Assignment 4

Designer: ZHU Yueming Tester: JIANG Chuan(Junit), ZHU Jiawen(Tester), XU Botian(Part of Code)

Question1 Polynomial

Design a class named Polynomial

- Design one **constructors**: A constructor with one parameter **coefficient**, which is an int array. The indexes in the array mean the exponent and the values in the array mean the coefficient. For example, if the array is {1, 0, 0, 3, 2}, it can express the polynomial as 1 + 3X^3 + 2X^4.
- Design a **toString()** method, which can return this polynomial in a specific format, which append all elements by the way as "exponent:coefficient" in ascending order of its exponent value, and except the elements the coefficient value of which is zero. For example, if the polynomial is **1 + 3X^3 + 2X^4**, the `toString()` method would return "**0:1 3:3 4:2**"; if the polynomial is **2X - 3X^2 + 4X^4**, the `toString()` method would return "**1:2 2:-3 4:4**"

Sample code:

```
int[] p1 = {3, 1, 4, 0, 3};
int[] p2 = {0, 0, 1, 3, 3, 1};
int[] p3 = {4, -2, 1, -4, 0, 3};
int[] p4 = {-2, 3, 1, -1, 0, 4};
int[] p5 = {-2};
Polynomial polynomial1 = new Polynomial(p1);
Polynomial polynomial2 = new Polynomial(p2);
Polynomial polynomial3 = new Polynomial(p3);
Polynomial polynomial4 = new Polynomial(p4);
Polynomial polynomial5 = new Polynomial(p5);
System.out.println(polynomial1);
System.out.println(polynomial2);
System.out.println(polynomial3);
System.out.println(polynomial4);
System.out.println(polynomial5);
```

Output:

```
0:3 1:1 2:4 4:3
2:1 3:3 4:3 5:1
0:4 1:-2 2:1 3:-4 5:3
0:-2 1:3 2:1 3:-1 5:4
0:-2
```

- Design a public method named `public Polynomial add(Polynomial polynomial)` with

an another Polynomial as a parameter. After executing the method, the original polynomial and the return value would be the sum of this polynomial + parameter polynomial.
- Design a public method named `public Polynomial minus(Polynomial polynomial)` with an another Polynomial as a parameter. After executing the method, the original polynomial and the return value would be the difference of this polynomial - parameter polynomial.
- **(bonus)**Design a public method named `public Polynomial multiply(Polynomial polynomial)` with an another Polynomial as a parameter. After executing the method, the original polynomial and the return value would be the product of this polynomial * parameter polynomial.
- Design three **static** methods

```
public static Polynomial add(Polynomial p1, Polynomial p2)
public static Polynomial minus(Polynomial p1, Polynomial p2)
(bonus) public static Polynomial multiply(Polynomial p1, Polynomial p2)
```

respectively with another two Polynomials as parameters. The return value of those three methods is a new Polynomial that represents the sum, difference (left polynomial - right polymial) and product of those two parameters, and the original polynmial is not changed.

Question 2 BusLine

It is just a simple exercise for our assignment that the busline in our exercise is only an unidirectional linear structure, however, the busline in real world would more complex than our exercise.

1. Design a Enum Class named District.

It describes the districts of Shen Zhen city, which includes:

- Two private field, **name(String)** and **stationCount(int)**, which means the name and the total number of Station of current district. The info data of District are shown in following table.

Object Name	name(String)	stationCount(int)
BAOAN	"Bao'an"	25
FUTIAN	"Futian"	51
LONGGANG	"Longgang"	22
LONGHUA	"Longhua"	9
LUOHU	"Luohu"	23
NANSHAN	"Nanshan"	49

- adding any methods that you think are necessary.

