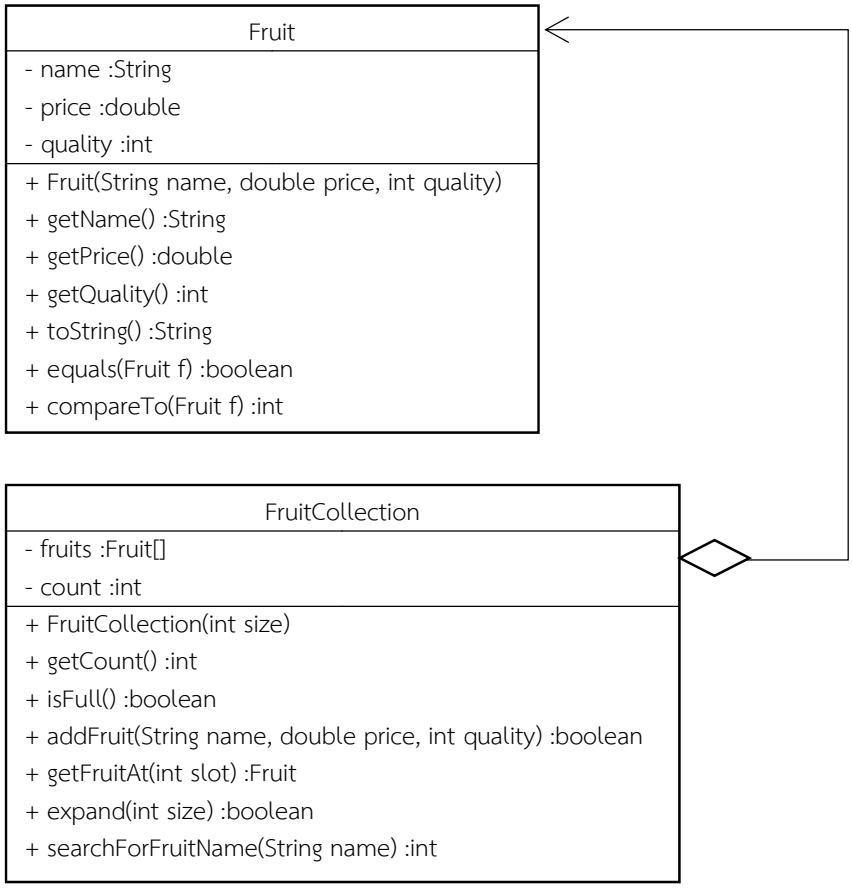
จงเขียนโปรแกรมตามข้อกำหนดใน class diagram โดยทำตามคำสั่งด้านล่างนี้



1) Fruit Class

1.1) Fruit class ประกอบด้วย (1) ชื่อ (**name**) เป็นข้อความ String (2) ราคา (**price**) เป็นจำนวนทศนิยม ที่ไม่มีค่าเป็นลบ และ (3) คุณภาพ (**quality**) เป็นจำนวนเต็มจาก 1 ถึง 10 โดย 1 หมายถึง คุณภาพต่ำ และ 10 หมายถึงคุณภาพสูง

1.2) Fruit มี 1 constructor ซึ่งรับชื่อ (name) ราคา (price) และคุณภาพ (quality) ของผลไม้เข้ามา เพื่อสร้างเป็น Fruit object โดยมีเงื่อนไขการสร้างดังนี้ คือ ถ้า name มีค่าเป็น null จะกำหนดให้ name เป็น "" (empty string), ถ้า price มีค่าน้อยกว่า 0.0 จะกำหนดให้ price มีค่าเป็น 0.0, ถ้า quality มีค่าน้อยกว่า 1 จะกำหนดให้ quality มีค่าเป็น 1 และถ้า quality มีค่ามากกว่า 10 จะกำหนดให้ มีค่าเป็น 10

1.3) Fruit มี 3 getters ได้แก่ getName(), getPrice(), และ getQuality() แต่ไม่มี setters นอกจากนี้แล้ว Fruit ยังมีอีก 3 methods ได้แก่ toString(), equals(), และ compareTo() ซึ่งมีรายละเอียดดังนี้

1.4) toString() method จะ return ค่าเป็น String ซึ่งประกอบด้วย ชื่อ (name) ราคา (price) และคุณภาพ (quality) ของผลไม้ โดย return ในรูปแบบดังนี้ `$ (price:#.00, quality:#)` เช่น `Apple (price:150.90, quality:10)`

1.5) equals(Fruit f) method รับอีก fruit หนึ่งเข้ามาเปรียบเทียบ โดย (1) จะ return **true** ถ้าผลไม้นี้ (this) และผลไม้ที่รับเข้ามา มีชื่อเหมือนกัน มีราคาเท่ากัน และมีคุณภาพเท่ากัน (เหมือนกันทุกประการ) หรือ (2) method นี้จะ return ค่าเป็น **false** ถ้าผลไม้นี้ (this) และผลไม้ที่รับเข้ามา มีชื่อต่างกัน หรือมีราคาต่างกัน หรือมีคุณภาพต่างกัน หรือผลไม้ที่รับเข้ามา มีค่าเป็น null

