

GeomFigur Dokumentaion

Kevin William Schnurpfeil

Dezember 2022

1 Ausarbeitung

1.1 Projektbeteiligte

Dokumentation, Ausarbeitung und Entwicklung durch:

Herr Kevin William Schnurpfeil
Auszubildender Fachinformatiker für Anwendungsentwicklung
EFI3A
Sycor GmbH
Heinrich-von-Stephan-Straße 1-5
37073 Göttingen

Aufgabenstellung:

Herr Dirk Rust
BBS II Göttingen
Godehardstraße 11
37081 Göttingen

1.2 Rahmen

Das Projekt wurde im Rahmen der Berufsausbildung im 3. Lehrjahr des Ausbildungsberufes “Fachinformatiker für Anwendungsentwicklung” an den Berufsbildenden Schulen 2 in Göttingen realisiert.

1.3 Umfang

Diese Dokumentation hat den Schwerpunkt der Benutzerschnittstellenentwicklung. Sie beinhaltet keine Benutzerdokumentation, sowie keine technische Dokumentation der Anwendung.

1.4 Projektmittel

- **Entwicklungsumgebung:** Visual Studio 2022
- **Programmiersprache:** C#
- **Applikation:** Windows Forms Applikation
- **Framework:** .NET Framework 4.7.2
- **Betriebssystem:** Microsoft Windows 10
- **Verwendete Versionsverwaltungssoftware:** GitHub

1.5 Repository

Der Quelltext ist zusätzlich noch unter dem folgenden Link zu erreichen:

Repository: <https://github.com/StringArrow/GeomFigur>

Programmieraufgabe Schuljahr 2022/2023

Erstellen Sie für ein Programm zur Körper- und Flächenberechnung von geometrischen Figuren **die Benutzerschnittstellen**.

Dieses Programm soll objektorientiert programmiert sein und das Vererbungskonzept verwenden.

Der Benutzer soll nach Eingabe der Daten (Farbe, Name, Figur, Radius...) die Berechnungsergebnisse (z.B.: A, U) angezeigt bekommen.

Das Programm ist beliebig wiederholbar und durch einen Beendigungsbefehl zu verlassen.

Auf Fehleingaben ist entsprechend zu reagieren. Achten Sie auf eine gute Softwareergonomie.

Mögliches Datenmodell:

Als Basisklasse verwenden Sie eine Klasse **CGeomFigur** mit den

Attributen: Farbe und Bezeichnung sowie den Methoden zum Lesen und Schreiben der Attribute.

Von dieser Klasse erben zwei Klassen (**CFlaeche** und **CKoerper**) mit eigenen Attributen und Methoden.

CFlaeche: Attribute: Flächeninhalt, Umfang Methoden: lesen und schreiben der Attribute

CKoerper: Attribute: Volumen, Oberfläche Methoden: lesen und schreiben der Attribute

Von der Klasse CFlaeche erben drei unterschiedliche Flächen (z.B.: Raute, ?, ?) mit ihren speziellen Attributen und Methoden.

Von der Klasse CKoerper erben drei unterschiedliche Körper (z.B.: Würfel, ?, ?) mit ihren speziellen Attributen und Methoden.

Die Benutzerschnittstelle soll in einem Designprozess entwickelt werden und verschiedene Formen von Entwürfen enthalten.

Abzugeben sind:

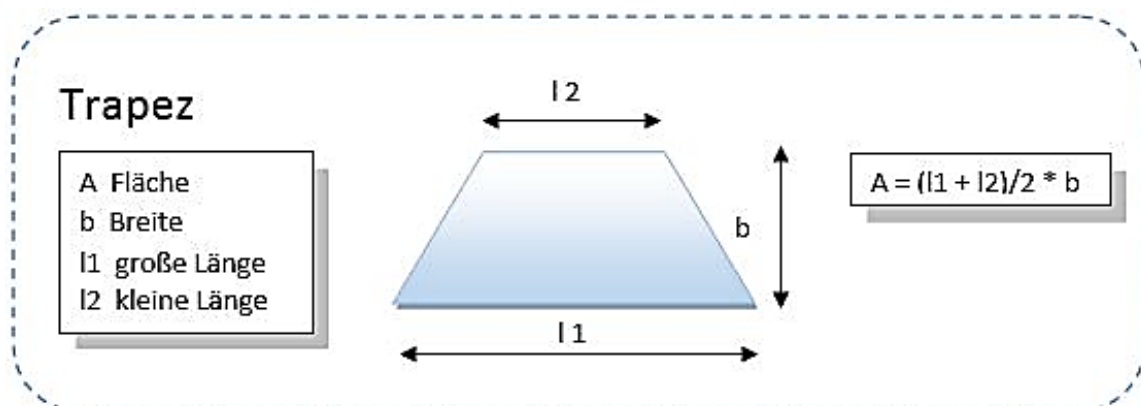
- verschiedene **Formen des Entwurfes** der Benutzerschnittstellen,
- die Benutzerschnittstellen als **Visual Studio Projekt**,
- das **UML Klassendiagramm** ihres Datenmodells und
- eine **Schnittstellendokumentation**.

Abgabetermin: 05.01.2023 bis 20:00 Uhr im IServ Aufgabentool

Bei Fragen zur Aufgabenstellung melden Sie sich bitte per Mail an mich!

Ich wünsche Ihnen viele kreative Ideen und die nötige Gelassenheit bei der Umsetzung dieser Aufgabe!

Dirk Rust, LF10A



2 Anforderungen an die Anwendung

2.1 Abweichung von der Aufgabenstellung

Die Anwendung wurde im Vergleich zur Aufgabenstellung zusätzlich noch um einige Funktionalitäten erweitert.

2.2 Funktionale Anforderungen

Die Anwendung soll:

- geometrische Figuren aus einer vordefinierten Auswahl erstellen können.
- erstellte Figuren entfernen können.
- mehrere geometrische Figuren anlegen können.
- eine Auswahl an geometrischen Figuren besitzen.
- zwischen verschiedenen Figuren umschalten können.
- die ausgewählte Figur in einem Vorschau-Fenster anzeigen können.
- je nach Figur-Typ unterschiedliche Parameter ausgeben können.

2.3 Nichtfunktionale Anforderungen

Die Anwendung sollte:

- möglichst performant sein.
- einfach um weitere Figur-Typen erweiterbar sein.
- Das Vererbungskonzept der objektorientierten Programmierung anwenden.
- Redundanz im Code vermeiden.
- eine gute grafische Benutzeroberfläche aufweisen, um den Anwender die Bedienung zu erleichtern.

3 Grafische Benutzeroberfläche

3.1 Planung

Um dem Anwender die Bedienung der Anwendung so einfach wie möglich zu gestalten, soll die GUI¹ den Benutzer durch die Anwendung führen. Des weiteren soll das GUI den Anforderungen unter Punkt 2 gerecht werden.

3.1.1 Anforderungen an die grafische Benutzeroberfläche

Idealerweise hat eine optimale Benutzeroberfläche die folgenden Charakteristiken:

- **Intuitiv:** Die Oberfläche soll einfach zu verstehen sein.
- **Effizient:** Die Oberfläche soll die zu erfüllenden Aufgaben einfach und effizient ohne mögliche Zusatzschritte oder Ablenkungen verwirklichen und soll die Informationen so minimal wie nötig und so organisiert wie möglich darstellen.
- **Flexibel:** Die Oberfläche kann für den Anwender personalisiert angepasst werden. In diesem Fall reicht die Skalierung des Anwendungsfensters aus.
- **Konsistent:** Die Oberfläche soll in den Punkten: “Layout, Sprache und Verhalten” möglichst konsistent sein, um dem Benutzer die Navigation zu vereinfachen und Verwirrungen zu verhindern.
- **Zugänglichkeit:** Die Oberfläche soll für alle Anwender zugänglich sein, selbst wenn diese noch keine Vorkenntnisse über die Anwendung besitzen.