2. Design a Class named Station

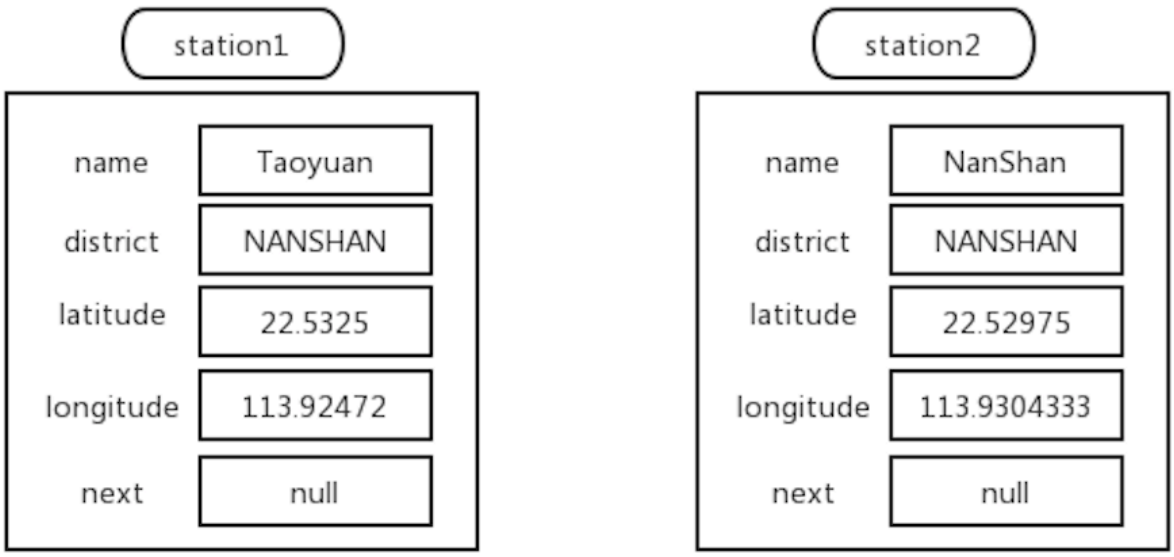
- Five private data fields
- **name(String)**: the name of Station
- **district(District)**: the district of Station
- **latitude(double)**: the latitude of Station
- **longitude(double)**: the longitude of Station
- **next(Station)**: it is a Station type, which means the next Station of the current one.
- Two **constructors**
    - A constructor with no parameter.
    - A constructor with four parameter as follows

```
public Station()
public Station(String name, District district, double latitude, double
longitude)
```

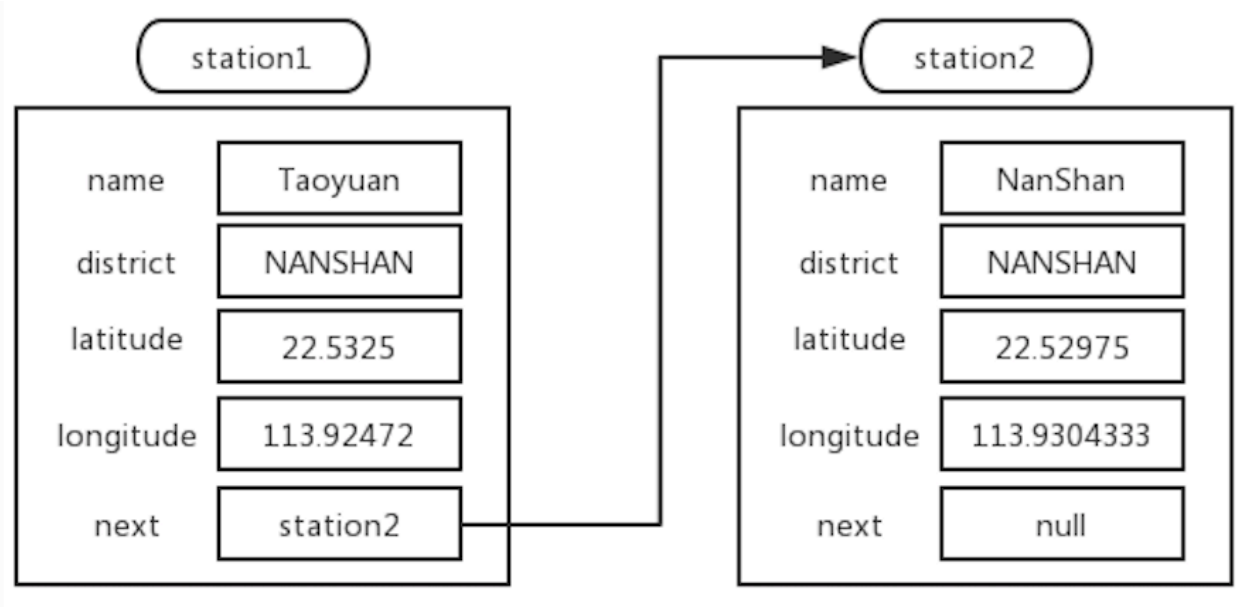
- **getter** and **setter** methods for each private fields. Please design your getter and setter methods in a standard way.

Hints: What is the next Station:

```
Station s1 = new Station("Taoyuan", District.NANSHAN, 22.5325, 113.92472);
Station s2 = new Station("NanShan", District.NANSHAN, 22.52975, 113.9304333);
```




```
s1.setNext(s2);
```



- A method **boolean equals(Station s)** to judge whether two Stations are identical according to their (all attributes except next). Here you need to take the case that the parameter s is null into consideration.
- A method **String toString()** as following type:

```
@Override
public String toString() {
    return "Station{" +
            "name='" + name + '\'' +
            ", district=" + district.getName() +
            ", latitude=" + latitude +
            ", longitude=" + longitude +
            ", next=" + next +
            '}';
}
```

