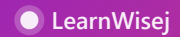


KI-Coding-Übung

Erstellen Sie das Beispielprojekt GlobalDesk 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 2 gerade vermittelt hat.

Worum Sie bitten

Lokalisieren Sie den GlobalDesk-Kundeneditor über den Visual Studio Designer. Stellen Sie zuerst das neutrale englische Layout des UserControl CustomerEditor fertig, setzen Sie dann Localizable auf true und fügen Sie über den Language-Selektor des Designers eine deutsche Variante hinzu, wobei Sie nur ändern, was sich unterscheiden muss: die Beschriftungen, die Breite der beiden längsten Schaltflächen und die Location eines Labels. Führen Sie die Anwendung unter de-DE aus und vergleichen Sie die tatsächliche Browserausgabe mit der Designeransicht, und fügen Sie dann eine Schaltfläche Recreate editor hinzu, die das Control verwirft und neu aufbaut, damit Sie selbst sehen, warum Designer-Ressourcen einem Kulturwechsel zur Laufzeit nicht folgen. Schließen Sie mit einer Lab-Notiz ab, die die überschriebenen Eigenschaften auflistet und diejenigen, die Sie bewusst dem neutralen Layout überlassen haben.

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

- ✓ UserControl CustomerEditor mit Localizable auf true über einem vollständigen neutralen englischen Layout
- ✓ Deutsche Designer-Sprachvariante, die die Beschriftungen sowie mindestens eine Size und eine Location überschreibt
- ✓ Anchoring, Docking oder AutoSize so eingesetzt, dass die deutschen Beschriftungen nicht abgeschnitten werden können
- ✓ Schaltfläche Recreate editor, die den Container nach einem Kulturwechsel verwirft und neu aufbaut
- ✓ Lab-Notiz mit den überschriebenen Eigenschaften und denen, die dem neutralen Layout überlassen wurden

Warum es so aufgebaut ist

Modul 1 hat GlobalDesk ein Dashboard gegeben, dessen Beschriftungen alle aus Strings.resx stammen, und das ist die richtige Heimat für Text, den der Code erzeugt. Beim Kundeneditor liegt der Fall anders: Er ist ein entworfener Bildschirm, nach Augenmaß im Visual Studio Designer angeordnet. Jede seiner Beschriftungen über einen Helper-Aufruf zu leiten hieße, die Designerdatei von Hand zu bearbeiten, und es löst das eigentliche Problem trotzdem nicht: dass ein deutsches Kundennummer speichern nicht in eine Schaltfläche passt, die für Save bemessen ist. Dieses Modul nutzt den Lokalisierungsworkflow, den Visual Studio ohnehin anbietet, sodass sich Beschriftungen und Geometrie eines entworfenen Bildschirms je Sprache unterscheiden können, ohne fest verdrahtete Zeichenfolgen und ohne eine zweite Layoutdatei.

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
Localizable	Eine Entwurfszeit-Eigenschaft auf einem Container der obersten Ebene; true versetzt den Designer in den Lokalisierungsworkflow, sodass Werte je Kultur abgelegt werden können.
Language -Selektor des Designers	Erscheint, sobald Localizable auf true steht; die ausgewählte Kultur ist die Fassung, die Sie bearbeiten, deshalb wird jede Änderung nur für diese Kultur aufgezeichnet.
Die neutrale Sprache	Die Standardfassung, auf die jede Kultur zurückfällt, und die einzige, in der Sie Controls hinzufügen, entfernen oder umstrukturieren; Kulturansichten enthalten Überschreibungen, keinen zweiten Bildschirm.
Lokalisierbare Eigenschaften	Nicht nur Text: Size, Location, Farben und Bilder können je Kultur einen anderen Wert tragen, denn eine Übersetzung verändert neben den Wörtern auch die Geometrie.
Generierte Ressourcendateien	Der Designer schreibt lokalisierte Werte in Ressourcendateien statt in den C#-Designercode und hält sprachspezifische Werte aus dem Code heraus, den Sie täglich lesen und zusammenführen.
Beim Erstellen angewendet	Im Designer lokalisierte Werte werden beim Aufbau der Controls angewendet, deshalb schreibt eine spätere Änderung der Sitzungskultur einen bereits bestehenden Container nicht neu.
Übersetzungstolerantes Layout	Docking, Anchoring, FlowLayoutPanel, TableLayoutPanel, FlexLayoutPanel, AutoSize und Padding lassen Text von selbst wachsen, sodass Sie weniger Geometrie-Überschreibungen brauchen.

