

Informatik 2 für Regenerative Energien

Klausur vom 23. September 2019: Lösungen

Jörn Loviscach

Versionsstand: 23. September 2019, 19:02



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.

1. Die Fehler:

Zeile	korrekter Programmtext
5	<code>Würfel.SchalteTestmodusEin();</code>
13	<code>string x = a.ErzeugeAbfragetext();</code>
23	<code>class Vokabeltrainer</code>
28	<code>public void FügeHinzu(Vokabel v)</code>
43	<code>... a => !a.WarRichtig ...</code>
49	<code>int indexDerRichtigenVokabel ...</code>
57	<code>Vokabel[] falscheVorschläge ...</code>
67	<code>if (Würfel.GibZufallszahl(2) == 0)</code>
104	<code>1 + Würfel.GibZufallszahl(...</code>
107	<code>public string ErzeugeAbfragetext()</code>
122	<code>v = richtigeVokabel;</code>
128	<code>abfrage += v.GibText(spracheNach);</code>
161	<code>switch(s)</code>
171	<code>return text;</code>
190	<code>public static void SchalteTestmodusEin()</code>

2. `x` ist "Was heißt day? 1. Tag / 2. Sonne / 3. Nacht", weil die Zufallsfunktion im Testmodus immer 0 liefert, so dass immer der erste verbleibende Eintrag genommen wird. `y` ist `false`, weil die zweite Antwort eine falsche ist. `z` ist `100.0`, weil 100 % der bisherigen Antworten (also eine einzige) falsch waren.

3. Am Anfang der Methode `SpeichereUndBewerteAntwort` zum Beispiel dies ergänzen:

```
if(a < 1 || a > falscheVorschläge.Length + 1)
{
    throw new ApplicationException("unmögliche Nummer");
}
```

4. Zum Beispiel in der Klasse `Vokalbeltrainer` dies ergänzen:

```
public double BerechneFehlerquote(Vokabel v)
{
    // Dies oder eine entsprechende for-Schleife:
    List<Abfrage> abfragenDieserVokabel =
        abfragen.FindAll(a => a.RichtigeVokabel == v);

    int gesamtanzahl = abfragenDieserVokabel.Count;
    if (gesamtanzahl == 0)
    {
        return 0.0;
    }
    else
    {
        return 100.0 *
            abfragenDieserVokabel.Count(a => !a.WarRichtig)
            / gesamtanzahl;
    }
}
```

und in der Klasse `Abfrage` dies ergänzen:

```
public Vokabel RichtigeVokabel
{ get { return richtigeVokabel; } }
```

5. Zum Beispiel so:

```
class UnregelmäßigesVerb : Vokabel
{
    string pastSimple;
    string pastParticiple;

    public UnregelmäßigesVerb(string deutsch, string englisch,
        string pastSimple, string pastParticiple)
        : base(deutsch, englisch)
    {
        this.pastSimple = pastSimple;
        this.pastParticiple = pastParticiple;
    }

    public override string GibText(Sprache s)
    {
        string text = "";

        switch (s)
        {
            case Sprache.Deutsch:
```

```

        text = deutsch;
        break;
    case Sprache.Englisch:
        text = englisch + ", " + pastSimple
            + ", " + pastParticiple;
        break;
    }

    return text;
}
}

```

Die Methode `GibText` der Klasse `Vokabel` muss dazu `virtual` gemacht werden, ihre Felder `deutsch` und `englisch` müssen `protected` werden oder passende `Getter-Propertys` erhalten.

6. Zum Beispiel so:

```

public List<Vokabel> FindeSchwierigeVokabeln()
{
    List<Vokabel> schwierigeVokabeln = new List<Vokabel>();
    foreach (Vokabel v in vokabeln)
    {
        // So oder mit einer weiteren foreach-Schleife:
        if(!abfragen.FindAll(a => a.RichtigeVokabel == v)
            .Last().WarRichtig)
        {
            schwierigeVokabeln.Add(v);
        }
    }
    return schwierigeVokabeln;
}

```

Die Klasse `Abfrage` benötigt dazu eine öffentliche Property `RichtigeVokabel`.

7. Die Klassen:

```

abstract class Raum
{
    bool istLichtAn;
    public void SchalteLicht(bool an)
    {
        istLichtAn = an;
    }
}

class Wohnzimmer : Raum
{

```

```
double solltemperatur;
public void MachWarmUndHell()
{
    solltemperatur = 21.0;
    SchalteLicht(true);
}
}

class Garage : Raum
{
    bool istTorGeöffnet;
    public void ÖffneTor()
    {
        istTorGeöffnet = true;
    }
    public void SchließeTor()
    {
        istTorGeöffnet = false;
    }
}
}
```

8. Die Werte sind 10, 2, 20.