Assessment 8

Jinglu.Yan09119408

Problem Statement:

This Java program is handling a kind of small data base system. This Java code provides operations for inserting, extracting, deleting and displaying information from the system. Meteorological offices keep records of average monthly rainfalls over a number of cities. The record for each city consists of the name of the city, the year the data refers to, and a list of twelve numbers describing respectively the average rainfall in each of the twelve months of the year. This program can insert the whole year values, one season of values and single value of the rainfall in the given month, year, and city. It also can delete value in that ways. In addition, there are very simple calculations. At last, print all values of rainfall.

Specifications:

Input

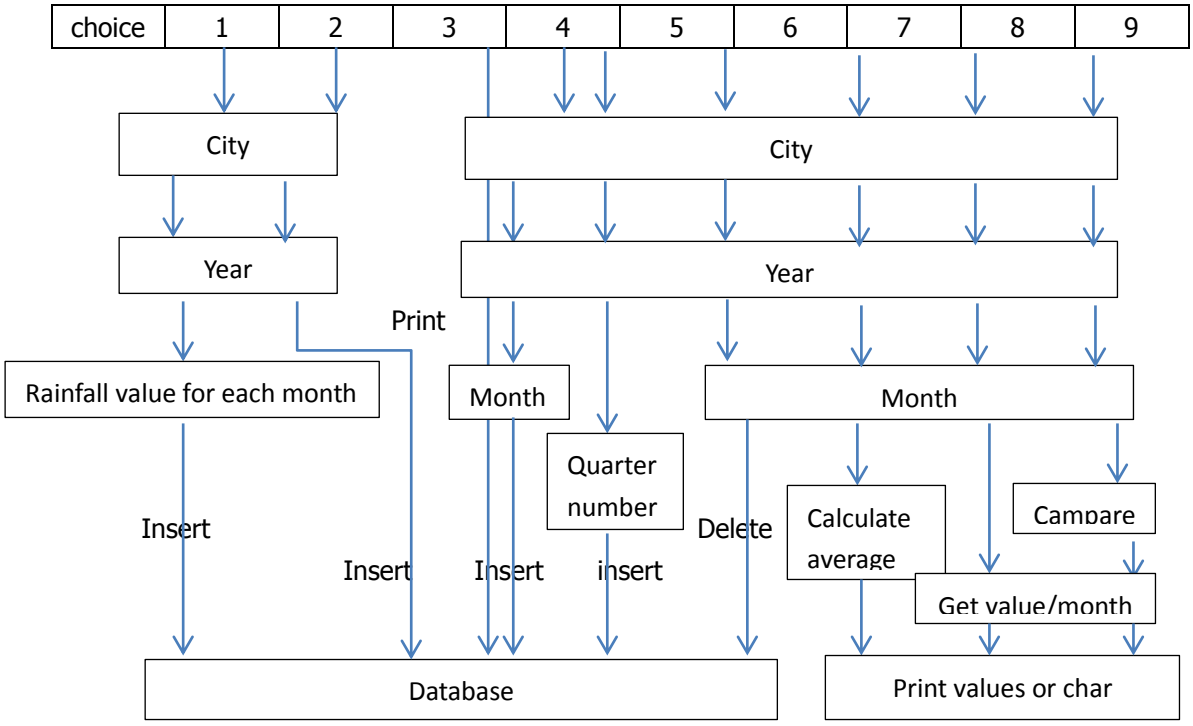
- ✓ Name of the city
- ✓ The year the data refers to
- ✓ A list of twelve numbers describing respectively the average rainfall in each of the twelve months of the year.
- ✓ Quarter number
- ✓ Rainfall values in one quarter

Output

- ✧ Average value in one year in one city.
- ✧ The just rainfall value user want
- ✧ The wettest month in one year in one city
- ✧ The whole database
- ✧ Other wrong information

Algorithm design:

