Lösungen

1. Willkommen in Android Studio

Fragen

1. Ein SDK (= Software Development Kit) ist eine Sammlung von Werkzeugen und Tools, um eine App zu erstellen, in unserem Fall für das Betriebssystem Android. Ein AVD (= Android Virtual Device) ist ein Programm oder Tool, das ein Android-Smartphone simuliert (es kann aber auch z.B. ein Tablet sein).
2. Eine Activity ist ein Benutzerinterface für eine Android-Applikation (App). Sie bildet sozusagen den Aktionsrahmen auf dem Display.

Keine Aufgaben

2. Das erste eigene Projekt

Fragen

1. Ein Projekt kann aus mehreren "Paketen" (`package`) bestehen. Ein Paket (`package`) ist sozusagen eine Obermenge, da es aus mehreren Klassen (`class`) bestehen kann. Ein Klasse selbst enthält die Definitionen von Eigenschaften und Methoden. Aus einer Klasse lässt sich ein Objekt erzeugen. (Komponenten sind auch Objekte.)
2. Bisher erst zwei: Schaltflächen (Buttons) und Anzeigefelder (TextViews).

Aufgaben

1. Schaue dir dazu das Projekt PROJEKT1A an.
2. Das neue Projekt heißt STERNE1.

3. Hallo mit Knopfdruck

Fragen

1. Erst kommt der Typ, dann der Name; z.B.:

```
String Text;  Button Knopf;  TextView Anzeige;
```

2. Ein Objekt wie z.B. `Knopf` vom Typ `Button` wird so verknüpft:

```
Button Knopf = (Button) findViewById(R.id.button);
```

Dabei sucht die Methode `findViewById` nach der ID der Komponente, die du im Layout erstellt hast. Dann werden beide verknüpft. Wenn du auf die abgebildete Komponente klickst oder tippst, wird das Objekt `Knopf` über die Methode `onClick()` aktiv.

Aufgabe

1. Hier findest du eine Lösung im Projekt HALLO2.
2. Das erweiterte Projekt heißt STERNE2.

4. Rechnen mit dem Zufall

Fragen

1. Durch die erste Zuweisung erhält Zahl1 eine zufällige ganze Zahl von 0 bis 5, in der zweiten Zuweisung sind es die Ganzzahlen von 1 bis 6. Könnte man für ein Würfelspiel benutzen.
2. Zum einen gibt es die Rechenoperatoren (`+ - * /`), die sich natürlich nur auf Zahlen anwenden lassen. Der Zuweisungsoperator (=) sorgt dafür, dass eine Variable einen Wert bekommt. Und der Verknüpfungsoperator (`.`) verbindet ein Objekt mit einer Methode (oder einer Eigenschaft, wie du noch sehen wirst).

Aufgabe

1. Schaue dir mal das Projekt MATHE3 an.

5. Bedingungen

Fragen

1. Vergleichsoperatoren dienen dazu, Werte bzw. Inhalte von Variablen zu vergleichen. Verknüpfungsoperatoren sind dazu da, um Bedingungen miteinander zu einer einzigen Gesamt-Bedingung zu verknüpfen.
2. Nein, in dieser `if`-Struktur werden **mehrere** Bedingungen miteinander verknüpft, in einer `switch`-Struktur lassen sich nur Werte **einer** Variablen überprüfen.

Aufgabe

1. Im Projekt MATHE4 gibt es was zum Eintippen.

6. Geld und Spiele

Fragen

1. Bei `while` wird erst die Bedingung überprüft, dann kommt der eigentliche Schleifenteil, bei `do-while` kommt die Abfrage der Bedingung am Schleifenende.
2. Wenn die Bedingung einer `while`-Schleife gleich zu Anfang nicht erfüllt ist, wird der Anweisungsblock niemals durchgeführt.
3. Wenn man im Million-Projekt für `Kapital` eine 0 eingibt, dann wiederholt sich die Schleife endlos. Das Programm muss »gewaltsam« beendet werden.
4. Wird die Zufallszahl direkt in `onClick()` erzeugt, dann gibt es bei jedem Klick auf OK eine neue Zahl. Die Chance, die herauszufinden, wäre fast aussichtslos.

Aufgaben

1. Das Projekt MILLION1A hat eine `do-while`-Schleife.
2. Dazu schaue dir mal das Projekt MILLION2 an.
3. Ohne den **OK**-Button kommt das Projekt RATEN7 aus.
4. Im Projekt RATEN6 kannst du deine Rate-Zahl eingeben.

7. Es bewegt sich was

Fragen

1. Eine Komponente vom Typ `ToggleButton` wirkt wie ein Ein-Aus-Schalter. Man kann beiden Schaltzuständen jeweils einen Text zuordnen. Im Prinzip lässt sich das auch mit einem `Button` machen, nur wird es da etwas umständlicher (Beispiel MOVIE4).
2. So wird ein Bild sichtbar bzw. unsichtbar:

```
Bild.setVisibility(View.VISIBLE|INVISIBLE); // GONE
```

3. Bei `oneshot="true"` wird die Animation genau einmal ausgeführt (und stoppt automatisch).
4. Ja, damit wird eine Bedingung ins Gegenteil verkehrt. Die Beispiele sind gleichbedeutend mit:

```
if (Gehen == false)  //oder
if (Zahl != 0)
```

