

**Name:**

**Matrikelnummer:**

**Aufgabe 1.** Gegeben seien die Klassen *Song* und *MP3Player*.

Die Klasse *MP3Player* verfügt über einen Konstruktor

```
public MP3Player(ArrayList<Song> songs),
```

der einen *MP3Player* mit entsprechender Liste von Songs erstellt.

Weiters verfügt die Klasse *MP3Player* über die Methoden

```
public void addSong(Song einSong)
```

zum Hinzufügen eines Songs *einSong* an das Ende der Songliste,

```
public void deleteSong(int i)
```

zum Löschen des *i*-ten Songs in der Songliste, und

```
public void playSong(int i)
```

zum Abspielen des *i*-ten Songs in der Songliste.

Schreiben Sie eine Klasse *MP3PlayerPlus*, die von *MP3Player* erbt. Neben einem Konstruktor Ihrer Wahl soll diese Klasse über eine Methode

```
public int gibAnzahlWiedergaben(int i)
```

verfügen, die angibt, wie oft der *i*-te Song bereits abgespielt wurde. (Dazu müssen Sie die Methode *playSong* überschreiben.) Diese Methode soll auch nach Löschen von Songs und nach Hinzufügen von neuen Songs die richtige Anzahl zurückgeben. (Dazu müssen Sie die Methoden *addSong* und *deleteSong* überschreiben.)

*Hinweis:* Falls ein Song zwischenzeitlich gelöscht wurde, soll die Zählung der Wiedergaben nach neuem Hinzufügen wieder bei 0 beginnen. Es ist also nicht nötig, sich die Anzahl der Wiedergaben für gelöschte Songs zu merken.

**Aufgabe 2.** Gegeben seien die Klassen *Anlage* und *Produkt*.

Die Klasse *Anlage* verfügt über die Methode

```
public boolean istUeberprueft(),
```

die genau dann *true* zurückgibt, wenn die Anlage überprüft wurde.

Die Klasse *Produkt* verfügt über die Methoden

```
public int gibStueckzahl(),
```

die die zu produzierende Stückzahl des Produkts zurückgibt, und

```
public Set<Anlage> gibAnlagen(),
```

die ein Set von jenen Anlagen zurückgibt, die das Produkt bearbeiten können. Weiters sind die Methoden *equals(Object)* und *hashCode()* aus *Object* in *Produkt* überschrieben.

Schreiben Sie eine Klasse *BottleneckAnalyse* mit einer statischen Methode

```
public static Map<Produkt, Anlage> gibRisikoprodukte(List<Produkt> dieProd),
```

die alle hochfrequentigen Produkte (Stückzahl > 100000) in *dieProd*, die nur auf einer einzigen überprüften Anlage bearbeitet werden können, in einer Map zurückgibt. Dabei soll in der Map für jedes entsprechende Produkt die einzige überprüfte Anlage zur Bearbeitung abgespeichert werden.