Algorithmische Kunst & Digitale Medien

Zur Geschichte, Ästhetik & Algorithmik digitaler Bilder

Wintersemester 2019/20 | Frieder Nake

Ein praxisorientierter Workshop
im Modul »Kultur, Geschichte, Ästhetik digitaler Medien«

Grundlegende Konzepte von Processing

FN 6.12.2019

Ein *Programm* ist die Beschreibung einer berechenbaren Funktion in einer für einen Computer bearbeitbaren Programmiersprache. Processing ist ein System zum Programmieren, das zwar keine selbständige Programmiersprache enthält (sondern auf Java aufsetzt). Aber es ist in vieler Hinsicht besonders einfach für das Erlernen des Programmierens und auch für dessen schon fortgeschrittene Anwendung. Wir können, ein wenig technischer, ein Programm auch als eine Folge von "Statements" auffassen. Wir unterscheiden grundlegend die folgenden syntaktischen Gegebenheiten.

Vereinbarung (Declaration)

Beispiel: int k = 0;

Eine Variable wird eingeführt mit ihrem Namen, ihrem Datentyp und evtl. einem Anfangswert. Sie hat zu jeder Zeit einen Wert, der zum vereinbarten Datentyp gehören muss. Dieser Datentyp ist hier "int", d.h. der Typ der ganzen Zahlen, { 0, 1, -1, 2, -2, 3, -3, ... }.

Datentypen (Data Types)

Beispiele: int (ganze Zahlen als Werte), float (Komma-Zahlen), boolean (logische Werte), char (alphanumerische Zeichen)

Ein Datentyp ist eine Menge von Elementen, die eine bestimmte Eigenschaft gemeinsam haben. Zum Datentyp gehören auch die für die Elemente zulässigen Operationen.

Zuweisung (Assignment)

Beispiel: x = 4.5 * sin(alfa)

Die Variable x muss hier den Datentyp float haben, da der Ausdruck auf der rechten Seite von diesem Typ ist. Die Variable alfa muss den Wert eines Winkels im Bogenmaß besitzen.

Der Wert des Ausdrucks auf der rechten Seite wird berechnet und der Variablen x zugewiesen. Das Zeichen "=" ist dabei *nicht* das Zeichen für "Gleichheit". Vielmehr steht es für den Operator der "Zuweisung" (der unsymmetrisch von rechts nach links wirkt):

Zuerst wird der Wert des Ausdrucks auf der rechten Seite von = bestimmt, dann wird dieser Wert der Variablen auf der linken Seite zugewiesen. Der bisherige Wert von x wird dabei überschrieben.

Funktionsaufruf (Function call)

Beispiel: cos (beta)

Der Wert der Funktion mit Namen cos (cosinus) an der Stelle beta (vom Typ float) wird berechnet.

Bedingtes Statement (Conditional Statement)

Beispiel: if (m==5) {statements1;} else {statements2;}

Falls die ganzzahlige Variable m den Wert 5 hat, werden die statements1 ausgeführt, andernfalls die statements2. Das doppelte (mathematische) Gleichheitszeichen "==" bedeutet tatsächlich die Gleichheit der Werte der Ausdrücke rechts und links von ==.

Iteriertes Statement (Iterative Statement)

Beispiel: for (int j = 0; j <= 7; j++) { statements; }

Die "Kontrollvariable" j wird lokal eingeführt und zunächst auf den Wert 0 gesetzt. Dann wird geprüft, ob die Abbruch-Bedingung erfüllt ist, ob also j > 7. Ist das nicht der Fall – d.h., ist j <= 7 – so werden die statements in { } ausgeführt. Danach wird j um 1 erhöht (durch die Operation j++). Nun wird wieder die Bedingung geprüft, etc. Die Iteration wird abgebrochen, wenn erstmals die Bedingung j <= 7 nicht mehr erfüllt ist.

Definition einer Funktion

Beispiel:

```
void setup( )
{
   size(600, 700); background(100);
}
```

Hier wird eine Funktion mit dem Namen "setup" und mit einer leeren Liste von Parametern "( )" vereinbart, die keinen Wert liefert ("void"), sondern nur Seiteneffekte erzeugt. In dem konkreten Fall setzt sie das Format des Bildes auf 600 x 700 Pixel fest, sowie die Hintergrundfarbe auf ein dunkles Grau ("100").

Die Besonderheit an dieser Funktion ist darüber hinaus, dass ihr Name ("setup") im System gegeben und mit der besonderen Eigenschaft verbunden ist, dass sie einmal nur ausgeführt wird.

Das allgemeine Schema für die Definition von Funktionen lautet:

```
<type> <frei gewählter Name> (<Liste von Parametern>)
{
   <Folge von Statements>
}
```

Strukturierte Datentypen. Datenstrukturen

Es gibt verschiedene Datentypen, die über die oben genannten elementaren Datentypen hinausgehen. Sie fassen Elemente einfacherer Datentypen zusammen. Der vielleicht wichtigste strukturierte Datentyp ist das "Feld" (*Array*).

Beispiel einer Deklaration: float [ ] x = new float[n];

Durch diese Vereinbarung wird eine Variable mit dem Namen x eingeführt. Der Name steht aber für einen "Vektor" mit n Elementen, die vom Typ float sind. Die Komponenten der Variable x werden mit einem Index bezeichnet: x[1], x[2], ..., x[i], ..., x[n].