**[โจทย์] Lab6-Static Fields**

ข้อ 1. กำหนดให้พนักงาน มีข้อมูลที่ต้องเก็บ คือ รหัสพนักงาน ชื่อพนักงาน เงินเดือน วัน-เวลาเข้างาน การสร้างวัตถุ ใหม่ทุกครั้งให้ได้รหัสที่รันเลขต่อไป เช่น

Employee e1 = new Employee(“John”,25000.00, LocalDateTime.now());

// วัตถุ ของ Employee ที่สร้างขึ้น จะมีรหัสเท่ากับ 001

System.out.print( e1.employeeInformation());

// ID: Emp-001 Name: John Salary: 25,000.00 Date: 16/02/2018 14:26:05

Employee e2 = new Employee(“Henry”,35000.00, LocalDateTime.now());

// วัตถุ ของ Employee ที่สร้างขึ้น จะมีรหัสเท่ากับ 002

System.out.print( e2.employeeInformation());

// ID: Emp-002 Name: Henry Salary: 35,000.00 Date: 16/02/2018 14:26:36

Employee e3 = new Employee(“Jim”,30000.00, LocalDateTime.now());

// วัตถุ ของ Employee ที่สร้างขึ้น จะมีรหัสเท่ากับ 003

System.out.print( e3.employeeInformation());

// ID: Emp-003 Name: Jim Salary: 30,000.00 Date: 16/02/2018 14:46:36

**# instance variables are id(String), name(String), salary, date**

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

ข้อ 2. กำหนดให้ใบกำกับภาษี (Invoice) มีข้อมูลที่ต้องเก็บ คือ เลขใบกำกับภาษี วันที่ออก จำนวนเงิน ภาษีมูลค่าเพิ่ม อัตราภาษีคงที่ 7% การสร้างวัตถุใหม่ทุกครั้งให้เลขใบใบกำกับภาษี มีการรันเลขต่อไป เช่น

Invoice inv1 = new Invoice(LocalDate.of(2018, Month.FEBRUARY, 14), 2500);

// วัตถุ ของ Invoice ที่สร้างขึ้น จะมีรหัสเท่ากับ “Inv-0001”

System.out.print( inv1.getInvoiceInformation() );

// ID: Inv-0001 Date: 2018-02-14 Amount: 2,500.00 Vat: 175

Invoice inv2 = new Invoice( LocalDate.of(2018, Month.FEBRUARY, 15),3500 );

// วัตถุ ของ ของ Invoice ที่สร้างขึ้น จะมีรหัสเท่ากับ “Inv-0002” Date: 2018-02-15 Amount: 3,500.00 Vat: 245

System.out.print( inv2.getInvoiceInformation() );

**# instance variables are id (String), date, amount, vat**

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

ข้อ 3. กำหนดให้ใบสั่งซื้อ (Order) มีข้อมูลที่ต้องเก็บ คือ เลขใบคำสั่งซื้อ วันที่สั่งซื้อ จำนวนเงินที่ซื้อ การสร้างวัตถุใหม่ทุกครั้งให้เลขใบคำสั่งซื้อมีการรันเลขต่อไป เช่น

Order o1 = new Order(LocalDate.of(2018, Month.FEBRUARY, 14),2500);

// วัตถุ ของ Order ที่สร้างขึ้น จะมีรหัสเท่ากับ “Order-0001”

System.out.print( o1.getOrderInformation());

// ID: Order-0001 Date: 2018-02-14 Amount: 2,500.00

Order e2 = new Order( LocalDate.of(2018, Month.FEBRUARY, 15),3500);

// วัตถุ ของ ของ Order ที่สร้างขึ้น จะมีรหัสเท่ากับ “Order-0002” Date: 2018-02-15 Amount: 3,500.00

System.out.print( o2.getOrderInformation());

**# Instance variables are id, date, amount.**

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

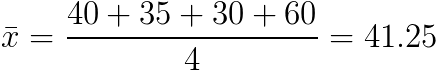
ข้อ 4. กำหนดให้สร้างอะเรย์ของคะแนนวิชาฟิสิกส์เป็นฟิลด์ภายในคลาส PhysicClass โดยกำหนดให้นักเรียนไม่เกิน 40 คน

เช่นคะแนน เป็น 40, 35, 30, 60

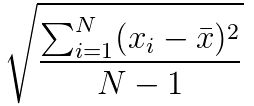
4.1 หาคะแนนสูงสุดได้ ผ่านเมธอด getMaximum() จะคืนค่า 40

4.2 หาคะแนนต่ำสุดได้ ผ่านเมธอด getMinimum() จะคืนค่า 30

4.3 หาคะแนนค่าเฉลี่ยได้ ผ่านเมธอด getAverage() จะคืนค่า



4.4 หาส่วนเบี่ยงเบนมาตรฐานได้ getSD()

ส่วนเบี่ยนเบนมาตรฐาน = 

4.5 เรียงคะแนนเรียงจากไปมากไปน้อยได้

4.6 ผสานคะแนนจากอะเรย์ภายนอกได้ เช่น มีคะแนน 5, 25 ก็จะนำมาแทรกข้างหน้าจะได้เป็น

5, 25, 10, 15, 20, 30

4.7 แทนที่อะเรย์ของคะแนนเดิมด้วยอะเรย์ของคะแนนใหม่

กรณีทดสอบ:

int [] scores = { 40,35,30,60};