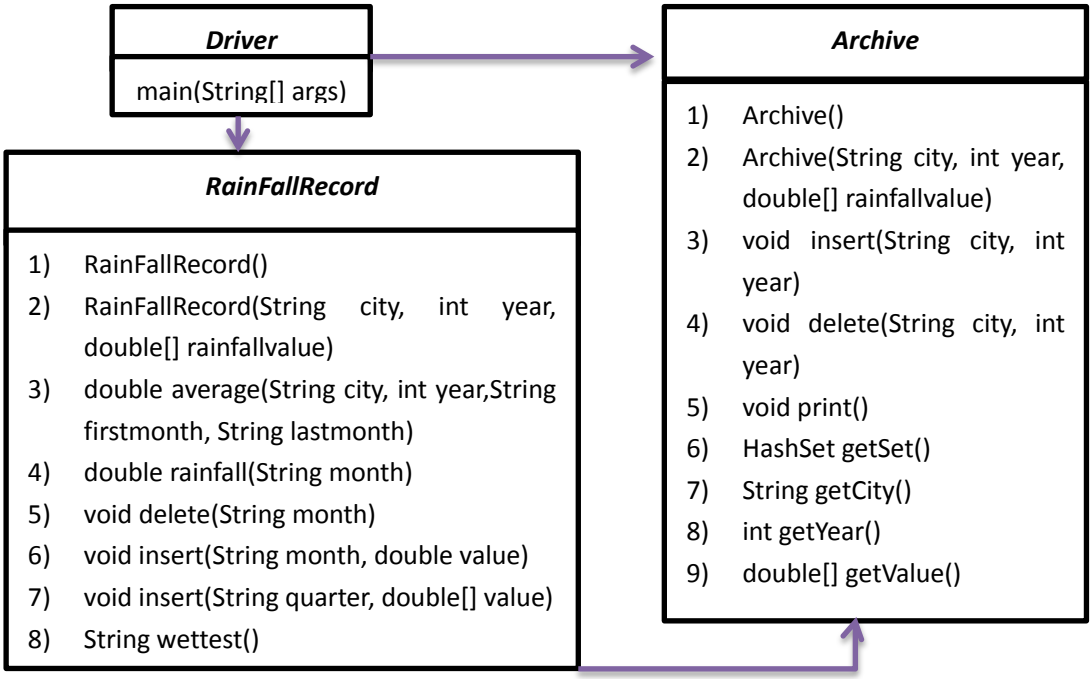
Read the choice which is inputted by user.



Implementation:

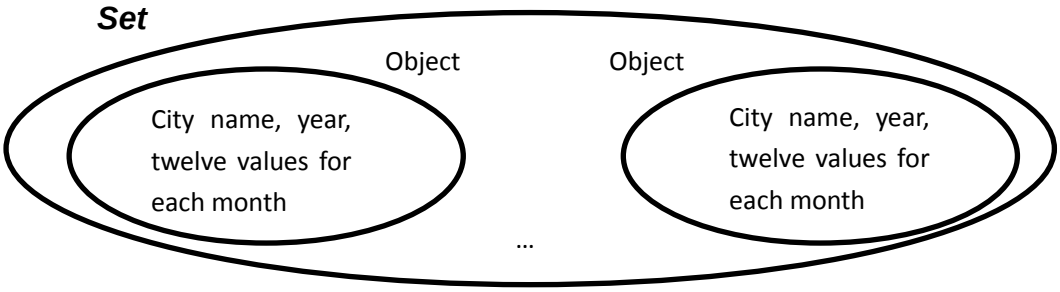
General Description:

Driver class which gives orders is a leader class for other two classes. One city, one year and a list of rainfall value will be saved in object in set. For inserting orders, when user input city name, year and values in one year, this record will be saved in database. When user chooses insert value in one quarter or one month, this program will search the corresponding city and year to insert the value. For deleting order is similar with inserting order. Calculating average value and finding the wettest month need to search corresponding object.



Class attributes or state variables:

One city, one year and a list of rainfall value will be saved in object in set. A list of twelve numbers describing respectively the average rainfall in each of the twelve months of the year will be saved in array in corresponding object in set. Archive class is a database. If some value in database will change through some methods in RainfallRecord class. The value will return to Archive class.



In Driver class, there are sixteen variables which type are integer, such as city, year, array rainfallcalue, choose, rainfall, month, value, quarter, first, second, third, array value, firstmonth, lastmonth and answer. *In RainFallRecord class*, there are eleven variables, such as city, year, array rainfallcalue, sum, first, last, c, rainfall, array value, array value1 and numbermonth. *In Archive class*, there are three variables, such as city, year and array rainfallcalue.

Methods:

There are three classes. One is ***Driver class*** which contains the *main ()* method which can run the whole program and create objects from other classes. First of all, there is a Scanner in order to read the input by user.

RainFallRecord class contains eight methods. The RainFallRecord() *and* RainFallRecord(String city, int year, double[] rainfallvalue) are constructors. The average(String city, int year,String firstmonth, String lastmonth) is the method which calculate average value in one year. The rainfall(String month) is a method to search value in a right position. The delete(String month) is a method to delete the value in right position. The insert(String month, double value) is a method to insert a value in right month. The insert(String quarter, double[] value) is a method to insert values in one quarter in right position. The wettest() method can find the wettest month in that city in right year.

