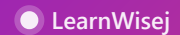


KI-Coding-Übung

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.



Der Assistent liefert die schnellen Hände. Der Architekt bleiben Sie.

Ein Assistent erstellt das Beispielprojekt dieses Moduls in wenigen Minuten. Was er nicht kann, ist zu entscheiden, ob das Ergebnis der richtige Entwurf für Ihre Anwendung ist – diese Entscheidung bleibt bei Ihnen. Das Urteilsvermögen bringen Sie mit: Instruieren Sie ihn sorgfältig und prüfen Sie dann, was zurückkommt, anhand dessen, was Ihnen Modul 7 gerade vermittelt hat.

Worum Sie bitten

Schließen Sie das Abschlussprojekt VisualOperationsStudio ab, indem Sie ein einziges Modell hinter alle vier Oberflächen stellen. Gliedern Sie das Projekt in die Ordner Models, Geometry, Renderers und einen Wisej-Adapter um, sodass das gezeichnete Gauge, die selbst gezeichneten Grid-Zellen, die Canvas-Topologie und der PNG-Export alle dasselbe OperationsModel lesen und die seiteneffektfreie Geometrieschicht teilen, ohne fachlichen Zustand pro Renderer zu duplizieren. Ergänzen Sie einen RenderMetrics-Helfer, der Renderdauer, Größe der Ausgabebitmap und die Anzahl der gezeichneten Szenenobjekte erfasst, und vergleichen Sie damit das server-seitig gezeichnete Gauge mit dem Off-Screen-Renderer – vor und nach einer Optimierung, die Sie wählen und begründen. Spielen Sie die Fehlerpfade durch – leere Daten, eine Oberfläche der Größe null, ein ungültiger Wertebereich, eine fehlende Schrift –, sodass jeder auf einen lesbaren Platzhalter statt auf eine Exception hinausläuft, ergänzen Sie eine barrierefreie Tabelle mit allen Werten, die die Grafik zeigt, und schreiben Sie die Abschlussnotizen zu Architektur, Browser-/Server-Grenze und Testmatrix.

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

- ✓ Models-, Geometry-, Renderers- und Adapterschichten mit einem OperationsModel, das alle vier Oberflächen speist
- ✓ RenderMetrics-Helfer, der Renderdauer, Bitmapgröße und Anzahl gezeichneter Objekte pro Oberfläche erfasst
- ✓ Vorher-nachher-Messungen für eine Optimierung samt Begründung der Wahl
- ✓ Fehlerpfade für leere Daten, Größe null, ungültigen Bereich und fehlende Schrift, die abstufen statt eine Exception auszulösen
- ✓ CapstoneNotes.md mit Architektur, Server-/Browser-Grenze, Testmatrix und einer barrierefreien Datentabelle

Warum es so aufgebaut ist

VisualOperationsStudio besitzt inzwischen vier Rendering-Oberflächen, und jede wurde in ihrem eigenen Modul mit ihrer eigenen privaten Vorstellung davon gebaut, was die Gesundheit einer Maschine bedeutet. So verrottet ein Zeichen-Subsystem: Jemand verschiebt eine Warnschwelle, drei Oberflächen sind sich einig, und das exportierte PNG widerspricht still dem Bildschirm, von dem es stammt. Dieses Modul führt diese vier Auslegungen zu einem Modell hinter einer Geometrieschicht zusammen und belegt dann mit Messungen, absichtlich durchgespielten Fehlerpfaden und einem semantischen Äquivalent, dass das Ergebnis echte Daten, echte Plattformen und echte Benutzer übersteht.

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
Visuelles Modell	Ausschließlich semantischer Zustand: Messwerte, Beschriftungen, Knoten und Kanten, Trendreihen, der Schweregrad als fachliche Entscheidung und die gemeinsame Auswahl – damit keine Oberfläche eine Kopie braucht.
Geometrieschicht	Weitgehend reine Mathematik, die Modellzustand plus die aktuelle Oberflächengröße in Rechtecke, Winkelbereiche, Beschriftungspositionen und Trefferbereiche verwandelt – prüfbar ohne jedes Control.
Oberflächenadapter	Ein Renderer pro Oberfläche statt einer gemeinsamen Schnittstelle: System.Drawing.Graphics und Wisej.Web.Canvas behalten über einer gemeinsamen Geometrie ihre eigenen Primitiven.
Leistungsbudget	Eine Zahl, die vor dem Optimieren vereinbart wird, etwa ein gezeichnetes Status-Control unter 5 ms und ein vollständiges Canvas-Neuzeichnen innerhalb von rund 16 bis 50 ms.
Rendering-Instrumentierung	Erfasste verstrichene Zeit, Bitmapabmessungen, Anzahl gezeichneter Objekte und Zuwachs der Datenmenge pro Oberfläche, damit eine Optimierung aus einem Vorher und einem Nachher begründet wird.
Golden-Image-Test	Ein Off-Screen-Rendering mit festen Eingaben, verglichen mit einer gespeicherten Referenz und einer Toleranz, denn Kantenglättung und Kodierer unterscheiden sich zwischen Plattformen.
Semantisches Äquivalent	Das Label, die AccessibleDescription, der Tooltip oder die begleitende Liste, die jeden wesentlichen Wert trägt, den die Grafik zeigt – damit nichts Wichtiges nur als Pixel existiert.