Dafür wurde beschlossen, die Oberfläche in nummerierten Schritten zu unterteilen:

1. Erstellen und auswählen der Figuren.
2. Anzeigen der ausgewählten Figur.
3. Tabellarisches Anzeigen der Parameter der erstellten Figur.

3.2 Design-Prozess

Der Design-Prozess der grafischen Benutzeroberfläche wurde in verschiedene Schritte unterteilt. Er umfasst die Erstellung von:

1. Skizzen
2. Wireframes
3. Mockups
4. Prototypen

¹Graphical User Interface (deutsch: Graphische Benutzeroberfläche)

3.2.1 Skizze

Für ein Konzept einer grafischen Benutzeroberfläche wurde zunächst Skizzen zu Papier gebracht, welche die Grundlage für den Design-Prozess bietet.

Diese Skizze ist eine grobe, handgezeichnete Repräsentation der grafischen Benutzeroberfläche, um die Idee des Aufbaus vorerst in der Work-In-Progress Phase zu veranschaulichen. Sie veranschaulicht lediglich das Design und hilft dabei, Probleme bei der Schnittstellenentwicklung frühzeitig zu erkennen.

Die Skizzen bieten eine gute Hilfe für den Beginn der Konzeptionsphase.

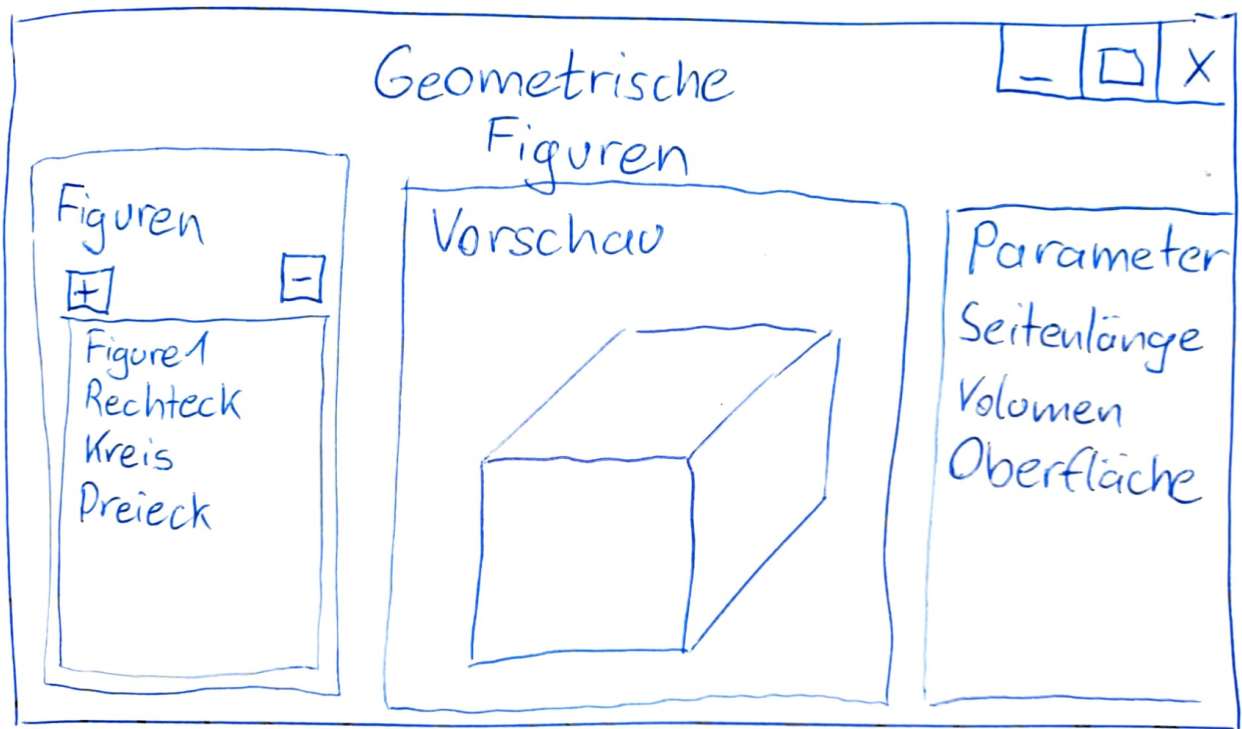


Abbildung 1: Skizze: Hauptbildschirms

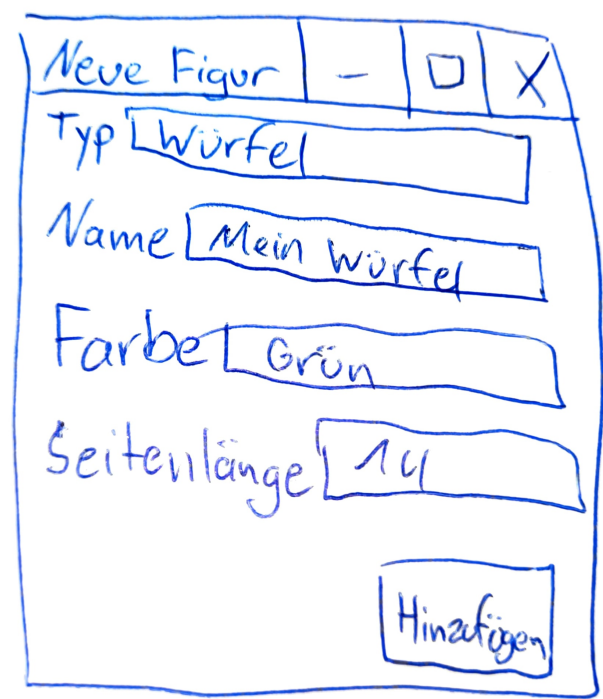


Abbildung 2: Skizze: Neue Figur Hinzufügen

3.2.2 Wireframes

Zusammenfassung

