**[เฉลย] Lab7-Array of Objects & Aggregation**

ให้สร้าง คลาส Player มีฟิลด์ id, name, และ score แทน รหัส ชื่อ คะแนน ตามลำดับ

แล้วให้สร้างคลาส Team มีอะเรย์ของผู้เล่น ไม่เกิน 15 คน ถ้าเราเพิ่มเกินจะฟ้องข้อผิดพลาดว่าเต็มแล้ว

เพื่อให้สามารถประมวลผลคำสั่งต่อไปนี้

Team m = new Team ();

Player s1= new Player (“6011”,”John”,8900);

Player s2= new Player (“6012”,”Bo”,5000);

m.addNew Player (s1); //

m.addNew Player (s2); //

int count = m.getNumberOf Player (); // คืนจำนวนนักเรียน คือ 2 คน

Player [] players = m.findAllPlayers (); // คืนอะเรย์ของนักเรียนทุกคน

Player[ ] ls = new Player [2];

ls[0] = new Player (“602”,”Koi”,60);

ls[1] = new Player (“603”,”Mono”,10);

m.merge(ls);

double avg = m.meanOfScore(); จะคืนค่าเฉลี่ยคะแนนของผู้เล่น

Player p = m.findByName(“Mono”);

จะคืนค่าของตัวแปรอ้างอิง เพื่อให้ p.getName() มีค่าเป็น “Mono” และ p.getScore() มีค่าเป็น 10

//------------------------------------------------------------------------------------------------

public class Player {

private String id;

private String name;

private double score;

public Player() {

this.id = null;

this.name = null;

this.score = 0.0;

}

public Player(String id, String name, double score) {

this.id = id;

this.name = name;

this.score = score;

}

public String getId() {

return id;

}

public void setId(String id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public double getScore() {

return score;

}

public void setScore(double score) {

this.score = score;

}

}

//------------------------------------------------------------------------------------------------

public class Team {

private Player[] player;

private int count;

public Team() {

this.player = new Player[15];

this.count = 0;

}

public void addNewPlayer(Player player) {

if(count==this.player.length) {

System.out.println("overflow!");

}

else {

this.player[this.count] = player;

this.count++;

}

}

public int getNumberOfPlayer() {

return this.count;

}

public Player[] findAllPlayers() {

Player[] tmp = new Player[this.count];

for(int i=0; i<tmp.length; i++) {

tmp[i] = this.player[i];

}

return tmp;

}

public void merge(Player[] players) {

if(this.count+players.length >= this.player.length) {

System.out.println("overflow!");

}

else {

for(int i=0; i<players.length; i++) {

this.addNewPlayer(players[i]);

}

}

}

public double meanOfScore() {

double sum = 0.0;

for(int i=0; i<this.count; i++) {

sum = sum+this.player[i].getScore();

}

double mean = sum/this.count;

return mean;

}

public Player findByName(String name) {

int index = -1;

for(int i=0; i<this.count; i++) {

if(this.player[i].getName().equals(name)) {

index = i;

}

}

if(index==-1) {

System.out.println("not found !! '"+name+"'");

return null;

}

else {

System.out.println("index "+index);

return this.player[index];

}

}

}

//------------------------------------------------------------------------------------------------

public class Main {

public static void main(String[] args) {

Team m = new Team ();

Player s1= new Player("6011","John",8900);

Player s2= new Player("6012","Bo",5000);

m.addNewPlayer(s1);

m.addNewPlayer(s2);

int count = m.getNumberOfPlayer(); // คืนจำนวนนักเรียน คือ 2 คน

Player[] players = m.findAllPlayers(); // คืนอะเรย์ของนักเรียนทุกคน

Player[] ls = new Player[2];

ls[0] = new Player("602","Koi",60);

ls[1] = new Player("603","Mono",10);

m.merge(ls);

double avg = m.meanOfScore(); //จะคืนค่าเฉลี่ยคะแนนของผู้เล่น

Player p = m.findByName("Mono"); //จะคืนค่าของตัวแปรอ้างอิง เพื่อให้ p.getName() มีค่าเป็น “Mono” และ p.getScore() มีค่าเป็น 10

//Player p = m.findByName2("Mono");

try {

System.out.println(p.getName());

}

catch(Exception e) {

System.out.println(e);

}

}

}