

Hinweise für den Prüfling

Auswahlzeit: 30 Minuten

Bearbeitungszeit (insgesamt): 180 Minuten

Auswahlverfahren

Es gibt zwei Aufgabengruppen A und B, aus denen jeweils ein Vorschlag zu bearbeiten ist. Der vorliegende Vorschlag aus der Gruppe A (Objektorientierte Modellierung) ist ein Pflichtvorschlag.

Wählen Sie von den zwei vorliegenden Vorschlägen der Gruppe B (Datenbanken) einen zur Bearbeitung aus. Der nicht ausgewählte Vorschlag muss am Ende der Auswahlzeit der Aufsicht führenden Lehrkraft zurückgegeben werden.

Erlaubte Hilfsmittel

1. ein Wörterbuch der deutschen Rechtschreibung
2. eine Liste der fachspezifischen Operatoren

Sonstige Hinweise

ohne PC-Nutzung

In jedem Fall vom Prüfling auszufüllen

Name: _____	Vorname: _____
Prüferin / Prüfer: _____	Datum: _____

Vehicle to Grid

Unter Vehicle to Grid (deutsch: Fahrzeug ans Netz) versteht man ein Konzept zur Speicherung und Abgabe elektrischer Energie aus dem öffentlichen Stromnetz in Elektro- und Hybridautos. Da durchschnittlich über 90% aller Autos geparkt sind, könnte man diese als Energiespeicher einsetzen. Bei Versorgungsengpässen wird ein Teil der in den Autobatterien gespeicherten Energie kurzfristig abgerufen; bei Überschüssen dienen sie als Energiespeicher.

Aufgaben

1. Im Folgenden soll ein vereinfachtes Grid für maximal zehn Elektroautos simuliert werden. Von jedem Auto werden das Kennzeichen, der Name des Halters und der momentane Ladezustand der Batterie in Energieeinheiten gespeichert. Diese Daten sollen jederzeit abgefragt und der Ladezustand um einen bestimmten Betrag geändert werden können. Das Grid selbst soll über Methoden zum Ausgeben der Daten aller Autos sowie zur Bestimmung des Indexes des Autos mit dem höchsten Ladezustand verfügen. Außerdem soll der Gesamtladezustand des Grids berechnet und gespeichert werden.

Modellieren Sie ein Grid durch ein UML-Klassendiagramm mit den beiden Klassen *TElektroauto* und *TAutogrid*. Sehen Sie in der Klasse *TElektroauto* alle Attribute und Methoden vor, mit denen die genannten Informationen gespeichert und abgefragt werden können. Bei der Modellierung der Klasse *TAutogrid* sind die im Material verwendeten Methoden und die sich daraus ergebenden Attribute zu berücksichtigen. Erläutern Sie die Beziehung zwischen den beiden Klassen.

(16 BE)

2. Analysieren und beschreiben Sie die beiden Methoden *bestimmeMaxIndex* und *bestimmeRelativenMaxIndex* (Material).

Erläutern Sie die inhaltliche Bedeutung der beiden Methoden.

(15 BE)

3. Implementieren Sie die Methode *holeEnergie(Betrag: integer)*, die dem Grid insgesamt den *Betrag* an Energieeinheiten entzieht. Die Energieentnahme erfolgt schrittweise und zwar immer bei dem Auto mit dem höchsten Ladezustand und beträgt stets eine Energieeinheit. Autos, deren Ladezustand 50 Energieeinheiten oder weniger beträgt, wird keine Energie entzogen. Die Energieentnahme endet, wenn der *Betrag* entnommen wurde oder der Ladezustand aller Autos kleiner gleich 50 ist.

(17 BE)

4. Entwerfen Sie das Struktogramm zu einem Algorithmus, der die Daten der Elektroautos des Grids nach ihrem Ladezustand in absteigender Reihenfolge ausgibt, ohne das Grid jedoch vorher zu sortieren, d. h. jedes Elektroauto behält seinen Feldindex.

(12 BE)

Material

```
01 type
02
03 TAutogrid = class
04     private
05         function bestimmeRelativenMaxIndex: integer;
06         ...
07     public
08         function bestimmeMaxIndex: integer;
09         procedure holeEnergie(Betrag: integer);
10         ...
11     end;
12
13 implementation
14
15 function TAutogrid.bestimmeMaxIndex: integer;
16     var i, MaxIndex: integer;
17 begin
18     MaxIndex := 1;
19     for i := 2 to Anzahl do
20         if Elektroautos[i].getLadezustand > Elektroautos[MaxIndex].getLadezustand
21             then MaxIndex := i;
22     Result := MaxIndex;
23 end;
24
25 function TAutogrid.bestimmeRelativenMaxIndex: integer;
26     var i, Index, Maximum: integer;
27 begin
28     Index := 0;
29     Maximum := -1;
30     for i := 1 to Anzahl do
31         if (Elektroautos[i].getLadezustand > Maximum) and
32             (Elektroautos[i].getAusgegeben = false)
33             then begin
34                 Index := i;
35                 Maximum := Elektroautos[i].getLadezustand;
36             end;
37     Elektroautos[Index].setAusgegeben(true);
38     Result := Index;
39 end;
40
41 procedure TAutogrid.holeEnergie(Betrag: integer);
42 begin
43     ...           // siehe Aufgabe 3
44 end;
```