

Informatik 2 für Regenerative Energien

Klausur vom 14. März 2012

Jörn Loviscach

Versionsstand: 15. März 2012, 10:20



This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Germany License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/3.0/de/> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.

15 Punkte für die erste Aufgabe; 3 Punkte für alle weiteren Aufgaben. Mindestpunktzahl zum Bestehen: 15 Punkte. Hilfsmittel: maximal vier einseitig oder zwei beidseitig beschriftete DIN-A4-Spickzettel beliebigen Inhalts, möglichst selbst verfasst oder zusammengestellt; kein Skript, keine anderen Texte, kein Taschenrechner, kein Computer, kein Handy und Ähnliches.

Name	Vorname	Matrikelnummer	E-Mail-Adresse

1. Im C#-Programmlisting im Anhang sind 15 Fehler, darunter keine Tippfehler und höchstens ein Fehler pro Zeile. Stellen Sie eine Liste dieser Art mit allen Fehlern auf:

Zeile	korrekter Programmtext
123	public void foo()
543	int a = 42;

2. Mit den (korrigierten) Klassen aus dem Anhang wird Folgendes ausgeführt. Welche Werte stehen danach in den Variablen a, b und c?

```
Fuhrpark fp2 = new Fuhrpark();
fp2.FügeHinzu(new Fahrzeug(1, "mein Einrad"));
fp2.FügeHinzu(new Fahrzeug(4, "BI-AUTO 11"));
fp2.FügeHinzu(new Fahrzeug(8, "BI-LASTER 12"));
bool a = fp2.IstInOrdnung();
Fahrzeug f2 = fp2.SucheKennzeichen("BI-AUTO 11");
Fahrzeug f3 = f2;
f3.Fahre(60000.0);
bool b = f3.IstInOrdnung();
f2.Fahre(60000.0);
bool c = fp2.IstInOrdnung();
```

3. Angenommen, die (korrigierte) Klasse `Fahrzeug` im Anhang soll auch Fahrräder ohne Motor und ohne Kennzeichen verwalten können. Dafür baut man noch diesen zweiten Konstruktor ein:

```
public Fahrzeug()
{
    motor = null;
    räder = new Rad[2];
    for (int i = 0; i < 2; i++)
    {
        räder[i] = new Rad();
    }
    this.kennzeichen = "";
}
```

Was muss dann noch an der Klasse `Fahrzeug` geändert werden?

4. Schreiben Sie eine öffentliche Methode `GibProblemFahrzeuge` für die (korrigierte) Klasse `Fuhrpark` im Anhang. Diese Methode soll einen `string` liefern, der die Kennzeichen aller Bielefelder (!) Fahrzeuge enthält, die nicht in Ordnung sind, jeweils durch Leerzeichen getrennt. Nach dem letzten Wort darf ein Leerzeichen stehen. Hinweis: `string` hat eine Methode `StartsWith(string v)`, die einen booleschen Wert zurückgibt, um anzugeben, ob der Anfang der Zeichenkette gleich `v` ist.
5. Erweitern Sie die Methode `FügeHinzu` der (korrigierten) Klasse `Fuhrpark` im Anhang so, dass er eine `Exception` wirft, wenn man versucht, ein Fahrzeug hinzuzufügen, dessen Kennzeichen schon bei einem Fahrzeug der Liste `fahrzeuge` vorkommt.
6. Welche Zahlen stehen nach Ausführung dieses C#-Programmfragments in den Variablen `d`, `e` und `f`? Geben Sie möglichst auch Zwischenschritte an, damit Ihr Gedankengang nachvollziehbar ist.

```
Stack<int> a = new Stack<int>();
a.Push(10);
a.Push(11);
a.Push(12);
Stack<int> b = new Stack<int>();
b.Push(20);
b.Push(21);
List<Stack<int>> c = new List<Stack<int>>();
c.Add(a);
c.Add(b);
int d = c[1].Count;
int e = c[1].Pop();
int f = c[1].Count;
```

7. Zeichnen Sie ein UML-Klassendiagramm für die (korrigierten) Klassen `Prüfobjekt`, `Motor` und `Rad` aus dem Anhang.

8. Die Klasse `Prüfobjekt` soll so erweitert werden, dass jedes Rad, jeder Motor usw. eine eindeutige Inventarnummer erhält, angefangen bei 0. Welche Änderungen sind dazu an `Prüfobjekt` nötig?

Dieses Listing enthält 15 Fehler!

Dies soll eine Verwaltung für die Fahrzeuge einer Firma werden.

```
1 abstract class Prüfobjekt
2 {
3     abstract public bool IstInOrdnung();
4 }
5
6 class Motor : Prüfobjekt
7 {
8     double gefahreneKilometer = 0.0;
9
10    public double Fahre(double kilometer)
11    {
12        gefahreneKilometer += kilometer;
13    }
14
15    override public bool IstInOrdnung()
16    {
17        return gefahreneKilometer < 100000.0;
18    }
19 }
20
21 class Rad : Prüfobjekt
22 {
23     int luftdruck = 2.3;
24
25    override public bool IstInOrdnung()
26    {
27        return luftdruck > 2.0 && < 3.0;
28    }
```

```
29 }
30
31 class Fahrzeug : Prüfobjekt
32 {
33     Motor motor;
34     Rad[] räder;
35     string kennzeichen;
36
37     public Fahrzeug(int anzahlRäder, string kennzeichen)
38     {
39         motor = new Motor();
40         räder[] = new Rad[anzahlRäder];
41         for (int i = 0; i < anzahlRäder; i++)
42         {
43             räder[i] = new Rad(4);
44         }
45         this.kennzeichen = kennzeichen;
46     }
47
48     public string Kennzeichen
49     {
50         get
51         {
52             return kennzeichen;
53         }
54     }
55
56     public void Fahre(double kilometer)
57     {
58         motor.Fahre(double kilometer);
59     }
60
61     public bool IstInOrdnung()
62     {
63         if(!motor.IstInOrdnung())
64         {
65             return false;
66         }
67
68         for (int i = 0; i < anzahlRäder; i++)
69         {
70             if(!räder[i].IstInOrdnung())
71             {
72                 return false;
73             }
74         }
75
76         return true;
77     }
78 }
79
```

```
80 class Fuhrpark
81 {
82     fahrzeuge = new List<Fahrzeug>();
83
84     public void FügeHinzu(Fahrzeug f)
85     {
86         fahrzeuge.Add(f);
87     }
88
89     public Fahrzeug SucheKennzeichen(string kennzeichen)
90     {
91         foreach(Fahrzeug f in fahrzeuge)
92         {
93             if(f.Kennzeichen == kennzeichen)
94             {
95                 return f;
96             }
97         }
98         return false;
99     }
100
101     override public bool IstInOrdnung()
102     {
103         foreach(Fahrzeug f in fahrzeuge)
104         {
105             if(!f.IstInOrdnung())
106             {
107                 return false;
108             }
109         }
110         return true;
111     }
112 }
113
114 class Test
115 {
116     public static void Main()
117     {
118         Fuhrpark fp = new Fuhrpark();
119         fp.FügeHinzu(new Fahrzeug(4, "BI-AB_123"));
120         Fahrzeug f = SucheKennzeichen("BI-AB_123");
121         f.Fahre("200000.0");
122         string zustand = "OK";
123         if(!fp.IstInOrdnung)
124         {
125             zustand = "nicht_OK";
126         }
127     }
128 }
```