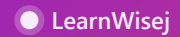


KI-Coding-Übung

Erstellen Sie das Beispielprojekt VisualOperationsStudio dieses Moduls mit ChatGPT oder Claude aus einem fertigen Prompt, der das Modell auf die Spezifikation des Moduls verweist – und prüfen, starten und erweitern Sie dann, was Sie erhalten.



Instruieren, lesen, dann auf die Probe stellen.

Diese Übung hat drei Teile, und nur der erste dreht sich um den Prompt. Sie instruieren einen Assistenten mit der Modulspezifikation und erhalten eine VisualOperationsStudio-Projektmappe zurück; Sie lesen, was er geschrieben hat, und entscheiden, ob es wirklich tut, was das Modul verlangt; dann treiben Sie es weiter, als das Modul verlangt hat. Im zweiten und dritten Teil steckt das eigentliche Lernen.

Worum Sie bitten

Geben Sie VisualOperationsStudio ein Betriebs-Grid mit zwei selbst gezeichneten Spalten. Fügen Sie gridOperations explizite Spalten hinzu, setzen Sie UserPaint = true ausschließlich für die Spalten Health und Trend und behandeln Sie CellPaint an einer einzigen Stelle: Kehren Sie sofort zurück, wenn e.RowIndex negativ ist oder e.ColumnIndex keine der beiden Zielspalten trifft, und zeichnen Sie dann innerhalb von e.ClipRectangle einen schwellenwertgefärbten Fortschrittsbalken für Health und eine Sparkline als Linienzug für Trend. Lesen Sie die Werte jeder Zeile aus dem gebundenen MachineStatus-Objekt, statt im Handler eine Liste zu durchsuchen, respektieren Sie die Farben der markierten Zeile und halten Sie den Handler frei von Seiteneffekten. Binden Sie 1.000 generierte Zeilen an, scrollen Sie und bauen Sie denselben Statusindikator anschließend ein zweites Mal als AllowHtml-Zelle, damit Sie schriftlich begründen können, welche Variante Sie ausliefern würden.

Das Modul verlangt diese Ergebnisse, und der Prompt unten fordert genau diese vom Assistenten an:

- ✓ gridOperations mit expliziten Spalten und UserPaint = true nur für die Spalten Health und Trend
- ✓ Ein einziger CellPaint-Handler, der e.RowIndex und e.ColumnIndex prüft, bevor irgendetwas gezeichnet wird
- ✓ Schwellenwertgefärbte Fortschrittszelle und eine Sparkline-Zelle als Linienzug, gezeichnet innerhalb von e.ClipRectangle
- ✓ Zeilenwerte, die aus dem gebundenen MachineStatus-Objekt gelesen werden, ohne Suche, Allokation oder Control-Erzeugung pro Zelle
- ✓ Eine parallel aufgebaute AllowHtml-Statusspalte samt Notiz, die beide beim Scrollen über 1.000 Zeilen vergleicht

Warum es so aufgebaut ist

Modul 2 hat VisualOperationsStudio ein TelemetryGauge gegeben: ein Control, eine Maschine, ein paar hundert Neuzeichnungen pro Stunde. Das Betriebs-Grid ist ein anderes Kaliber. Eine Schichtleiterin öffnet eine Liste mit tausend Maschinen und will auf einen Blick sehen, welche nahe an ihrer Grenze sind und welche den ganzen Vormittag abgedriftet sind. Zahlen in einer Spalte leisten das nicht, und ein Gauge pro Zeile wären tausend serverseitige Controls in einer einzigen Sitzung. Das Grid selbst muss zeichnen, einmal pro sichtbarer Zelle, und in dem Moment aufhören, in dem die Zelle aus dem Bild scrollt.

Diese Begriffe muss der Assistent richtig verwenden. Geht die Antwort nachlässig mit ihnen um, hat er das Modul nicht verstanden:

Begriff	Was er hier bedeutet
UserPaint	Der ausdrückliche Schalter: DataGridViewCell.UserPaint = true für eine einzelne Zelle, DataGridViewColumn.UserPaint = true für eine ganze Spalte. Ohne ihn fordert das Grid Sie nie zum Zeichnen auf.
CellPaint	Das Event auf Grid-Ebene, das für selbst gezeichnete Zellen ausgelöst wird, wenn sie gezeichnet werden müssen; Sie abonnieren es einmal am Grid, nicht einmal pro Zeile, und verzweigen im Handler.
DataGridViewCellPaintEventArgs	Leitet sich von PaintEventArgs ab, trägt also Graphics und ClipRectangle, und ergänzt RowIndex und ColumnIndex, damit ein Handler mehrere Spalten bedienen kann.
keine WinForms-Helfer	Die dokumentierten Wisej.NET-Ereignisargumente bieten die WinForms-Komfortmethoden PaintBackground() und PaintContent() nicht. Ihr Handler besitzt das gesamte visuelle Ergebnis dieser Zelle.
e.ClipRectangle als Fläche	Das lokale Zielrechteck der Zelle. Spalten lassen sich in der Breite ändern und das Grid kann responsiv angeordnet werden, also muss jede gezeichnete Koordinate daraus abgeleitet sein und nicht aus einer Konstanten.
seiteneffektfreie Render-Helfer	Methoden in der Form DrawProgressCell(Graphics g, Rectangle rect, double value), sodass der Handler zu Verteilercode wird und derselbe Helfer später in eine Off-Screen-Bitmap rendern kann.
AllowHtml -Zellinhalt	Die browsereigene Alternative: Icons, Spans, Labels und CSS, die der Client darstellt – und der Grund, warum die Dokumentation CellPaint für Darstellungen reserviert, die für HTML zu komplex sind.

Selbst bauen – wählen Sie Ihren Assistenten

Beide Buttons öffnen einen neuen Tab, in dem **derselbe fertige Prompt** bereits eingetragen ist. Der Prompt verweist das Modell auf die Spezifikation dieses Moduls – eine einzelne Markdown-Datei mit der Lektion, der Lab-Aufgabe und den geforderten Ergebnissen –, sodass es mit dem echten Material arbeitet, statt zu raten, worum es in Modul 3 geht. Sie können [module-03.md](#) selbst öffnen, um genau zu sehen, was es lesen wird.



ChatGPT verwenden

Öffnet chatgpt.com mit dem Prompt dieses Moduls. Funktioniert mit GPT-5 oder Codex; jeder Tarif mit Websuche ruft die Spezifikation ab.



Claude verwenden

Öffnet claude.ai mit demselben Prompt. Wenn Sie Claude Code nutzen, können Sie ihn stattdessen in einem leeren Ordner in ein Terminal einfügen.



Unverändert abgeschickt, erhalten Sie ein Projekt, kein Code-Fragment

Der Prompt verlangt am Ende die gesamte Projektmappe als einzelne [.zip](#) -Datei mit Anleitung zum Kompilieren und Starten. Laden Sie sie herunter, öffnen Sie die Projektmappe in Visual Studio, drücken Sie F5, und VisualOperationsStudio startet auf [localhost](#) . Sie benötigen das .NET SDK und das NuGet-Paket [Wisej-4](#) ; alles andere soll der Assistent laut Prompt selbst schreiben.

Ich arbeite gerade Modul 3 – „Zellenzeichnen in der DataGridView“ – des LearnWisej-Kurses „Zeichnen und Grafik in Wisej.NET“ durch und möchte das Beispielprojekt des Moduls gemeinsam mit dir erstellen. Bitte antworte auf Deutsch.

Lies zuerst die Modulspezifikation. Sie ist eine einzelne Markdown-Datei (auf Englisch) mit der vollständigen Lektion, der Lab-Aufgabe und den geforderten Ergebnissen:

<https://learnwisej.com/assets/courses/drawing-and-painting/specs/module-03.md>

Erstelle dann ein vollständiges Wisej.NET-4-Projekt, das ich in Visual Studio öffnen kann:

