

Wetterstationssimulator

Prüfung 3, MLZ

Fach	CS1
Klasse	20I/21NI
Lehrperson	Sagi Nedunkanal
Semester	Herbst 2021
Prüfungsbeginn	24.01.2022
Abgabetermin	29.01.2022, 23:59 Uhr

1. Ausgangslage

Es soll ein Wetterstationssimulator zur Generierung und Aufzeichnung von Wetterdaten realisiert werden. Die Wetterdaten sollen vom Benutzer mittels GUI erfasst, verändert, aber auch automatisch simuliert werden können. Jede Sekunde soll ein neuer Datensatz erfasst werden. Alle Datensätze werden in einem zweiten Bereich in Form eines Logs angezeigt.

Es soll eine Windows-Forms-Applikation entwickelt werden.

Die Aufgabenstellung ist nicht in allen Details ausspezifiziert. Bei Unklarheiten wird erwartet, dass beim Dozenten nachgefragt wird.

Symbolbild (keine Piktogramme und Diagramme nötig)



Quelle: <https://www.interdiscount.ch/de/haushalt-k%C3%BCche/haustechnik-stromversorgung/wetterstationen--c688000/tfa-wetterstation-funk-life--p0002361751> (23.01.2022)

2. Anforderungen

2.1. Menüführung

Schreibe eine WinForms-Applikation. Die Applikation soll auf der Oberfläche folgende Funktionen anbieten:

1. Wetterdaten erfassen
2. Wettersimulation starten und stoppen
3. Wetterdaten-Logbuch anzeigen
4. Wetterdaten editieren
5. Programm beenden

Erstelle eine Bedienoberfläche, die optisch ansprechend und intuitiv zu bedienen ist.

2.2. Wetterdaten erfassen

In diesem Teil der Applikation kann man Wetterdaten eines bestimmten Tages erfassen. Ein Datensatz besteht aus:

1. Datum [Tag]
2. Temperatur [°C]
3. Windgeschwindigkeit [km/h]
4. Niederschlagsmenge [l/m²]
5. Niederschlagswahrscheinlichkeit [%]
6. Luftfeuchtigkeit [%]

Über einen Knopf kann man die eingegebenen Werte erfassen.

2.3. Wettersimulation starten und stoppen

Vorher unter “Wetterdaten erfassen” konnte der Benutzer einen Datensatz von Hand eingeben und erfassen. Nun sollen auch Zufallszahlen erstellt und als Datensatz hinzugefügt werden können.

Beim Klick auf einen Knopf “Starten” werden die Felder mit Zufallszahlen abgefüllt.

Jede Sekunde wird ein Datensatz für ein bestimmtes Datum mit neuen Zufallszahlen erfasst.

Die Zufallszahlen sollen maximal +/-5 Einheiten von dem zuletzt erfassten Wert abweichen, damit wir keine riesigen Sprünge in der Zeitreihe haben.

Beispiel:

Diese Reihe ist ok 2, 7, 6, 4, 9.

Diese Reihe ist nicht ok 2, 7, 20, 24, 21, weil der Sprung von 7 auf 20 grösser 5 ist.

Beim Klick auf einen Knopf “Stoppen” stoppt der Zufallszahlengenerator.

2.4. Wetterdaten-Logbuch anzeigen

In einem Bereich sollen alle erfassten Datensätze kontinuierlich angezeigt werden. Als Control kann man eine Multiline-Textbox verwenden.

Beispiel:

```
Datum;Temperatur [°C];Windgeschwindigkeit  
[km/h];Niederschlagsmenge [l/m²];  
Niederschlagswahrscheinlichkeit [%];Luftfeuchtigkeit [%]  
23.01.2022;10,1;0;10;30  
24.01.2022;15,2;1;15;35  
25.01.2022;12,3;0;13;32  
26.01.2022;11,0;1;17;34
```

2.5. Wetterdaten editieren

In einem Bereich hat es eine ComboBox, mit welcher man ein Datum auswählen kann, z.B. 24.01.2022 und dann wird der Datensatz angezeigt. Der Datensatz kann editiert und gespeichert werden.

Die ComboBox und die Anzeigefelder sollen mittels Data-binding verbunden werden, damit sich alles automatisch anpasst.

2.6. Programm beenden

Es gibt einen Menüeintrag Datei → Beenden über welche die Applikation beendet werden kann.

2.7. Listen von Objekten

Zur Datensätzen sollen in einer geeigneten Collection-Klasse gehalten werden.

3. Code

Die bei den Hausaufgabenbesprechungen und Codereviews besprochenen Regeln sollen eingehalten werden.

4. Softwaredesign

Es sollen geeignete Klassen entworfen werden, um die nötigen Funktionen, Daten und GUI-Controls zu verwalten.

5. Qualitätssicherung

5.1. Testdaten

Beim Applikationsstart soll bereits eine Ausgangslage mit mindestens 3 Datensätzen erfasst sein.

Vereinfachung: Die Testdaten dürfen hartkodiert aus einer Fabrikmethode kommen.

5.2. Validierung

Es sollen geeignete Validierungen durchgeführt werden, so dass der Benutzer immer weiss, wenn etwas falsch ist und so, dass die Applikation nicht abstürzt.

Die Validierungen sollen mit den im Unterricht kennengelernten Methoden umgesetzt werden, z.B. Casting, TryParse oder try-catch.

6. Dokumentation

Es soll eine minimale Dokumentation erstellt werden:

- Lösungsskizze (kann von Hand gezeichnet, fotografiert und als JPEG abgelegt werden; kein im Nachhinein generiertes Klassendiagramm)

- Screenshots aller Fenster/Masken
- Mindestens 3 Commits

7. Bewertungskriterien

1. Umsetzung der funktionalen Anforderungen (1/4)
 2. Bewertung des GUIs (1/4)
 3. Bewertung des Codes (1/4)
 4. Testen der Applikation (1/4)
- Erlaubt: Internet als Quelle, Beispiele aus dem Unterricht usw.
 - Nicht erlaubt: Kopieren grösserer Codemengen, Mitarbeit Dritter usw.
 - Selbstkontrolle: «Kann ich jede Zeile Code erklären?»
 - Missachtung der Prüfungsregeln führen zu einer Note 1.
 - Verspätete Abgabe führt zu Notenabzug:
 - Bis 6h Verspätung → 0.5 Note Abzug
 - Bis 12h Verspätung → 1.0 Note Abzug
 - >12h Verspätung → Note 1.0

8. Abgabe

Die Applikation wird via GitHub abgegeben:

1. Den Quellcode ablegen unter
`CS1\02 Prüfungen\Prüfung 3 - MLZ\Quellcode\<Vorname Nachname> CS1 Prüfung 3 MLZ.sln`
2. Das Kompilat als ZIP ablegen unter
`CS1\02 Prüfungen\Prüfung 3 - MLZ\Kompilat\Wetterstationssimulator.zip`
3. Die Dokumentation ablegen unter
`CS1\02 Prüfungen\Prüfung 3 - MLZ\Dokumentation`
4. Alles hochladen

Verwende die Versionsverwaltung, um regelmässig Sicherungen des aktuellen Standes zu erstellen (mindestens drei Commits) und hochzuladen (Push).