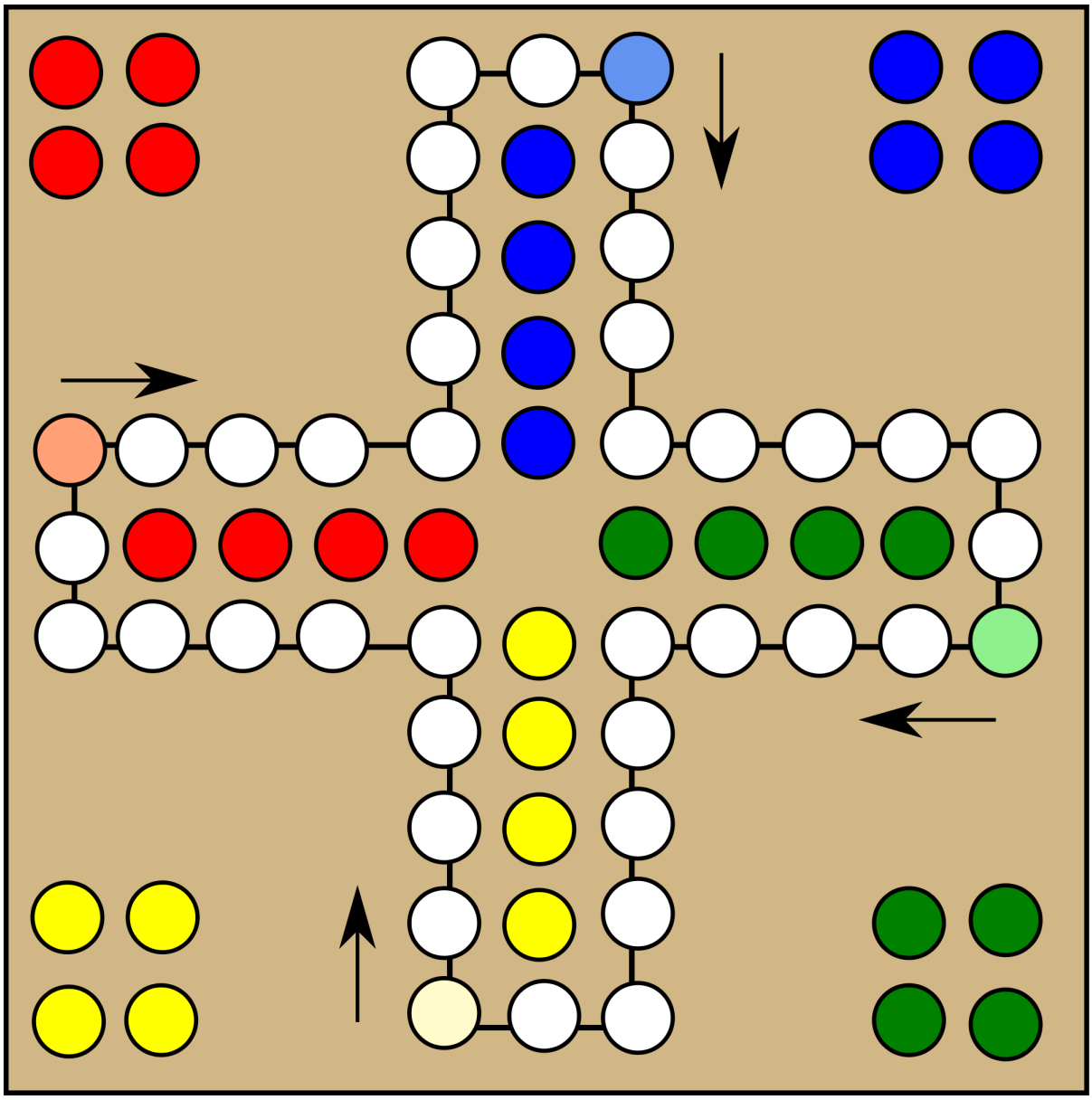
Mensch ärgere dich nicht



Quelle: https://upload.wikimedia.org/wikipedia/commons/thumb/9/91/Menschenaergern.svg/2000px-Menschenaergern.svg.png