Archive class which contains nine methods. Archive() and Archive(String city, int year, double[] rainfallvalue) are constructors. The insert(String city, int year) and delete(String city, int year) are methods to insert and delete records. The print() is a method to print all data. The getSet(), getCity(), getYear() and getValue() are methods to get set and values in this class.

Testing:

1. Input records and then print the results like following:

City	Year	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	AVE
Beijing	2010	0.1	0.1	0.2	0.2	0.3	0.3	0.1	0.5	0.7	0.3	0.1	0.2	0.258
Beijing	2009	0.05	0.04	0.1	0.2	0.2	0.3	0.6	0.8	0.4	0.5	0.2	0.04	0.286
Suzhou	2010	0.1	0.3	0.2	0.3	0.4	0.5	0.6	0.5	0.7	0.9	0.7	0.4	0.467

```
You choose: 3
Suzhou 2010 0.1 0.3 0.2 0.3 0.4 0.5 0.6 0.5 0.7 0.9 0.7 0.4
Beijing 2010 0.1 0.1 0.2 0.2 0.3 0.3 0.1 0.5 0.7 0.3 0.1 0.2
Beijing 2009 0.05 0.04 0.1 0.2 0.2 0.3 0.6 0.8 0.4 0.5 0.2 0.04
If you want to stop, input No. Otherwise, input anything:
```

According to this, the insert(String city, int year) method and print() method are right. This program put these two records as object into set. Therefore, the statements like add(), remove() can be used.

2. Calculating the average rainfall value in Beijing 2010. The value in that picture is right answer. Therefore, the average(String city, int year,String firstmonth, String lastmonth) method in RainfallRecord class is right. It can read values in array with a special number. Logic in it is also right.

```
You choose: 7
City Name: Beijing
Year: 2010
First Month: Jan
Last Month: Dec
The average rainfall value in Beijing in 2010 is 0.25833333333333336
```

3. Delete Beijing 2009 records, and then print out database. The results like following:

```
You choose: 3
Suzhou 2010 0.1 0.3 0.2 0.3 0.4 0.5 0.6 0.5 0.7 0.9 0.7 0.4
Beijing 2010 0.1 0.1 0.2 0.2 0.3 0.3 0.1 0.5 0.7 0.3 0.1 0.2
Beijing 2009 0.05 0.04 0.1 0.2 0.2 0.3 0.6 0.8 0.4 0.5 0.2 0.04
If you want to stop, input No. Otherwise, input anything: 2
You choose: 2
City Name: Beijing
Year: 2009
If you want to stop, input No. Otherwise, input anything: 2
You choose: 3
Suzhou 2010 0.1 0.3 0.2 0.3 0.4 0.5 0.6 0.5 0.7 0.9 0.7 0.4
Beijing 2010 0.1 0.1 0.2 0.2 0.3 0.3 0.1 0.5 0.7 0.3 0.1 0.2
```

According to that picture, the delete(String city, int year) method in Archive class is right. If program cannot find the corresponding object in set, it will print" That city or that year is not in database, so the records do not change". This is showed in appendix.

4. Insert a rainfall value of January in Beijing 2010. Rewrite the value to 12, and insert the second quarter in Suzhou 2010 by 10, 11 and 12. At the end, delete the value of November in Beijing 2010.

```
You choose: 4
City Name: Beijing
Year: 2010
Month: Jan
Value: 12
If you want to stop, input No. Otherwise, input anything: 5
You choose: 5
City Name: Suzhou
Year: 2010
Which quarter(one,two,three,four): two
Input the first value in this quarter: 10
Input the second value in this quarter: 11
Input the third value in this quarter: 12
If you want to stop, input No. Otherwise, input anything: 6
You choose: 6
City Name: Beijing
Year: 2010
Month: Nov
If you want to stop, input No. Otherwise, input anything: 3
You choose: 3
Suzhou 2010 0.1 0.3 0.2 10.0 11.0 12.0 0.6 0.5 0.7 0.9 0.7 0.4
Beijing 2010 12.0 0.1 0.2 0.2 0.3 0.3 0.1 0.5 0.7 0.3 0.0 0.2
```

According to this picture, the insert(String month, double value), insert(String quarter, double[] value) and delete(String month) methods in RainFallRecords class are right. This program can read right value in right position of array, and it can rewrite the value by 0.0, other value or other array.

5. Find the rainfall value of March in Suzhou 2010. The result is like following:

```
You choose: 8
City Name: Suzhou
Year: 2010
Month: Mar
The rainfall value in Suzhou in Mar 2010 is 0.2
```

According to this picture, the rainfall(String month) method in RainFallRecord class is right. This program can read the right value in wanted position, and print it out. If program cannot find the corresponding object in set, it will print" Sorry. That city or that year is not in database. Cannot find the right value". This is showed in appendix.

