Aufgabe zu Abstrakten Klassen und Interfaces (Parkhaus)

Überblick über die Quellen:
- [**?**]
- [**?**, Seite 11, Thema 2, Teilaufgabe 2, Aufgabe 1]

In dieser Aufgabe werden Sie Datentypen für die Verwaltung eines Parkhauses mit Hilfe objektorientierter Methoden definieren. Bearbeiten Sie die folgenden Teilaufgaben in einer objektorientierten Programmiersprache Ihrer Wahl (geben Sie diese an)! Solange nicht anders definiert, sollen Eigenschaften und Methoden *uneingeschränkt sichtbar* sein.

(a) Erzeugen Sie eine *Klasse* `Fahrzeug`, deren Instanzen folgende Eigenschaften besitzen (wählen Sie geeignete Typen):

- Ein amtliches Kennzeichen (Buchstaben- und Zahlenkombination).
- Die Dimensionen des Fahrzeugs (Länge, Breite, Höhe) in Metern.
- Das Datum der Erstzulassung. Definieren Sie hierfür entweder einen eigenen Datentyp oder machen Sie Gebrauch von der Standardbibliothek Ihrer gewählten Programmiersprache.

Die Eigenschaften sollen für *Unterklassen nicht sichtbar* sein. Schreiben Sie außerdem einen Konstruktor, der eine Instanz erzeugt und die Eigenschaften setzt!

```java
import java.util.Date;

public class Fahrzeug {
  @SuppressWarnings("unused")
  private String kennzeichen;
  private double laenge;
  private double breite;
  private double hoehe;
  @SuppressWarnings("unused")
  private Date erstzulassung;

  public Fahrzeug(String kennzeichen, double laenge, double breite,
  ↪  double hoehe, Date erstzulassung) {
    this.kennzeichen = kennzeichen;
    this.laenge = laenge;
    this.breite = breite;
    this.hoehe = hoehe;
    this.erstzulassung = erstzulassung;
  }

  public double gibLaenge() {
    return laenge;
  }

  public double gibBreite() {
    return breite;
  }

  public double gibHoehe() {
    return hoehe;
  }
```

```
33
34  }
```

(b) Schreiben Sie eine *Klasse Parkplatz*, in der ebenfalls Eigenschaften für die Dimension (Länge, Breite, Höhe) in Metern vorgesehen sind! Die Eigenschaften sollen ebenfalls im *Konstruktor* initialisiert werden können. Außerdem soll eine Objektmethode hinzugefügt werden, die prüft, ob ein gegebenes Fahrzeug in den Parkplatz passt.

```
3   public class Parkplatz {
4     public double laenge;
5     public double breite;
6     public double hoehe;
7     public boolean frei;
8     public boolean reserviert;
9
10    public Parkplatz(double laenge, double breite, double hoehe) {
11      this.laenge = laenge;
12      this.breite = breite;
13      this.hoehe = hoehe;
14      frei = true;
15      reserviert = false;
16    }
17
18    public boolean groessePruefen(Fahrzeug f) {
19      if (f.gibLaenge() < this.laenge && f.gibBreite() < this.breite &&
        ↪  f.gibHoehe() < this.hoehe) {
20        return true;
21      }
22      return false;
23    }
24
25  }
```

(c) Ein *Interface* `Parkhaus` soll Objektmethoden für folgende Anwendungsfälle deklarieren:

(i) Alle freien Parkplätze sollen (z.B. als *Array* oder als Instanz einer in der Standardbibliothek Ihrer verwendeten Sprache definierten Kollektionsklasse) zurückgegeben werden.

(ii) Der erste freie Parkplatz, der zu einem gegebenen Fahrzeug passt, soll zurückgegeben werden.

(iii) Ein gegebener Parkplatz soll für ein gegebenes Fahrzeug reserviert werden. Deklarieren Sie das Interface und geben Sie geeignete Signaturen die *Objektmethoden* an!

```
3   public interface Parkhaus {
4     public Parkplatz[] freieParkplaetze();
5
6     public Parkplatz ersterFreierPlatz(Fahrzeug f);
7
8     public void reservieren(Parkplatz p, Fahrzeug f);
9   }
```

(d) Schreiben Sie eine *abstrakte Klasse*, die das Interface `Parkhaus` partiell implementiert! Geben Sie außerdem eine geeignete *Implementierung* für die Objektmethode (c.ii) unter Verwendung der existierenden Objektmethoden aus (b) und (c.i) an!

```
3  public abstract class MeinParkhaus implements Parkhaus {
4
5    public abstract Parkplatz[] freieParkplaetze();
6
7    public Parkplatz ersterFreierPlatz(Fahrzeug f) {
8      Parkplatz freierPlatz = null;
9      Parkplatz[] freiePlaetze = freieParkplaetze();
10     for (int i = 0; i < freiePlaetze.length; i++) {
11       Parkplatz p = freiePlaetze[i];
12       if (p.groessePruefen(f)) {
13         freierPlatz = p;
14       }
15     }
16     return freierPlatz;
17   }
18
19   public abstract void reservieren(Parkplatz p, Fahrzeug f);
20 }
```