Name:

Matrikelnummer:

Bearbeitungszeit: 120 min.

Aufgabe 1.
Für die auf einer Landkarte dargestellten Länder ist die Klasse `Land` mit der Methode

```
public List<Land> getNachbarn()
```

gegeben, die die Nachbarländer eines Landes zurück gibt. Schreiben Sie eine Klasse `Landkarte` mit dem Konstruktor

```
public Landkarte(List<Land> laender),
```

dem die Liste der darzustellenden Länder übergeben wird. Die Klasse `Landkarte` soll weiters die Methode

```
public boolean istZweifaerbig(),
```

enthalten, die überprüft, ob jedem darzustellenden Land eine von zwei Farben so zugeordnet werden kann, dass benachbarte Länder jeweils verschieden gefärbt sind.

Dieses Beispiel kann mittels Exhaustive Search gelöst werden, dafür erhalten Sie aber nur 8 von 10 möglichen Punkten. Die volle Punkteanzahl erhalten Sie nur für eine schnellere Lösung.

Aufgabe 2. Für die Benutzerverwaltung in einem IT-System sind die Klassen `User` und `ITsystem` gegeben. Die Klasse `ITsystem` verfügt über den Konstruktor

```
public ITsystem(String bezeichnung)
```

und die Methode

```
public User anmelden(String name, String password),
```

die einen Benutzer im System anmeldet. Ist die Anmeldung erfolgreich, dann wird das entsprechende `User`-Objekt zurückgegeben, und ansonsten `null`.

Schreiben Sie eine Unterklasse `SicheresITsystem` von `ITsystem` mit dem Konstruktor

```
public SicheresITsystem(String bezeichnung, int anzLoginFehler).
```

In `SicheresITsystem` soll nach `anzLoginFehler` *aufeinander folgenden* und fehlgeschlagenen Anmeldeversuchen für einen Benutzernamen dieser Benutzername gesperrt werden, also keine Anmeldung für diesen Benutzernamen mehr möglich sein.

Hinweis: Zwischen den erfolglosen Anmeldeversuchen für Benutzer `"Bob"` könnten erfolgreiche Anmeldungen des Benutzers `"Ann"` stattfinden. Dennoch soll nach der entsprechenden Anzahl von Fehlversuchen der Benutzer `"Bob"` gesperrt werden.

Aufgabe 3. Gegeben ist die Klasse `Spielkarte` mit der Methode

```
public boolean passtZu(Spielkarte sk).
```

Der Methodenaufruf `sk1.passtZu(sk2)` liefert genau dann `true`, wenn die Spielkarte `sk1` zur Spielkarte `sk2` passt.

Schreiben Sie eine Klasse `Spiel` mit der Methode

```
public static void entfernePassende(List<Spielkarte> kartenliste),
```

die aus `kartenliste` solange Paare `sk1!=sk2` mit `sk1.pastZu(sk2)==true` entfernt, bis es kein solches Paar gibt.