3. Design a Class named BusLine

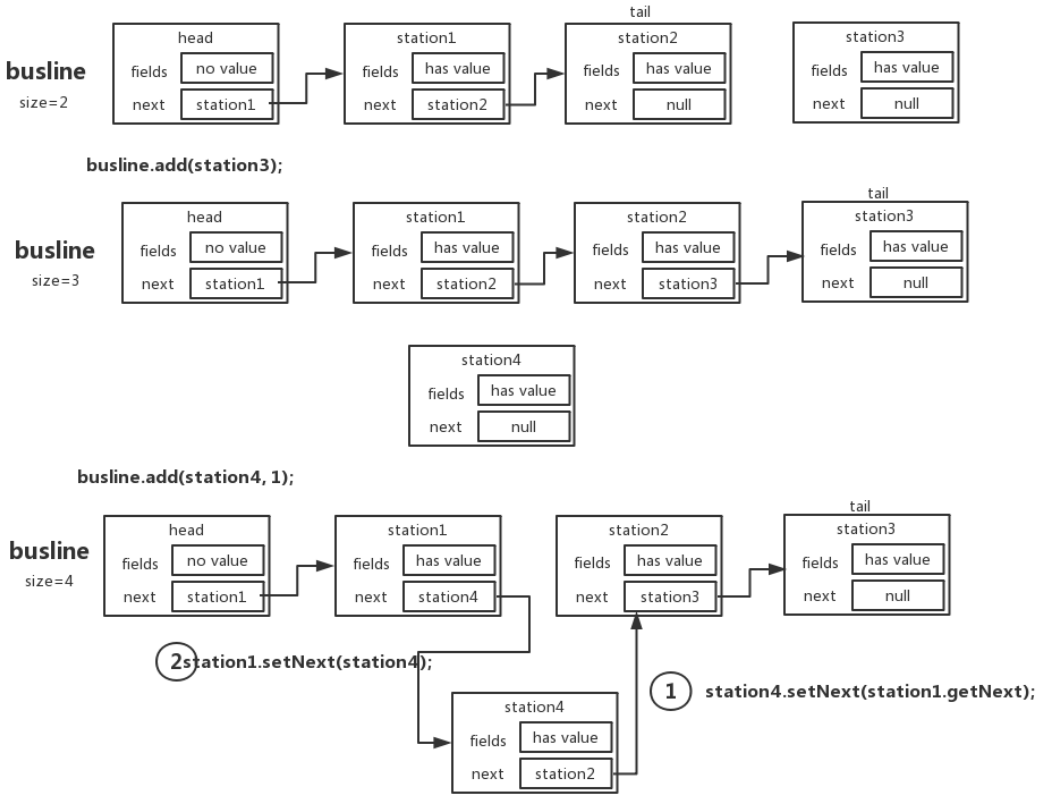
It describes an unidirectional linear busline that is composed by many Station.

You can only design those four private data fields, do not add another fields in this class. The fields includes:

- **head(Station)**: the head station of busline, and usually it doesn't store any data except the next Node.
- **tail(Station)**: the last station in the busline.
- **number(String)**: the number of current busline, such as 81 or 74
- **size(int)**: the size of current busline, and the initial value of which is 0, when adding a station, the value of size would increase by 1.
- Two **constructors**
- A constructor with no parameter.
- A constructor with one pareameter as follows.

```
public BusLine()
public BusLine(String number)
```

- **getter** and **setter** methods for each private fields **except size**.Please design your getter and setter methods in a standard way.
- Add method `public void addStation(Station station)`: Linked the specified Station to the end of this list. The value of size would increase by 1.
- **(bonus)**Add method `public void addStation(Station station, int index)`:Inserts the specified Station at the specified position (index) in this busline. The value of size would increase by 1.



- Add method `public boolean isEmpty()`: return true if this busline contains no Station, otherwise return false.
- Add method `public int size()`: return how many Stations in this busline.
- Add method `public void printStation()`: print the name of each Station from the first node to the tail. **Separate names with a space**
- Add method `public Station nearestStation(Station station)`: Return the **nearest station** of the **parameter station** in the busline, by calculate the straight-line distance according to the **latitude** and **longitude**.

  distance = Math.sqrt(difference of latitude^2 + difference of longitude ^2)

  The nearest station **cannot equal the parameter station itself**.

- Add method `public double ratioOfDistrict(District district)`: Return the ratio of total number of all Stations in parameter district in this busline to the total number of Stations in this district.

Submission of Assignment:

(1) You should submit all the source code files (with an extension **".java"**).

(2) You need submit those four java files **".java" Polynomial.java, District.java, Station.java, BusLine.java**

(3) You should submit all source code **directly** into your sakai system, **do not compress** them into one folder.

(4) **No Chinese characters** are allowed to appear in your code.

(5) No **package** included.

(6) The output must strictly follow the description and the sample in this document and the Junit Test, and you can only get points for each task only when you pass the test case.

(7) The assignment should be submitted before the deadline **(Nov. 24th). Late submissions within 24 hours after the deadline (even a few minutes) will incur a 50% penalty**, meaning that you can only get 50% of the score, which you could get if the assignment was submitted before the latest deadline. **Assignments submitted after the latest deadline will not be graded** (meaning your will get a zero for the assignment).