1. Projekt „VisualOperationsStudio“, C#, net8.0 oder neuer, NuGet-Paket Wisej-4 4.1.x, Kestrel-Host, der `app.UseWisej()` aufruft.
2. Setze um, was das Lab der Spezifikation verlangt: Geben Sie `VisualOperationsStudio` ein Betriebs-Grid mit zwei selbst gezeichneten Spalten. Fügen Sie `gridOperations` explizite Spalten hinzu, setzen Sie `UserPaint = true` ausschließlich für die Spalten `Health` und `Trend` und behandeln Sie `CellPaint` an einer einzigen Stelle: Kehren Sie sofort zurück, wenn `e.RowIndex` negativ ist oder `e.ColumnIndex` keine der beiden Zielspalten trifft, und zeichnen Sie dann innerhalb von `e.ClipRectangle` einen schwellenwertgefärbten Fortschrittsbalken für `Health` und eine Sparkline als Linienzug für `Trend`. Lesen Sie die Werte jeder Zeile aus dem gebundenen `MachineStatus`-Objekt, statt im Handler eine Liste zu durchsuchen, respektieren Sie die Farben der markierten Zeile und halten Sie den Handler frei von Seiteneffekten. Binden Sie 1.000 generierte Zeilen an, scrollen Sie und bauen Sie denselben Statusindikator anschließend ein zweites Mal als `AllowHtml`-Zelle, damit Sie schriftlich begründen können, welche Variante Sie ausliefern würden.
3. Liefere diese Ergebnisse und sag mir, wo sich jedes davon in der Projektmappe befindet:
 - `gridOperations` mit expliziten Spalten und `UserPaint = true` nur für die Spalten `Health` und `Trend`
 - Ein einziger `CellPaint`-Handler, der `e.RowIndex` und `e.ColumnIndex` prüft, bevor irgendetwas gezeichnet wird
 - Schwellenwertgefärbte Fortschrittszelle und eine Sparkline-Zelle als Linienzug, gezeichnet innerhalb von `e.ClipRectangle`
 - Zeilenwerte, die aus dem gebundenen `MachineStatus`-Objekt gelesen werden, ohne Suche, Allokation oder Control-Erzeugung pro Zelle
 - Eine parallel aufgebaute `AllowHtml`-Statusspalte samt Notiz, die beide beim Scrollen über 1.000 Zeilen vergleicht
4. Halte dich an die Spezifikation, wo immer sie eine Wisej.NET-Klasse, -Eigenschaft oder ein -Event nennt – verwende genau diese. Wo sie eine Wahl offenlässt, sag mir, welche API du gewählt hast und warum.
5. Der Server bleibt maßgeblich: Zustand und Validierung leben in .NET, der Browser stellt dar und meldet zurück. Verlagere nie eine Entscheidung nach JavaScript, nur weil es dort einfacher ist.
6. Mach das Ergebnis vorführbar: Ich muss es starten und das Verhalten sehen können, um das es in diesem Modul geht – auch, was passiert, wenn etwas schiefgeht.

Bevor du Code schreibst, fasse in eigenen Worten zusammen, was du bauen wirst und wo jedes Teil liegen wird, damit ich den Entwurf korrigieren kann, solange das noch wenig kostet.

Wenn der Code fertig ist, packe die gesamte Projektmappe in eine einzelne .zip-Datei zum Herunterladen und nenne mir die genauen Schritte, um sie zu kompilieren und auszuführen.

Beantworte zum Schluss diese Fragen aus der Spezifikation und nutze dabei deine eigene Implementierung als Beleg:

- Ein Kollege fügt ein WinForms-Beispiel zum Zellenzeichnen ein, das mit einem Aufruf von `e.PaintBackground(...)` beginnt. Erklären Sie, was die Ereignisargumente von Wisej.NET Ihnen stattdessen geben und was der Handler nun selbst entscheiden muss.
- Ihr Sparkline-Helfer verdrahtet eine Plotbreite von 100 Pixeln fest und sieht auf Ihrem Bildschirm gut aus. Beschreiben Sie zwei Benutzeraktionen, die ihn zerstören, und was der Helfer stattdessen auslesen sollte.
- Die HTML-Statusspalte und die gezeichnete Health-Spalte zeigen beide eine kritische Maschine. Begründen Sie, welche Sie für ein Grid ausliefern würden, das Support-Mitarbeitende durchsuchen und mit einem Screenreader lesen, und was Sie jeweils verlieren.

Frag mich nach allem, was unklar ist, bevor du anfängst.

Prüfen Sie dann, was zurückkommt

Code, der kompiliert, ist noch lange kein richtiger Code. Die spannenden Fragen sind Designfragen: Ist das wirklich das Wisej.NET-Control, das die Aufgabe verlangt? Folgt der Code dem Lebenszyklus, den das Framework vorgibt? Liegt der maßgebliche Zustand auf dem Server – oder wurde er im Browser abgelegt, weil das bequemer war? Lesen Sie das Ergebnis als Reviewer, nicht als Kunde.

