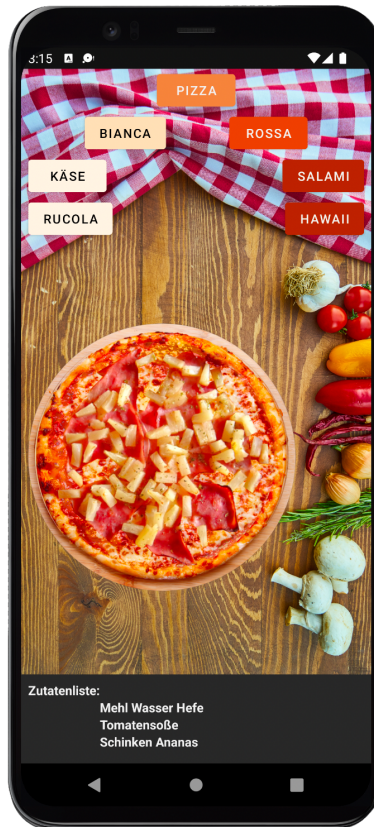


Hinweis: Zu bearbeiten sind Aufgaben 1 und 2

1. PIZZA - Klassen und Vererbung

In dieser Aufgabe programmieren wir eine Pizza App, welche uns für jede Art von Pizza die Zutatenliste liefert.



- Öffne das Projekt "Pizza"
- Erstelle für jede PizzaVariante eine eigene Datei und füge dort den dazugehörigen Code, d.h. die Funktionen und die Variablen der entsprechenden Pizzavariante ein.
- Jede Art von Pizza beinhaltet ein Grundrezept mit den gleichen Zutaten. In unserem Fall vereinfachen wir dieses, sodass der Teig einer Pizza aus "Mehl", "Wasser" und "Hefe" hergestellt werden kann.
 - Erstelle eine neue Klasse namens "Pizza", sie enthält im Konstruktor 3 String Attribute, repräsentativ für die drei Grundzutaten. Diese haben jedoch noch keinen Wert zugewiesen!

- Die Klasse enthält eine Funktion namens `bake`, die eine `ImageView` als Parameter übergeben bekommt. Innerhalb der Funktion wird die `ImageResource` der `ImageView` auf die drawable `teig` gesetzt.
- Die Klasse enthält zudem eine weitere Funktion, die drei `TextViews` übergeben bekommt (die drei Zeilen der Zutatenliste). Innerhalb der Funktion wird der Text der ersten Zeile mit den Zutaten beschrieben. Der Text der zweiten und der dritten Zeile soll ein leerer String sein

Hinweis: Da von der Klasse geerbt werden soll und die Funktionen überschrieben werden sollen, musst du das Keyword `open` einsetzen

- Erstelle nun eine Klasse `PizzaBianca`, die eine weiße Pizza repräsentiert und eine Klasse `PizzaRossa`, die die klassische Pizza repräsentiert
 - Beide Klassen erweitern die Klasse `Pizza` und erben von ihr
 - Zusätzlich haben beide Klassen jedoch ein weiteres Attribut im Konstruktor, die Zutat "Ricotta" für die Pizza Bianca und die Zutat "Tomatensoße" für die Pizza Rossa
 - Beide Klassen überschreiben die Funktionen aus der `Pizza` Klasse
 - Die Funktion `bake` ist identisch zu der Funktion aus Klasse `Pizza`, setzt jedoch das drawable `pizza_bianca` bzw. `pizza_rossa`
 - Die Funktion `ingredients` ruft mit dem entsprechenden Keyword die Funktion der Klasse `Pizza` auf und erweitert die Funktion, indem nun auch die zweite Zeile der Zutatenliste die entsprechende Zutat enthält.
- Erstelle analog die Klassen `PizzaRossaSalami` und `PizzaRossaHawaii`, welche beide von Klasse `PizzaRossa` erben. Hierbei enthält der Konstruktor für die Pizza Salami nun auch noch die Zutat "Salami" und der Konstruktor der Pizza Hawaii die Zutaten "Schinken" und "Ananas".
 - Die Funktion `bake` setzt die entsprechende drawable
 - Die Funktion `ingredients` ruft die übergeordnete Funktion auf und füllt nun auch die dritte Zeile
- Erstelle nun genauso zwei Klassen `PizzaBiancaKaese` und `PizzaBiancaRucola` mit den Extrazutaten "Käse" bzw "Rucola"
- Abschließend programmieren wir in der Klasse "MainActivity" noch einen `onClickListener` pro Button, in denen immer Folgendes passiert:
 - Ein Objekt der entsprechenden Klasse wird erstellt
 - beide Funktionen werden mit dem Objekt aufgerufen

Viel Erfolg!



