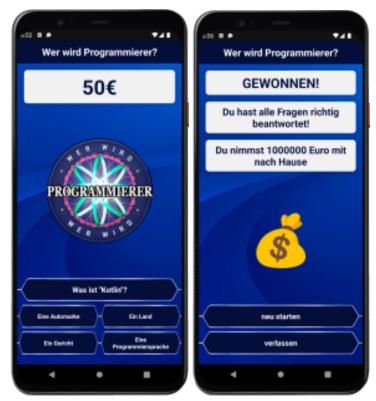
Aufgaben



Hinweis: Zu bearbeiten ist Aufgabe 1.

1. WER WIRD PROGRAMMIERER? - Shared ViewModel

In dieser Aufgabe stellen wir die App "Wer wird Programmierer?" fertig, indem wir den Ergebnis-Screen mit den Ergebnis-Informationen programmieren.



- Öffne das Projekt "Wer wird Programmierer"
- Bisher wurden auf dem Ergebnis Bildschirm keine nützlichen Informationen angezeigt (Was war die richtige Antwort? Wie viel Geld wurde erspielt?). Das wollen wir jetzt ändern. Dafür ist ein Shared ViewModel sehr nützlich, da wir so in unserem ResultFragment auf die Informationen zugreifen können, die vorher nur im QuizViewModel des QuizFragments sichtbar waren.

SHAREDVIEWMODEL

Dazu hat sich die Struktur im ui Ordner geändert. Anstelle von zwei unterschiedlichen ViewModels für die einzelnen Fragmente, gibt es nun ein einziges **SharedVieModel** für die Programm Ablauflogik und Daten.

Außerdem gibt es keine einfache _gameOver Variable mehr, die nur auf das Ende des Spiels verweist. Vielmehr muss die SharedViewModel Klasse hier um drei Live Data Variablen erweitert werden.

- moneyWon Variable speichert die bisher erreichten Preisgelder
 - Defaultwert: 0
- lastAnswer Variable zeigt, ob die letzte Frage korrekt beantwortet wurde
 - Defaultwert: true
- wonTheMillion variable zeigt, ob der Teilnehmer alle Fragen korrekt beantwortet hat und die Million gewonnen hat.
 - Defaultwert: false

Hinweis: Bekannte Verschachtelung für LiveData beachten.

Nun muss auch die alt bekannte *checkAnswer(answerIndex: Int)* Funktion angepasst werden, denn wir können jetzt einige Unterscheidungen machen, wie das Spiel endet oder in welchem Zustand es sich zum Ausführungszeitpunkt befindet.

Uns ist bekannt, wenn die Frage richtig beantwortet ist springen wir zur nächsten Frage.

- Doch nun muss vorher noch die moneyWon Variable auf das aktuelle Preisgeld der Frage erhöht werden.
- Wenn es sich um die letzte Frage handelt, die richtig beantwortet wurde, muss die Variable wonTheMillion dementsprechend korrekt gesetzt werden.

Wurde die Frage falsch beantwortet setzten wir die Variable lastAnswer mit dementsprechenden Wert.

Zum Schluss ergänzen wir auch die resetGame() Funktion.

• setze die drei Variablen auf ihre Standardwerte zurück.

Hinweis: Es handelt sich um LiveData die verändert wird. Bitte beachte die Besonderheit.

Die Logik ist nun fertig implementiert im Anschluss kümmern wir uns um die Fragmente und die Darstellung.

QUIZFRAGMENT

Zuerst schauen wir uns das **QuizFragment** an. Wir kennen den Aufbau der Klasse und auch das dazugehörige Layout fragment_quiz.xml

Das Layout braucht keine Anpassungen, sondern kann wie vorgegeben genutzt werden.

In der Klasse QuizFragment müssen folgende Anpassungen stattfinden

- speichere in der Variable viewModel eine Instanz des SharedViewModel, achte darauf, dass das ViewModel dieses mal aus der Activity geladen wird.
 - Hinweis: siehe Vorlesung 3.8 Folie 6
- in der *onViewCreated()* Funktion
 - weise der binding Variable viewmodel des Layouts, das viewModel der Klasse zu. (Achte auf die Schreibweisen der beiden Variablen)
 - o rufe über das viewModel die resetGame() Funktion auf
 - führe in den onClickListenern die checkAnswer(index) Funktion mit den korrekten indizes aus
 - o Beobachte die LiveData Variablen wonTheMillion und lastAnswer.
 - Wenn das Spiel beendet ist (Gewonnen oder Verloren)
 navigiere zum ResultFragment

RESULTFRAGMENT

Nun implementieren wir das **ResultFragment** und führen folgende Anpassungen durch

- speichere in der Variable viewModel eine Instanz des SharedViewModel, achte darauf, dass das ViewModel dieses mal aus der Activity geladen wird.
- in der onViewCreated() Funktion
 - weise der binding Variable viewmodel des Layouts, das viewModel der Klasse zu. (Achte auf die Schreibweisen der beiden Variablen)
 - o Prüfe ob die Million gewonnen wurde:
 - wenn gewonnen:
 - setzte den Text der TextView tvResult zu "GEWONNEN!"
 - setzte den Text der TextView tvRightAnswer zu "Du hast alle Fragen richtig beantwortet"
 - wenn verloren:
 - erstelle eine lokale Variable rightAnswer
 - hole aus dem viewModel, die korrekte Antwort der aktuellen Frage, dieser gibt einen Integer Wert zurück, welche ein Verweis auf die korrekte Antwort ist
 - Wenn die korrekte Antwort
 - 1 ist dann setzte rightAnswer = answerA der aktuellen Frage
 - 2 ist dann setzte rightAnswer = answerB
 der aktuellen Frage
 - 3 ist dann setzte rightAnswer = answerC
 der aktuellen Frage
 - 4 ist dann setzte rightAnswer = answerD der aktuellen Frage
 - setzte den Text der TextView tvResult zu "Spiel beendet!"
 - setzte den Text der TextView tvRightAnswer zu "Die richtige Antwort war rightAnswer"

○ In beiden Fällen (gewonnen oder verloren) muss die TextView tvAmountWon das erreichte Preisgeld mitteilen, setzte dafür den korrekten Text

Hinweis: Für die Strings der Text views existieren string Ressourcen die formatiert werden können siehe **getString()** Funktion

Viel Erfolg! \$

