

Name: _____ Abgabetermin: KW 17

Mat.Nr: _____ Punkte: _____

Übungsgruppe: _____ korrigiert: _____

Aufwand in h: _____

Beispiel 1 (8 Punkte) Klasse Wetterstation: Schreiben Sie eine Klasse zur Darstellung der Daten einer Wetterstation (Temperatur und Luftfeuchtigkeit) mit folgender Schnittstelle:

```
1 class WeatherStation {
2 public:
3
4     // Constructor
5     WeatherStation(std::string const &name="", double celsius=0, double humidity=0);
6
7     // Accessor methods
8     std::string const &GetName() const;
9     void SetName(std::string const &name);
10    double GetCelsius() const;
11    void SetCelsius(double c);
12    double GetFahrenheit() const;
13    void SetFahrenheit(double f);
14    double GetHumidity() const;
15    void SetHumidity(double h);
16
17    // Print the data of the weather station (Name, °C, °F, humidity)
18    void Print(std::ostream &out) const;
19
20 private:
21     std::string mName;    // Name of the weather station
22     double mCelsius;     // Temperature in °C
23     double mHumidity;    // Humidity in %
24 };
25
26 inline std::ostream &operator <<(std::ostream &out, WeatherStation const &ws) {
27     ws.Print(out);
28     return out;
29 }
```

Formeln zur Umrechnung zwischen Celsius (T_C) und Fahrenheit (T_F):

$$T_C = (T_F - 32)/1.8 \quad T_F = T_C * 1.8 + 32$$

Testen Sie die Klasse `WeatherStation` ausführlich und unabhängig vom 2. Beispiel.

Beispiel 2 (16 Punkte) Verwaltung von Wetterstationen: Schreiben Sie eine Klasse zur Verwaltung von Wetterstationen mit der folgenden Schnittstelle:

```
1 class WeatherStations {
2 public:
3
4     // Constructor
5     WeatherStations(size_t maxNr=10);
6
7     // Copy constructor
8     WeatherStations(WeatherStations const &ws);
9
10    // Destructor
11    ~WeatherStations();
12
13    // Assignment operator
14    WeatherStations &operator=(WeatherStations const &ws);
15
16    // Adds a new weather station at the end (if possible)
17    bool Add(WeatherStation const &ws);
18
19    // Returns the number of weather stations
20    size_t GetNrStations() const;
21
22    // Prints all weather stations
23    void PrintAll(std::ostream &out) const;
24
25    // Prints coldest/warmest weather station
26    void PrintColdest(std::ostream &out) const;
27    void PrintWarmest(std::ostream &out) const;
28
29 private:
30
31     // member variables
32     ...
33
34     // private helper methods
35     ...
36 };
```

Speichern Sie die Wetterstationen in einem dynamisch erzeugten Array ab, dessen Größe (=maximale Anzahl von Wetterstationen) im Konstruktor übergeben wird.

Achten Sie im Testtreiber darauf, dass alle Methoden ausführlich getestet werden.

Allgemeine Hinweise: Legen Sie bei der Erstellung Ihrer Übung großen Wert auf eine **saubere Strukturierung** und auf eine **sorgfältige Ausarbeitung!** Verwenden Sie immer **Module**, um den Testtreiber und die eigentliche Implementierung zu trennen! Dokumentieren Sie alle Schnittstellen und versehen Sie Ihre Algorithmen an entscheidenden Stellen ausführlich mit **Kommentaren!** Testen Sie ihre Implementierungen ausführlich! Geben Sie **Lösungsideen** an!