

.NET Programmierung für industriennahe Anwendungen

Wintersemester 2022

(c) 2022

Roland J. Graf / Gerald Lochner

Labor 01: Einführung in C#/.NET

1.1. Themen

- Einführung in C#/.NET
- Unterschiede von C# zu C++
- Design Guidelines
- Dokumentation

1.2. Projektaufgaben

Im ersten Schritt lernen Sie C# und .NET kennen. Hierfür schreiben Sie einige Klassen und eine kleine Applikation zum Testen Ihrer Klassen.

1.2.1. Erstellungen von applikationsunabhängigen Klassen

Der Fokus sollte auf sowohl einfach zu verwendende, sichere und fehlertolerante Klassen als auch eine "schöne Implementierung der Klassen" gelegt werden (\Rightarrow *Quick & Dirty* ist hier nicht gefragt!!).

- **Würfel**
 - Klasse zum Verwalten eines beliebigen Würfels (die Anzahl der Würfelseiten $s \geq 4$)
 - Bezeichnung der Würfelseiten ist beliebig (z.B. 1,2,...,6 oder *Neun*, *Zehn*,..., *König*, *Ass* oder beliebig andere Bezeichnungen)
 - Simulation des Vorgangs *Würfeln* \Rightarrow Liefert eine zufällig ausgewählte Würfelseite
 - Möglichkeit zur Abfrage der Würfeigenschaften
- **Würfelsatz**
 - Klasse zum Verwalten mehrerer Würfel (z.B. ein Satz von 5 Würfel für Poker)
 - Würfeln mit dem gesamten Würfelsatz (z.B. mit allen 5 Würfel)
 - Würfeln eines einzelnen beliebigen Würfels aus dem vorhandenen Würfelsatz
 - Möglichkeit zur Abfrage der Würfelsatzeigenschaften und der Eigenschaften der Würfel
- **PokerHand**
 - Klasse zur Verarbeitung einer Poker-Hand; z.B. zum Ermitteln, ob mit einem Satz aus 5

Würfel mit je 6 Seiten ein *FiveOfKind*, *FullHouse*, *FourOfKind*,... gewürfelt wurde; hierfür sollten auch Würfel unterstützt werden, deren Seiten nicht 1,2...,6 sind, sondern auch beliebige andere 6-seitige Würfel :-)

1.2.2. Erstellung einer TestApplikation

- Schreiben Sie eine **TestApp**, welche die Verwendung der obigen Klassen zeigt und deren Funktionsfähigkeit vollumfänglich belegt
- Testen Sie alle Klassen, Methoden und Properties
- Abschließender Endtest, ob alles funktioniert:
Würfeln Sie 10.000.000-mal mit einem 5er-Satz und zählen Sie dabei die unterschiedlichen Kartenkombinationen; am Ende sollte die Verteilung der Kombination in xx.xx% ermittelt werden
- Erstellen Sie eine formatierte Ausgabe einer Ergebnistabelle und vergleichen Sie Ihre Ergebnisse mit den erwarteten Wahrscheinlichkeiten (siehe [Wikipedia](#) ⇒ [Würfelpoker](#) ⇒ [Kombinationen](#))

Allgemeine Anforderungen:

für alle Ihre Klassen und Quelltexte gelten folgende Regeln:

- Jede Klasse ist in einer eigenen Datei
- Jeder Quelltext beschreibt am Beginn der Datei in 1-2 Zeilen, was die Klasse macht/bietet
- am Beginn der Datei ist immer auch der Name des Autors der Klasse vermerkt
- Achten Sie auf einen *well-formed* Quelltext (Einrückungen, Zeilenumbrüche, Ausrichtungen,...); der Quelltext sollte einfach zu lesen sein
- Achten Sie auf selbstdokumentierenden Code (Bezeichner für Variablen, Methoden,...)
- Bestimmen Sie Richtlinien zur Entwicklung Ihrer Klassen/Quelltexte (z.B. für Formatierung, einheitliche Namenskonventionen, Typenbeschreibungen, Ausnahmebehandlungen,...); stellen Sie damit eine (API-)Konsistenz her; alle Quelltexte sollten *wie aus einer Hand* aussehen, auch wenn Sie im Team erarbeitet wurden; (als Anhaltspunkt siehe [C# Coding Conventions](#))
- Alle Methoden sollten so kommentiert werden, dass Visual Studio bei deren Verwendung im Editor eine Hilfestellung anzeigen kann (siehe [Dokumentationskommentare](#), verwenden Sie zumindest die Tags `<summary>`, `<param>` und `returns`)

Projektarbeiten

- Deadline, Abgabebedingungen und -formate: siehe Moodle-Kurs