

UML Dokumentation

Damien Flury

27. Februar 2019

Inhaltsverzeichnis

1	Einführung	3
2	Klassendiagramme	3
2.1	Vererbung	3
2.2	Assoziationen	3
3	Use Case Diagram	3
4	Aktivitätsdiagramm	8

```
public class Person {
    private final String firstName;
    private final String lastName;
    public Person(String firstName, String lastName) {
        this.firstName = firstName;
        this.lastName = lastName;
    }
}

public final class Employee extends Person {
    private final String teamName;
    public Employee(String firstName, String lastName, String teamName) {
        super(firstName, lastName);
        this.teamName = teamName;
    }
}
```

Abbildung 1: Vererbung in Java

1 Einführung

2 Klassendiagramme

2.1 Vererbung

Bei der Vererbung (engl. Inheritance) von Klassen werden Unterklassen generalisiert. Java verfügt im Gegensatz zu zum Beispiel C++ nur über Single Inheritance, das heisst, dass eine Klasse jeweils nur von maximal einer anderen Basisklasse erben kann. Dies führt zu einigen Limitierungen, vereinfacht den Code jedoch. Somit muss in Java gegebenenfalls auf andere Beziehungen ausgewichen werden, wie zum Beispiel die Komposition. Codebeispiel: 1, grafische Beispiele: Abbildung 2 und Abbildung 3.

2.2 Assoziationen

Assoziationen bezeichnen sowohl Besitz- als auch Kennbeziehungen. Dies funktioniert im Beispiel von Java häufig anhand des Konstruktors oder mithilfe von Setter-Methoden. Codebeispiel: Abbildung 4, UML-Diagramme sind unter Abbildung 5 und Abbildung 6 zu finden.

3 Use Case Diagram

Das Use Case Diagram bietet die Möglichkeit, Benutzerinteraktion mit der Applikation zu repräsentieren. Zum ersten Mal vorgestellt wurde es an einer Konferenz in 1987 von

