

Sven Bauder

Das große Programmier- handbuch Band 3

Erste Buchauflage – für IoT und Mobile

Inhaltsverzeichnis

Kapitel 1: Raspberry Pi Linux Installation

Kapitel 2: Android SDK

Kapitel 3: NAS und Router

Einleitung

Hallo und herzlich Willkommen zu diesem brandneuen Buch von Sven Bauder.

Es ist wichtig, dass die Linux-Entwicklung voranschreitet, da die Konzerne Apple und Microsoft immer deutlicher nicht mehr ökonomisch und sozial unterstützbar sind. Nur leider sind wir auf diese Systeme von den beiden Konzernen angewiesen.

Wenn wir viel mehr Programme auf Linux-Systemen laufen lassen könnten, würde diese Nachfrageblase platzen und viele Leute würden sich auf ein Linux-System freuen, als wie auf Systeme mit Windows oder MacOS.

Dieses Buch ist sowohl für den experimentellen Einsatz von Freizeit und Hobby konzipiert, als auch eine wertvolle Erfahrung für Unternehmer kleiner Größe.

Es baut auf dem ersten Buch „Das große Programmierbuch“ von Sven Bauder auf.

Dieses gibt es hier zu kaufen:

URL: shop.tredition.com

Suchwort: Sven Bauder Das große Programmierbuch

Direktlink: https://shop.tredition.com/booktitle/Das_gro%3fe_Programmierbuch/W-773-204-266

Dieses Buch ist in deutscher Sprache geschrieben. Wenn Sie aber die Programmierbefehle nicht lesen können, da diese in englischer Sprache geschrieben werden, empfehle ich Ihnen wärmstens, diese 2 Produkte:

-RosettaStone.de Englisch

-Vera F. Birkenbihl Englisch für Einsteiger und Wiedereinsteiger

Nach all meinen didaktischen Abwägungen bin ich schlussendlich dazu gekommen, dass ein tägliches Training mit diesen beiden Medien gemeinsam (ich verdiene nichts an der Empfehlung dieser 2 Produkte) sehr gut gelingen kann, da ich diese selbst benutzt habe.

Ich wünsche Ihnen viel Freude mit dem nachfolgenden Buch!

© Sven Bauder - Frogproject

Kapitel 1:

Raspberry Pi Linux Installation

Los geht's!

Um auf einem Raspberry Pi Programmierungsprojekte laufen zu lassen, müssen Sie erstmal eines der neuesten Geräte besitzen, die mit aktueller Hardware kompatibel sind.

Wie wir bereits aus Band 1 wissen (es gibt verschiedene Assembler-Sprachen) gibt es für viele unterschiedliche Verwendungszwecke einen anderen jeweiligen Raspberry Pi.

Schauen Sie sich einfach auf der folgenden Webseite einmal um:

<https://www.raspberrypi.com/documentation>

Sie brauchen für die Verwendung eines Raspberry Pi-Minicomputers einen angeschlossenen funktionierenden Bildschirm, eine funktionierende Tastatur und evtl. auch eine funktionierende Maus.

Ein Linux-Betriebssystem kann auf einem externen Computer mit dem folgenden Programm auf eine kleine SD-Karte übertragen werden.

<https://www.raspberrypi.com/software/>

Diese SD-Karte muss nun in den Raspberry Pi gesteckt werden. Stellen Sie dabei sicher, dass der Raspberry Pi abgeschaltet ist bevor Sie die SD-Karte hereinsticken.

Nachdem die SD-Karte eingelegt ist, kann der Raspberry Pi eingeschaltet werden. Folgen Sie einfach den Anweisungen auf dem Bildschirm und installieren Sie Ihr gewünschtes Betriebssystem.

Das wars. Einfach oder?

Kapitel 2:

Android SDK

Los geht's!

Kommen wir nun zum nächsten Kapitel, was allerdings wieder etwas schwieriger wird. Die Programmierung für Android. Dem Linux-Betriebssystem von Google für Smartphones und Tablets.

Installation der SDK-Programme

Um für Android Apps programmieren zu können, brauchst du folgende Programme:

Java Standard Edition (JDK & JRE) <https://www.oracle.com/java/technologies/downloads/>
Android Studio <https://developer.android.com/studio>
Eclipse <https://eclipseide.org/>
Android Development Tools (ADT) Plugin <https://www.eclipse.org/downloads/packages/release/2024-06/r/eclipse-ide-enterprise-java-and-web-developers>

Eine neue Sprache: Java

Die Sprache Java ist unter Windows, Mac und Linux für ihre Sicherheitslücken und Hacking-Anfälligkeit bekannt. Aus einer ehemals bekannten Sprache mit der man viele verschiedene Nuancen der Programmierung ausdrücken konnte wurde ein Flickenteppich der von der Firma Oracle (die Firma hinter Java) trotzdem immer weiter entwickelt worden ist. Java ist für Google Android deshalb nur beliebt, wegen der Möglichkeit der feinen Einstellungen. Java für Desktop-Geräte zu verwenden ist nicht zu empfehlen. Das Betriebssystem Google Android gilt jeher sowieso als unsicher, da seit Jahren Ton und Video mit geschnitten werden, sowie sämtliche anderen persönlichen Daten von den Geräten an Google und andere Werbetreibende, sowie Software-Entwickler automatisch gesendet werden.

