

Aufgaben 12 Wiederholung C++ Klassen

1.

Erstellen Sie die Klasse **Akte**, die als Datenfelder die Angaben zum Aktenzeichen (std:string) und der Laufzeit des erfassten Vorgangs (in Jahren) enthält.

Eine weitere von der Klasse Akte abgeleitete Klasse **AkteX** soll neben den bereits in der Klasse Akte enthaltenen Angaben einen Aktencode (eine dreistellige ganze Zahl) enthalten. Ergänzen Sie beide Klassen um die Konstruktoren und weitere Methoden, die die folgenden Anweisungen in der main-Funktion ermöglichen würden:

```
Akte akte1("08-15",2), akte2;
std::cout << "Voraussichtliche Laufzeit: "<<akte1.getLaufzeit()<<"\n";
akte1.setLaufzeit(4);
akte2.ausgabe();
AkteX aktex("08-16",99,816);
aktex.ausgabe();
std::cout <<aktex.getAktencode()<<"\n";
```

Die Funktion `ausgabe` beider Klassen soll alle in einem Objekt gespeicherten Angaben auf dem Standard-Output ausgeben. In der Klasse **AkteX** sollen aber sowohl die Funktion `ausgabe` als auch die Funktion `getAktencode` statt des gespeicherten Aktencodes dessen Quersumme ausgeben bzw. liefern.

Hinweis: bei den Objekten der Klasse `std:string` ist (anders als bei `char[]`) die Zuweisung mit `=` möglich.

2.

Im Rahmen der Messdatenerhebung mit Sensoren soll eine Klasse für Temperatur-Messungen erstellt werden. Für eine bessere Erweiterbarkeit wird außerdem die allgemeine Basisklasse für Sensoren jeder Art benötigt.

In der Basisklasse **Sensor** sollen folgende Daten gespeichert werden:

- eine ID,
- maximal und minimal messbare Werte und
- ein aktuell gemessener Wert

Um den Zugriff auf die private-Attribute zu gewährleisten sind die entsprechenden `get-` und `set-`Methoden zu definieren.

Definieren Sie weiterhin für die Klasse einen Konstruktor und die Funktion `status`, die die Sensor-Informationen auf die Standard-Ausgabe ausgibt.

Die Klasse **TemperaturSensor** verfügt außerdem über die Methode `messen`, wo der aktuell gemessene Wert zufällig erzeugt wird.

An den Konstruktor der Klasse **TemperaturSensor** ist nur die ID zu übergeben. Sonst ist davon auszugehen, dass der messbare Bereich die Temperaturen zwischen -20 und 100 Grad umfasst.

Erstellen Sie in der `main`-Funktion zwei Objekte der Klasse **TemperaturSensor** und lassen Sie diese in einer Schleife die Temperatur „messen“ und ihren Sensor-Status ausgeben.

3. (Zusatz)

Die Klasse **Moor** enthält als Datenfelder die Angaben zu den Faktoren, die die Bildung und das Bestehen von Mooren beeinflussen:

- die Dauer des Bodenfrostes (in Tagen pro Jahr),
- die maximale Eindringtiefe des Bodenfrostes (in cm) und
- den pH-Wert des Wassers

sowie die Methode `getPrognose()`, welche die prognostizierte Fortdauer des Moors mit Hilfe der Methode `simulation` der Klasse `Utility` berechnet.

```
#include "utility.h"
class Moor
{
    private:
        int frost_dauer;
        double frost_tiefe,pH_wert;
    public:
        double getPrognose()
        {
            return Utility::simulation(frost_dauer,frost_tiefe,pH_wert);
        }
};
```

Für das Bestehen eines Hangmoors ist außerdem die Hangneigung wichtig. Definieren Sie die Klasse **Hangmoor**, die von der Klasse `Moor` abgeleitet wird und zusätzlich die Hangneigung (in Grad) beinhaltet.

Vervollständigen Sie außerdem die Klasse **Moor** um die Methoden, die folgende Anweisungen in der `main`-Funktion ermöglichen würden:

```
Moor mo1(30,30.0,4.5),
      mo2(29,10.0,5.5);

mo1.setPHWert(6.0);
mo1.ausgabe();
Hangmoor mo3(29,10.0,5.5,15);
mo3.ausgabe();
std::cout<< mo3.getNeigung()<<" "<<mo3.getPrognose()<<"\n";
```

Die Funktion `ausgabe()` beider Klassen soll alle in einem Objekt gespeicherten Angaben auf dem Standard-Output ausgeben.

Die Funktion `getPrognose()` der Klasse `Hangmoor` soll den mit der Funktion `getPrognose()` von der Klasse `Moor` berechneten Wert mit dem Faktor $\cos(\text{neigung})$ multiplizieren. Falls die Neigung 45 Grad übersteigt, soll der Wert **0** zurückgeliefert werden. Beachten Sie die erforderliche Umrechnung von Grad in Radiant (Bogenmaß).

Tipp: Im Übungsordner finden Sie eine `utility.h`-Datei zum testen (ohne wissenschaftlichen Sinn oder Korrektheit :-D).