Nach der Erstellung der Skizzen, wurden die Schnittstellen miteinander in Verbindung gebracht.

Dies dient der Auswertung der Struktur und sorgt für Verständnis, wie die Schnittstellen aufgerufen werden.

Es wurde aber auf die Erstellung eines Schaubildes auf Grund der Einfachheit verzichtet und stattdessen die Aufrufreihenfolge als Text beschrieben.

Der Bildschirm unter fig. 2 soll aufgerufen werden, wenn der Benutzer die Schaltfläche “+” in der Skizze fig. 1 betätigt.

Die Navigation soll zurück zum Bildschirm unter: fig. 1 führen, wenn der Benutzer eine neue Figur erstellt hat, bzw. die Erstellung durch Betätigen der Schaltfläche “X” Auf dem Bildschirm fig. 2 abgebrochen werden.

3.2.3 Mockups

Als der grobe Entwurf aus der Skizze feststand, wurde unter Zuhilfenahme der Software “Krita” eine Mock-Up-Skizze von der grafischen Benutzeroberfläche erstellt.

Sie beinhaltet alle nötigen Ausgabeflächen, sowie Schaltflächen, welche auf dem Prototypen der Oberfläche vorhanden sein muss.

Die Mockup-Zeichnung soll dabei so nahe wie möglich der tatsächlichen Darstellung kommen, was sie von der Skizze unterscheidet, da sie so realitätsnah wie möglich abgebildet wird, um die finale grafische Benutzerschnittstelle darstellen zu können.

