Informatik 2 für Regenerative Energien

Probeklausur vom 13. Juni 2011

Jörn Loviscach

Versionsstand: 16. Juli 2011, 08:48

15 Punkte für die erste Aufgabe; 3 Punkte für alle weiteren Aufgaben. Mindestpunkt[c1]zahl 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.

[c1] *text added by jl*

Name	Vorname	Matrikelnummer	E-Mail-Adresse, falls nicht in Mailingliste

1. Im C#-Programmlisting im Anhang sind 15 Fehler, darunter keine Tippfehler und höchstens ein Fehler pro Zeile. Stellen Sie[c2] eine Liste dieser Art mit allen Fehlern auf:

   [c2] *text added by jl*

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

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

   ```
   Queue<Bilanzposten> q = new Queue<Bilanzposten>();
   q.Enqueue(new Gebäude());
   q.Enqueue(new Fahrzeug());
   double a = q.Dequeue().GibWert();
   int b = q.Count;
   double c = q.Dequeue().GibWert();
   ```

3. In der Klasse `Bilanzposten` im Anhang wird dies ergänzt:

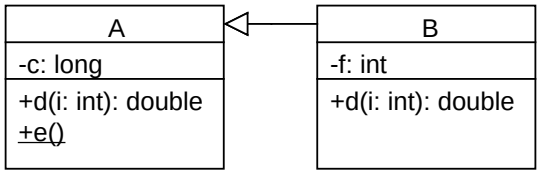
   ```
   TimeSpan abschreibungsdauer;
   public Bilanzposten(TimeSpan t)
   {
       abschreibungsdauer = t;
   }
   ```

Das verlangt Änderungen an der Klasse `Gebäude`. Wie kann man die Klasse `Gebäude` zum Beispiel reparieren? (Programmcode angeben) Hinweis: `TimeSpan` hat eine statische Methode `FromDays`.

4. In der Methode `SetzeWert` der Klasse `Bilanzposten` im Anhang soll sicher gestellt werden, dass der Wert nie auf negative Zahlen gesetzt werden kann. Ein solcher Fehler soll im Zweifelsfall zum Programmabbruch führen. Welche Zeilen Programmcode kann man dieser Methode dafür hinzufügen?

5. In einer Textdatei steht eine Liste ganzer Zahlen, eine Zahl pro Zeile. Schreiben Sie eine C#-Methode, die als Parameter den Namen der Datei erwartet und als Rückgabewert die Summe dieser Zahlen liefert. Hinweis: `System.IO.File.ReadAllLines`, `int.Parse` und die `Split`-Methode der Zeichenketten.

6. Die Klassen `A` und `B` sind so gebaut:



Finden und korrigieren Sie die zwei Fehler in folgendem Ausschnitt eines C#-Programms, das diese beiden Klassen verwendet:

```
A x = new A();
x.e();
A y = new B();
double z = 42;
double w = y.d(z);
```

7. Gegeben ist folgende Struct:

```
struct Messung
{
    public double Wert;
    public DateTime Zeit;
}
```

Eine[c1] `List<Messung> messungen` enthalte solche Messungen – allerdings nicht unbedingt zeitlich geordnet. Schreiben Sie einige Zeilen C#, um aus der Liste den `Wert` der jüngsten Messung herauszufinden, also der Messung mit der größten Zeit. Gehen Sie davon aus, dass die Liste immer mindestens eine Messung enthält.

[c1] jl: Messung statt Messungen

8. Geben Sie die Namen dreier statischer[c2] Elemente (Methoden oder Properties) aus der .NET-Klassenbibliothek an. Zu welchen Klassen bzw. Structs gehören die jeweils?

[c2] *text added by jl*

Dieses Listing enthält 15 Fehler![c1]

[c1] jl: Zeile 87: double statt int

```
1  class MainWindow : Window
2  {
3                  // ... diverses Anderes
4
5      private void buttonClick(object sender, RoutedEventArgs e)
6      {
7          Anlagevermögen v;
8          v.FügeHinzu(new Gebäude());
9          v.FügeHinzu(new Fahrzeug());
10         v.FügeHinzu(new Fahrzeug());
11
12         Bilanzsummierer b = new Bilanzsummierer();
13         v.NehmeAn(b);
14         double s = b.GibResultat();
15
16         Abschreibung a = new Abschreibung();
17         v.NehmeAn(a);
18
19         v.NehmeAn(b);
20         s = b.GibResultat();
21     }
22 }
23
24 class Bilanzposten
25 {
26     double wert;
27     public abstract void NehmeAn(Buchhaltungsjob job);
28     public GibWert()
29     {
30         wert;
31     }
32     public void SetzeWert(double wert)
33     {
34         wert = this.wert;
35     }
36 }
37
38 class Gebäude : Bilanzposten
39 {
40     public Gebäude()
41     {
42         wert = 1000000.0;
43     }
44     public void NehmeAn(Buchhaltungsjob job)
45     {
46         job.Bearbeite(this);
47     }
48 }
49
```

```
50  class Fahrzeug : Bilanzposten
51  {
52      public Fahrzeug()
53      {
54          wert = 5000.0;
55      }
56      public override void NehmeAn(Buchhaltungsjob job)
57      {
58          job.Bearbeite(this);
59      }
60  }
61
62  class Anlagevermögen
63  {
64      List<Bilanzposten> anlagen;
65      public void FügeHinzu(Bilanzposten posten)
66      {
67          anlagen.Add(posten);
68      }
69      public void NehmeAn(Buchhaltungsjob job)
70      {
71          for (int i = 0; i < anlagen; i++)
72          {
73              anlagen.NehmeAn[i](job);
74          }
75      }
76  }
77
78  abstract class Buchhaltungsjob
79  {
80      public void Bearbeite(Gebäude g);
81      public void Bearbeite(Fahrzeug f);
82  }
83
84  class Bilanzsummierer
85  {
86      double summe;
87      protected double GibResultat()
88      {
89          return summe;
90      }
91      public override void Bearbeite(Gebäude g)
92      {
93          summe += g.GibWert();
94      }
95      public override void Bearbeite(Fahrzeug f)
96      {
97          summe += f.GibWert();
98      }
99  }
100
```

```
101 class Abschreibung : Buchhaltungsjob
102 {
103     public override void Bearbeite(Gebäude g)
104     {
105         g.SetzeWert(0.9 * g.GibWert());
106     }
107     public override void Bearbeite(Fahrzeug f)
108     {
109         f.SetzeWert(0.75 * g.GibWert());
110     }
111 }
```