PhysicClass c1 = new PhysicClass( scores);

double m = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ เก็บค่าเฉลี่ย

double sd = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ เก็บค่าเบี่ยงเบนมาตรฐาน

int maxScore = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ คะแนนสูงสุด

int minScore = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ คะแนนต่ำสุด

int scoreAt\_i = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ คะแนน ณ ตำแหน่ง i

c1.sort(); //เรียงคะแนน

int [] s = {15,70};

c1.merge(s);

scores = c1.getScores();

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

System.out.println(scores[i]);

}

int [ ] ns = { 10,20,35,15, 60,70};

c1.replaseScoreBy(ns);

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

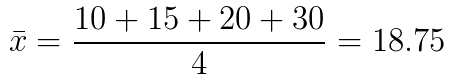
ข้อ 5. กำหนดให้สร้างอะเรย์ของคะแนนวิชาคณิตศาสตร์ เป็นฟิลด์ภายในคลาส MathClass โดยกำหนดให้นักเรียนไม่เกิน 40 คน

เช่นคะแนน เป็น 10, 15, 20, 30

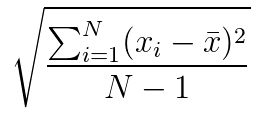
5.1 หาคะแนนสูงสุดได้ ผ่านเมธอด getMax() เพื่อหาค่ามากที่สุดและจะคืนค่า 30

5.2 หาคะแนนต่ำสุดได้ ผ่านเมธอด getMin() เพื่อหาค่าน้อยที่สุดและจะคืนค่า 10

5.3 หาคะแนนค่าเฉลี่ยได้ ผ่านเมธอด mean() เพื่อหาค่าเฉลี่ยและจะคืนค่า



5.4 หาส่วนเบี่ยงเบนมาตรฐานได้ sd()

ส่วนเบี่ยนเบนมาตรฐาน = 

5.5 คืนค่าคะแนนตามดัชนีได้

5.6 เรียงคะแนนเรียงจากมากไปน้อยได้

5.7 แทรกคะแนนจากอะเรย์ภายนอกได้ เช่น มีคะแนน 5,25 ก็จะนำมาแทรกตรงกลางได้เป็น

10,15, **5, 25**, 20, 30

กรณีทดสอบ:

int [] scores = { 10,15,20,30};

MathClass c1 = new MathClass( scores);

double m = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ เก็บค่าเฉลี่ย

double sd = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ เก็บค่าเบี่ยงเบนมาตรฐาน

int maxScore = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ คะแนนสูงสุด

int minScore = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ คะแนนต่ำสุด

int scoreAt\_i = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ คะแนน ณ ตำแหน่ง i

c1.sort(); //เรียงคะแนน

int [] newscores = {5,25};

c1.insert(newScore); // 10,15, **5, 25**, 20, 30

int [ ] allScore = c1.getScores();

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

System.out.println(allScore[i]);

}

ถ้าผ่านการเรียงแล้วจะได้

30

25

20

15

10

5

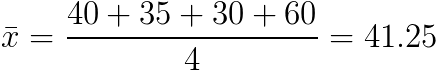
//---------------------------------------------------------------------------------------------------------------

ข้อ 6. กำหนดให้สร้างอะเรย์ของคะแนนวิชาฟิสิกส์เป็นฟิลด์ภายในคลาส PhysicClass โดยกำหนดให้นักเรียนไม่เกิน 40 คน เช่น คะแนนเป็น 40, 35, 30, 60

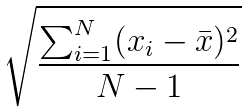
6.1 หาคะแนนสูงสุดได้ ผ่านเมธอด findMax() จะคืนค่า 40

6.2 หาคะแนนต่ำสุดได้ ผ่านเมธอด findMin() จะคืนค่า 30

6.3 หาคะแนนค่าเฉลี่ยได้ ผ่านเมธอด findMean() จะคืนค่า



6.4 หาส่วนเบี่ยงเบนมาตรฐานได้ findSTD()

ส่วนเบี่ยนเบนมาตรฐาน = 

6.5 เรียงคะแนนเรียงจากไปน้อยไปมากได้

6.6 ผสานคะแนนจากอะเรย์ภายนอกได้ เช่น มีคะแนน 5, 25 ก็จะนำมาต่อกันได้เป็น

10, 15, 20, 30, 5, 25

6.7 แทนที่อะเรย์ของคะแนนเดิมด้วยอะเรย์ของคะแนนใหม่

กรณีทดสอบ:

int [] scores = { 40,35,30,60};

PhysicClass c1 = new PhysicClass( scores);

double m = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ เก็บค่าเฉลี่ย

double sd = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ เก็บค่าเบี่ยงเบนมาตรฐาน

int maxScore = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ คะแนนสูงสุด

int minScore = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ คะแนนต่ำสุด

int scoreAt\_i = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ คะแนน ณ ตำแหน่ง i

c1.sort(); //เรียงคะแนน

int [] s = {15,70};

c1.merge(s);

scores = c1.getScores();

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

System.out.println(scores[i]);

}

int [ ] ns = { 10,20,35,15, 60,70};

c1.replaseScoreBy(ns);