

Objektorientierte Modellierung und Programmierung

Dr. Christian Schönberg

Großübung Fehlerbehandlung und Generics

- Meldet euch als Teilnehmer mit Mikrofon an
- Sobald ihr eingewählt seid, deaktiviert Mikrofon und Video, indem ihr unten die beiden Symbole verwendet (auf den Symbolen müssen Mikrofon und Kamera durchgestrichen sein, dann sind sie deaktiviert)
- Aktiviert ausschließlich das Mikrofon, wenn ich euch drannehme. Deaktiviert es nach eurem Beitrag wieder
- Schreibt Wortmeldungen oder kurze Zwischenfragen in den Chat (einfach kurzes "meldung" oder "frage" o.ä.)

- Es gibt Spielkarten
- Es gibt Spielkarten, die aufgrund von bestimmten Werten miteinander vergleichbar sind
- Es gibt Spielkarten mit französischem Blatt
 - Farben: Karo < Herz < Pik < Kreuz
 - Werte: Ass < 2 < 3 < 4 < 5 < 6 < 7 < 8 < 9 < 10 < Bube < Dame < König < Joker
 - die Joker haben keine Farbe
- Für Skat werden nur folgende Werte in je allen vier Farben benötigt
 - Ass, 7 – 10, Bube, Dame, König
- Für Canasta werden alle Werte von allen Farben benötigt, sowie 6 Joker

Aufgabe 1

- Definieren Sie
 - Spielkarten
 - vergleichbare Spielkarten
 - Spielkarten mit franz. Blatt
 - Werte und Farben von Spielkarten
- Verwenden Sie dabei die bekannten objektorientierten Konzepte, z.B.
 - Interfaces, abstrakte Klassen
 - Generics
 - Enumerations
 - Exceptions
 - ...

- Definieren Sie
 - Kartenstapel
 - hier können Karten hinzugefügt werden
 - das Hinzufügen von ungültigen Karten (s.u.) führt zu einem Fehler
 - Skat-Kartenstapel
 - nur für Skat gültige Karten können hinzugefügt werden
 - Canasta-Kartenstapel
 - analog
- Verwenden Sie dabei die bekannten objektorientierten Konzepte, z.B.
 - Interfaces, abstrakte Klassen
 - Generics
 - Enumerations
 - Exceptions
 - ...

Aufgabe 3

- Definieren Sie
 - eine Klasse zur Erzeugung von Kartenstapeln
 - eine Klasse zur Erzeugung von Skat-Kartenstapeln
 - eine Klasse zur Erzeugung von Canasta-Kartenstapeln
- Verwenden Sie dabei geeignete Entwurfsmuster (Designpatterns)
- Verwenden Sie dabei die bekannten objektorientierten Konzepte, z.B.
 - Interfaces
 - abstrakte Klassen
 - Generics
 - Enumerations
 - Exceptions
 - ...