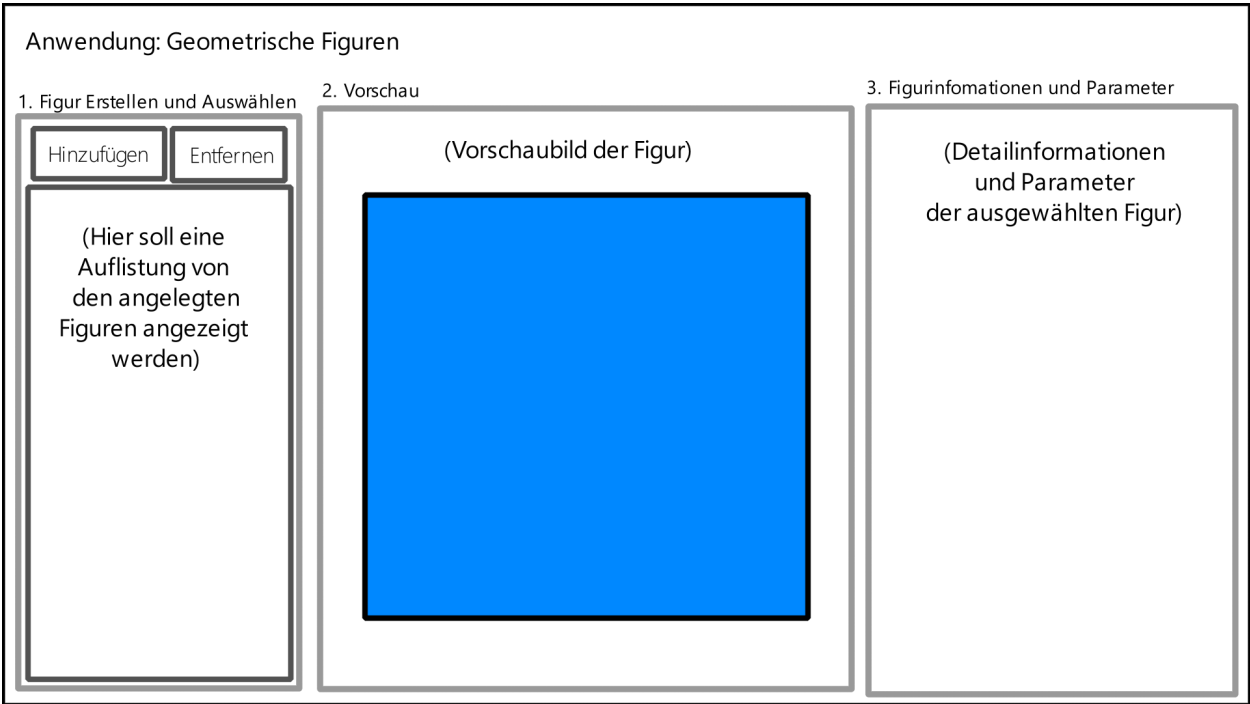


Abbildung 3: Mockup des Hauptbildschirms

3.2.4 Prototyp

Als der Entwurf feststand, wurde die grafische Benutzeroberfläche mit den angegebenen Mitteln unter Punkt: [1.4 Projektmittel](#) realisiert.

Dafür wurde eine Windows-Forms-Applikation erstellt, welche die nötigen Tools aufweist, um die fig. [3](#). Mock-Up-Skizze² leicht umsetzen zu können.

Die Oberfläche wurde mit der Mockup-Zeichnung aus fig. [3](#) erstellt und rustifiziert.

