

01418211 / 01418217 ปีการศึกษา 2562 Lab Assignment 1

พิจารณาโจทย์ปัญหาแต่ละข้อ แล้วตอบคำถามต่อไปนี้

- A. มี Object ไต ปรากฏอยู่บ้าง
- B. แต่ละ Object ควรมี attribute อะไรบ้าง จะตั้งชื่อว่าอะไร เพื่อเก็บค่าอะไร
- C. แต่ละ Object ควรมี method อะไรบ้าง จะตั้งชื่อว่าอะไร method ดังกล่าว มีไว้เพื่ออะไร
- D. ต้องสร้าง Class กี่คลาส ชื่อ Class อะไรบ้าง
- E. เขียน UML Diagram

โจทย์ปัญหา 1

จงเขียนโปรแกรมเพื่อรับค่าเศษ (numerator) และค่าส่วน (denominator) ของเศษส่วน (fraction) สองจำนวน แล้วคำนวณหาผลรวมของเศษส่วนทั้งสองจำนวน
สมมติให้เศษส่วนแรกอยู่ในรูป a/b และเศษส่วนที่สองอยู่ในรูป c/d ในที่นี้ให้แสดงผลลัพธ์ p/q ซึ่งเป็นผลรวมที่ได้ในรูปเศษส่วนเช่นกัน

A:

f1 สำหรับเศษส่วนตัวที่หนึ่ง

f2 สำหรับเศษส่วนตัวที่สอง

f3 สำหรับเศษส่วนที่เป็นผลรวม

B:

numerator - เก็บค่าเศษ

denominator - เก็บค่าส่วน

C:

Fraction - สร้าง Object รับค่าเศษและส่วน

add - รวมเศษส่วน 2 ตัว และสร้าง Object ใหม่

getNumerator - คืนค่าเศษ

getDenominator - คืนค่าส่วน

toString - แสดงค่าของเศษส่วน

D: 1 คลาส - Fraction

E:

Fraction
- numerator: int - denominator: int
+ Fraction (numerator:int, denominator:int) + add (other:Fraction) : Fraction + getNumerator () : int + getDenominator () : int + toString () : String

โจทย์ปัญหา 2

Write a program to calculate the number of stamps that customer gets after customer spent every 50 Bahts.

A: c1 สำหรับสร้างรายการคำนวณยอดซื้อ

B:

payments - เก็บยอดชำระเงิน

stamps - เก็บจำนวนสแตมป์

C:

CalculateStamps - สร้าง Object รับยอดชำระเงิน และคำนวณสแตมป์ที่ได้รับ

getStamps - คืนค่าจำนวนสแตมป์

D: 1 คลาส - CalculateStamps

E:

CalculateStamps
- payments: double - stamps: int
+ CalculateStamps (payments:double) + getStamps () : int

โจทย์ปัญหา 3

จงเขียนโปรแกรมสำหรับเล่นเกมทายตัวเลข โดยกำหนดให้โปรแกรมสร้างเลขเป้าหมาย แล้วรับตัวเลขจากผู้เล่นที่ทายเข้ามา

โดยให้เพิ่มการเปรียบเทียบเพื่อบอกผลการทายตัวเลขในกรณีที่ทายไม่ถูกเป็น 3 กรณี คือ มากกว่าค่าเป้าหมาย (too high), น้อยกว่าค่าเป้าหมาย (too low) และเท่ากับค่าเป้าหมาย

A: g1 สำหรับสร้างเลขเป้าหมาย

B: target - เก็บเลขเป้าหมาย

C:

GuessNumbers - สร้าง Object รับเลขเป้าหมาย

guessTarget - เปรียบเทียบค่าเลขเป้าหมายกับเลขที่ทาย

D: 1 คลาส - GuessNumbers

E:

GuessNumbers
- target: int
+ GuessNumbers (target:int)
+ guessTarget (guess:int) : String

โจทย์ปัญหา 4

เกม Master Mind เป็นเกมทายตัวเลขของผู้เล่น 2 คน ผู้เล่นคนแรกกำหนดจำนวนเต็มขึ้นมา 1 จำนวน ซึ่งกำหนดให้แต่ละหลักมีค่าไม่ซ้ำกัน แล้วให้ผู้เล่นอีกคนหนึ่งทาย

ให้เขียนโปรแกรมเพื่อรับจำนวนเต็ม 2 จำนวน ที่มีขนาด 4 หลัก จำนวนแรกเป็นของผู้เล่น 1 และจำนวนที่สองเป็นของผู้เล่น 2

ให้โปรแกรมแสดงผลว่าผู้เล่น 2 ทายตัวเลขได้ถูกต้องกี่ตัว (digits) และกี่ตำแหน่ง (positions)

ในกรณีที่ทายถูกทุกตัวและทุกตำแหน่ง ให้แสดงข้อความ “Congratulations, you just mastered my mind!!”