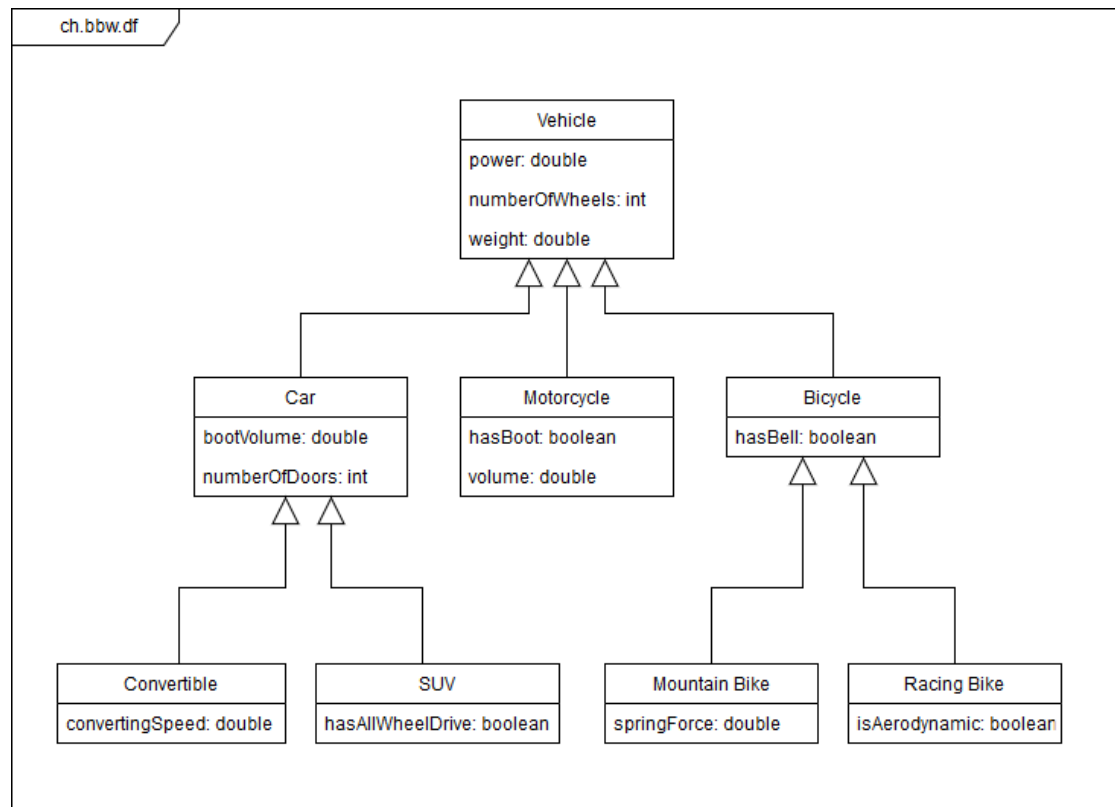


Abbildung 2: UML-Diagramm mit Vererbung (draw.io)

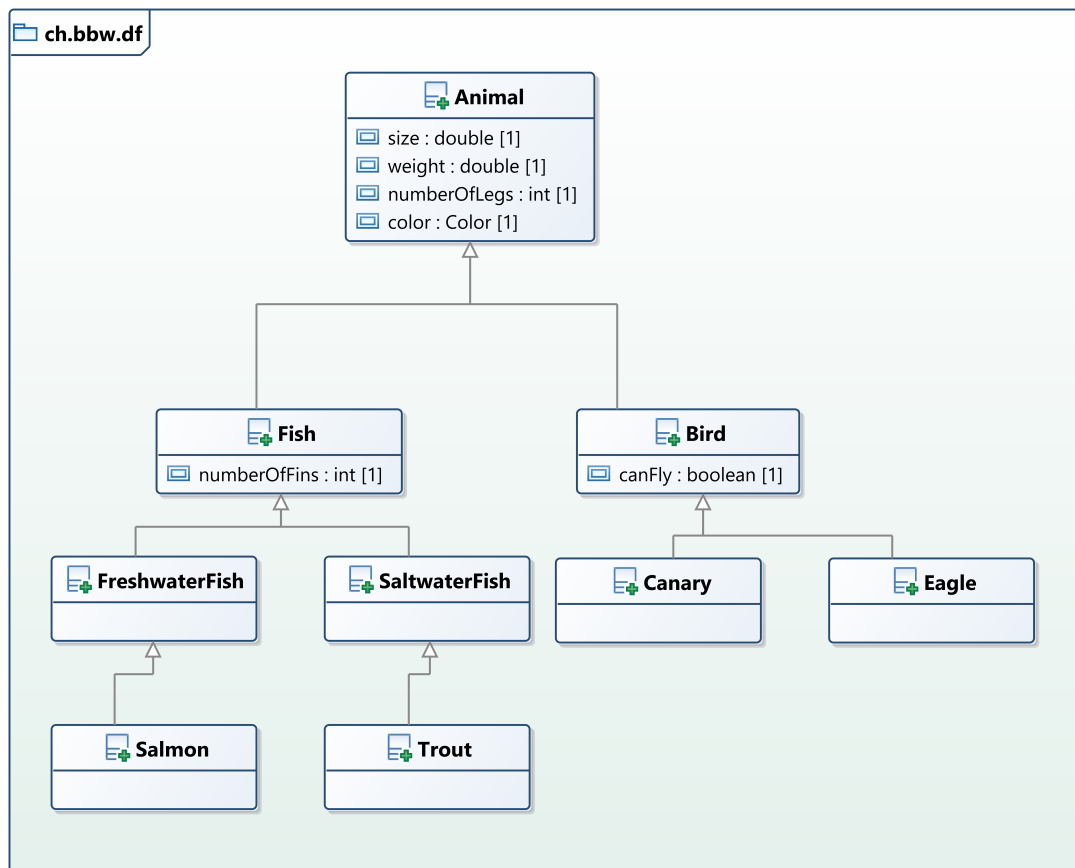


Abbildung 3: UML-Diagramm mit Vererbung (UML Designer)

```

public final class Application {
    public static void main(String[] args) {
        var bike = new Bike();

        // Hier wird dem Driver ein Bike uebergeben:
        var driver = new Driver(bike);
    }
}

public final class Driver {
    private final Bike bike;
    public Driver(Bike bike) {
        this.bike = bike
    }
}

