



QUIZ PROGRAMM

Modul 226a

Muriel Rüegg, Arda Öztug

Inhaltsverzeichnis

1	Projektbeschreibung.....	2
1.1	Github	2
2	Use-Case Diagramm.....	2
3	Klassendiagramm.....	3
3.1	Vor der Programmierung.....	3
3.2	Nach der Programmierung.....	4
4	JDK.....	5
5	IDE	5
6	Gradle.....	5
7	API	5
8	OO-Concepts im Code.....	6
8.1	Encapsulation.....	6
8.2	Delegation.....	7
8.3	Static	8
8.4	Final.....	9
8.5	Exceptions	10
9	Anwendung.....	11
10	Reflexion	12

1 Projektbeschreibung

In unserem Projekt erstellen wir Quiz über verschiedene Themen. Dazu wird ein Ranking System implementiert, in dem man seinen Namen zuerst eingeben kann und danach ein scoreboard erstellt wird. Über ein simples und verständliches UI kann man seine Antwort entweder per Multiple Choice auswählen oder selbst eingeben, es variiert auf das Quiz.

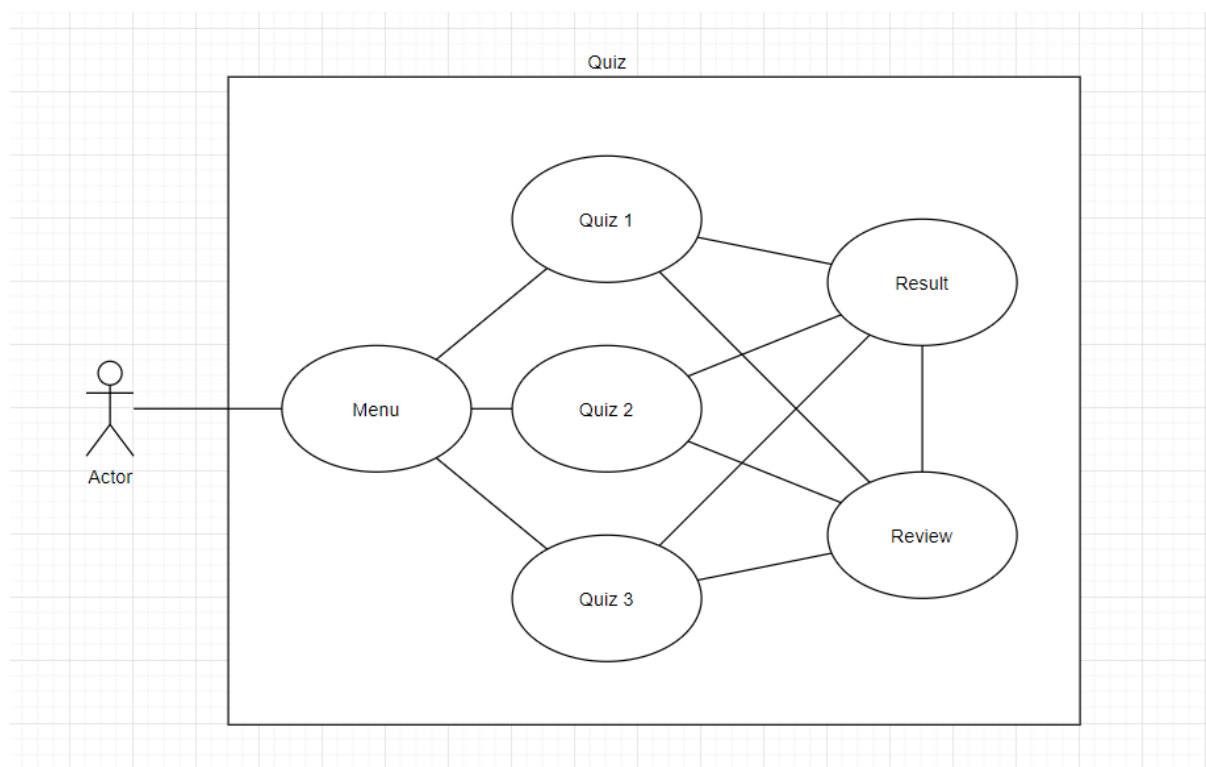
1.1 Github

Unser Projekt wurde auf Github veröffentlicht und ist unter dem folgenden Link zu finden:

<https://github.com/Freyja21/226a>

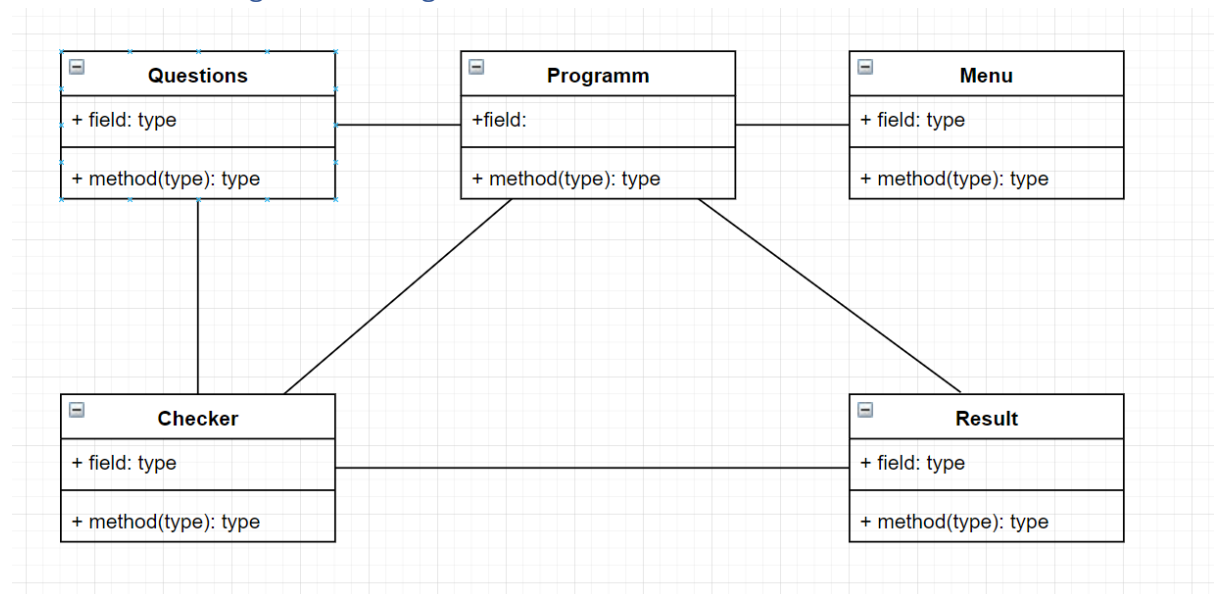
2 Use-Case Diagramm

Am **Anfang** des Projekts haben wir dieses Use-Case Diagramm erstellt.