A: m1 สำหรับสร้างเลขเป้าหมาย

B:

target - เก็บเลขเป้าหมาย

digits - เก็บจำนวนเลขที่ทายถูก

positions - เก็บจำนวนเลขที่ทายตำแหน่งถูก

C:

MasterMind - สร้าง Object รับเลขเป้าหมาย

guessTarget - เปรียบเทียบค่าเลขเป้าหมายกับเลขที่ทาย

D: 1 คลาส - MasterMind

E:

MasterMind
- target: String - digits: int - positions: int
+ MasterMind (target:String) + guessTarget (guess:String) : String

โจทย์ปัญหา 5

Hangman เป็นเกมทายคำศัพท์ โดยผู้เล่นจะทราบเพียงแค่ จำนวน ตัวอักษรของศัพท์ที่ให้มาเท่านั้น ให้ผู้เล่น

เลือกทายอักษรที่ผู้เล่นคาดว่ามีอยู่ในคำที่โจทย์ให้มา หากอักษรที่ผู้เล่นทาย ไม่มีในศัพท์ที่โจทย์ให้มา ผู้เล่นก็จะถูกแขวนคอ เริ่มจาก หัว ลำตัว แขนซ้าย แขนขวา ขาซ้าย ขาขวา เมื่อผิดครบทั้ง 6 ครั้ง จึงถือว่า เป็นการแขวนคออย่างเต็มตัว ผู้เล่นก็จะแพ้ในเกมนั้น หากผู้เล่นสามารถทายศัพท์ถูกต้องก่อนขาขวาจะโดนแขวน ผู้เล่นก็จะชนะในเกมนั้น

ให้เขียนโปรแกรมจำลองเกม Hangman

A: h1 สำหรับสร้างคำศัพท์เป้าหมาย

B:

target - เก็บคำศัพท์เป้าหมาย

history - เก็บตัวอักษรที่เคยทาย

correct - เก็บจำนวนครั้งที่ทายถูก

wrong - เก็บจำนวนครั้งที่ทายผิด

length - เก็บความยาวคำศัพท์

C:

Hangman - สร้าง Object รับคำศัพท์เป้าหมาย

guessTarget - เปรียบเทียบตัวอักษรคำศัพท์เป้าหมายกับตัวอักษรที่ทาย

D: 1 คลาส - Hangman

E:

Hangman
<ul style="list-style-type: none">- target: String- history: String- correct: int- wrong: int- length: int
<ul style="list-style-type: none">+ Hangman (target:String)+ guessTarget (guess:char) : String

โจทย์ปัญหา 6

เขียนโปรแกรมเพื่อเก็บข้อมูล Monster Ball

Monster Ball (ใน Pokémon) มีหลายประเภท ยกตัวอย่าง 4 ประเภท ดังนี้

Monster Ball Type	Price	Sell for	Ball Value
Poké Ball	200	100	1
Great Ball	600	300	1.5
Ultra Ball	1200	600	2
Master Ball	Cannot be brought	Cannot be sold	255

A: m1, m2, สำหรับ Monster Ball 1 ลูก

B:

ballType - ชนิดของลูกบอล

price - เก็บราคาซื้อของบอล

sell - เก็บราคาขายต่อของบอล

value - เก็บมูลค่าของบอล

C:

MonsterBall - สร้าง Object รับค่าต่างๆ ของบอล

setPrice - กำหนดราคาซื้อของบอล

setSell - กำหนดราคาขายต่อของบอล

toString - แสดงรายละเอียดทั้งหมดของบอล

D: 1 คลาส - MonsterBall

E:

MonsterBall
<ul style="list-style-type: none"> - ballType: String - price: double - sell: double - value: double
<ul style="list-style-type: none"> + MonsterBall (ballType:String, price:double, sell:double, value:double) + setPrice (price:double) : void + setSell (sell:double) : void + toString () : String : void

โจทย์ปัญหา 7

ต้องการออกแบบโปรแกรมพจนานุกรมเพื่อเก็บรวบรวมคำศัพท์ และความหมาย พร้อมตัวอย่างการใช้งาน โดยโปรแกรมสามารถเพิ่มคำศัพท์ และค้นหาความหมายของคำศัพท์ที่ต้องการได้

คำศัพท์	ประเภท	ความหมาย	ตัวอย่างการใช้งาน
iterate	Verb	To say, or do again	She kept reiterating her request.
recursive	Adjective	Of or relating to a repeating process whose output at each stage is applied as input in the succeeding stage.	It is easy to implement Fibonacci with recursive function.

A:

w1, w2, สำหรับเก็บข้อมูลคำศัพท์ สร้างจาก Words
db สำหรับรวมคำศัพท์ สร้างจาก Dictionary

B:

----- w1, w2, -----

word - เก็บคำศัพท์

partsofSpeech - เก็บประเภท

meaning - เก็บความหมาย

example - เก็บตัวอย่างการใช้งาน

----- db -----

name - เก็บชื่อพจนานุกรม

words - เก็บที่อยู่ของคำศัพท์

C:

----- w1, w2, -----

Words - สร้าง Object รับค่าต่างๆ ของคำศัพท์

toString - แสดงรายละเอียดทั้งหมดของคำศัพท์

----- db -----

Dictionary - สร้าง Object รับชื่อหนังสือ

addWords - เพิ่มคำศัพท์ลงหนังสือ

findWords - ค้นหาความหมายคำศัพท์

D: 2 คลาส - Words และ Dictionary

E:

Words
<ul style="list-style-type: none">- word: String- partsofSpeech: String- meaning: String- example: String
<ul style="list-style-type: none">+ Words (word:String, partsofSpeech:String, meaning:String, example:String)+ toString () : String

Dictionary
<ul style="list-style-type: none">- name: String- words: Words**
<ul style="list-style-type: none">+ Dictionary (name:String)+ addWords (newWord:Words*, i:int) : void+ findWords (searchWords:String) : void