6. Find the wettest month in Beijing 2010. The result is like following:

```
You choose: 9
City Name: Beijing
Year: 2010
In Beijing, the wettest month in 2010 is Jan
```

According to this picture, the wettest() method is right. Math.max() is used to find the maximum value in value array, and this program can change integer which is the position for the maximum value into String(this String months is in another array) to print out, successfully. If program cannot find the corresponding object in set, it will print" Sorry. That city or that year is not in database. Cannot find the right month". This is showed in appendix.

7. If the city and year has already in database, record will be rewritten.

Beijing 2010 1 1 1 1 1 1 1 1 1 1 1 1 ⟶ Beijing 2010 2 2 2 2 2 2 2 2 2 2 2 2

```
This record has already in database, old one will be rewrotten.
If you want to stop, input No. Otherwise, input anything: 3
You choose: 3
Beijing 2010 2.0 2.0 2.0 2.0 2.0 2.0 2.0 2.0 2.0 2.0 2.0 2.0
```

Appendixes: (Java Code):

Input Style

```
What do you want to do now?
Insert record---1               Delete record---2                    Print records---3
Insert value---4                Insert a quarter value---5           Delete value---6
Calculate average value---7     Find one month rainfall value---8    Find wettest month---9
*************************************************************************************************
If you need to input Month, please input like this:
Jan   Feb   Mar   Apr   May   Jun   Jul   Aug   Sep   Oct   Nov   Dec
*************************************************************************************************
You choose: 1
City Name: Beijing
Year: 2010
The value of the 1th month: 0.1
The value of the 2th month: 0.1
The value of the 3th month: 0.2
The value of the 4th month: 0.2
The value of the 5th month: 0.3
The value of the 6th month: 0.3
The value of the 7th month: 0.1
The value of the 8th month: 0.5
The value of the 9th month: 0.7
The value of the 10th month: 0.3
The value of the 11th month: 0.1
The value of the 12th month: 0.2
If you want to stop, input No. Otherwise, input anything: w
You choose: 1
City Name: Beijing
Year: 2009
The value of the 1th month: 0.05
The value of the 2th month: 0.04
The value of the 3th month: 0.1
The value of the 4th month: 0.2
The value of the 5th month: 0.2
The value of the 6th month: 0.3
The value of the 7th month: 0.6
The value of the 8th month: 0.8
The value of the 9th month: 0.4
The value of the 10th month: 0.5
The value of the 11th month: 0.2
The value of the 12th month: 0.04
If you want to stop, input No. Otherwise, input anything: 4
You choose: 1
City Name: Suzhou
Year: 2010
The value of the 1th month: 0.1
The value of the 2th month: 0.3
The value of the 3th month: 0.2
The value of the 4th month: 0.3
The value of the 5th month: 0.4
The value of the 6th month: 0.5
The value of the 7th month: 0.6
The value of the 8th month: 0.5
The value of the 9th month: 0.7
The value of the 10th month: 0.9
The value of the 11th month: 0.7
The value of the 12th month: 0.4
```

1. Strong program testing

a) If user choose a wrong number, this will be showed:

```
What do you want to do now?
Insert record---1               Delete record---2                    Print records---3
Insert value---4                Insert a quarter value---5           Delete value---6
Calculate average value---7     Find one month rainfall value---8    Find wettest month---9
********************************************************************************************************
If you need to input Month, please input like this:
Jan   Feb   Mar   Apr   May   Jun   Jul   Aug   Sep   Oct   Nov   Dec
********************************************************************************************************
You choose: 10
You choose is wrong, please try again!
You choose:
```

b) There is no Beij city in database, so inserting a single value is not allowed.

```
You choose: 4
City Name: Beij
Year: 2010
Month: Jan
Value: 2
Sorry. That city or that year is not in database. Nothing is changed in database.
```

c) Quarter should be put in a right way. If user in a wrong way, this will be showed.

```
You choose: 5
City Name: Beijing
Year: 2010
Which quarter(one,two,three,four): 2
Input the first value in this quarter: 2
Input the second value in this quarter: 3
Input the third value in this quarter: 4
Sorry, only one, two, three, four are allowed.
```

d) User wants to delete a value in Tianjin 2010. Tianjin is not in database, so nothing is changed.

```
You choose: 6
City Name: Tianjin
Year: 2010
Month: Jun
Sorry. That city or that year is not in database. Nothing is changed in database.
```

