Übungen zu Kapitel 10

10.1 Urlaubsverwaltung:

Eine Anwendung soll den Jahresurlaub verwalten. Dazu dient die nichtausführbare Klasse `Urlaubskonto` und die ausführbare Klasse `Planung`.

Die nichtausführbare Klasse `Urlaubskonto`:

Attribute:

```
private int restUrlaub;
private static final int ANSPRUCH = 30;
private String vertreter = "N.N.";
private boolean genehmigt = false;
```

Methoden:

`public Urlaubskonto( int vorjahr )` : Der Konstruktor setzt das Attribut `restUrlaub` auf den Wert `ANSPRUCH+vorjahr`. Der Parameter `vorjahr` enthält den noch nicht genommenen Resturlaub des Vorjahres.

`public boolean buchen( int tage, String stv )` : Sofern `tage<=resturlaub` wird der verbleibende Resturlaub um die Zahl `tage` verringert. In diesem Fall wird das Attribut genehmigt auf `true` gesetzt, andernfalls auf `false`. Die Methode `setVertreter()` soll mit `stv` aufgerufen werden. Der Wert des Attributs `genehmigt` wird zurückgegeben

`private void setVertreter( String v )` : Das Attribut vertreter wird auf den Wert `v` gesetzt, fall das Attribut `genehmigt` den Wert `true` hat. Andernfalls wird `vertreter` auf "N.N." gesetzt.

`public String getVertreter( )` : Das Attribut `vertreter` wird zurückgeliefert.

Die ausführbare Klasse `Planung`:

In ihrer `main()`-Methode soll der Benutzer nach der Höhe des Vorjahresresturlaubs gefragt und ein `Urlaubskonto`-Objekt angelegt werden. Solange (Schleife!) noch ein Urlaubsanspruch besteht soll der Benutzer Urlaub buchen können. Es soll ausgegeben werden, ob der Urlaub buchbar war oder nicht und wieviel Resturlaub jeweils verbleibt. Auch der Name des Vertreters soll ausgegeben werden.

10.2 Automat:

Es soll ein Getränkeautomat simuliert werden, der mit Flaschen und Geld gefüllt ist. Flaschen können gezogen und nachgefüllt werden, Geld wird beim Kauf eingegeben und Rückgeld wird zurückgegeben. Eine Flasche kostet 2 GE. Die nichtausführbare Klasse Automat verwaltet den Automateninhalt an Geld und Flaschen.

Die nichtausführbare Klasse `Automat`:

Attribute:

```
private int flaschen;
private double geld;
private static final double PREIS = 2.0;
private static final int MAXFLASCHEN = 10;
```

Methoden:

`public Automat( int f, double g )`: Der Konstruktor setzt den anfänglichen Flaschen- und Geldbestand. Mehr als `MAXFLASCHEN` können nicht aufgenommen werden.

`public int getFlaschen()`: Liefert die aktuelle Anzahl Flaschen im Automaten.

`public double getGeld()`: Liefert die aktuelle Geldmenge im Automaten.

`public double entnehmen( double einWurf )`: Der Parameter `einWurf` ist die eingegebene Geldmenge. Wenn `einWurf >= PREIS` ist, wird eine Flasche (durch Konsolenausgabe) geliefert, d.h. `flaschen` um 1 verringert und `geld` um `PREIS erhöht`. Der Betrag des Rückgeldes (`einWurf-PREIS`) wird zurückgegeben. Andernfalls wird die Annahme des Geldes verweigert und der `einWurf` zurückgegeben, ohne dass sich `flaschen` und `geld` ändern.

Die ausführbare Klasse `Betrieb`:

In der `main()`-Methode wird ein Automatenobjekt mit Anfangsinhalt initialisiert. Solange (Schleife!) der Automat noch Flaschen enthält, können Kunden bedient werden.

Die eingeworfene Geldmenge wird abgefragt. Der aktuelle Bestand an Flaschen und Geld wird gemeldet.

10.3 Studenten:

Entwerfen und implementieren Sie eine Klasse `Student`.

Studenten haben einen Namen, gehören zu einer Hochschule, studieren ein Fach, befinden sich in einem Semester und haben sind zur Prüfung bereits angemeldet oder auch nicht. Eindeutiges Merkmal ist die Matrikelnummer, bei der es sich um eine fünfstellige Zahl handeln soll. Die Klasse `Student` soll über die Zahl ihrer Objekte informiert sein. Alle Attribute seien privat.

Lesender und schreibender Zugriff darauf soll durch Set- und Get-Methoden realisiert werden. Neben dem Konstruktor soll eine statische Methode zur Objekterzeugung vorhanden sein, die vom Benutzer (mittels `IO`) alle nötigen Daten erfragt und ein `Student`-Objekt zurückliefert. Eine Methode zur kompletten Datenausgabe eines `Student`-Objekts soll vorhanden sein. Eine statische Methode soll zwei `Student`-Objekte als Parameter entgegennehmen und deren Semesterzahl auf Gleichheit prüfen. Das Ergebnis (`true` oder `false`) soll zurückgegeben werden.

Geben Sie der Klasse `Student` zu Testzwecken auch eine `main()`-Methode. Testen Sie in dieser die Implementation durch Erzeugen von `Student`-Objekten und Aufruf aller Methoden.