

Unterprogramme (A)

Diese Station soll es einem ermöglichen im MIT App Inventor Unterprogramme zu erstellen, die anschließend an beliebigen Stellen im Code aufgerufen werden können.

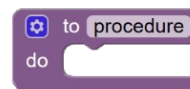
Theorie: Unterprogramme fassen Code zusammen, der an mehreren Stellen eines Programms benötigt wird. Anstatt diesen mehrfach zu implementieren, werden die entsprechenden Codeteile in ein Unterprogramm ausgelagert, das beliebig oft aufgerufen werden kann. Dies verbessert neben der Wiederverwendbarkeit auch die Lesbarkeit komplexerer Programme.

Nach der Abarbeitung des Unterprogramms findet ein Rücksprung zu der Stelle statt, die das Unterprogramm aufgerufen hat. Beim Aufruf des Unterprogramms können auch Werte- oder Referenzparameter übergeben werden (*call-by-value* bzw. *call-by-reference*). Zudem kann ein Unterprogramm einen berechneten Wert an die aufgerufene Stelle zurückgeben. Unterprogramme werden oftmals auch Funktionen, Prozeduren oder Routinen genannt. In der objektorientierten Programmierung spricht man hingegen von Methoden.

Aufgabe 1

Mache dich mit den Codeblöcken der Kategorie *Procedures* vertraut. Erstelle dafür leere Unterprogramme mit folgenden Eigenschaften:

- Name: „Ausgabe“, keinen Werteparameter, keinen Rückgabewert
- Name: „MeineAusgabe“, einen Werteparameter, keinen Rückgabewert
- Name: „Summe“, zwei Werteparameter, einen Rückgabewert




Mithilfe der *call*-Blöcke können Unterprogramme aufgerufen werden. Woran kann man bereits an diesen Blöcken erkennen, ob das aufgerufene Unterprogramm einen Rückgabewert hat oder nicht?

Aufgabe 2

Erstelle eine App, die einen zufälligen Wert aus drei Usereingaben auswählt und zurückgibt.

Erstelle dafür eine neue App, die aus drei TextBoxen, einem Button und einem Label besteht. In die drei TextBoxen soll der User Texte (z.B. Namen) eintragen können. Wenn er dann den Button anklickt, werden die drei Texte an ein Unterprogramm weitergegeben. Dieses wählt einen der drei Texte zufällig aus und gibt diesen anschließend zurück. Danach wird dieser Text über das Label angezeigt.

Tipp: Der Block  aus der Kategorie *Math* ist für die Implementierung nützlich. Zudem wird eine Verzweigung benötigt, um die Aufgabe zu lösen!


Unterprogramme (B)

Diese Station dient zur Vertiefung der Station *Unterprogramme (A)* und setzt voraus, dass die Konzepte dieser Station und aller weiteren Stationen im Niveau A verinnerlicht wurden.

Aufgabe 1

Erstelle eine App, die für eine eingegebene Zahl die Quersumme ermittelt.

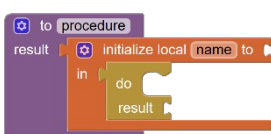
Erstelle dafür eine neue App, die aus einer TextBox, einem Button und einem Label besteht. In die TextBox soll vom User eine Ganzzahl eingegeben werden. Sobald dieser dann den Button anklickt wird die eingegebene Zahl an ein Unterprogramm übergeben, ihre Quersumme berechnet und diese daraufhin zurückgegeben. Anschließend soll die berechnete Quersumme über das Label ausgegeben werden.

Tipp: Der Block  aus der Kategorie *Math* ist für die Implementierung nützlich. Zudem werden eine Verzweigung und ein rekursiver Aufruf des Unterprogramms benötigt.

Aufgabe 2

Erstelle eine App, die für ein gegebenes Zahlenarray den Mittelwert berechnet.

Erstelle dafür zunächst eine neue App, die aus einem Button und einem Label besteht. Erstelle zudem ein Zahlenarray, welches mindestens fünf Zahlen enthält. Klickt man den Button an, so soll einem Unterprogramm das Array übergeben werden. Dieses Unterprogramm berechnet daraufhin den Mittelwert der im Array gespeicherten Zahlen und gibt diesen zurück. Dieser Wert wird daraufhin über das Label ausgegeben.

Tipp: 

Aufgabe 3

Erstelle eine App, die das Zeichnen eines regelmäßigen n -Ecks ermöglicht.

Die App soll lediglich aus einem Canvas, einer TextBox und einem Button bestehen. Über die TextBox soll die Eckenanzahl n eingegeben werden. Über den Button soll lediglich das Canvas gelöscht werden. Über einen Klick in den Canvas, soll zunächst der Zahlenwert der TextBox ausgelesen werden und an ein Unterprogramm übergeben werden. Dieses Unterprogramm berechnet dann die Eckpunkte des regelmäßigen n -Ecks und gibt diese als Liste zurück. Anhand der Punkteliste wird das regelmäßige n -Eck dann auf das Canvas gezeichnet.

Tipp: Die Eckpunkte eines regelmäßigen n -Ecks auf dem Kreis mit Radius 1 um den Ursprung sind gegeben durch $(x_y, y_k) = \left(\cos\left(\frac{360^\circ \cdot k}{n}\right), \sin\left(\frac{360^\circ \cdot k}{n}\right) \right)$.