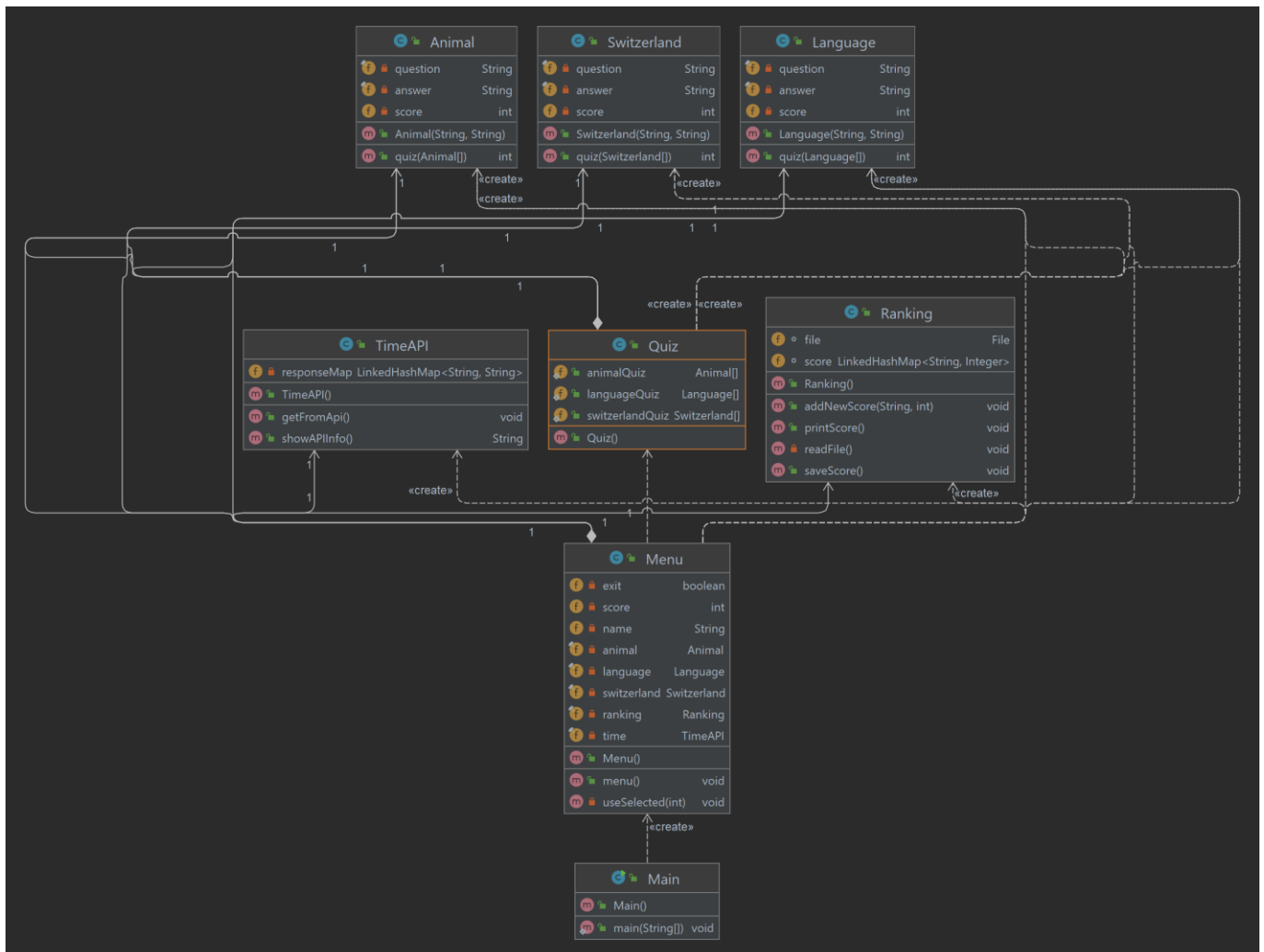
3 Klassendiagramm

3.1 Vor der Programmierung



3.2 Nach der Programmierung

erstellt wurde dieses Bild mit IntelliJ Ultimate Edition



4 JDK

Für unser Projekt haben wir Java JDK 16.0.2 verwendet.

5 IDE

Als IDE haben wir den IntelliJ IDEA Ultimate verwendet.

Durch ein Angebot von JetBrains, kann man sich als Schüler registrieren und die Lizenz für die Ultimate gratis erhalten.

<https://www.jetbrains.com/shop/eform/students>

<https://www.jetbrains.com/de-de/idea/download/#section=windows>

6 Gradle

Für unser Projekt haben wir das Dependency Manager Gradle verwendet.

