Aleksandr Stepanov

27th July 2021

cLASS dIAGRAMS

Assignment 1, 2021

Introduction to Class Diagram

A Class Diagram in Software engineering is a static structure that gives an overview of a software system by displaying classes, attributes, operations, and their relationships between each other. Class Diagram helps construct the code for the software application development. (Class Diagram Tutorial, 2021)

## What is a class diagrams used for?

Class diagrams give you the ability to create models with the help of UML using attributes, relationships, operations and intersections. A class diagram visualizes the paths between classes in the form of aggregations and associations as well as through the passing on of properties and behavior between classes. These take the form of generalizations.

Class diagrams are the most important kind of UML diagram and are vitally important in software development. Class diagrams are the best way to illustrate a system’s structure in a detailed way, showing its attributes, operations as well as its inter-relationships. Classes play a significant role in object orientated programming languages – they are indispensable when it comes to software modelling. (TheCLassDiagramm, 2021)

## Relationships

In a class diagram, it is necessary that there exists a relationship between the classes for representing a connection between structural, behavioral, or grouping things. It is also called a link that describes how two or more things can relate to each other during the execution of a system. Type of UML Relationship are Association, Dependency, Generalization, and Realization.

## Benefits

* Any simple or complex data model could be illustrated using the class diagram to gain maximum information.
* The schematics of an application could be understood with the help of it.
* Any system need could be visualized and passed across the business for specific action to be taken.
* Any requirement to implement a specific code could be highlighted through charts and programmed to the described structure.
* A description that is implementation-independent could be provided and passed on to the components. (Class Diagrams, 2021)

## Disadvantages of Class Diagram

Like many other data structuring methods, the class diagrams has drawbacks

* The class diagrams might often take a longer time to manage, and maintain. It requires time for the synchronization with the software code to set it up and maintain.
* A lack of clarity in understanding the beneficiary of the diagram is also a disadvantage. Hence, there is often an argument to not waste time on the class diagrams and focus rather on using a whiteboard or paper to draw the diagram. (Class Diagrams, 2021)
* An overcomplicated or overwhelming diagram doesn’t help software developers in their work. Mapping out every single scenario could make the diagram messy and hard to work with.
* Putting overemphasis on the design could cause a hindrance to the developers and companies. People need to get down on the actual work rather than spending time looking into the diagram and solving issues. (Class Diagrams, 2021)

## Conclusion

UML class diagrams are useful when modeling business data. By accurately modeling attributes and associations of class entities, we can easily map these class diagram specifications to entity beans with CMP. Class attributes map to abstract access methods for persistent fields, and association roles map to abstract access methods for relationship fields. Navigability determines whether relationship access methods appear in both related entity beans or just one. Furthermore, multiplicity notation determines the correct type for relationship fields, life cycle issues, and cascading delete characteristics. (MappingUML, 2021)

Class diagram from Iteration 2Diagram

Description automatically generated

Class diagram from Iteration 3

A picture containing diagram

Description automatically generated

Class diagram from Iteration 4/5

Graphical user interface, application

Description automatically generated

|  |  |  |
| --- | --- | --- |
| **Rule** | **checklist** | **Example** |
| added all needed classes | yes | See diagram |
| all class names written in the singular | yes | See diagram |
| avoided using such naming elements as 'data', 'record', or 'info' | yes | See diagram |
| named each class so its meaning can be precisely understood | yes | See diagram |
| included only classes for which data will actually need to be stored or manipulated | yes | See diagram |
| avoided plural attributes | yes | See diagram |
| made model as general as you can | yes | See diagram |
| attributes in one class less than 10 | yes | See diagram |
| any ways to simplify your model | no | See diagram |
| Do the attributes represent simple data that each instance must have (String, Integer, Float, Date, Time, Boolean etc.)? | yes | See diagram |

Reflection

Class diagrams helped me to illustrate the Vending Machine structure in a detailed way, showing its attributes, operations. thanks to this method, I was able to see the full picture, the relationship between the classes, the missing methods.

# References

*Class Diagram Tutorial*. (2021, 07 27). Retrieved from guru99: https://www.guru99.com/uml-class-diagram.html

*Class Diagrams*. (2021, 07 27). Retrieved from educba: https://www.educba.com/class-diagram/

guru99. (2021). *UML Class Diagram Tutorial.* Christchurch: https://www.guru99.com/uml-class-diagram.html.

*MappingUML*. (2021, 07 27). Retrieved from informit: https://www.informit.com/articles/article.aspx?p=26272&seqNum=6