1.6) compareTo(Fruit f) method รับอีก fruit หนึ่งเข้ามาเปรียบเทียบ โดย (1) จะ return ค่าเป็น **1** ถ้าผลไม้นี้ (this) มีคุณภาพ (quality) สูงกว่าผลไม้ที่รับเข้ามา หรือผลไม้ที่รับเข้ามา มีค่าเป็น null หรือ (2) จะ return ค่าเป็น **-1** ถ้าผลไม้นี้ (this) มีคุณภาพ (quality) ต่ำกว่าผลไม้ที่รับเข้ามา แต่ (3) ถ้าผลไม้ทั้งสองมีคุณภาพ (quality) เท่ากัน จะ return ค่าเป็น **0**

2) FruitCollection Class

2.1) FruitCollection class ประกอบด้วย (1) **fruits** ซึ่งเป็น array ของ Fruit สำหรับเก็บข้อมูลเกี่ยวกับผลไม้ และ (2) **count** เป็นจำนวนผลไม้ที่มีอยู่ใน array

2.2) FruitCollection มี 1 constructor ซึ่งรับขนาด (size) เข้ามา เพื่อสร้าง Fruit array ที่มีขนาดเท่ากับ size ให้กับ fruits แต่ถ้า size ที่รับเข้ามา มีค่าน้อยกว่า 1 จะสร้าง Fruit array ที่มีขนาดเท่ากับ 1 ให้กับ fruits ขนาดของ array นี้บ่งบอกถึงความจุของ FruitCollection ว่าสามารถใส่ Fruit ลงไปได้กี่ครั้งจึงจะเต็ม

2.3) FruitCollection มี 1 getter คือ getCount() และไม่มี setter ใด ๆ นอกจากนี้แล้ว FruitCollection ยังมีอีก 5 methods ได้แก่ isFull(), addFruit(), getFruitAt(), expand(), และ searchForFruitName() ซึ่งมีรายละเอียดดังนี้

2.4) isFull() method ตรวจสอบว่า FruitCollection นี้ ได้บรรจุ Fruit เต็มแล้วหรือไม่ (1) method นี้ return **true** ถ้า FruitCollection นี้เต็มแล้ว หรือ (2) return **false** ถ้า FruitCollection นี้ยังไม่เต็ม

2.5) addFruit(String name, double price, int quality) method จะ (1) เพิ่มผลไม้ที่มีชื่อเป็น **name** ราคามีค่าเป็น **price** และคุณภาพมีค่าเป็น **quality** เข้าไปใน FruitCollection และนับจำนวนผลไม้ใน FruitCollection เพิ่มขึ้นอีก 1 ก่อนจะ return ค่าเป็น **true** แต่ (2) ถ้า FruitCollection นี้เต็มแล้ว หรือ **price** ที่รับเข้ามา มีค่าน้อยกว่า **0** หรือ **quality** ที่รับเข้ามา ไม่ได้มีค่าอยู่ในช่วงของ **1** ถึง **10** จะ return เป็น **false** และไม่เพิ่มผลไม้เข้าไปใน FruitCollection

2.6) getFruitAt(int slot) method จะ (1) return **Fruit** ที่อยู่ในช่องที่ **slot** ของ FruitCollection ถ้าช่อง (**slot**) ที่รับเข้ามา มีค่าตั้งแต่ 0 ขึ้นไป และไม่เกินช่องที่มีผลไม้อยู่ มิฉะนั้น (2) method นี้จะ return ค่าเป็น null

2.7) expand(int size) method จะ (1) ขยายขนาดของ FruitCollection ขึ้นอีก **size** ช่อง (ขนาดใหม่ มีค่าเท่ากับ ขนาดเดิม + size) และ return ค่าเป็น **true** แต่ (2) ถ้าค่า **size** ที่รับเข้ามา มีค่าน้อยกว่า 1 จะไม่ขยายขนาดของ FruitCollection และจะ return ค่าเป็น **false** (หมายเหตุ: การขยายขนาดของ FruitCollection สามารถทำได้โดยการสร้าง Fruit array ขึ้นใหม่ที่มีขนาดใหญ่ขึ้น แล้วย้าย Fruit ที่มีอยู่ใน array เดิม ไปใส่ไว้ใน array ใหม่ จากนั้นจึงนำ array ใหม่ไปแทนที่ array เดิม)

2.8) searchForFruitName(String name) method จะ (1) return ค่าตำแหน่งช่องใน FruitCollection ที่มีผลไม้ที่มีชื่อ name ตามที่รับเข้ามา หากมีผลไม้ที่มีชื่อ **name** ซ้ำกันหลายช่อง ให้ return ตำแหน่งช่องแรกสุดที่พบ โดยนับจากตำแหน่ง 0 เป็นช่องแรก แต่ (2) หากว่าไม่มีผลไม้ในชื่อ **name** หรือ name ที่รับเข้ามา มีค่าเป็น null ให้ return ค่า **-1**

ตัวอย่างของการเรียกใช้งาน class มีดังนี้

```
Fruit a = new Fruit("Apple", 150.9, 10);
Fruit b = new Fruit("Banana", 48.52, 6);
System.out.println(a);
System.out.println(a.equals(b));
System.out.println(a.compareTo(b));

FruitCollection fc = new FruitCollection(2);
fc.addFruit("Cherry", 85.179, 8);
fc.addFruit("Date", 94.62, 9);
fc.expand(1);
fc.addFruit("Kiwi", 64.93, 7);
int s = fc.searchForFruitName("Date");
if (s>=0) {
    System.out.println(fc.getFruitAt(s));
}
```

หมายเหตุ: การเปรียบเทียบ String ว่ามี content เหมือนกันหรือไม่นั้น ต้องใช้ equals เช่น

```
String a = new String("123");
String b = new String("123");
a.equals(b);
```