e) Calculating the average must guarantee that record has already in database.

```
You choose: 7
City Name: Shanghai
Year: 2020
First Month: Jan
Last Month: Mar
Sorry. That city or that year is not in database.Average value can not be calculated.
The average rainfall value in Shanghai in 2020 is 0.0
```

f) Month must input in right way.

```
You choose: 8
City Name: Beijing
Year: 2010
Month: 1
Please input month in right way!
```

g) Database cannot find the city, so it cannot find the right value.

```
You choose: 8
City Name: Neimenggu
Year: 2010
Month: Dec
Sorry. That city or that year is not in database. Cannot find the right value.
```

h) The year cannot find database.

```
You choose: 9
City Name: Beijing
Year: 2088
Sorry. That city or that year is not in database. Cannot find the right month.
In Beijing, the wettest month in 2088 is Unk
```

2. Codes

Class Driver:

```
/*
 * Author:Jinglu.Yan
 * Date:2010/12/4
 * Aim:This Java program is handling a kind of small data base system. This Java code provides operations for inserting, extracting, deleting and displaying information from the system.
 * This is Driver class which contains only one method to control all process
 * There are three classes in this program.
 */
import java.util.*;
public class Driver{
	public static String city = "";
	public static int year = 0;
	public static double[] rainfallvalue = {0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0};
	/*
	 * This is main class.It is the begining for every program.
	 */
	public static void main(String[] args){
		Scanner input = new Scanner(System.in);
		System.out.println("What do you want to do now?");
		System.out.println("Insert record---1 Delete record---2 Print records---3");
		System.out.println("Insert value---4 Insert a quarter value---5 Delete value---6");
		System.out.println("Calculate average value---7 Find one month rainfall value---8 Find wettest month---9");

System.out.println("**********************************************************************************");
		System.out.println("If you need to input Month, please input like this:");
		System.out.println("Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec");

System.out.println("**********************************************************************************");
		while(true){//loop this procee many times
			System.out.print("You choose: ");
			int choose = input.nextInt();

if(choose==1||choose==2||choose==3||choose==4||choose==5||choose==6||choose==7||choose==8||choose==9){
				if(choose==1){
					double[]rainfallvalue = {0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0};
```

```
                    System.out.print("City Name: ");
                    String city = input.next();
                    System.out.print("Year: ");
                    int year = input.nextInt();
                    for(int n=1; n<13;n++){
                        System.out.print("The value of the " +n+"th month: ");
                        double rainfall = input.nextDouble();
                        rainfallvalue[n] = rainfall;//input values in array
                    }
                    Archive A = new Archive(city, year, rainfallvalue);
                    A.insert(city, year);//call insert method from Archive class
                }
                if(choose==2){
                    System.out.print("City Name: ");
                    String city = input.next();
                    System.out.print("Year: ");
                    int year = input.nextInt();
                    Archive A =new Archive(city, year, rainfallvalue);
                    A.delete(city,year);//call delete method from Archive class
                }
                if(choose==3){
                    Archive A =new Archive(city, year, rainfallvalue);
                    A.print();//call pring method from Archive class
                }
                if(choose==4){
                    System.out.print("City Name: ");
                    String city = input.next();
                    System.out.print("Year: ");
                    int year = input.nextInt();
                    System.out.print("Month: ");
                    String month = input.next();

if(month.equals("Jan")||month.equals("Feb")||month.equals("Mar")||month.equals("Apr")||month.equals("May")||month.equals("Jun")||month.equals("Jul")||month.equals("Aug")||month.equals("Sep")||month.equals("Oct")||month.equals("Nov")||month.equals("Dec")){
                        System.out.print("Value: ");
                        double value = input.nextDouble();
                        RainFallRecord R = new RainFallRecord(city, year, rainfallvalue);
                        R.insert(month,value);//call insert method from RainFallRecord class
                    }
                    else{
                        System.out.println("Please input month in right way!");//if user input a wrong month
                    }
```

```
                }
                if(choose==5){
                        System.out.print("City Name: ");
                        String city = input.next();
                        System.out.print("Year: ");
                        int year = input.nextInt();
                        System.out.print("Which quarter(one,two,three,four): ");