2. LOTTO - Kotlin Basics (Bonus)

In dieser Aufgabe programmieren wir eine Lottomaschine mit zugehöriger Logik.



- Öffne das Projekt "Lotto" und hier die Klasse "MainActivity". Zudem findest du die Klasse `Lottokugel`, in der bereits eine Funktion zum "ziehen" der Kugel enthalten ist. Du programmierst in Klasse "MainActivity"
- Erstelle bei den Klassenvariablen eine veränderliche Liste in der 9 Lottokugel Objekte enthalten sind, je eine Kugel pro mögliche Zahl von 1 bis 9
- Erstelle zwei weitere Listen mit Elementen vom Typ `Int`, eine Liste in der alle gezogenen Zahlen gespeichert werden sollen und eine Liste in der alle getippten Zahlen gezogen werden sollen. Beide Listen sind noch leer.
- Führe nun in der `onCreate` Funktion die Funktion `setOnClickListener` für jedes TippFeld einmal aus und übergebe das TippFeld als Parameter
- Hole den `btnDraw` aus dem Layout, mit dem die Ziehung ausgelöst wird. Setze einen `onClick`Listener darauf, in dem Folgendes passiert:

- Eine Schleife wiederholt sich 6 Mal und führt jedes Mal eine Reihe von Dingen aus:
 - Ein zufälliger Index wird erstellt, der zwischen 0 und der Größe der Liste mit den Lottokugeln liegt
 - Die Lottokugel an dieser Stelle wird aus der Liste geholt und die Funktion `drawBall` auf die Kugel ausgeführt

Hinweis: als Parameter `activity` kannst du einfach `this` übergeben

 - Die Kugel an der Stelle des Index wird aus der Liste mit Lottokugeln entfernt
 - Die Zahl des `number` Attributs der Kugel wird in die Liste der gezogenen Zahlen hinzugefügt
- Außerhalb der Schleife wird der Button deaktiviert (nicht klickbar) und die Funktion `checkIfWon` wird ausgeführt
- Nun müssen wir nur noch die Funktionen `setOnClickListener` und `checkIfWon` programmieren
- In der Funktion `setOnClickListener` gehst du mit einer Schleife durch jeden Button im übergebenen `tippFeld` und führst jedes Mal Folgendes aus:
 - die `backgroundTintList` des Buttons wird auf `tintYellow` gesetzt
 - die Textfarbe wird auf Schwarz gesetzt
 - In die Liste mit den getippten Zahlen wird die Zahl des Buttons hinzugefügt

Hinweis: um einen String in ein `Int` zu konvertieren, kannst du die Funktion `Integer.parseInt()` verwenden

 - Alle Buttons im `tippFeld` werden auf nicht klickbar gesetzt
- Die Funktion `checkIfWon` wird nach der Ziehung ausgeführt und vergleicht die Liste der getippten Zahlen mit der Liste der gezogenen Zahlen
 - Gehe mit einer Schleife die indices der getippten Zahlen Liste durch und vergleiche die Zahl mit der Zahl an der gleichen Stelle der gezogenen Zahlen Liste
 - Zähle mit einer vorher angelegten Variable mit, wie oft dieser Vergleich klappt, wenn er 6 Mal klappen sollte, ist der Jackpot geknackt
 - gib in der TextView namens `tvEnd` einen Text aus, in dem steht, wie viele der Zahlen am Ende richtig getippt wurden
 - Falls der Jackpot geknackt ist, kannst du die ImageView namens `geldregen` zudem auf sichtbar stellen
 - Das Spiel sollte nun funktionieren, viel Glück ;)

Viel Erfolg!