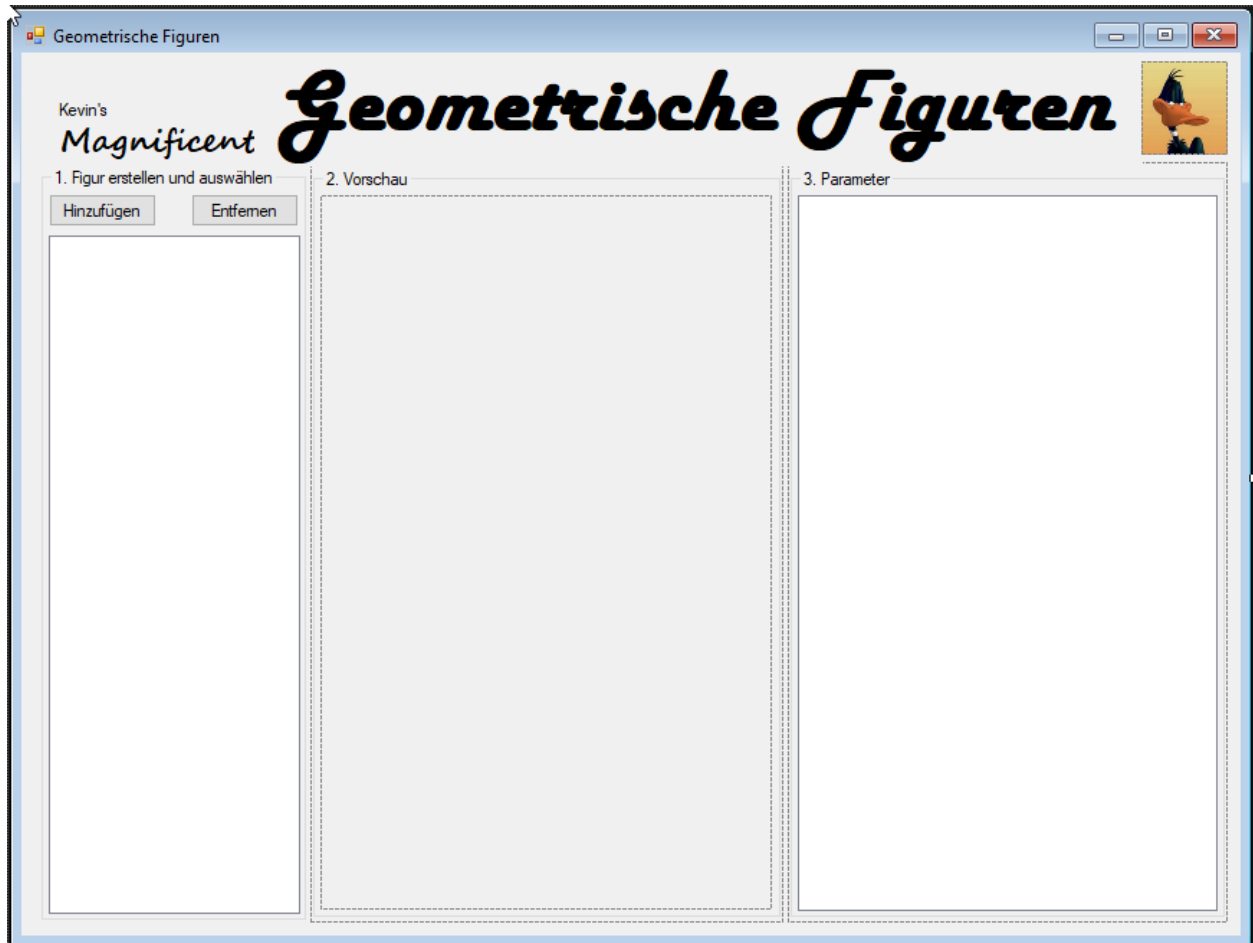


Abbildung 4: Prototyp Hauptoberfläche

4 Funktionalität

Des Weiteren wurde die Logik hinter der grafischen Benutzerschnittstelle implementiert.

Die Anwendung kann somit den funktionalen Anforderungen unter section [2](#) gerecht werden.

