

Informatik 2 für Regenerative Energien

Klausur vom 7. Juli 2023: Lösungen

Jörn Loviscach

Versionsstand: 7. Juli 2023, 08:42



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
7	Wechselakku w1 = new Wechselakku(100.0);
11	verwaltung.FügeERollerHinzu(eRoller);
45	... akku.Ladezustand ...
50	public void Update ...
66	: base(kennzeichen, letzteWartung, position)
76	public void TauscheAkkuGegen ...
87	class Akku
89	public double Ladezustand ...
97	class Wechselakku : Akku
141	return Math.Sqrt(...
155	... Finde(string name)
163	... new List<ERoller>();
183	z.B. double entfernung = double.PositiveInfinity;
190	if (neueEntfernung < entfernung)
197	return nächsterFreierERoller;

2. `x` ist `true`, weil beide Bedingungen der Methode `IstFahrbereit` erfüllt sind. `y` ist `98.0`, weil jener Akku mit den `Update` auf diesen Wert gesetzt worden ist. `z` ist `null`, weil die aktuelle Position keinen Ortsangabe enthält.

3. *Zum Beispiel:* Am Anfang der besagten Methode dies ergänzen:

```
if (this.roller != null)
{
    throw new ApplicationException(
        "Steckt noch in einem anderen Roller!");
}
```

4. *Zum Beispiel:* Wenn die Position unbekannt ist, könnte man `Breitengrad` und `Längengrad` auf `double.NaN` setzen und vor jeder Verwendung dieser Felder prüfen, ob sie diesen Sonderwert enthalten.

Oder: Man könnte den Typ von Breitengrad und Längengrad auf `double`? ändern und vor jeder Verwendung dieser Felder prüfen, ob sie einen Wert enthalten.

Oder: Wenn die Position unbekannt ist, könnte man das Feld `Position` auf `null` setzen und vor jeder Verwendung der Position prüfen, ob es eine echte Referenz enthält.

Oder: Man könnte die Klasse `Geokoordinaten` um eine öffentlich lesbare `bool-Property` `IstGültig` erweitern und vor jeder Verwendung einer Instanz von `Geokoordinaten` prüfen, ob diese `Property` `true` ist.

5. Auf der obersten Ebene dies ergänzen:

```
enum Akkuhersteller { A, B, C }
```

In der Klasse `Wechselakku` dies ergänzen:

```
public Akkuhersteller Hersteller { get; private set; }
```

Den Konstruktor ersetzen durch:

```
public Wechselakku(double ladezustand,
                    Akkuhersteller hersteller)
    : base(ladezustand)
{
    Hersteller = hersteller;
}
```

Und diesen Konstruktor dann nach diesem Muster aufrufen:
`new Wechselakku(100.0, Akkuhersteller.A).`

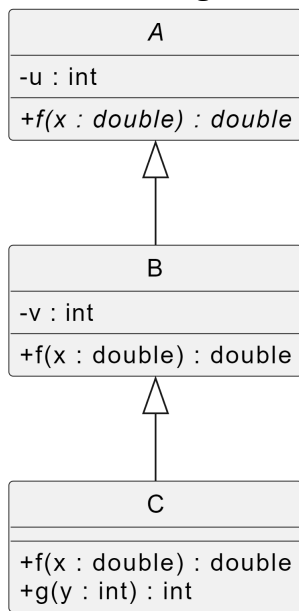
6. *Zum Beispiel:* Zum Beispiel in der Klasse `ERoller` dem Feld `akku` einen öffentlichen Getter geben:

```
public Akku? Akku { get { return akku; } }
```

Und dann in der Klasse `ERollerVerwaltung` diese Methode hinzufügen (alternativ mit einer entsprechenden `for-Schleife`):

```
public List<ERoller>
    FindeZuLadende(Geokoordinaten p)
{
    return rollerflotte.FindAll(roller =>
        roller.Position.EntfernungZu(p) < 0.1
        && (roller.Akku == null
            || roller.Akku.Ladezustand < 50.0));
}
```

7. Das UML-Diagramm:



8. Die Werte sind 0, 0, 20.