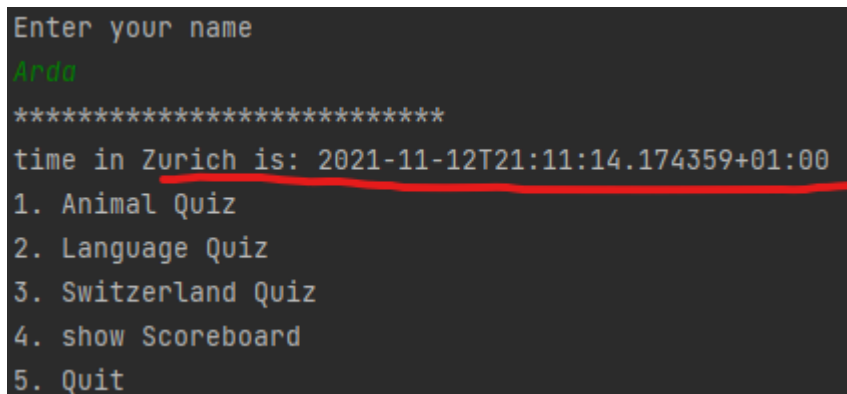
7 API

Wenn man seinen Namen eingibt am Anfang des Quizes, wird noch die Uhrzeit und das Datum angezeigt.

Anhand dieser API konnten wir die Uhrzeit und das Datum anzeigen:

<http://worldtimeapi.org/>

Hier noch ein kleiner Screenshot:



```
Enter your name
Arda
*****
time in Zurich is: 2021-11-12T21:11:14.174359+01:00
1. Animal Quiz
2. Language Quiz
3. Switzerland Quiz
4. show Scoreboard
5. Quit
```

8 OO-Concepts im Code

8.1 Encapsulation

Anhand dieses Screenshots ist erkennbar, dass die Variable `question` privat in der Klasse `public Switzerland` verkapselt wird. Dazu sind noch viele weitere Beispiele für Encapsulation enthalten.

```
1  package ch.tbz;
2
3  import java.util.Scanner;
4
5  public class Switzerland {
6      final private String question;
7      private final String answer;
8      private int score = 0;
9
10     public Switzerland(String question, String answer) {
11         this.question = question;
12         this.answer = answer;
13     }
14
15     @ public int quiz(Switzerland[] quiz) {
16         Scanner s = new Scanner(System.in);
17         for (Switzerland switzerland : quiz) {
18             System.out.println(switzerland.question);
19             String userInput = s.nextLine();
20             if (userInput.equals(switzerland.answer)) {
21                 score++;
22             }
23         }
24         System.out.println("score is: " + score + "/" + quiz.length);
25     }
```

8.2 Delegation

Anhand dieses Screenshots ist erkennbar, dass die Selection weiter delegiert wird, an das richtige Quiz.

```
import java.util.Scanner;

4
5 public class Menu {
6     private boolean exit = false;
7     private int score = 0;
8     private String name = "";
9     private final Animal animal = new Animal( question: "", answer: "");
10    private final Language language = new Language( question: "", answer: "");
11    private final Switzerland switzerland = new Switzerland( question: "", answer: "");
12    private final Ranking ranking = new Ranking();
13    private final TimeAPI time = new TimeAPI();
14
15    private void useSelected(int selected){
16        switch (selected) {
17            case 1 -> score += animal.quiz(Quiz.animalQuiz);
18            case 2 -> score += language.quiz(Quiz.languageQuiz);
19            case 3 -> score += switzerland.quiz(Quiz.switzerlandQuiz);
20            case 4 -> {
21                ranking.addNewScore(name, score);
22                ranking.printScore();
23            }
24            case 5 -> {
25                ranking.saveScore();
26                exit = true;
27            }
28        }
29    }
30 }
```


8.3 Static

Anhand dieses Screenshots ist erkennbar, dass die Quizfragen statisch codiert worden sind und ohne Instanziierung der Klasse Quiz aufgerufen werden können.

```
package ch.tbz;

public class Quiz {
    public static Animal[] animalQuiz = {
        new Animal( question: "How many legs does a dog have?", answer: "4"),

        new Animal( question: "Which animal lives in the northpole? \n (a) penguin / (b) Ice Bear / (c) dolphin", answer: "b"),
        new Animal( question: "What is the biggest thing a whale can eat? \n (a) a car / (b) little crabs / (c) big fishes", answer: "b"),
    };

    public static Language[] languageQuiz = {
        new Language( question: "What is house in german?", answer: "Haus"),
        new Language( question: "What is Kitchen cabin in german?", answer: "chuchichästli", // >:D
        new Language( question: "What is Love in korean?", answer: "사랑해"), // >:D
    };

    public static Switzerland[] switzerlandQuiz = {
        new Switzerland( question: "What is the capital of switzerland? \n (a) Berne / (b) Zürich ", answer: "b"), // >:D
        new Switzerland( question: "Who is the best IT guy in switzerland?", answer: "Gregor Muheim", // with a lot of lööy from BI18a
        new Switzerland( question: "Who is the apple guy? \n (a) Manuel Müller / (b) Louis Define / (c) Wilhelm Tell", answer: "c"),
    };
}
```