²Prototyp-Skizze von der grafischen Benutzeroberfläche

5 Objektorientierung

5.1 Klassendiagramm der Anwendung

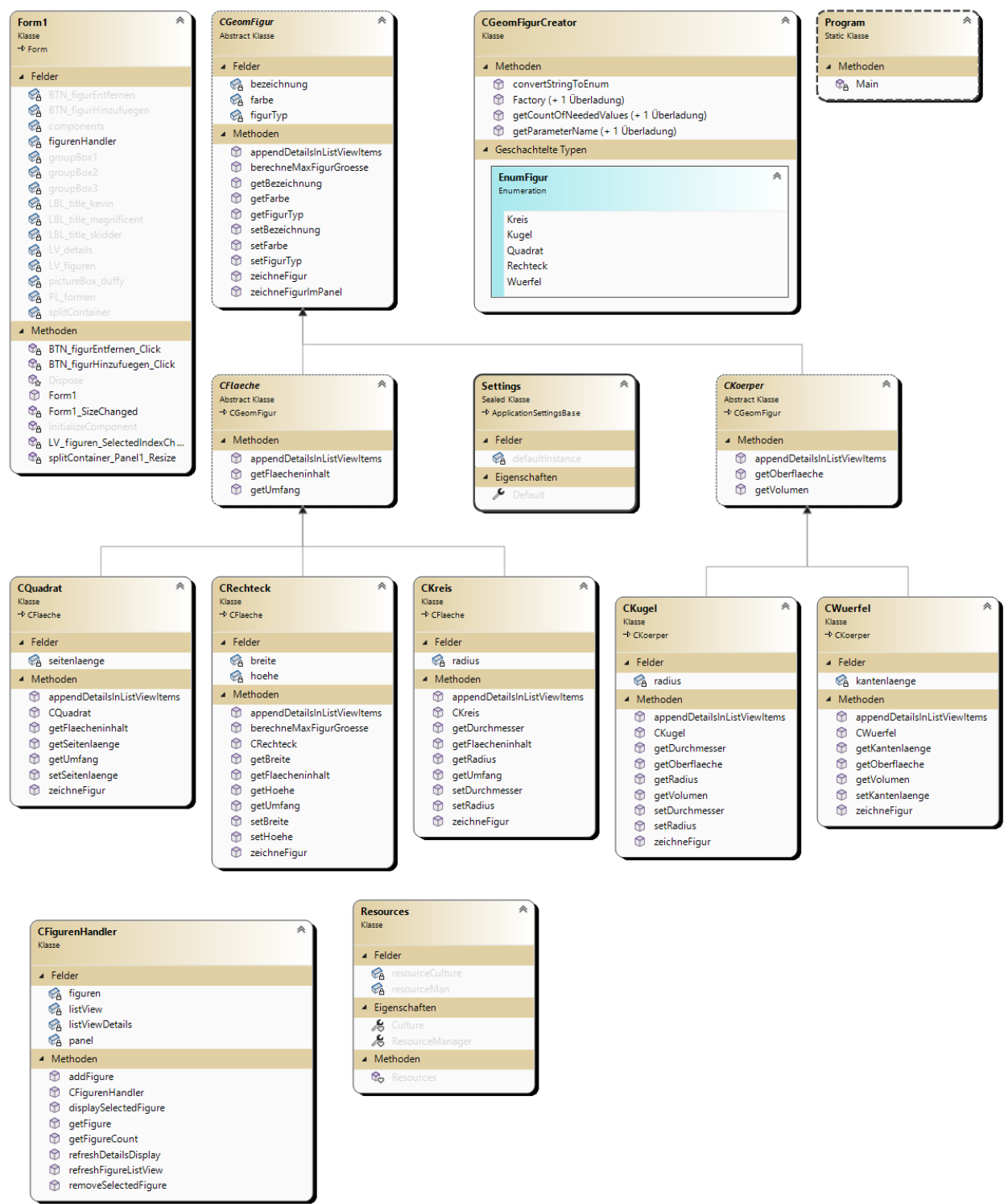


Abbildung 5: Klassendiagramm der Anwendung

5.1.1 Datenmodell

Das Datenmodell der Figuren umfasst die abstrakten Klassen: **CGeomFigur**, von dem die Klassen: **CFlaeche**, sowie **CKoerper** erben. Alle Klassen, welche die Generalisierungen **CFlaeche** oder **CKoerper** spezialisieren lassen sich instanzieren. Die Spezialisierungen implementieren entweder eine spezifische Fläche, bzw. einen Körper.

Des Weiteren wurde eine die Klasse **CGeomFigurCreator** implementiert, welche mittels des Design-Patterns “Factory” die unterschiedlichen Objekte des Typs **CGeomFigur** erzeugen kann.

Das erleichtert die Einbindung von neuen Geometrischen Figuren, welche je nach Bedarf hinzugefügt werden können.