Sachverhalt

In dem Spiel Mensch ärgere dich nicht, geht es darum, dass Spielfiguren über ein Spielbrett bewegt werden. Es können zwei bis vier Spieler gleichzeitig miteinander spielen. Jeder Spieler hat vier Initialpositionen außerhalb des Spielfeldes, eine Startposition und vier Endpositionen in der Mitte. Der Rundkurs hat 40 Positionen.
Jeder Spieler erhält einen Würfel (1-6), mit dem er an seinem Zug würfelt und danach entscheidet welche seiner Spielfiguren er bewegt. Um eine Spielfigur aus der Initialposition auf die Startposition zu bewegen, wird eine gewürfelte 6 benötigt.

Von Johannes Unterstein – unterstein@me.com

Sollte ein Spieler eine Figur auf eine Position bewegen, welche bereits von einer anderen Figur besetzt ist, so wird die bereits dort stehende Figur zurück auf eine Initialposition bewegt.
Das Spiel ist dann zu ende, wenn ein Spieler alle Figuren in den Endposition untergebracht hat. Selbstverständlich müssen die Figuren mit einem passenden Würfelwurf in eine freie Endposition bewegt werden.

Anmerkungen

In der Modellierung des Datenmodells sollte auf die in der Vorlesung vorgestellten Werkzeuge der Objektorientierung zurückgegriffen werden. So sollte es zum Beispiel eine Modellierung des Spieles geben, welche das Spielfeld und die Spieler beinhaltet. Das Spielfeld sollte die Belegung der Figuren auf den jeweiligen Position modellieren und so weiter …
Weiterhin sollten Funktionalitäten nahe an den zugehörigen Daten erfolgen, z.B. sollte die Wahl der nächsten Figur auf dem Spieler geschehen oder das Bewegen der Figur auf dem Spielfeld.
Zur Vereinheitlichung wird Position 0 des Spielfeldes auf die Startposition von blau (rechts oben) festgelegt.

Slack

https://join.slack.com/dhbw-java/shared_invite/MjA2MDgwMjUyMDgyLTE0OTg4MDU1MjEtN2EwOGZhNWY1Ng

Projekt Teil 1 - Datenmodell

Das Objektmodell von Mensch ärgere dich nicht soll zuerst mittels UML entworfen werden. Dabei sollen sowohl die Beziehungen der Klassen dargestellt werden, als auch die Funktionalität der Klassen.
Anschließend soll das Objektmodell in Java implementiert werden. Dabei ist die Klassenbibliothek von Java immer einzusetzen, wenn dies möglich ist. Grundlegend soll das Objektmodell die Daten gut kapseln, das heißt keinen direkten Zugriff auf die einzelnen Attribute erlauben, sondern stattdessen den Zugriff über get/set-Methoden zur Verfügung zu stellen.

Achtung wichtig: Die fachliche Logik des Programmes soll noch nicht implementiert werden. Also genau jene Logik, welche entscheidet, wann eine Figur wieder in den initial Bereich zurück gestellt werden muss und so weiter. Dies geschieht in Teil 3.

Projekt Teil 2 - Persistenz

Die Daten des Objektmodells der Mensch ärgere dich nicht Anwendung sollen vollständig in eine Datei persistiert werden. Beim Beenden der Anwendung sollen die Daten dazu automatisch in die Datei geschrieben und beim Start wieder automatisch ausgelesen werden. Dies ist vor dem Anwender zu verbergen.

Schritt 1 - Binärbasiert

Bitte schreibt eine neue Klasse `BinarySerializer`, welcher eine exemplarische Belegung des Spielfeldes über eine binäre Serialisierung auf die Festplatte schreibt und wieder auslesen kann.