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 7 geht. Sie können [module-07.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 7 – „Zeichenarchitektur für den Produktivbetrieb, Performance und Abschlussprojekt“ – 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-07.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: Schließen Sie das Abschlussprojekt VisualOperationsStudio ab, indem Sie ein einziges Modell hinter alle vier Oberflächen stellen. Gliedern Sie das Projekt in die Ordner Models, Geometry, Renderers und einen Wisej-Adapter um, sodass das gezeichnete Gauge, die selbst gezeichneten Grid-Zellen, die Canvas-Topologie und der PNG-Export alle dasselbe OperationsModel lesen und die seiteneffektfreie Geometrieschicht teilen, ohne fachlichen Zustand pro Renderer zu duplizieren. Ergänzen Sie einen RenderMetrics-Helfer, der Renderdauer, Größe der Ausgabebitmap und die Anzahl der gezeichneten Szenenobjekte erfasst, und vergleichen Sie damit das server-seitig gezeichnete Gauge mit dem Off-Screen-Renderer – vor und nach einer Optimierung, die Sie wählen und begründen. Spielen Sie die Fehlerpfade durch – leere Daten, eine Oberfläche der Größe null, ein ungültiger Wertebereich, eine fehlende Schrift –, sodass jeder auf einen lesbaren Platzhalter statt auf eine Exception hinausläuft, ergänzen Sie eine barrierefreie Tabelle mit allen Werten, die die Grafik zeigt, und schreiben Sie die Abschlussnotizen zu Architektur, Browser-/Server-Grenze und Testmatrix.
3. Liefere diese Ergebnisse und sag mir, wo sich jedes davon in der Projektmappe befindet:
 - Models-, Geometry-, Renderers- und Adapterschichten mit einem OperationsModel, das alle vier Oberflächen speist
 - RenderMetrics-Helfer, der Renderdauer, Bitmapgröße und Anzahl gezeichneter Objekte pro Oberfläche erfasst
 - Vorher-nachher-Messungen für eine Optimierung samt Begründung der Wahl
 - Fehlerpfade für leere Daten, Größe null, ungültigen Bereich und fehlende Schrift, die abstufen statt eine Exception auszulösen
 - CapstoneNotes.md mit Architektur, Server-/Browser-Grenze, Testmatrix und einer barrierefreien Datentabelle
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:

- Ihre Geometrieschicht nimmt das Modell und eine Größe entgegen und gibt Formen zurück, doch ein Teammitglied möchte, dass sie einen GraphicsPath liefert, damit der Gauge-Renderer kürzer wird. Was kostet das den Canvas-Renderer und die Unit-Tests, und wo würden Sie stattdessen die Grenze ziehen?
- Das Export-PNG und die Canvas auf dem Bildschirm weichen auf dem Linux-Buildagenten um ein paar Pixel voneinander ab, stimmen unter Windows aber überein. Welche Messungen und welche Art von Test würden Ihnen sagen, ob das ein Defekt oder ein erwarteter Plattformunterschied ist?
- Ein Reviewer sagt, die barrierefreie Tabelle sei überflüssig, weil jedes Gauge bereits einen Tooltip hat. Erklären Sie, was die Tabelle einer Person gibt, die nur die Tastatur oder einen Screenreader nutzt, und was der Tooltip nicht gibt – und was daraus für die rein farblich codierten Schweregrad-Chips im Grid folgt.

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:

- ✓ Ihre Geometrieschicht nimmt das Modell und eine Größe entgegen und gibt Formen zurück, doch ein Teammitglied möchte, dass sie einen GraphicsPath liefert, damit der Gauge-Renderer kürzer wird. Was kostet das den Canvas-Renderer und die Unit-Tests, und wo würden Sie stattdessen die Grenze ziehen?
- ✓ Das Export-PNG und die Canvas auf dem Bildschirm weichen auf dem Linux-Buildagenten um ein paar Pixel voneinander ab, stimmen unter Windows aber überein. Welche Messungen und welche Art von Test würden Ihnen sagen, ob das ein Defekt oder ein erwarteter Plattformunterschied ist?
- ✓ Ein Reviewer sagt, die barrierefreie Tabelle sei überflüssig, weil jedes Gauge bereits einen Tooltip hat. Erklären Sie, was die Tabelle einer Person gibt, die nur die Tastatur oder einen Screenreader nutzt, und was der Tooltip nicht gibt – und was daraus für die rein farblich codierten Schweregrad-Chips im Grid folgt.

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.

- ✓ **Beweisen Sie, dass es wirklich funktioniert.** *"Geh dieses Projekt Ergebnis für Ergebnis anhand der Modulspezifikation unter <https://learnwisej.com/assets/courses/drawing-and-painting/specs/module-07.md> durch und zeig mir den Beleg, dass jedes wirklich erledigt ist – den Codepfad, worauf ich klicken muss und was ich sehen sollte. Wo ein Ergebnis nur teilweise erfüllt ist, sag das klar, statt es schönzureden. Ergänze dann, was fehlt."*
- ✓ **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."*



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.