



MCI DiBSE-B-4-SE2

Software Engineering II

Vaadin Flow Framework

RKI Projekt Essen auf Rädern

Code Standards

Juni-Juli 2021

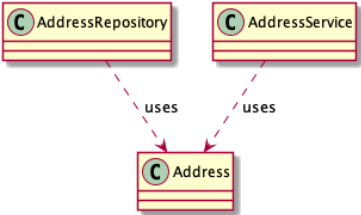
(gültig für alle Klassen außer TestDataLoader.java)

Squad 4:

- Andreas Oberhofer Quality Engineer / Testing
- Arthur Waldner Backend Services Engineer
- Michael Mösl Build Engineer
- Sandra Weigl Frontend GUI Engineer
- Thomas Widmann Technology Engineer

Coding Guidelines:

Naming Convention

| Element | Guideline-Standard | Example |
|--------------------------|--|--|
| Component Classes | Must end with postfix „Component” | OrderDeliveryComponent |
| Form Classes | Must end with postfix „Form” | AdditionalInformationForm |
| Interfaces | Must start with prefix “I” | IRoutingService |
| Interface Implementation | Must contain the name of the interface and info about implementation details | RoutingServiceGraphHopper |
| UI/View Classes | Must end with postfix „View” | KitchenView |
| Service Classes | Must end with postfix „Service” | EmployeeService |
| Repository Classes | Must end with postfix „Repository” | AddressRepository |
| Entity Classes | Must contain the name of the Entity and for every Entity must exist the corresponding Repository Class | Address  |

Naming Conventions (from Uncle Bob's Clean Code Philosophy)

1. Use Intention-Revealing Names
2. Avoid Disinformation and Encodings
3. Make Meaningful Distinctions
4. Use Pronounceable Names
5. Use Searchable Names
6. Don't Be Cute/Don't Use Offensive Words
7. Pick One Word per Concept
8. Don't Pun
9. Solution Domain Names vs Problem Domain Names
10. Add Meaningful Context as a Last Resort

Comments rules

- **Always try to explain yourself in code.**
- Don't be redundant.
- Don't add obvious noise.
- Don't use closing brace comments.
- Don't comment out code. Just remove.
- Use as explanation of intent.
- Use as clarification of code.
- Use as warning of consequences.

Structure of Classes

| | |
|--|---|
| <ol style="list-style-type: none">1. Member variables2. Constructor's3. Public functions/methods4. Protected functions/methods5. Private functions/methods | <pre>6 public class AdjacencyMatrix { 7 8 private final int dimension; 9 private final double[][] matrixData; 10 11 public AdjacencyMatrix(int dimension) { 12 this.dimension = dimension; 13 matrixData = new double[dimension][dimension]; 14 } 15 16 public double[][] getMatrix() { return matrixData; } 17 18 public void setElement(int x, int y, double dVal) { 19 if(x > dimension - 1 x < 0) { 20 throw new IllegalArgumentException("Index X out of range!"); 21 } 22 23 if(y > dimension - 1 y < 0) { 24 throw new IllegalArgumentException("Index Y out of range!"); 25 } 26 27 matrixData[x][y] = dVal; 28 } 29 } 30</pre> |
|--|---|

Tests

- Readable
- Fast
- Independent
- Repeatable

Additional sources

- <https://vaadin.com/docs/latest/guide/contributing-docs/style-guidelines>
- <https://dzone.com/articles/naming-conventions-from-uncle-bobs-clean-code-phil>
- <https://gist.github.com/wojtekl/73c6914cc446146b8b533c0988cf8d29>
- <https://www.baeldung.com/cs/clean-code-comments>
- <https://www.oracle.com/technical-resources/articles/java/javadoc-tool.html>
<https://www.oracle.com/java/technologies/javase/codeconventions-comments.html>