Protokoll

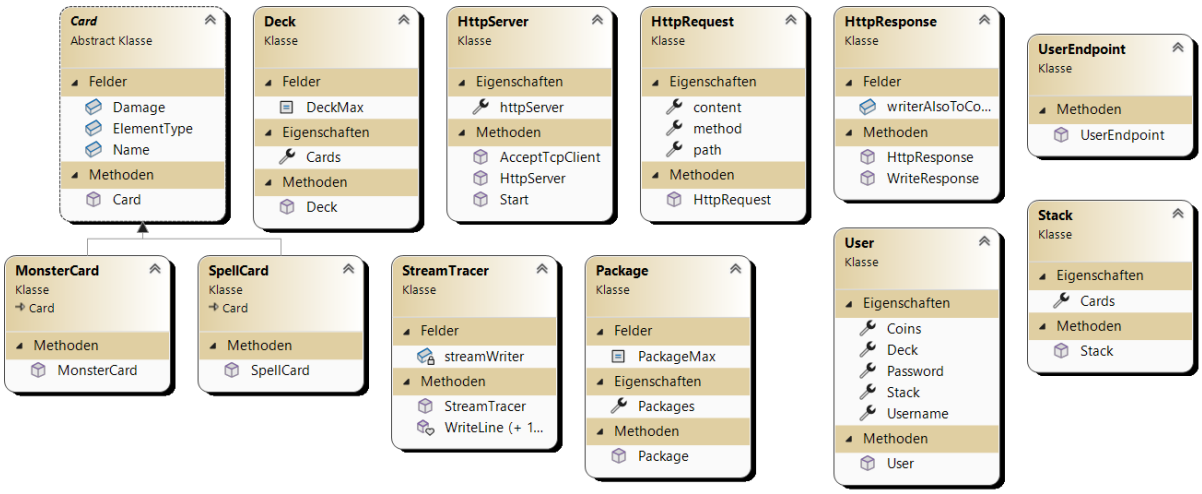
Decisions

Ich habe mich entschieden die httpserverdemo als Vorlage für den http Server zu verwenden und sie in Http Server, http Response und http Request aufgeteilt. Ich habe einen Userendpoint, um Login und Registrierung zu behandeln erschaffen. Die Request habe ich geparst um die Method, den Path und den Content aus der Request zu extrahieren und die Response mit variablen Eingabewerten und einer WriteResponse Methode versehen, um sie im Userendpoint nutzen zu können. Der Userendpoint behandelt alle Requests, die an den Pfaden /users und /sessions geschickt werden und führt bei /users die Registrierung durch und bei /sessions das Login. Im Userendpoint wird aus dem Content der Username und Passwort extrahiert, sollte es möglich sein. Im Programm befindet sich ein Dictionary, um die Datenbank zu simulieren, welches bei der Registrierung abgefragt wird, um zu schauen, ob ein User bereits existiert. Die Models beinhalten alle Attribute, die in der Spezifikation verlangt waren. Den Cards verwendet ich ein Enum um die drei Elementtypen darzustellen. MonsterCard und Spellcards habe ich als Kinder von Card erschaffen, weil sie die meisten Attribute teilen. Card ist eine abstrakte Klasse, weil ich davon ausgegangen bin, dass die Card Klasse als Objekt nicht zu existieren braucht, weil sie nur dient, um ihre Attribute an die Tochterklassen zu vererben und die Tochterklassen in Listen gemeinsam zu sammeln. Bei den Monstertypen bin ich davon ausgegangen, dass es reicht die Namen zu vergleichen, um die Spezialitäten darzustellen und daher keine Spezies oder Typ dafür definiert. Die Businesslogik habe ich noch nicht implementiert.

Structure

Das Projekt ist in Http Server und Models unterteilt.

Class diagramm



Gitlink

https://github.com/peziwezi/MTCG_Patrick_Rohrweckh.git