                        String quarter = input.next();
                        System.out.print("Input the first value in this quarter: ");
                        double first = input.nextDouble();
                        System.out.print("Input the second value in this quarter: ");
                        double second = input.nextDouble();
                        System.out.print("Input the third value in this quarter: ");
                        double third = input.nextDouble();
                        double[] value = {first, second, third};//new array to save three values
                        RainFallRecord R = new RainFallRecord(city, year, rainfallvalue);
                        R.insert(quarter, value);//call insert method from RainFallRecord class
                }
                if(choose==6){
                        System.out.print("City Name: ");
                        String city = input.next();
                        System.out.print("Year: ");
                        int year = input.nextInt();
                        System.out.print("Month: ");
                        String month = input.next();

if(month.equals("Jan")||month.equals("Feb")||month.equals("Mar")||month.equals("Apr")||month.equ
als("May")||month.equals("Jun")||month.equals("Jul")||month.equals("Aug")||month.equals("Sep")||
month.equals("Oct")||month.equals("Nov")||month.equals("Dec")){
                                RainFallRecord R = new RainFallRecord(city, year, rainfallvalue);
                                R.delete(month);//call delete method from RainFallRecord class
                        }
                        else{
                                System.out.println("Please input month in right way!");
                        }
                }
                if(choose==7){
                        System.out.print("City Name: ");
                        String city = input.next();
                        System.out.print("Year: ");
                        int year = input.nextInt();
                        System.out.print("First Month: ");
                        String firstmonth = input.next();
                        System.out.print("Last Month: ");
```

```
                    String lastmonth = input.next();

if(firstmonth.equals("Jan")||firstmonth.equals("Feb")||firstmonth.equals("Mar")||firstmonth.equals("Apr")||firstmonth.equals("May")||firstmonth.equals("Jun")||firstmonth.equals("Jul")||firstmonth.equals("Aug")||firstmonth.equals("Sep")||firstmonth.equals("Oct")||firstmonth.equals("Nov")||firstmonth.equals("Dec")||lastmonth.equals("Jan")||lastmonth.equals("Feb")||lastmonth.equals("Mar")||lastmonth.equals("Apr")||lastmonth.equals("May")||lastmonth.equals("Jun")||lastmonth.equals("Jul")||lastmonth.equals("Aug")||lastmonth.equals("Sep")||lastmonth.equals("Oct")||lastmonth.equals("Nov")||lastmonth.equals("Dec")){
                        RainFallRecord R = new RainFallRecord(city, year, rainfallvalue);
                        System.out.println("The average rainfall value in "+city+" in "+year+" is "+R.average(city, year, firstmonth, lastmonth));//call average method from RainFallRecord class
                    }
                    else{
                        System.out.println("Please input month in right way!");
                    }
                }
                if(choose==8){
                    System.out.print("City Name: ");
                    String city = input.next();
                    System.out.print("Year: ");
                    int year = input.nextInt();
                    System.out.print("Month: ");
                    String month = input.next();

if(month.equals("Jan")||month.equals("Feb")||month.equals("Mar")||month.equals("Apr")||month.equals("May")||month.equals("Jun")||month.equals("Jul")||month.equals("Aug")||month.equals("Sep")||month.equals("Oct")||month.equals("Nov")||month.equals("Dec")){
                        RainFallRecord R = new RainFallRecord(city, year, rainfallvalue);
                        System.out.println("The rainfall value in "+city+" in "+month+" "+year+" is "+R.rainfall(month));//call rainfall method from RainFallRecord class
                    }
                    else{
                        System.out.println("Please input month in right way!");
                    }
                }
                if(choose==9){
                    System.out.print("City Name: ");
                    String city = input.next();
                    System.out.print("Year: ");
                    int year = input.nextInt();
                    RainFallRecord R = new RainFallRecord(city, year, rainfallvalue);
                    System.out.println("In "+city+", the wettest month in "+year+" is "+R.wettest());//call wettest method from RainFallRecord class
```

```
                    }
                    System.out.print("If you want to stop, input No. Otherwise, input anything: ");
                    String answer = input.next();
                    if(answer.equals("No")){
                        System.exit(0);//stop this program
                    }
                }
                else{
                    System.out.println("You choose is wrong, please try again!");//if input a wrong choice
                }
            }
        }
}
```