```

Abbildung 4: Aggregation in Java

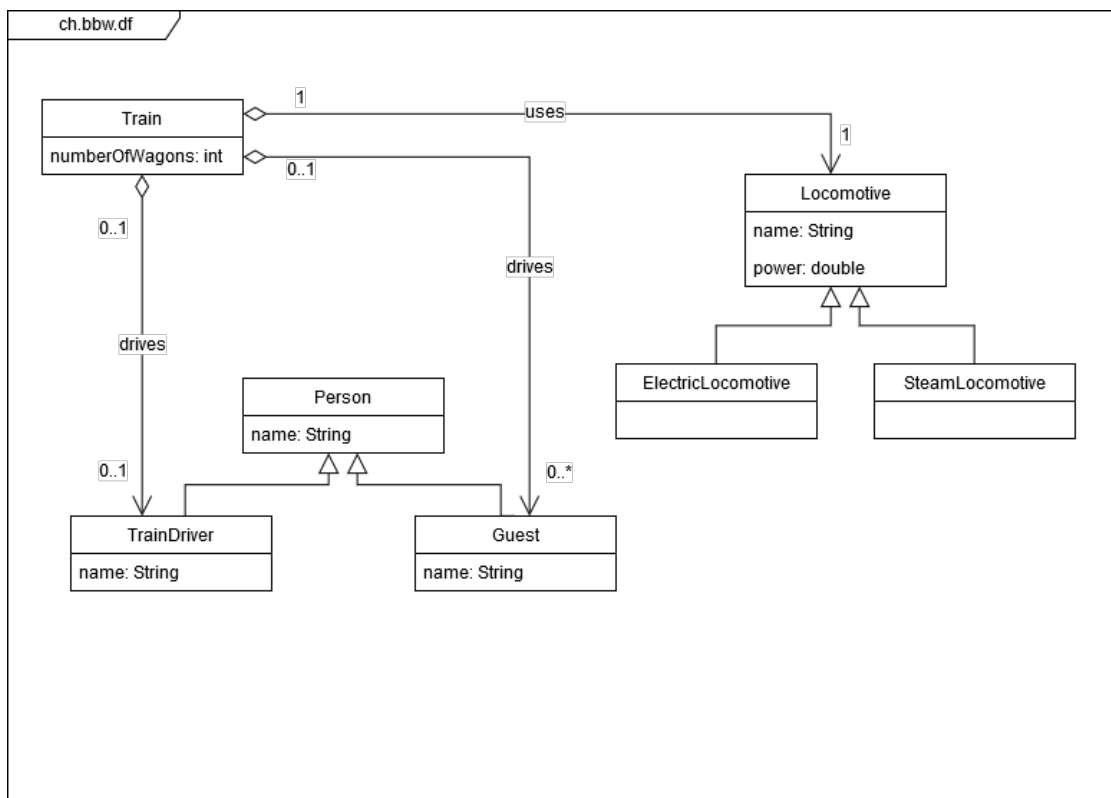


Abbildung 5: Klassendiagramm mit Assoziationen (draw.io)

