

Aufgabe und Lösungsansatz

Aufgabe ist die Erstellung eines kleinen Programms in einer funktionalen Sprache. In meinem Fall habe ich mich für die Umsetzung der Aufgabe mit F# entschieden. Hierfür sprechen mehrere Faktoren.

Zum einen handelt es sich bei F# um eine "richtige" funktionale Sprache, wenn auch um keine Sprache, die nur rein funktional genutzt werden kann. F# verfügt auch über objektorientierte Konzepte. Zum anderen handelt es sich um eine Sprache im Umfeld von .NET, weshalb die CLR zugreif- und verwendbar ist. Die Ergebnisse der Arbeit können so einfach durch andere Sprachen des .NET Ökosystems genutzt werden.

Für mich als aktuell eher im Windows und C# Bereich agierender Entwickler ist dies daher eine sehr gute Option.

Lösung

Als Lösung habe ich ein einfaches NIMM-Spiel in F# als Konsolenanwendung umgesetzt. Der Code ist auf Git Hub verfügbar: <https://github.com/ChristianKitte/NimmSpiel>

Spielregeln

Zu Anfang muss ein Anfangswert zwischen aktuelle 5 und 20, dann eine Zahl als Schrittweite zwischen aktuell 3 und 6 eingegeben werden.

Dann spielen zwei Spieler im Wechsel miteinander. Ist ein Spieler am Zug, so gibt er eine Zahl im Bereich der Schrittweite ein. Diese Zahl wird von der aktuell vorhandenen Punktzahl abgezogen. Der Spieler, welcher als erster genau 0 erreicht, gewinnt. Rutscht die Punktzahl durch den Zug eines Spielers unter 0, so verliert dieser Spieler.

Ziel ist es somit, möglichst schnell genau alle anfänglichen Punkte abzuziehen, ohne unter 0 Punkte zu gelangen. Hierbei kann jedoch ein Spieler den anderen stören.

Erfahrungen

Das Programmieren in einer funktionalen Sprache unterscheidet sich doch sehr von der gewohnten Programmierung in einer objektorientierten, wie beispielsweise C#.

Abgesehen vom Einstieg, der doch auf Grund anderer Konzepte und Verhaltens etwas knifflig war, gewöhnt man sich jedoch leicht an das Vorgehen und die Syntax. Ich persönlich finde Gefallen an Ihr, auch wenn Visual Studio öfters eine Bereinigung gebrauchte und die IDE Unterstützung bei weitem nicht so gut wie für C# ist. Es fühlt sich

an, als wenn man “geradlinig” zu einer Lösung hinarbeite. In Teilen erinnert mich F# hierbei an Python.