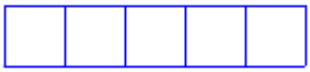
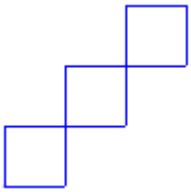
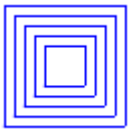
Übungen zu Unterprogrammen

1. *Ein Unterprogramm was_mache_ich(zahl1, zahl2) ist gegeben durch:*
   falls (zahl1>zahl2)
   dann ergebnis = zahl1 – zahl2
   sonst ergebnis = zahl2 – zahl1
   sage ergebnis
   Was passiert beim Aufruf des Unterprogramms durch was_mache_ich(5, 10)?
   Was passiert bei was_mache_ich(25, 13)?
   Was passiert bei was_mache_ich(7, 7)?
2. *Ein Unterprogramm du_nervst(zahl, wort) ist gegeben durch:*
   wiederhole (zahl)-mal
   sage wort
   Was passiert beim Aufruf von du_nervst(5, 'ohje')?
   Rufe das Unterprogramm auf, so dass 'blablablablablablabla' gesagt wird.
3. *Ein Unterprogramm quadrat(zahl) zeichnet ein Quadrat mit Seitenlänge zahl.*
   Schreibe den Code (in Worten) für folgende Bilder. Nutze „quadrat", „ändere x um..." und „ändere y um..." und natürlich die Kontrollstrukturen und evtl. Variable, die du brauchst.

Übungen zu Unterprogrammen im MIT-App-Inventor

1. *Schreibe ein Unterprogramm „zeichne_quadrat", dem man die Koordinaten x und y der linken oberen Ecke u. die Seitenlänge eines Quadrats übergeben kann und das dieses zeichnet.*
2. *Beim Berühren des Bildschirms im Bereich der Leinwand (Canvas) soll an der Stelle ein Quadrat mit Seitenlänge 50 gezeichnet werden. Nutze „zeichne_quadrat".*
3. *Schreibe ein Unterprogramm „begruessung", dem man einen Vornamen übergeben kann, und das „Hallo," +Vorname zurückgibt.*
4. *Beim Klick auf einen Button wird die Begrüßung auf einem Label ausgegeben. Der Vorname kann dazu in einer TextBox eingegeben werden. Nutze „begruessung".*
5. *Schreibe ein Unterprogramm „zauberzahl", dem eine Zahl übergeben werden kann und das die Zahl verdoppelt, wenn sie kleiner als 50 ist und halbiert, wenn sie größer ist.*
6. *Teste das Unterprogramm aus Aufgabe 5 mit einigen Zahlen.*
7. *Schreibe ein Unterprogramm „groesste_zahl", dem man drei Zahlen übergeben kann und das die größte der drei Zahlen zurückgibt.*
8. *Teste das Unterprogramm aus Aufgabe 7 geeignet.*
9. ****** Programmiere ein Rechenspiel: Zwei Zufallszahlen (z.B. zwischen -1000 und 1000) werden angezeigt und man kann eine Antwort eingeben. Diese wird überprüft und es wird ausgegeben, ob die Antwort richtig oder falsch war. Schreibe dazu ein Unterprogramm „pruefe_summe", dem man drei Zahlen (z1, z2 und eine mögliche Summe) übergeben kann und das überprüft, ob die dritte Zahl tatsächlich die Summe der beiden ersten ist.*

Mögliche Lösungen zu den Übungen:

1. *Beim ersten Aufruf was_mache_ich(5, 10) ist zahl1 (5) < zahl2 (10), also wird die Anweisung nach „sonst“ ausgeführt: ergebnis = 10 – 5 = 5. Die Katze (oder wer auch immer) sagt also „5“.*
   Beim zweiten Aufruf was_mache_ich(25, 13) ist die erste Zahl größer, also wird die erste Anweisung ausgeführt und in der Variable ergebnis wird der Wert 12 gespeichert. Gesagt wird „12“.
   Beim dritten Aufruf ist die Bedingung der Verzweigung nicht erfüllt, da nach zahl1 > zahl2 gefragt wird und das Gleichheitszeichen nicht dabei ist. Das Ergebnis ist 0. (Wäre es aber auch im umgekehrten Fall.)
   Allgemein berechnet das Unterprogramm die Differenz der beiden übergebenen Zahlen.
2. *Beim Aufruf des Unterprogramms katze_nervt(5, ‘ohje‘) wird die Zahl 5 an die Variable zahl übergeben und ‘ohje‘ an die Variable wort. Die Schleife wird also 5 mal durchlaufen, somit sagt die Katze ‘ohjeohjeohjeohjeohje‘.*
   Den zweiten Teil erhält man durch den Aufruf katze_nervt(7, ‘bla‘).
3. ***Abbildung 1:***
   wiederhole 5 mal
   * quadrat(30)*
   * ändere x um 30*
   Abbildung 2:
   wiederhole 3 mal
   * quadrat(30)*
   * ändere x um 30*
   * ändere y um 30*
   Abbildung 3:
   setze seite auf 60
   wiederhole 5 mal
   * quadrat(seite)*
   * ändere x um 5*
   * ändere y um 5*
   * ändere seite um -10*

Scratch wird von der Lifelong-Kindergarten-Group am MIT-Media-Lab entwickelt. Siehe *http://scratch.mit.edu*.
Scratch ist lizenziert unter *CC BY-SA 2.0* (*https://creativecommons.org/licenses/by-sa/2.0/deed.en*).

Mögliche Lösung der Programmieraufgaben:

Aufgabe 1 und 2:

```
to zeichne_quadrat x y seitenlaenge
do
  call Canvas1 .DrawLine
    x1 get x
    y1 get y
    x2 get x + get seitenlaenge
    y2 get y
  call Canvas1 .DrawLine
    x1 get x + get seitenlaenge
    y1 get y
    x2 get x + get seitenlaenge
    y2 get y + get seitenlaenge
  call Canvas1 .DrawLine
    x1 get x + get seitenlaenge
    y1 get y + get seitenlaenge
    x2 get x
    y2 get y + get seitenlaenge
  call Canvas1 .DrawLine
    x1 get x
    y1 get y + get seitenlaenge
    x2 get x
    y2 get y
```

```
when Canvas1 .TouchDown
  x y
do
  call zeichne_quadrat
    x get x
    y get y
    seitenlaenge 50
```

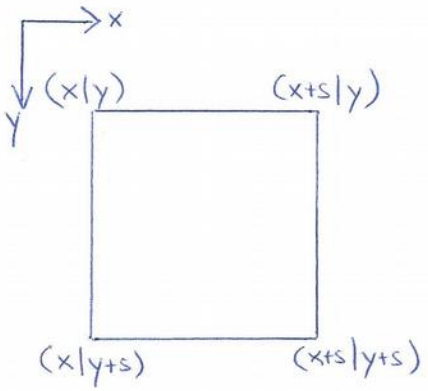
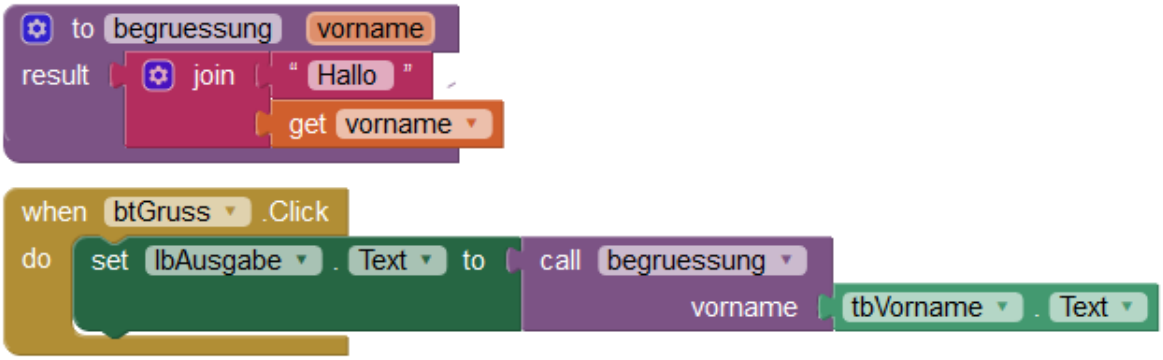
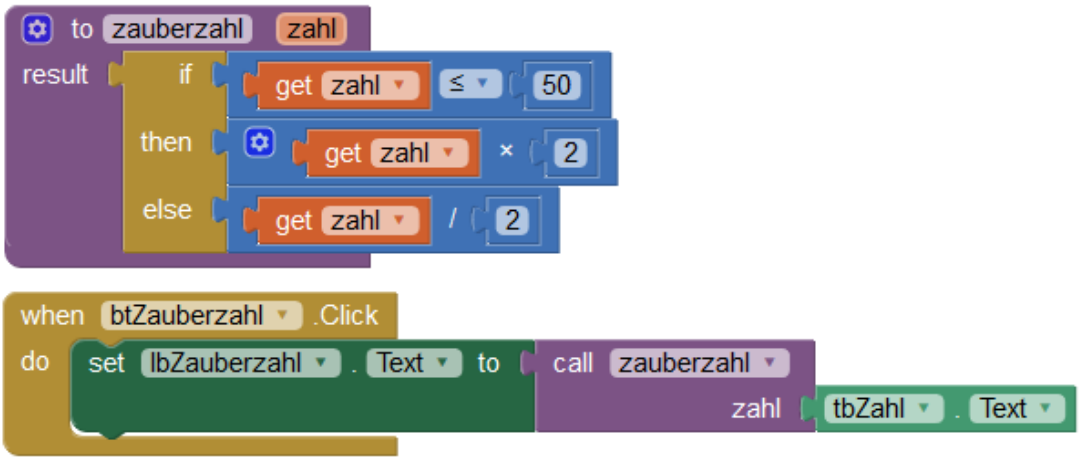