8.4 Final

Anhand dieses Screenshots ist erkennbar, dass die Werte der Variable Animal kann keiner neuen Instanz zugewiesen werden.

```
package ch.tbz;

import java.util.Scanner;

public class Menu {
    private boolean exit = false;
    private int score = 0;
    private String name = "";
    private final Animal animal = new Animal( question: "", answer: "");
    private final Language language = new Language( question: "", answer: "");
    private final Switzerland switzerland = new Switzerland( question: "", answer: "");
    private final Ranking ranking = new Ranking();
    private final TimeAPI time = new TimeAPI();
}
```

8.5 Exceptions

Anhand dieses Screenshots ist erkennbar, dass jede Exception abgefangen wird und nach dem try oder dem Catch, noch der finally ausgeführt.

```
public void menu(){
    Scanner input = new Scanner(System.in);
    try{
        time.getFromApi();
        System.out.println("Enter your name");
        name = input.nextLine();
        while (!exit){
            System.out.println("*****\n" + time.showAPIInfo() + "\n1. Animal Quiz\n2. Language Quiz\n3. Switzerland Quiz\n4. show Scoreboard");
            useSelected(input.nextInt());
        }
    }catch (Exception e){
        System.out.println(e);
    }
    finally {
        input.close();
    }
}
```

9 Anwendung

Beim Start: Namen eingeben

```
> Task :Main.main()
Enter your name
Arda|
```

Nachdem man seinen Namen eingeben hat, kann man zwischen den fünf Optionen wählen.

```
time in Zurich is: 2021-11-12T21:31:22.459100+01:00
1. Animal Quiz
2. Language Quiz
3. Switzerland Quiz
4. show Scoreboard
5. Quit
```

Option 4 als Beispiel:

```
Arda
*****
time in Zurich is: 2021-11-12T21:31:22.459100+01:00
1. Animal Quiz
2. Language Quiz
3. Switzerland Quiz
4. show Scoreboard
5. Quit
4
Player: gregor    Score: 3
Player: Muriel    Score: 3
Player: Arda     Score: 0
*****
```

Für dieses Projekt haben wir Java JDK 16.0.2 verwendet.

Achtung bei Gradle müssen die Dependencis hinzugefügt werden, um den Code zu kompilieren!

10 Reflexion

Beim Start dieses Modul, sah Java wie eine unüberwindbare Mauer aus, aber nach viel Zusammenarbeit und Geduld konnten wir diese Mauer bezwingen. Mithilfe Googles und guten Freunden konnten wir unser Wissen über Java erweitern und dieses Projekt durchführen mit Erfolg.

Ein kniffliger Teil des Projektes war die Sortierung der Rangliste. Umsetzen konnten wir es mithilfe vieler Beispiele im Internet in Kombination eines Kollegen konnten wir das perfekte Scoreboard für das Projekt programmieren. Gegenseitig haben wir uns motiviert und am Projekt gearbeitet. Mit gutem Fortschritt konnten wir unsere Motivation auch oben halten.

Wir hatten ein Stack Overflow Problem am Anfang, dass durch ein Rekursiven Aufruf der eigenen Klassen beim Instanzieren entstanden ist. Das Problem haben wir durch die Klasse Quiz behoben.

Wir sind sehr zufrieden mit dem Ergebnis, wir konnten unsere Ziele erreichen und haben sogar noch weiteres hinzugefügt, was wir am Anfang nicht geplant haben.

Ein grosser Dank geht raus an unseren Kollegen Gregor, ohne seine Hilfe hätten wir niemals so viel gelernt und geschafft.