Schritt 2 - Textbasiert

Bitte schreibt eine neue Klasse `TextSerializer`, welcher eine exemplarische Belegung des Spielfeldes über eine textbasierte Serialisierung auf die Festplatte schreibt und wieder auslesen kann.
Das Format der Datei soll textbasiert sein und ist vor der Implementierung in Java zuerst zu designen. Zum Einlesen und Schreiben des eigenen Formats sollen Streams verwendet werden. Dies kann entweder in einem XML-Format oder einem eigenen Dateiformat geschehen. Bei einem eigenen Dateiformat könnte ein Zielausgabe folgendermaßen aussehen:

```
Spiel;
Spieler;1,Peter
Initial;1
End;1,3
Spiel;36
AmZug;false
Spieler;2,Klaus
Initial;0
End;3
Spiel;4,24,10
AmZug;true
```

Diese serielle Darstellung würde ein Spielfeld beschreiben von zwei Spielern. Spieler 1 heißt Peter und Spieler 2 Klaus. Spieler 1 hat eine Figur im initialen Bereich und 2 im Endbereich (an der zweiten und letzten Position). Weiterhin hat er eine Figur an Position 37. Spieler 2 hat keine Figur im initialen Bereich, dafür aber eine Position im letzten Endbereich und 3 Figuren auf dem Rundkurs.

Projekt Teil 3 - Fachliche Logik und erste Eingaben

Schritt 1
Bitte implementiert die fachliche Logik des Programms.
Implementiert nun Methoden, wie zum **Beispiel** die Methode `bewegeFigur(figur: Figur, anzahl: Integer)` der Klasse Spielfeld. In dieser Methode muss geprüft werden, ob der Zug gültig ist, ob eine andere Spielfigur bereits die Position belegt, oder ob sich die Figur in den Zielbereich bewegt und das Spiel ggf. beendet wird.

Schritt 2
Bitte implementiert eine rudimentäre Eingabeverarbeitung.
Eingaben sind: Spielernamen und Nachfrage, welche Figur bewegt werden soll. Ablauf eines Spielzuges könnte wie folgt laufen:
1.) Es wird automatisch der Würfel geworfen, wenn ein Spieler am Zug ist
2.) Der Spieler wird gefragt, welche Figur er bewegen oder ins Spiel nehmen möchte bzw. ob er passen möchte, wenn kein Zug möglich ist
3.) Der nächste Spieler ist am Zug

Ein Spiel würfe dann so lange laufen, bis ein Spieler alle Figuren in seinen Zielbereich bewegt hat.

Ziel dieses Teiles ist es ein funktionsfähiges Spiel bereitzustellen, mit welchem 2-4 menschliche Spieler gegeneinander spielen können.

Mit folgendem Code kann man einen String vom Terminal einlesen:

```
	static String readString() {
		String eingabe = new String();
		try {
			BufferedReader i = new BufferedReader(new
InputStreamReader(System.in));
			eingabe = i.readLine();
		} catch (IOException e) {
			e.printStackTrace();
		}
		return eingabe;
	}
```

Projekt Teil 4 - UI

Gefordert ist eine Oberfläche für unser Mensch-Ärgere-Dich-Nicht-Spiel. Die Oberfläche muss die Möglichkeit bereitstellen ein neues Spiel zu starten oder ein bestehendes fortzusetzen. Weiterhin muss es möglich sein die Spielernamen einzugeben um danach zu spielen. Bevor jedem Zug wird automatisch gewürfelt und der aktuelle Spieler hat die Möglichkeit eine Figur zum ziehen auszuwählen.

Beim beenden des Spieles soll der aktuelle Spielstand persistiert werden und beim Starten soll dieser wieder geladen werden, so dass ein Spiel fortgesetzt werden kann.

Die Oberfläche sollte im besten Fall in JavaFX umgesetzt werden, da dies in der Vorlesung am umfangreichsten vorgestellt wurde. Eine Umsetzung in eine anderen Oberflächen-Technologie ist allerdings ebenfalls möglich.

Von Johannes Unterstein – unterstein@me.com