Aufgabenstellung Wetterstation

**Generelle Information:**  
Beachten Sie die gestellte Hilfeleistung sowie die einheitliche Benennung der Steuerelemente.

In der Stadt Altenberg werden im Winter einige Skihänge und Sessellifte betrieben. Aufgrund der klimatischen Lage ist Altenberg aber nicht „schneesicher“. Die Stadtverwaltung plant, zusammen mit dem Deutschen Wetterdienst, eine Wetterstation am Bahnhof zu installieren. So soll die Schneesicherheit besser vorausgesagt werden und die Zufriedenheit der Besucher im Winter verbessert werden.

# Aufgabe 1 - Formulardesigner

Um die Daten für potenzielle Besucher besser auslesbar zu machen, sollen sie testweise in eine Formularanwendung implementiert werden. Da zuerst ein Prototyp entwickelt wird werden vorerst nur folgende Daten gemessen:

* Temperatur in °C (Grad Celsius)
* Luftfeuchtigkeit in % (Prozent)
* Luftdruck in hPa (Hektopascal)
* UV-Stärke in µW/cm (micro Watt pro cm)
* Lichteinfall in lx (Lux)
* Abfrage Uhrzeit in hh:mm (24-Stunden:Minuten)

## Aufgabe 1.1 (3 BE)

Für den generellen Entwurf liegt bereits die Vorarbeit eines Kollegen vor (Vorlage 1).

Erweitern Sie das Formular so, dass alle gemessenen Parameter angezeigt werden können.

## Aufgabe 1.2 (2 BE)

Der DWD plant in Zukunft ein ganzes Netz an Wetterstationen im Erzgebirge zu implementieren.

Erweitern Sie das Formular, um eine Möglichkeit jederzeit während der Laufzeit die Abfrage URL zu verändern.

# Aufgabe 2 – Implementierung

Um die ausgelesenen Daten einfach zu verarbeiten, sollen sie als Objekt der Klasse „WeatherData“ gespeichert werden.

Da Informatiker beim Deutschen Wetterdienst und Freistaat rar sind, wurde durch einen Praktikanten des BSZET bereits ein Klassendiagram entworfen (Vorlage 2).

## Aufgabe 2.1 (5 BE)

Setzen sie das Klassendiagram „WeatherData“ im Projekt um.

## Aufgabe 2.2 (3 BE)

Damit ein Mitarbeiter nun den Prototyp der Wetterstation testen kann, stellt ein Mikrocomputer (Raspberry PI) diese nun als JSON – String unter folgender URL bereit:

http://api.norku.de/api/v1/senseBox/weather

Implementieren sie innerhalb der Klasse „WeatherData“ die Methode „GetData()“ die den String herunterlädt und während der Laufzeit in einer Variable speichert. Hierfür soll die von Microsoft vorgegebene Klasse „System.Net“ und verwendet werden.

**Zusätzliches Material:**

<https://learn.microsoft.com/en-us/dotnet/api/system.net.webclient.downloadstring?view=net-8.0>

**Tipp:**

Um Laufzeitfehler zu vermeiden, nutzen Sie folgende Quelle.

<https://learn.microsoft.com/en-us/dotnet/csharp/language-reference/statements/using>

## Aufgabe 2.3 (2 BE)

Damit die Daten nun besser verarbeitet werden können, soll aus dem gespeicherten String ein Objekt der Klasse „WeatherData“ erzeugt werden.  
Verwenden Sie hierfür die Bibliothek „Newtonsoft.Json“, diese wird über die NUGET-Paketverwaltung installiert.

Zusätzliches Material:

<https://www.newtonsoft.com/json/help/html/DeserializeObject.htm>

## Aufgabe 2.4 (3 BE)

Überprüfen sie das nun erstellte Objekt in jedem seiner Eigenschaften auf falsche bzw. unrealistische Sensorwerte.