Class RainFallRecord*:*

```
/*
 * Author:Jinglu.Yan
 * Date:2010/12/4
 * Aim:This class can calculate average value and find the wettest month. It also can change single value in records.
 * There are eight methods.
 */
import java.util.*;
public class RainFallRecord
{
    private String city;
    private int year;
    private double[] rainfallvalue;//they are different type of variables.
    private static String months[] = {"Unk", "Jan", "Feb", "Mar", "Apr","May", "Jun", "Jul", "Aug","Sep", "Oct", "Nov", "Dec"};
    Archive A = new Archive(city, year,rainfallvalue);//connect with Archive class. A is an object for Archive
    /*
     * This is the first constructor.Only give a initial values to that variables.
     */
    public RainFallRecord(){
        String city = "Beijing";
        int year = 2000;
        double[] rainfallvalue = {0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0};
    }

    /*
     * This is the second constructor. This constructor can get the values from other class into this class to process.
```

```
     */
    public RainFallRecord(String city, int year, double[] rainfallvalue){
        this.city = city;
        this.year = year;
        this.rainfallvalue = rainfallvalue;
    }

    /*
     * This method calculates average value, and judges if that record in database.
     */
    public double average(String city, int year,String firstmonth, String lastmonth){
        HashSet records = A.getSet();
        double sum = 0.0;
        int first=0, last=0;
        int c=0;//they are different type of variables, and give a initial values to that variables.
        for(Object o: records){//records set is an object o. use for sstatement to find elements in set.
            Archive a = (Archive) o;//cast object o to object a which type is Archive
            double[] value = a.getValue();//get array values from Archive class
            for(int m=0; m<13;m++){//loop 12 times
                if(firstmonth.equals(months[m])){//find the String month position to get the integer
                    first = m;//find the first month from String to integer
                }
            }
            for(int z=0;z<13;z++){
                if(lastmonth.equals(months[z])){
                    last = z;//find the second month from String to integer
                }
            }
            if((a.getCity()).equals(city)&&a.getYear()==year){
                for(int n=first; n<last+1;n++){
                    sum = sum + value[n];//add all elements in array
                }
                c=c+1;//judge if this if process
            }
        }
        if(c==0){//if that judge does not process
            System.out.println("Sorry. That city or that year is not in database.Average value can not be
calculated.");
        }
        return sum/(last-first+1);//return average value
    }

    /*
     * This method can find the right value, and judges if that record in database.
```

```
     */
    public double rainfall(String month){
        HashSet records = A.getSet();//get set from Archive class
        double rainfall =0.0;
        int c=0;
        for(Object o: records){
            Archive    a = (Archive ) o;
            double[] value = a.getValue();
            if((a.getCity()).equals(city)&&a.getYear()==year){
                for(int n=0; n<13;n++){
                    if(month.equals(months[n])){//find the right position to find the corresponding
value
                        rainfall =    value[n];//give the value to rainfall variable
                    }
                }
                c=c+1;
            }
            if(c==0){
                System.out.println("Sorry. That city or that year is not in database. Cannot find the
right value.");
            }
        }
        return rainfall;//return the right value

    }
    /*
     * This method can delete single value in records, and judges if that record in database.
     */
    public void delete(String month){
        HashSet records = A.getSet();
        int c = 0;
        for(Object o: records){
            Archive a = (Archive)o;
            double[] value = a.getValue();
            if((a.getCity()).equals(city)&&a.getYear()==year){
                for(int n=0; n<13;n++){
                    if(month.equals(months[n])){
                        value[n] = 0.0;//give 0.0 to that value in order to let use know that value has
been deleted
                    }
                }
                c=c+1;
            }
        }
```

```
            if(c==0){
                System.out.println("Sorry. That city or that year is not in database. Nothing is changed in
database.");
            }

        }
        /*
         * This method can insert single value in records, and judges if that record in database.
         */
        public void insert(String month, double value){
            HashSet records = A.getSet();
            int c=0;
            for(Object o: records){
                Archive a = (Archive)o;
                double[] value1 = a.getValue();
                if((a.getCity()).equals(city)&&a.getYear()==year){
                    for(int n=0; n<13;n++){
                        if(month.equals(months[n])){
                            value1[n] = value;//give(insert) the parameter value to that value in record
                        }
                    }
                    c=c+1;
                }
            }
            if(c==0){
                System.out.println("Sorry. That city or that year is not in database. Nothing is changed in
database.");
            }

        }
        /*
         * This method inserts value by quarter,and judges if that record in database.
         */
        public void insert(String quarter, double[] value){
            HashSet records = A.getSet();
            int c=0;
            for(Object o: records){
                Archive a = (Archive)o;
                double[] value1 = a.getValue();

if(quarter.equals("one")||quarter.equals("two")||quarter.equals("three")||quarter.equals("four")){//if
user input the right quarter
                    if((a.getCity()).equals(city)&&a.getYear()==year){
                        if(quarter.equals("one")){
```

```
                        for(int n=1; n<4; n++){
                            value1[n]=value[n-1];//rewrite the first quarter value
                        }
                    }
                    if(quarter.equals("two")){
                        for(int n=4; n<7; n++){
                            value1[n]=value[n-4];//rewrite the second quarter value
                        }
                    }
                    if(quarter.equals("three")){
                        for(int n=7; n<10; n++){
                            value1[n]=value[n-7];//rewrite the third quarter value
                        }
                    }
                    if(quarter.equals("four")){
                        for(int n=10; n<13; n++){
                            value1[n]=value[n-10];//rewrite the fourth quarter value
                        }
                    }
                    c=c+1;
                }
            }
            else{//user input wrong quarter
                System.out.println("Sorry, only one, two, three, four are allowed.");
            }
        }
        if(c==0){
            System.out.println("Sorry, that city or that year is not in database. Therefore, there is no
change.");

        }
    }
    /*
     * This method finds wettest month, and judges if that record in database.
     */
    public String wettest(){
        HashSet records = A.getSet();
        double wettest = rainfallvalue[0];
        int numbermonth = 0;
        int c=0;
        for(Object o: records){
            Archive a = (Archive)o;
            double[] value = a.getValue();
            if((a.getCity()).equals(city)&&a.getYear()==year){
```

```
                    for(int n=1; n<13; n++){
                        wettest = Math.max(value[n],wettest);//find the maximum value in array
                    }
                    for(int m=0;m<13; m++){
                        if(wettest==value[m]){
                            numbermonth = m;//find corresponding position to two array
                        }
                    }
                    c=c+1;
                }
            }
            if(c==0){
                System.out.println("Sorry. That city or that year is not in database. Cannot find the right
month.");
            }
            return months[numbermonth];//return the String in months array
        }
}
```