Bild: Eigenes

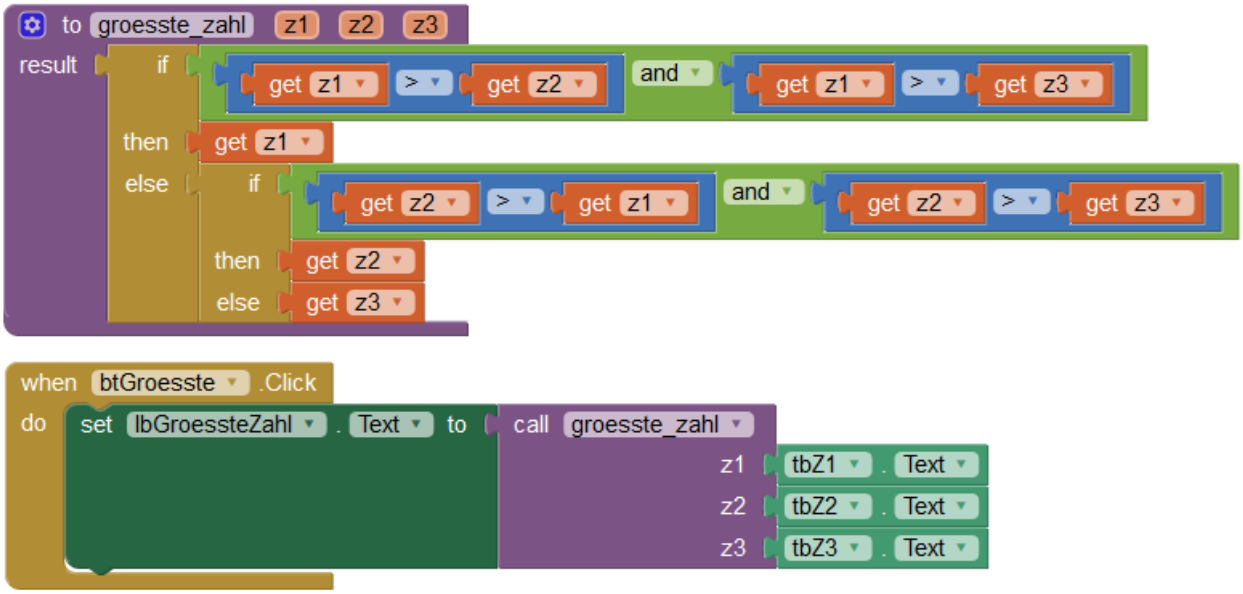
Aufgabe 3 und 4:



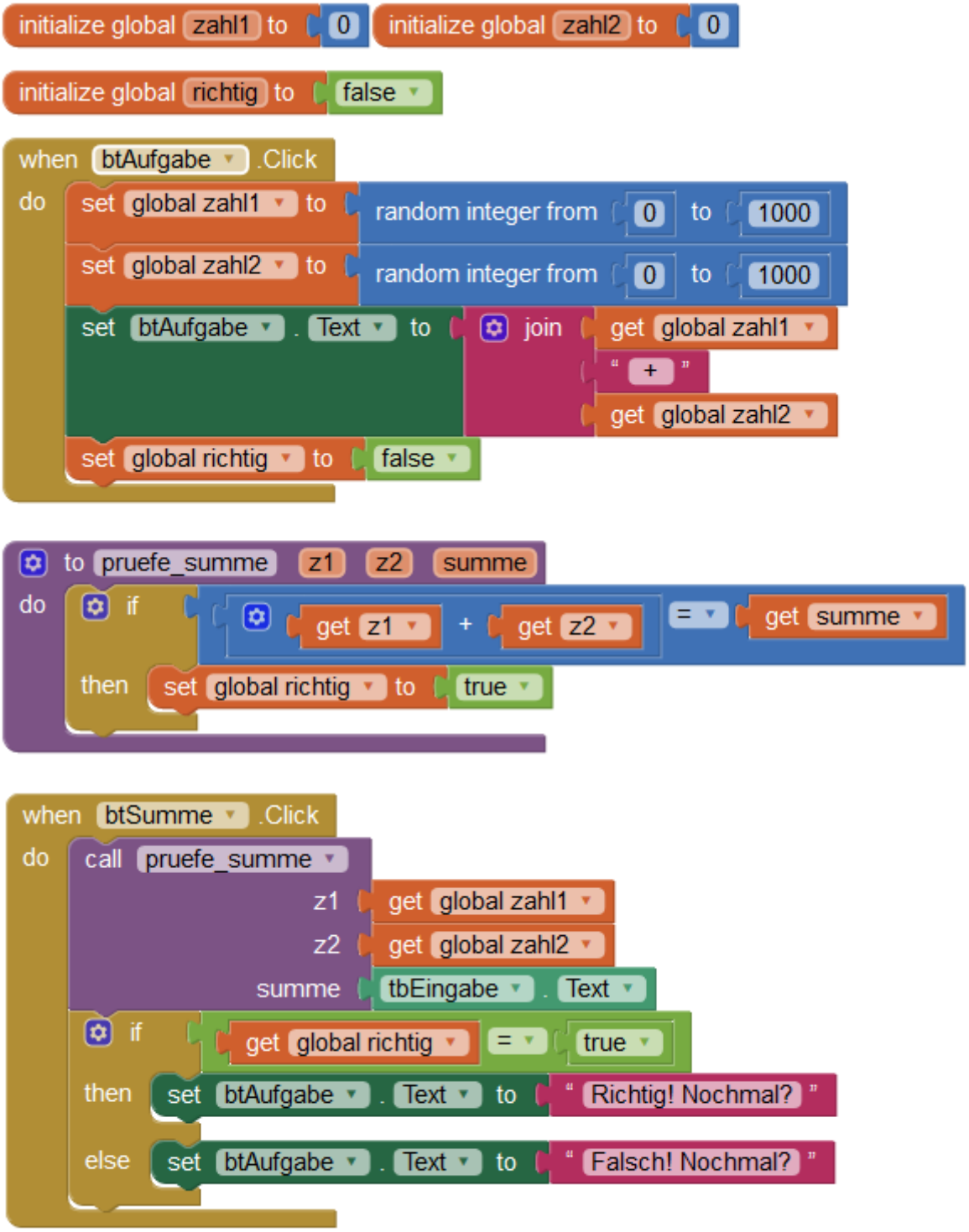
Aufgabe 5 und 6:



Aufgabe 7 und 8:



Aufgabe 9:



Der MIT App Inventor (*http://appinventor.mit.edu*) wurde ursprünglich von einem Entwicklerteam um Mark Friedman und Hal Abelson bei Google entwickelt und 2012 an das MIT übergeben.

Der MIT App Inventor wird unter der Creative Commons Attribution-ShareAlike 3.0 Unported License veröffentlicht: *https://creativecommons.org/licenses/by-sa/3.0*