Das sind die Gründe warum Java, ehemals Universalwerkzeug, im Band 1 und im Band 2 noch keine Erklärung bekommen hat. Hier versuchen wir dich trotzdem auf den Klassiker fit zu machen, falls Sie harmlose Daten zu transportieren hast. Für alle diskreten Dinge empfehlen wir keine Android-Apps zu verwenden.

In Java gibt es 4 wichtige Begriffe:

Klassen - Eine Klasse ist eine Sammlung an Objekten. Beispielsweise macht die Katze Miau und ist schwarz-weiß gefleckt, der Hund ist aber braunfarben und macht stattdessen „Wuff!“. Das sind zwei verschiedene Klassen.

Objekte - Objekte sind die Schlüsselwörter mit denen die Informationen abrufbar sind. So kann „Fellfarbe“ das Schlüsselwort für „braunfarben“ oder „schwarz-weiß gefleckt“ sein.

Methoden - Mit Methoden können Sie Informationen mathematisch verarbeiten. Mehr dazu später.

Instanzen Variablen - Instanzen sind die Schlüsselwörter der Objekte.

Dein erster Java-Befehl, den Sie in die Eclipse-Umgebung eingeben können, ist:

```
System.out.println("Hallo Welt");
```

Sollte der Code ausgeführt werden, wird

Hallo Welt

Auf dem Bildschirm angezeigt.

Ziemlich unspektakulär, aber mehr kommt ja auch noch.

Wie auch in anderen Programmiersprachen gibt es folgende Variabtentypen:
Int, double, string, boolean, byte, short, char, long, float

Diese müssen erst definiert werden:

```
string meineersteVariable;
```

Und können dann mit Inhalt gefüllt werden:

```
meineersteVariable = „Bingo!“;  
System.out.println(meineersteVariable);
```

Auch sind die Kontrollstrukturen wieder vorhanden:

```
if(bootscher-Ausdruck)  
    AnweisungFallA;  
else  
    // falls boolean false war;  
    5 AnweisungFallB;
```

Neu sind bspw. Die switch-Anweisungen:

```
switch(variable) {  
    case Konstante1:  
        Anweisungen;  
        break;  
  
    case Konstante2:  
        Anweisungen2;  
  
    Kontrollstrukturen  
    break;  
  
    // mehr Faelle  
  
    case KonstanteN:  
        AnweisungenN;  
        break;  
  
    default:  
        // was sonst geschehen soll  
        Anweisungen;  
}
```

For-Schleifen sind auch mit dabei:

```
for(int i=1; i<10; i=i+1) {  
    Anweisungen;  
}  
  
// oder mit nur einer Anweisung  
  
for(int i=1; i<10; i=i+1)  
    Anweisung;  
  
// oder in Kurzschreibweise  
  
for(int i=1; i<10; i++) {  
    Anweisung;  
}
```

Auch while-Schleifen dürfen hier nicht fehlen:

```
boolean pruefungen = true;  
  
while(pruefungen) {  
    // lernen  
    // hier pruefungen auf false setzen  
}
```

Arrays werden wie folgt festgelegt:

```
teilnehmer[0] = 123456;  
teilnehmer[1] = 123457;  
teilnehmer[2] = 123458;  
    //usw.
```

Mehrdimensionale Arrays werden wie folgt festgelegt:

```
int[][] matrix = new int[3][3];  
matrix[0][0] = 1;  
matrix[0][1] = 2;  
matrix[0][2] = 3;  
matrix[1][0] = 4;  
matrix[1][1] = 5;  
matrix[1][2] = 6;  
matrix[2][0] = 7;  
matrix[2][1] = 8;  
matrix[2][2] = 9;
```

Methoden sind ein Weg um Rechnungen immer wieder und kurz befohlen auszuführen.
Eine Methode legt man, wie folgt, fest:

```
public static int f(int x){  
    return x * x;  
}
```

Die Methode heißt nun f und kann immer wieder in folgender Form aufgerufen werden:

```
f()
```

Wie jetzt Apps „gebaut“ werden können, ist nach ein wenig Übung relativ selbsterklärend:

```
public class Konto {  
    String inhaber;  
    double guthaben;  
    int nummer;  
    int blz;  
    double limit;  
  
    // Konstruktor 1: Mit Guthaben
```

```
Konto(String i, double g, int n, int b, double l) {  
    inhaber = i;  
    guthaben = g;  
    nummer = n;  
    blz = b;  
    limit = l;  
}
```

```
// Konstruktor 2: Kein Guthaben
```

```
Konto(String i, int n, int b, double l) {  
    inhaber = i;  
    guthaben = 0.0;  
    nummer = n;  
    blz = b;  
    limit = l;  
}
```

Mit Objekten natürlich. Nun kannst du in Eclipse in Ruhe deinen Code schreiben und in Android Studio deine App designen.