

Code Review

Roger:

Vorschlag 1:

Ich würde ihm empfehlen, konsistenter Kommentare hinzuzufügen, besonders JavaDoc Kommentare, die beschreiben, was für Argumente entgegengenommen werden oder zurückgegeben werden.

Vorschlag 2:

Ich würde ihm empfehlen, dass er bei String Vergleichen die null-safe .equals() Methode verwendet anstatt '=='.

Vorschlag 3:

Ich würde ihm empfehlen, dass er bei Instanzvariablen, für die es unterstützen, die Variabel auf final zu setzen.

Vorschlag 4:

Ich würde ihm empfehlen, dass er die korrekten Zugriffskontrollen für die Instanzvariablen implementiert. Bspw. nur public zu verwenden, wenn dies wirklich auch so genutzt werden muss.

Levyn:

Fehlerbehandlung beim Starten des Controllers

Füge eine Fehlerbehandlung hinzu, falls die Initialisierung des HeatCalculatorController fehlschlägt oder bei der Ausführung von controller.run() eine Ausnahme auftritt.

Begründung: Ohne Fehlerbehandlung könnte die Anwendung bei einem unerwarteten Fehler abrupt abstürzen, was die Benutzerfreundlichkeit beeinträchtigt.

```
1 package ch.i.et_gibb;
2
3 import ch.i.et_gibb.controller.HeatCalculatorController;
4
5 /**
6  * @author Levyn Schneider
7  * @version 3.0.0
8  */
9 public class HeatCalculator {
10     /**
11      * Main method
12      *
13      * @param args command line arguments
14      */
15     public static void main(String[] args) {
16         HeatCalculatorController controller = new HeatCalculatorController();
17         controller.run();
18     }
19 }
```

Klassendokumentation spezifizieren

Ergänze in der Klassendokumentation eine genauere Beschreibung der Klasse CubicTank, einschliesslich ihrer Nutzung und Einschränkungen. Begründung: Eine detaillierte Beschreibung macht die Funktion und den Anwendungsbereich der Klasse für andere Entwickler klarer.

```
public CubicTank(float maxTemperature, String name, float dailyEnergy, float
    super(maxTemperature, name, dailyEnergy);
    setLength(length);
    setWidth(width);
    setHeight(height);
    super.setVolume(calculateSquareTankVolumeInLiters());
}
```

JavaDoc für den Konstruktor erweitern

Ergänze die JavaDoc-Beschreibung des Konstruktors mit Informationen zur erwarteten Einheit für Parameter (z. B. Meter oder Celsius) und was bei ungültigen Werten geschieht.

Überprüfung der Null-Werte bei Strings

Ersetze die Überprüfung `if (name == "")` in `setName` durch `if (name == null || name.isEmpty())`. Die aktuelle Überprüfung stellt nur sicher, dass der Name kein leerer String ist, prüft jedoch nicht auf null.

```
public void setName(String name) { 5 usages  Roger Widmer
    if (name == "") {
        throw new IllegalArgumentException("Name kann nicht null sein");
    }
    this.name = name;
}
```

Berechnung der gespeicherten Energie als double

Gib die Methode `calculateStoredEnergy` als `double` statt `int` zurück, und lasse die gesamte Berechnung in double-Präzision erfolgen.

```
public float calculateStoredEnergy() { return (int) (volume * 4.18 * (maxTemperature - 23) / (double) 3600); }
```