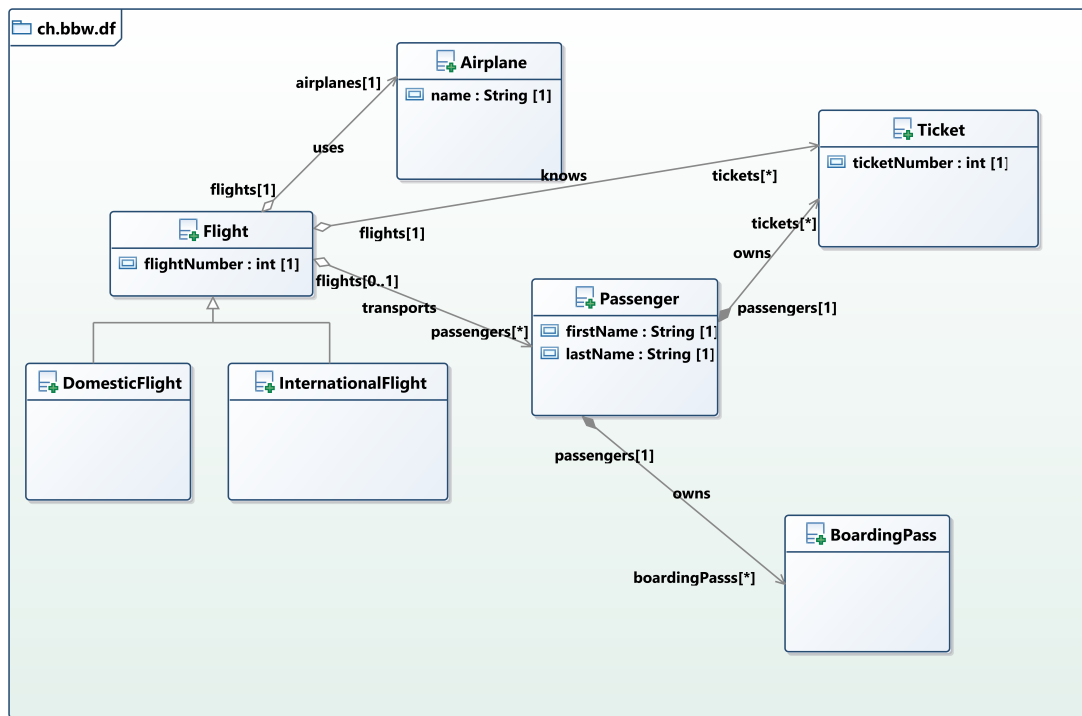


Abbildung 6: Klassendiagramm mit Assoziationen (UML Designer)

Jacobson und findet heutzutage häufig einen Platz in unterschiedlichen Softwareteams mit diversen Programmiersprachen. Beispiele: Abbildung 7 und Abbildung 8

4 Aktivitätsdiagramm

Mithilfe des Aktivitätsdiagramms kann man Abläufe, Aktionen und Kontrollflüsse beschreiben.

Häufig werden durch Aktivitätsdiagramme einzelne Use Cases genauer beschrieben. Das Aktivitätsdiagramm ermöglicht hier die Darstellung von ein wenig komplexeren Abläufen mit Falls-Verzweigungen, Wiederholungen und anderen Ausnahmen. Beispiele: Abbildung 9 und Abbildung 10