Die Interpretation von unrealistisch ist freigestellt sollte sich jedoch an der Realität, dem Beispiel und mathematischer Eindeutigkeit orientieren.

Sind alle Werte valide, geben sie das Objekt zurück. Bei einem falschen Wert geben sie NULL zurück.

## Aufgabe 2.5 (5 BE)

Um das Auslesen in Zukunft auch fremden Programen zu ermöglichen, soll die Methode „GetData“ und die Klasse „WeatherData“ in eine eigene DLL verschoben werden.

Erstellen sie eine eigene DLL mit dem oben genannten Inhalt. Ändern sie ihr Formular nun so ab, das dieses nur noch eine „Anfrage“ an die DLL stellt und diese die Daten zurück gibt.

# Aufgabe 3 – Debugging und Benutzersteuerung

Da nicht nur Informatiker am Projekt arbeiten, sollen Laufzeitfehler bestmöglich abgefangen werden. Dem Benutzer soll hierbei einfach und verständlich erklärt werden, wo der Fehler liegt.

## Aufgabe 3.1 (1 BE)

Erweitern Sie die Formularanwendung so, dass Laufzeitfehler korrekt abgefangen werden.

## Aufgabe 3.2 (3 BE)

Um dem Nutzer die Fehler besser verständlich zu machen, soll ein Fenster mit dem Fehlertext erscheinen. Verwenden sie hierfür die von „Windows.Froms“ vorgegebene Klasse „MessageBox“.

Erweitern sie ihr Projekt um die genannten Fenster. Nutzen sie hierfür folgendes Material zur Erstellung:

<https://www.c-sharpcorner.com/uploadfile/mahesh/understanding-message-box-in-windows-forms-using-C-Sharp/>

<https://learn.microsoft.com/en-us/dotnet/api/system.windows.forms.messagebox?view=windowsdesktop-8.0>

## Aufgabe 3.3 (4 BE)

Damit Laufzeitfehler übersichtlicher strukturiert sind sollen zwei Programmspezifische Fehler erstellt werden. Dafür gibt es bei C# die Klasse „Exception“, von ihr werden alle spezifischen Fehlermeldungen vererbt.

Implementieren sie eine Klasse mit dem Namen „SensorException“ und eine mit dem Namen „NetworkParsingException“.

Passen sie danach ihr Programm so an, dass bei falschen Sensorwerten eine „SensorException“ und bei fehlerhafter Verbindung zum Webserver eine „NetworkParsingException“ ausgelöst werden.

Nutzen sie hierfür folgendes Material:

<https://stackoverflow.com/questions/2200241/in-c-sharp-how-do-i-define-my-own-exceptions>

<https://stackoverflow.com/questions/5318723/how-to-create-a-custom-c-sharp-exception>

<https://learn.microsoft.com/en-us/dotnet/standard/exceptions/how-to-create-user-defined-exceptions>

<https://www.tutorialsteacher.com/csharp/custom-exception-csharp>

# Aufgabe 4 – Webcam (Zusatz für schnelle)

Für Skifahrer sind die Sichtverhältnisse sowie der allgemeine Zustand der Piste enorm wichtig. Um ihnen weitere Informationen zu liefern, wird eine Kamera an den Mikrocomputer angeschlossen. Sie macht aller 10 Sekunden ein Bild.

## Aufgabe 4.1 (2 BE)

Erweitern sie ihre Formularanwendung, um eine Möglichkeit ein Bild anzuzeigen.

## Aufgabe 4.2 (6 BE)

Das aktuelle Bild wird unter folgender URL bereitgestellt:

http://api.norku.de/api/v1/senseBox/liveview

Erweitern Sie ihre Formularanwendung so, dass bei Datenabfrage auch das aktuelle Bild durch die DLL abgefragt wird.

Erweitern sie ihre DLL um eine Methode „GetPicture“ welche das aktuelle Bild herunterlädt und als Bitmap übergibt.

Gesamt: / 39  
 Notenpunkte: