

Name:

Matrikelnummer:

Aufgabe 1. Gegeben sei eine Klasse *Datei* mit Konstruktor

```
public Datei(String t),
```

der eine Datei erstellt, die den Text *t* enthält. Weiters gebe es in der Klasse *Datei* die Methoden

```
public String leseDatei()
```

und

```
public void schreibeDatei(String text),
```

mit denen die Datei ausgelesen bzw. neu geschrieben werden kann.

Schreiben Sie eine Klasse *GeschuetzteDatei*, die von *Datei* erbt. Der Konstruktor

```
public GeschuetzteDatei(String ttt),
```

soll eine den Text *ttt* enthaltende Datei erstellen, die sowohl für Lese- als auch Schreibzugriff gesperrt ist.

Implementieren Sie Methoden, mit denen der Lese- und Schreibzugriff neu gesetzt werden kann. Dabei soll eine Datei ohne Lesezugriff automatisch auch schreibgeschützt sein.

Überschreiben Sie die Methoden *leseDatei* und *schreibeDatei* so, dass die Datei nur dann gelesen bzw. neu geschrieben werden darf, wenn der Lese- und Schreibzugriff entsprechend gesetzt ist.

Aufgabe 2. Gegeben sei eine Klasse *Urne* mit Konstruktor

```
public Urne(),
```

der eine leere Urne erzeugt. Weiters verfügt die Klasse *Urne* über eine Methode

```
public void add(String color),
```

die der Urne einen Ball der Farbe *color* hinzufügt.

Schreiben Sie eine Klasse *Urnexperiment* mit Konstruktor

```
public Urnexperiment(),
```

der ein *Urnexperiment* mit einer leeren Urne erzeugt. Schreiben Sie in diese Klasse außerdem eine Methode

```
public boolean fuell(int anzRot, int anzBlau, int maxGesamt),
```

die der Urne solange zufällig (mit gleich hoher Wahrscheinlichkeit) jeweils einen blauen oder roten Ball hinzufügt, bis sich mindestens *anzRot* viele rote Bälle und mindestens *anzBlau* viele blaue Bälle in der Urne befinden. Sollte allerdings die Gesamtzahl an Bällen *maxGesamt* erreichen, bevor sich genügend rote und blaue Bälle in der Urne befinden, soll dieser Vorgang abgebrochen und *false* zurückgegeben werden. Andernfalls soll die Methode *true* zurückgeben.