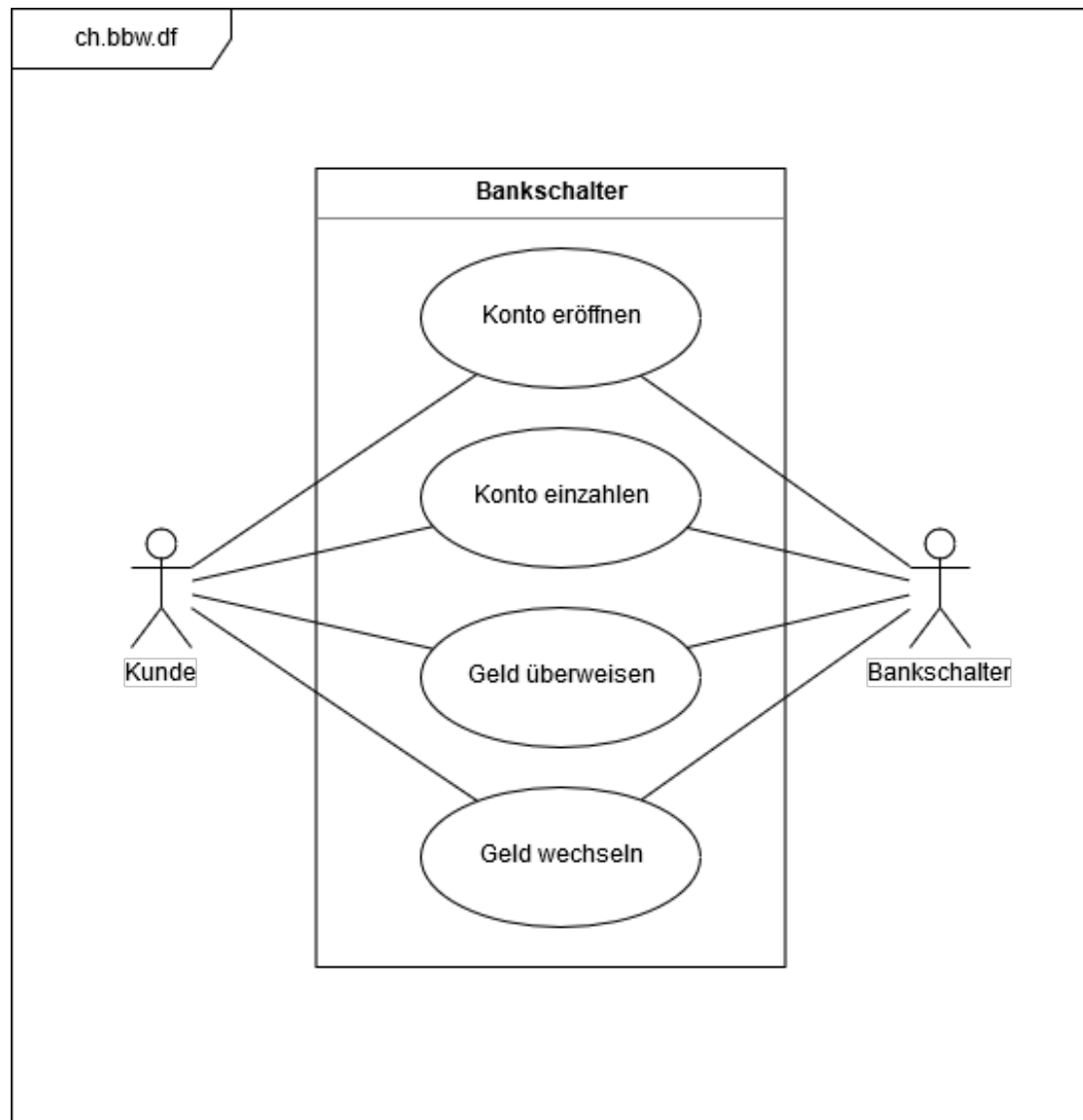


Abbildung 7: Use Case Diagram (draw.io)