Die Checkliste liefert Ihnen dieses Modul. Beantworten Sie jeden Punkt anhand des Codes, den Sie gerade erhalten haben:

- ✓ Ein Kollege fügt ein WinForms-Beispiel zum Zellenzeichnen ein, das mit einem Aufruf von `e.PaintBackground(...)` beginnt. Erklären Sie, was die Ereignisargumente von Wisej.NET Ihnen stattdessen geben und was der Handler nun selbst entscheiden muss.
- ✓ Ihr Sparkline-Helfer verdrahtet eine Plotbreite von 100 Pixeln fest und sieht auf Ihrem Bildschirm gut aus. Beschreiben Sie zwei Benutzeraktionen, die ihn zerstören, und was der Helfer stattdessen auslesen sollte.
- ✓ Die HTML-Statusspalte und die gezeichnete Health-Spalte zeigen beide eine kritische Maschine. Begründen Sie, welche Sie für ein Grid ausliefern würden, das Support-Mitarbeitende durchsuchen und mit einem Screenreader lesen, und was Sie jeweils verlieren.

Alles, was Sie korrigieren mussten, sollten Sie festhalten. Eine kurze Notiz zu den Konventionen und Entscheidungen, auf die Sie sich festgelegt haben – eingefügt in die erste Nachricht Ihrer nächsten Sitzung –, ist mehr wert als jeder noch so clevere Prompt.



Für diesen Kurs gibt es noch kein Referenzbeispiel – Ihres ist das erste

Das [LearnWisej-Beispiel-Repository](#) enthält ausgearbeitete Builds für die meisten Kurse, für diesen aber noch nicht. Es gibt also nichts zum Vergleichen und keine Abkürzung: Die Spezifikation und Ihr eigenes Review sind das ganze Sicherheitsnetz. Behalten Sie, was Sie bauen – und wenn es gut wird, würden wir es gern sehen.

Drei Prompts für den nächsten Schritt

Richten Sie Ihren Assistenten auf das Projekt, das Sie gerade gebaut haben, und probieren Sie diese Prompts aus. Jeder ist eine Übung zum Prüfen und Erweitern, kein Neuschreiben.

- ✓ **Prüfen Sie es wie einen Pull-Request.** *"Führ mich durch diesen VisualOperationsStudio-Code, als würdest du den Pull-Request eines Kollegen prüfen und hättest ihn nicht selbst geschrieben. Wo lebt der Zustand, was passiert bei einer einzelnen Benutzeraktion von Anfang bis Ende, was würdest du zur Überarbeitung zurückgeben, und welche Teile würdest du nicht guten Gewissens einem Kunden vorlegen? Sei konkret und zitiere die Zeilen, die du meinst – fasse mir nicht den Entwurf zusammen."*
- ✓ **Tauschen Sie ein Teil aus, ohne die Schnittstellen zu verschieben.** *"Wähle den Teil dieses VisualOperationsStudio-Projekts, der am offensichtlichsten nur ein Platzhalter für etwas Echtes ist – eine Fake-Datenquelle, ein In-Memory-Speicher, ein vereinfachter Service – und ersetze ihn durch eine praxisnahe Implementierung. Der Vertrag auf beiden Seiten darf sich nicht ändern: dieselben Typen, dieselben Aufrufe, dasselbe Verhalten im Fehlerfall. Sag mir, was du durch die echte Implementierung behandeln musstest, was der Platzhalter nicht behandelt hat."*
- ✓ **Machen Sie es fit für einen schlechten Tag.** *"Untersuche dieses VisualOperationsStudio-Projekt auf Fehlerszenarien, die es noch nicht behandelt – fehlerhafte oder fehlende Eingaben, eine Abhängigkeit, die langsam oder nicht erreichbar ist, Arbeit, die eintrifft, nachdem der Benutzer die Seite verlassen hat, und zwei Aktionen, die sich gegenseitig in die Quere kommen. Sag mir für jedes zuerst, ob es in Wisej.NET 4 ein echtes Risiko ist und warum, und schreib erst dann die Korrektur. Wo du entscheidest, dass nichts behandelt werden muss, sag das und begründe es, statt defensiven Code hinzuzufügen."*



Nehmen Sie es mit ins Lab

Das praktische Lab am Ende dieses Moduls verlangt dieselben Ergebnisse, die Sie gerade vom Assistenten haben erstellen lassen. Nutzen Sie, was Sie hier haben, als Gerüst für Ihre Antwort und als Beleg dafür – nicht als die Antwort selbst. Bewertet wird im Lab Ihre Argumentation, und dieser Teil muss von Ihnen stammen.