้อยู่ที่ <https://github.com/MUICT-SERU/zoo-project> ให้นักศึกษาทำการศึกษาโค้ดของคลาส Zoo และใช้เทคนิค Path Analysis เพื่อหา test case ที่เป็น Linearly Independent Set

3.1 จงวาดกราฟของเมธอด get\_ticket\_price

A screenshot of a diagram

Description automatically generated

3.2 จงลิสต์ linearly independent set ของเมธอดนี้ทั้งหมด

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| Path 1 | 1 | 1 |  |  |  |  |  |  |
| Path 2 | 1 |  | 1 | 1 |  |  |  |  |
| Path 3 | 1 |  | 1 |  | 1 | 1 |  |  |
| Path 4 | 1 |  | 1 |  | 1 |  | 1 | 1 |

Path 1: S1-C1-S2 (segment 1, 2)

Path 2: S1-C1-C2-S3 (segment 1, 3, 4)

Path 3: S1-C1-C2-C3-S4 (segment 1, 3, 5, 6)

Path 4: S1, C1, C2, C3, C4, S5 (segment 1, 3, 5, 7, 8)

ข้อที่ 4: ให้นักศึกษานำ test case ทั้งหมดที่ได้จากข้อที่ 1, 2, และ 3 มาแปลงเป็น unit test case โดยการ Fork โปรเจคท์ Zoo ไปยัง GitHub Account ของนักศึกษา และทำการเขียน unit test cases ดังกล่าวด้วยภาษา Python เมื่อทำเสร็จแล้วให้ตรวจสอบว่า test case ทั้งหมดรันผ่าน หากรันไม่ผ่านให้แก้ไขโค้ดของเมธอด get\_ticket\_price ให้ถูกต้อง (มี fault อยู่ 4 จุด ในโปรแกรมนี้)

เมื่อนักศึกษาทำเสร็จแล้ว ให้ commit & push และดาวน์โหลด zip file ของโปรเจคท์​ Zoo เพื่อส่งพร้อมกับคำตอบในไฟล์นี้บน MyCourses