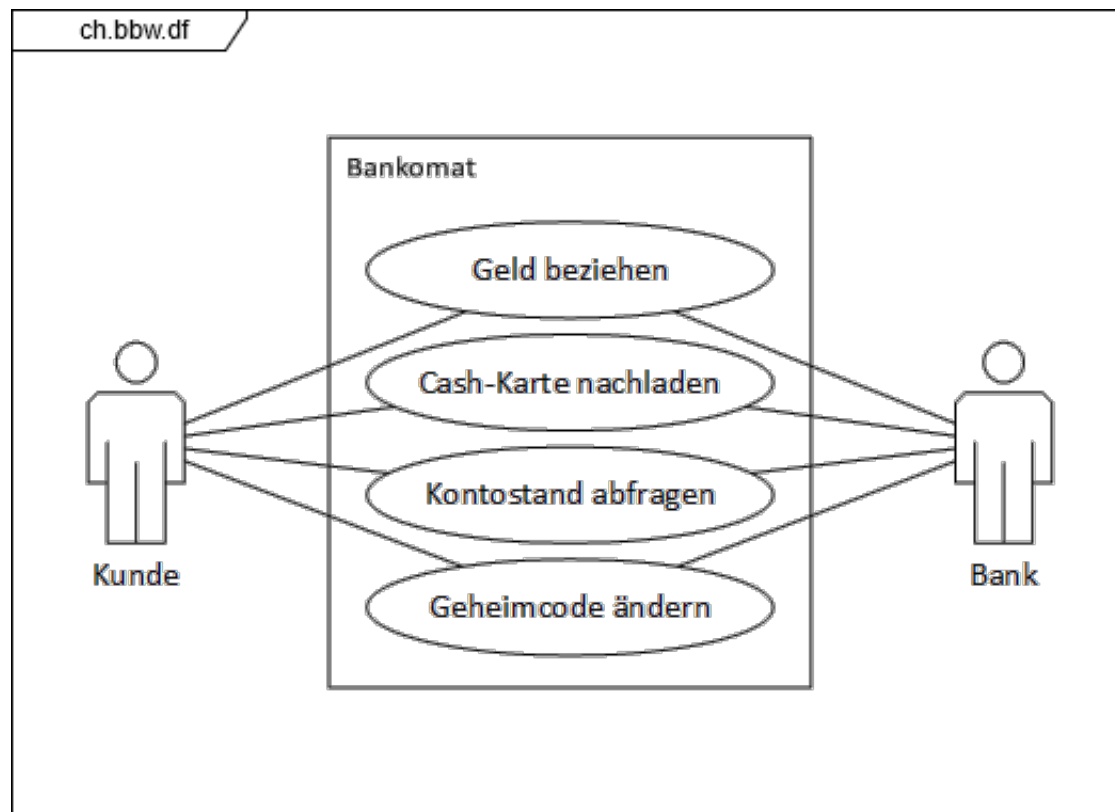


Abbildung 8: Use Case Diagram (Microsoft Visio)

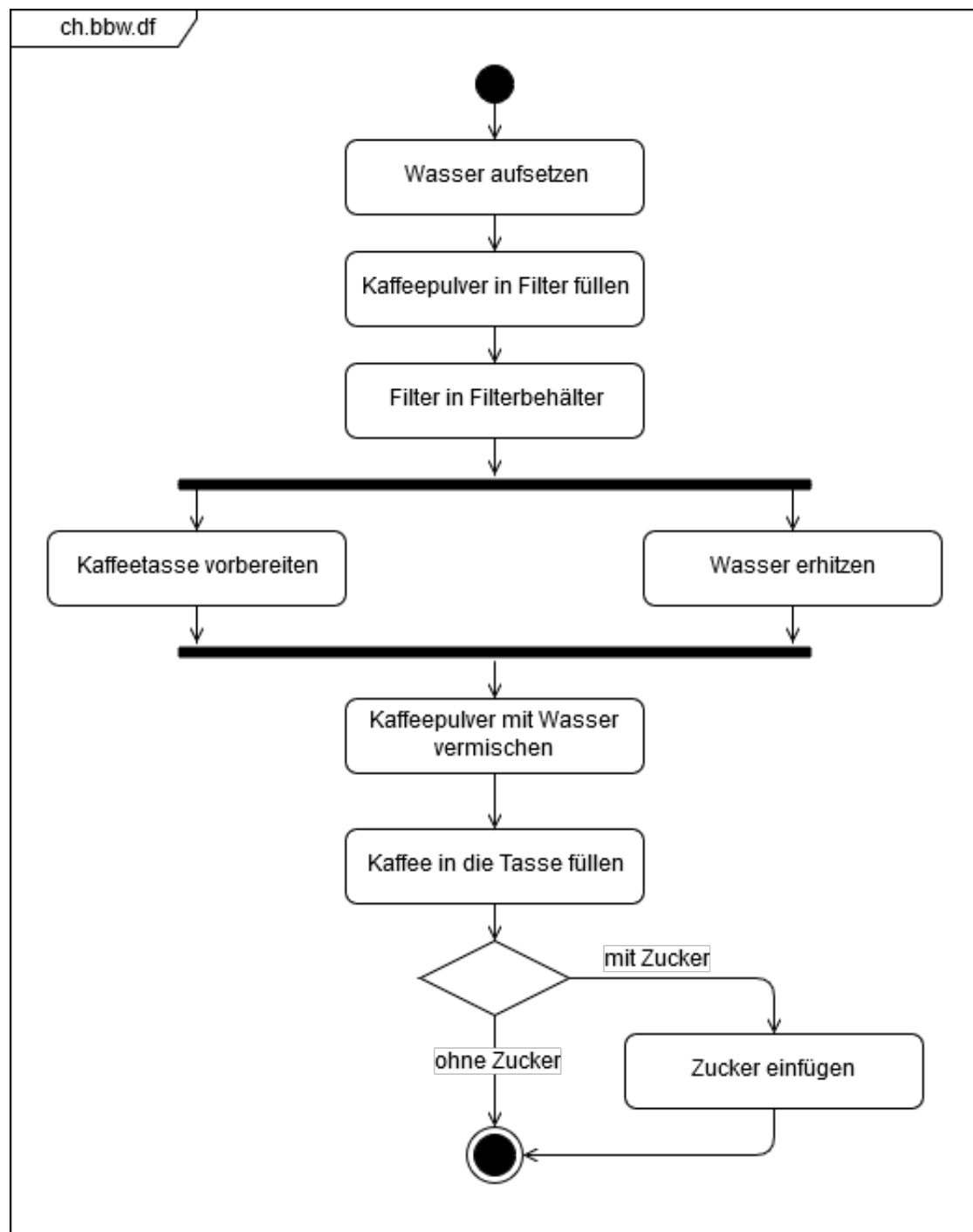


Abbildung 9: Aktivitätsdiagramm (draw.io)