Aufgabe

1. Eine Lösungsmöglichkeit (mit einem normalen Button statt drittem ToggleButton) findet sich im Projekt MOVIE4.

8. Animationen

Fragen

1. Beide Methoden `onClick()` und `onTouch()` reagieren auf »Druck«, z.B. Mausklicken oder Fingertippen. Beide informieren über das berührte Objekt. `onTouch()` aber kann noch zahlreiche Zusatzinformationen wie genaue Klick-Tipp-Position, genauere Art der Berührung usw. liefern.
2. `animate()` ist eine Art Tausendsassa-Methode von `ImageView`. Man kann damit ein Bild mehrfach verändern: Einblenden, Ausblenden, Verkleinern, Vergrößern, Verschieben (und mehr).
3. Ohne die genauen Displaymaße in Pixeln lässt sich eine App nicht an alle Smartphones oder Tablets anpassen. Wichtig aber ist: In der Regel stehen nicht alle Pixel zur Verfügung, oft gibt es unerreichbare Ränder.
4. Eine selbst vereinbarte Methode wird aktiviert, indem man sie aufruft:

```
// Methode vereinbaren
void machMal() {
}
// Methode aufrufen
machMal();
```

Aufgaben

1. Schaue dir mal die Wachs-Schrumpf-Variante MOVIE6 an.
2. Hier gibt es eine Lösungsmöglichkeit im Projekt FADEGROW1.
3. Die neue Methode `setLimits()` findest du in KUGEL5.

9. Ein Käfer krabbelt sich frei

Fragen

1. Größe und Lage verändern geht mit der `animate`-Methode, zum Drehen sollte man `setRotation()` benutzen:

```
Bild.animate.scaleX().scaleY(); // größer-kleiner
Bild.animate().x().y();         // verschieben
Bild.setRotation();             // drehen
```

2. Ein `Runnable`-Objekt ist ein ausführbarer Prozess wie z.B. eine Methode. Es kann von Handlern oder »Exekutoren« verarbeitet werden.
3. Ein Konstruktor ist eine Methode, um ein Objekt aus einer Klasse zu erzeugen.
4. Ein Objekt aus einer selbst vereinbarten Klasse wird erzeugt, indem man den

Konstruktor über `new` zuweist:

```
// Klasse vereinbaren
class Klasse {
  Klasse() {  // Konstruktor
  }
}
// Objekt erzeugen
Klasse Ding = new Klasse();
```

Aufgaben

1. Das Ergebnis meiner Bemühungen liegt im Projekt KUGEL5.
2. Eine springende Schaltfläche findest du in JUMPBUTTON1.

10. Vom Käfer zur Wanze

Keine Fragen

Aufgabe

1. Das Projekt KUGEL6 hat eine eigene `GameView`-Klasse.

11. Springen oder Ducken?

Fragen

1. Bei kleinen Smartphones geht es wie bei großen Tablets darum, dass die Größe von Bildobjekten zu den Bildschirmmaßen passt. Also muss sie über das Layout flexibel angepasst werden.
2. Die Klick-Tipp-Position ermittelt man mit `getX()` und `getY()`, und wenn `getAction()` den Wert `action_up` zurückgibt, wurde die Maustaste losgelassen bzw. der Finger vom Display abgehoben.

Aufgabe

1. Die Lösung zu dieser Aufgabe findest du quasi »versteckt« im Projekt DODGER2A.

12. Kontaktvermeidung

Frage

1. Die Bedingung muss hier heißen: `Ding.getRight()-overlap > Figur.getLeft()`

Aufgaben

1. Im Projekt DODGER6B kommt der Ball von rechts auf die links stehende Figur zu.
2. Das Projekt DODGER7 zeigt eine Startinformation, bevor es losgeht, und am Schluss Zusatzinformationen. Dafür sind zwei Textfelder im Einsatz.
3. Die modifizierten Methoden `setPosition()` und `moveObject()` und weitere Anpassungen findest du ebenfalls im Projekt DODGER7.

13. Arrays

Fragen

1. Arrays mit jeweils zehn Strings und Dezimalzahlen kann man z.B. so erzeugen:

```
String[] Text; Text = new String [10];
float[] Zahl; Zahl = new float[10];
```

2. Um den Elementen eines Arrays Werte zuzuweisen, kann man eine `for`-Schleife benutzen:

```
int[] Zahl = new int[10];
for (int i = 0; i < 99; i++) Zahl[i] = i;
```

Aufgaben

1. Im Projekt OWNVIEWS werden jeweils ein Textfeld und eine Schaltfläche erzeugt. (Beide bekommen auch einen Text zum Anzeigen.)
2. Eine Lösung findest du in DEBUG3A.

14. Das große Krabbeln

Fragen

1. Würde man `FTyp` in `GameView` vereinbaren, dann müsste man von außen z.B. so darauf zugreifen:

```
Spiel.showObject(nr, Rdraw[Spiel.FTyp[nr]]);
```

   Das meiste, was mit `FTyp` geschieht, spielt sich im Hauptprogramm ab. Es wäre also auf jeden Fall deutlich umständlicher, wenn dieses Array in `GameView` vereinbart

wäre. Deshalb »liegt« es praktischerweise in MAINACTIVITY.

2. Das wäre ein `String`-Array mit einigen Farbnamen:

```
String[] Farbe = new String[]
  {"Rot", "Blau", "Grün", "Gelb", "Türkis", "Lila",
"Braun",
   "Schwarz", "Weiß", "Grau"};
```

Aufgabe

1. Das Auftauchen von Fliegen und Ameisen kannst du im Projekt DEBUG7A beobachten. Jede tote Fliege bringt einen Pluspunkt, jede tote Ameise zwei Minuspunkte.

15. Buntes Allerlei

Keine Fragen

Aufgabe

1. Dazu gibt es keine Beispiele mehr.