

**Name:**

**Matrikelnummer:**

### Aufgabe 1.

Gegeben seien die Klassen *Spielkarte* und *Kartenstapel*.

Die Klasse *Spielkarte* verfügt über die Methoden

```
public int getWert(),  
public String getFarbe(),
```

die den Wert bzw. die Farbe der Karte zurückgeben.

Die Klasse *Kartenstapel* verfügt über einen parameterlosen Konstruktor sowie über eine Methode

```
public Spielkarte getNext(),
```

die die nächste Karte vom Stapel entfernt und zurückgibt.

Schreiben Sie eine Klasse *UnoKartenstapel*, die von *Kartenstapel* erbt. Der parameterlose Konstruktor soll einen *UnoKartenstapel* anlegen, sowie die erste Karte vom Stapel nehmen und auflegen. Weiters soll die Menge der aufgenommenen Karten zu Beginn leer sein (siehe unten).

Die Methode

```
public ArrayList<Spielkarte> getAufgenommene()
```

soll die aufgenommenen Karten in einer *ArrayList* zurückgeben.

Die Methode

```
public Spielkarte getAufgelegte()
```

soll die aufgelegte Karte zurückgeben.

Überschreiben Sie weiters die Methode *getNext* in *UnoKartenstapel* so, dass die zurückgegebene Karte denselben Wert oder dieselbe Farbe wie die gerade aufgelegte Karte hat. Dazu soll zunächst unter den aufgenommenen Karten nach einer solchen Karte gesucht werden. Falls es eine solche Karte gibt, soll sie aus der Menge der aufgenommenen Karten entfernt und zurückgegeben werden. Falls es unter den aufgenommenen Karten keine passende Karte gibt, soll solange eine Karte vom Stapel aufgenommen werden, bis eine passende Karte gefunden wurde (d.h. nicht passende Karten sollen den aufgenommenen Karten hinzugefügt werden).

Die passende Karte soll von der Methode zurückgegeben werden und die aufgelegte Karte ersetzen.

## Aufgabe 2.

Gegeben seien die Klassen *Bauteil* und *Produkt*, in denen die Methoden *equals* und *hashCode* aus *Object* überschrieben sind.

In *Bauteil* gibt es weiters die Methode

```
public String getTyp(),
```

die den Typ des Bauteils zurückgibt.

In *Produkt* gibt es eine Methode

```
public List<String> getBauplan(),
```

die eine Liste von Bauteiltypen zurückgibt, die zum Bau des Produkts nötig sind.

Entsprechend verfügt *Produkt* über einen Konstruktor

```
public Produkt(List<Bauteil> dieBauteile),
```

der aus den Bauteilen ein entsprechendes Produkt erstellt.

Schreiben Sie eine Klasse *Produktion* mit Konstruktor

```
public Produktion(Produkt prototyp)
```

die eine Produktion für Produkte mit vorgegebenem Prototyp erstellt.

Schreiben Sie in die Klasse *Produktion* eine Methode

```
public Set<Produkt> produziere(List<Bauteil> dieBauteile),
```

die aus der übergebenen Liste von Bauteilen soviele dem Prototyp entsprechende Produkte wie möglich erzeugt und in einem Set zurückgibt. Die verwendeten Bauteile sollen aus der Liste *dieBauteile* entfernt werden.