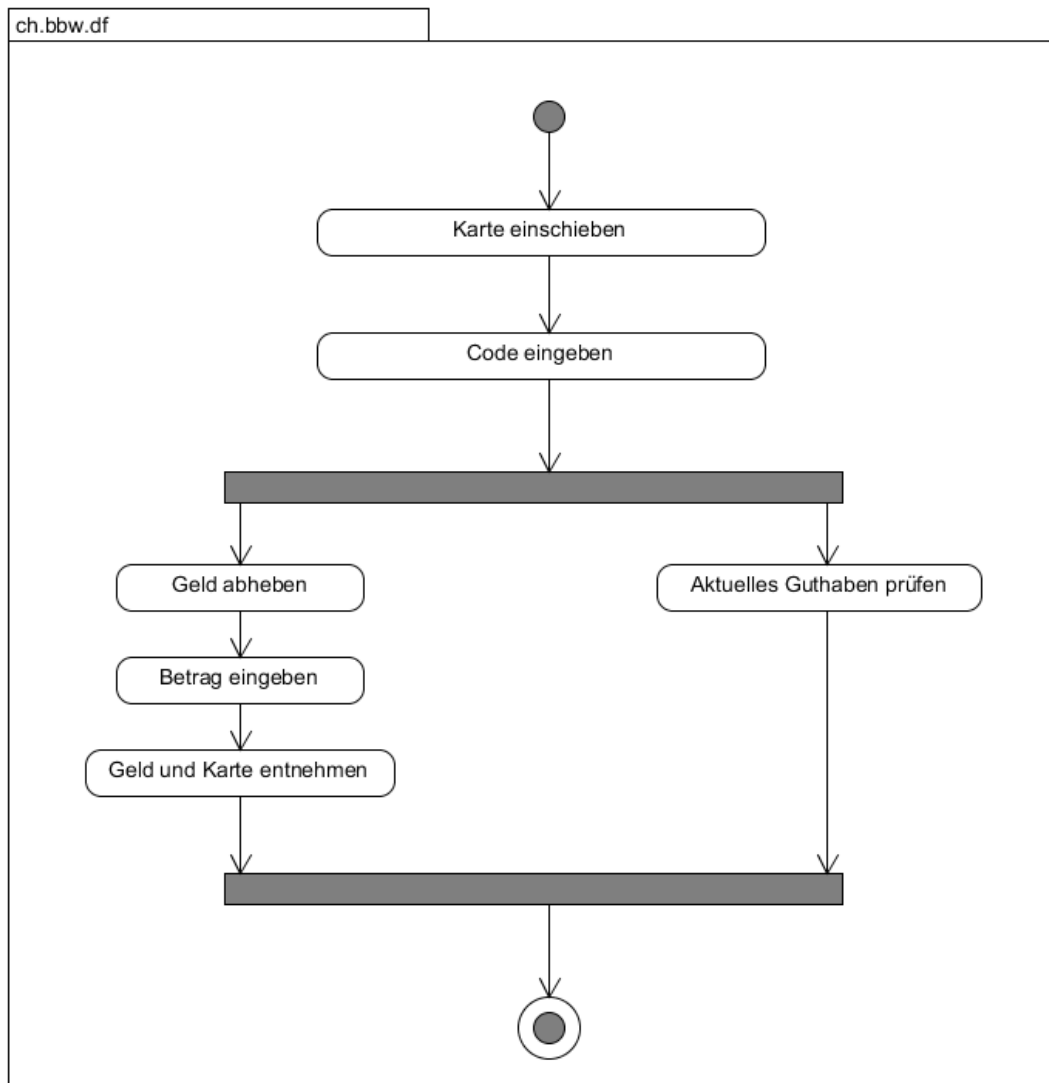


Abbildung 10: Aktivitätsdiagramm (UMLet)