Class Archive:

```
/*
 * Author:Jinglu.Yan
 * Date:2010/12/4
 * Aim:this class is a mini database, records are all saved in this calss.
 * There are nine methods.Two of them are constructor, others four are get the value or set in this class.
 */
import java.util.*;
public class Archive
{
    static HashSet records = new HashSet();//create a new HashSet named records
    private String city;
    private int year;
    private double[] rainfallvalue;//they are different type of variavles.
    /*
     * This is a first constructor. Only give a initial values to that variables.
     */
    public Archive(){
        String city = "Beijing";
        int year = 2000;
        double[] rainfallvalue = {0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0};
    }
    /*
     * This is the second constructor. This constructor can get the values from other class into this class
to process.
```

```
     */
    public Archive(String city, int year, double[] rainfallvalue){
        this.city = city;
        this.year = year;
        this.rainfallvalue = rainfallvalue;//there are three parameters.
    }
   /*
    * This method can insert records, and it also can judge if the record has already in database.
    */
    public void insert(String city, int year){
        int c =0;
        for(Object o: records){//records set is an object o. use for sstatement to find elements in set.
            Archive a = (Archive)o;//cast object o to object a which type is Archive
            if((a.getCity()).equals(city)&&a.getYear()==year){//if there is an object which has the same
city and year in set
                System.out.println("This record has already in database, old one will be rewrotten.");
                records.remove(o);//remove that record firstly
                records.add(new Archive(city,year,rainfallvalue));//rewrite that record as an object use
add()
                c=c+1;//count and judge if this if statement run
            }
        }
        if(c==0){//if that if do not run
            records.add(new Archive(city,year,rainfallvalue));//add that record directly
        }
    }
    /*
     * This method can delete records, and judge if that record has already in database
     */
    public void delete(String city, int year){
        int c =0;
        for(Object o: records){//records set is an object o. use for sstatement to find elements in set.
            Archive a = (Archive)o;//cast object o to object a which type is Archive
            if((a.getCity()).equals(city)&&a.getYear()==year){//if there is an object which has the same
city and year in set
                records.remove(o);//delete the record
            }
            c=c+1;//count and judge if this if statement run
        }
        if(c==0){//if that if do not run
            System.out.println("That city or that year is not in database, so the records do not
change.");
        }
    }
```

```
    /*
     * This method can print all records in database.
     */
    public void print(){
        getSet();//get the set firstly.
        for(Object o: records){//records set is an object o. use for sstatement to find elements in set.
            Archive s = (Archive) o;//cast object o to object a which type is Archive
            System.out.print(s.city+" ");//print city once for one object
            System.out.print(s.year+" ");//print year once for one object
            for(int n =1; n<13;n++){//print the elements in array seperately
                System.out.print(s.rainfallvalue[n]+" ");
            }
            System.out.println();
        }
    }
    /*
     * This method can get set.
     */
    public HashSet getSet(){
        return records;
    }
    /*
     * This method gets city name
     */
    public String getCity(){
        return city;
    }
    /*
     * This method gets year
     */
    public int getYear(){
        return year;
    }
    /*
     * This method gets the array
     */
    public double[] getValue(){
        return rainfallvalue;
    }
}
```