

Informatik 2 für Regenerative Energien

Klausur vom 8. Juli 2019: Lösungen

Jörn Loviscach

Versionsstand: 15. Juli 2019, 21:16



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
12	Acker a = new Acker();
15	... Bodenprobe(Bodenart.Lehm, 8.0);
26	class GeoPunkt
39	return phi * Math.PI / 180.0;
42	public double BerechneAbstandZu...
45	double b2 = BogenmaßAusGrad(p.geoBreite);
61	public Beobachtung...
86	abstract class Pflanze
88	...new List<Beobachtung>()
102	public abstract bool IstVerträglichMit
105	class Zwiebel : Pflanze
132	return b.Bodenart == Bodenart.Sand;
146	double abstand = pflanzen[i].Wo...
159	...pf.Wo.BerechneAbstandZu...
190	...pf => !pf.IstVerträglichMit...

2. `x` ist `true`, weil der Abstand des ersten vom dritten Punkt offensichtlich viel größer ist als der vom zweiten zum dritten. `y` ist `true`, weil die nächstliegende Pflanze die Steckzwiebel ist und diese mit diesem Boden verträglich ist. `z` ist `false`, weil die Möhre nicht mit dem Boden verträglich ist.

3. Zum Beispiel am Anfang der Methode `FügeBeobachtungHinzu` dies ergänzen:

```
if(beobachtungen.Count > 0 && beobachtungen[0].Art != b.Art)
{
    throw new ApplicationException("falsche Art");
}
```

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

```
public double BerechneWachstumsrate()
{
    if (beobachtungen.Count > 1)
    {
        Beobachtung b1 = beobachtungen[0];
        Beobachtung b2 = beobachtungen[beobachtungen.Count - 1];
        return (b2.Höhe - b1.Höhe) / (b2.Wann - b1.Wann).TotalDays;
    }
    else
    {
        return double.NaN;
    }
}
```

Die Klasse `Beobachtung` ist dann um öffentliche `Getter-Propertys` `Höhe` und `Wann` zu ergänzen.

5. Zum Beispiel so:

```
class GeoPunktMitHöhe : GeoPunkt
{
    double geoHöhe;

    public GeoPunktMitHöhe(double geoBreite, double geoLänge,
                           double geoHöhe)
        : base(geoBreite, geoLänge)
    {
        this.geoHöhe = geoHöhe;
    }
}
```

Ja, man kann `p1.BerechneAbstandZu(p2)` aufrufen, aber das Ergebnis berücksichtigt nicht die Höhen der Punkte, sondern ist der Abstand auf Nullniveau.

6. Zum Beispiel dies in der Klasse `Pflanze` ergänzen:

```
public Pflanze FindeNördlichstePflanze()
{
    double bisherMaximaleGeoBreite = double.NegativeInfinity;
    Pflanze pf = null;
    for (int i = 0; i < pflanzen.Count; i++)
    {
        double br = pflanzen[i].Wo.GeoBreite;
        if (br > bisherMaximaleGeoBreite)
        {
            bisherMaximaleGeoBreite = br;
        }
    }
}
```

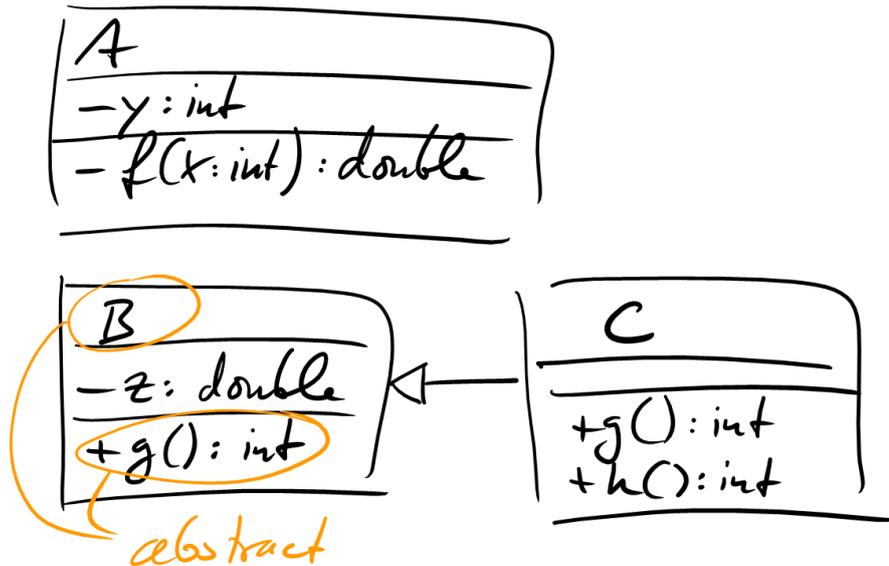
```

        pf = pflanzen[i];
    }
}
return pf;
}

```

Die Klasse GeoPunkt benötigt dann noch eine öffentliche Getter-Property GeoBreite.

7. Das Diagramm:



8. Die Werte sind 10, 7, 11.