

**Задание:** Разработать класс для реализации указанного объекта. Построить его диаграмму и составить программный код.

Все поля классов должны быть частными (private) или защищенными (protected). Методы не должны содержать операций ввода/вывода, за исключением процедуры, единственной задачей которой является вывод информации об объекте на экран.

Объект – результат однодневного наблюдения за погодой. Поля: дата, температура в градусах Цельсия и относительная влажность в процентах. Методы: конструктор, процедура вывода на экран результатов наблюдения и функции, возвращающие содержимое полей по запросу.

### ***Часть 1. Массив объектов***

Разработать тестирующую программу, в которой дневник наблюдений за погодой за N дней представлен массивом объектов (наблюдений). Для заданной совокупности наблюдений должна рассчитываться средняя температура и влажность.

### ***Часть 2. Композиция***

Разработать и реализовать диаграмму классов для описанных объектов предметной области, используя механизмы композиции.

Объект – дневник погоды. Содержит результаты наблюдений за N дней. Методы объекта должны позволять: инициализировать дневник списком наблюдений, выводить на экран результаты всех наблюдений, возвращать по запросу число дней наблюдений, определять максимальную и среднюю температуру и влажность.

**Выполнение:** Диаграммы классов WeatherData и WeatherDiary изображены на рисунке 1

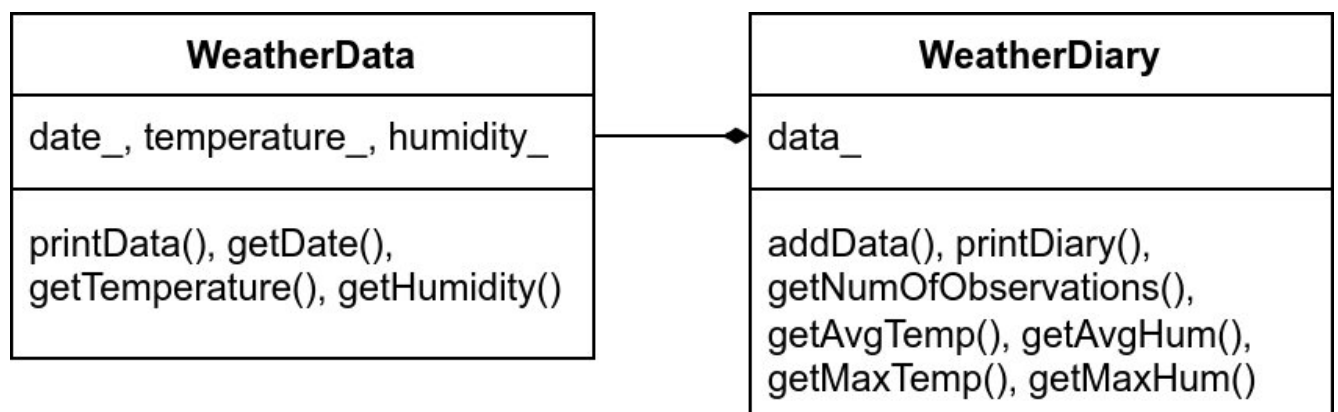


Рисунок 1 — диаграммы классов WeatherData и WeatherDiary

Заголовочный файл класса WeatherData изображен на рисунке 2

```
#ifndef WEATHERDATA_H
#define WEATHERDATA_H
#include <string>

using namespace std;

class WeatherData {
public:

    WeatherData(const string& date, int temperature, int humidity);

    void printData();

    string getDate();

    int getTemperature();

    int getHumidity();

private:
    string date_;
    int temperature_;
    int humidity_;
};

#endif //WEATHERDATA_H
```

Рисунок 2 — Заголовочный файл класса WeatherData

Заголовочный файл класса WeatherDiary изображен на рисунке 3

```
#ifndef WEATHERDIARY_H
#define WEATHERDIARY_H
#include <vector>

#include "WeatherData.h"

using namespace std;

class WeatherDiary {
public:
    WeatherDiary();
    WeatherDiary(WeatherData weatherObservations[], int N);

    void addData(const WeatherData& weather_data);

    void printDiary();

    int getNumOfObservations();

    double getAvgTemp();

    double getAvgHum();

    int getMaxTemp();

    int getMaxHum();

private:
    vector<WeatherData> data_;
};
#endif //WEATHERDIARY_H
```

Рисунок 3 — Заголовочный файл класса WeatherDiary

Основная тестирующая программа изображена на рисунке 4

```
#include <iostream>

#include "WeatherData.h"
#include "WeatherDiary.h"
using namespace std;

int main() {
    cout << "Enter N days: ";
    int N;
    cin >> N;
    if (N ≤ 0) {
        cout << "N must be > 1" << endl;
        return -1;
    }

    WeatherDiary wd;
    string date;
    int temperature, humidity;
    for (int i = 0; i < N; ++i) {
        cout << "Enter date: ";
        cin >> date;
        cout << "Enter temperature: ";
        cin >> temperature;
        cout << "Enter humidity: ";
        cin >> humidity;
        wd.addData(WeatherData(date, temperature, humidity: humidity));
    }

    cout << "Diary: " << endl;
    wd.printDiary();

    cout << "Average temperature: " << wd.getAvgTemp() << endl;
    cout << "Max temperature: " << wd.getMaxTemp() << endl;
    cout << "Average humidity: " << wd.getAvgHum() << endl;
    cout << "Max humidity: " << wd.getMaxHum() << endl;
    return 0;
}
```

Рисунок 4 — Основная тестирующая программа

**Результат выполнения программы:** результат выполнения программы изображен на рисунке 5

```
/home/sergey/МГТУ/АиП/НВ/04/cmake-build-debug/04
Enter N days: 3
Enter date: 10/11/22
Enter temperature: 10
Enter humidity: 50
Enter date: 11/11/22
Enter temperature: 30
Enter humidity: 60
Enter date: 12/11/22
Enter temperature: 50
Enter humidity: 70
Diary:
Weather date 10/11/22
Temperature C: 10
Humidity: 50
Weather date 11/11/22
Temperature C: 30
Humidity: 60
Weather date 12/11/22
Temperature C: 50
Humidity: 70
Average temperature: 30
Max temperature: 50
Average humidity: 60
Max humidity: 70

Process finished with exit code 0
```

Рисунок 5 — результат выполнения программы

**Вывод:** В ходе лабораторной работы были успешно применены принципы объектно-ориентированного программирования. Реализованы классы, обеспечивающие хранение, обработку и анализ данных наблюдений за погодой. Работа со списком объектов и использование композиции позволили разработать функциональный и гибкий инструмент для работы с погодными данными.