

### 1. Wolverine

Ziel dieses Beispiels ist die Implementierung der Suche nach einer Lösung des Wolf – Bänderinnen – Kohlkopf – Schaf(Ziege) - Problems, wie in der Vorlesung besprochen.

Erster Teil ist die speichereffiziente Implementierung eines GameState. Verwenden Sie für die interne Darstellung Bitmaskierung. Nach Außen soll der GameState sinnvolle Methoden zur Verfügung stellen, um die Position der Figuren zu ermitteln, Folgezustände zu generieren (eine Liste von Nachbarn). Die Methoden isValid() zum Prüfen, ob ein Zustand erlaubt ist, sowie isTerminationState(), die angibt, ob es sich um einen gültigen Endzustand handelt, sind ebenso sinnvoll. Implementieren Sie auch die Methoden clone() und hashCode().

Lösen Sie das Spiel „uninformiert“ einerseits durch Tiefensuche, andererseits durch Breitensuche. Beachten Sie bei der Tiefensuche, dass bereits besuchte Zustände erkannt werden müssen, um einen „ewigen“ Zyklus zu vermeiden (Sie haben ja schon die Methode hashCode()).

Ersetzen Sie in einer Variante der Implementierung die Bitmaskierung durch ein objektorientiert eleganteres Feld von Position- (Enumerations-) objekten. Beschreiben Sie kurz die Vor- und Nachteile der beiden Varianten.