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 2 geht. Sie können [module-02.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 GlobalDesk 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 2 – „Die Wisej.NET-Oberfläche im Visual Studio Designer lokalisieren“ – des LearnWisej-Kurses „Lokalisierung 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/localization-in-wisej-net/specs/module-02.md>

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

1. Projekt „GlobalDesk“, 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: Lokalisieren Sie den GlobalDesk-Kundeneditor über den Visual Studio Designer. Stellen Sie zuerst das neutrale englische Layout des UserControls CustomerEditor fertig, setzen Sie dann Localizable auf true und fügen Sie über den Language-Selektor des Designers eine deutsche Variante hinzu, wobei Sie nur ändern, was sich unterscheiden muss: die Beschriftungen, die Breite der beiden längsten Schaltflächen und die Location eines Labels. Führen Sie die Anwendung unter de-DE aus und vergleichen Sie die tatsächliche Browserausgabe mit der Designeransicht, und fügen Sie dann eine Schaltfläche Recreate editor hinzu, die das Control verwirft und neu aufbaut, damit Sie selbst sehen, warum Designer-Ressourcen einem Kulturwechsel zur Laufzeit nicht folgen. Schließen Sie mit einer Lab-Notiz ab, die die überschriebenen Eigenschaften auflistet und diejenigen, die Sie bewusst dem neutralen Layout überlassen haben.
3. Liefere diese Ergebnisse und sag mir, wo sich jedes davon in der Projektmappe befindet:
 - UserControl CustomerEditor mit Localizable auf true über einem vollständigen neutralen englischen Layout
 - Deutsche Designer-Sprachvariante, die die Beschriftungen sowie mindestens eine Size und eine Location überschreibt
 - Anchoring, Docking oder AutoSize so eingesetzt, dass die deutschen Beschriftungen nicht abgeschnitten werden können
 - Schaltfläche Recreate editor, die den Container nach einem Kulturwechsel verwirft und neu aufbaut
 - Lab-Notiz mit den überschriebenen Eigenschaften und denen, die dem neutralen Layout überlassen wurden
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 Teammitglied fügt CustomerEditor ein neues Telefonfeld hinzu, während in der Language

-Liste Deutsch ausgewählt ist. Was sieht der englische Benutzer wahrscheinlich, und welche Regel hätte das verhindert?

- Die Speicherbestätigung, die der Editor nach einer erfolgreichen Aktualisierung zeigt, stammt weiterhin aus Strings.resx und nicht aus den Designer-Ressourcen. Nennen Sie zwei Gründe, warum das der richtige Ort dafür ist, und benennen Sie eine Beschriftung auf diesem Bildschirm, die tatsächlich in den Designer gehört.
- Ihre Schaltfläche Recreate editor funktioniert, aber ein Kollege argumentiert, der Bildschirm solle die lokalisierten Eigenschaften einfach erneut anwenden, statt neu aufzubauen. Was müsste dieser Code wissen, was der Neuaufbau nicht wissen muss, und wann wäre sein Ansatz der bessere Kompromiss?

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 Teammitglied fügt CustomerEditor ein neues Telefonfeld hinzu, während in der Language - Liste Deutsch ausgewählt ist. Was sieht der englische Benutzer wahrscheinlich, und welche Regel hätte das verhindert?
- ✓ Die Speicherbestätigung, die der Editor nach einer erfolgreichen Aktualisierung zeigt, stammt weiterhin aus Strings.resx und nicht aus den Designer-Ressourcen. Nennen Sie zwei Gründe, warum das der richtige Ort dafür ist, und benennen Sie eine Beschriftung auf diesem Bildschirm, die tatsächlich in den Designer gehört.
- ✓ Ihre Schaltfläche Recreate editor funktioniert, aber ein Kollege argumentiert, der Bildschirm solle die lokalisierten Eigenschaften einfach erneut anwenden, statt neu aufzubauen. Was müsste dieser Code wissen, was der Neuaufbau nicht wissen muss, und wann wäre sein Ansatz der bessere Kompromiss?

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/localization-in-wisej-net/specs/module-02.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 GlobalDesk-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 GlobalDesk-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.