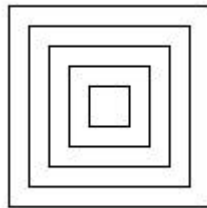


Thema: Rekursion;
 Software: WinLOGO;
 »Altersstufe«: Anfangsunterricht (Unterstufe);
 Voraussetzungen: Kenntnis elementarer LOGO-Graphikbefehle; Konzept der Modularisierung (konkret: Erstellung und Nutzung einer Prozedur zum Zeichnen eines Quadrats mit vorgegebener Seitenlänge);
 Literatur: [1] Otte G.: Einführung in LOGO; Dümmler; Bonn, 1996;
 [2] Papert S.: Gedankenblitze; rororo; Reinbek bei Hamburg, 1985;
 Ausarbeitung: P. K. Antonitsch (nach einem Beispiel in [1])

Aufgabenstellung 1 (zur Demonstration des rekursiven Lösungsansatzes):

Die folgende Figur ist mit Hilfe der bereits erstellten Prozedur »QUADRAT :LÄNGE« zu zeichnen:



Erster Programmansatz für »QUADRATE :LÄNGE«:

```
PR QUADRATE :LÄNGE
QUADRAT :LÄNGE
;STIFTHOCH hebt den Stift von der Zeichenfläche
STIFTHOCH
;RE dreht um die angegebenen Grad nach rechts
RE 45
;VW geht um die angegebene Strecke vorwärts
VW 14
;LI dreht um die angegebenen Grad nach links
LI 45
;STIFTAB setzt den Stift auf die Zeichenfläche
STIFTAB
QUADRAT :LÄNGE - 20
STIFTHOCH RE 45 VW 14 LI 45 STIFTAB
QUADRAT :LÄNGE - 40
STIFTHOCH RE 45 VW 14 LI 45 STIFTAB
QUADRAT :LÄNGE - 60
etc.
ENDE
```

did. Hinweis: Vor der Präsentation dieses Programmes sollten die Schüler diese Figur tätig nachbilden (zeichnen oder entlanggehen) und eine genaue Beschreibung geben, welche Bewegungen sie ausführen! Im Idealfall kann das Programm von den Schülern selbst gefunden werden.

Beobachtung: Die grau unterlegten Anweisungen werden mit immer neuem Übergabeparameter wiederholt. Da sich der Wert des Übergabeparameters ändert, ist die Verwendung der »WIEDERHOLE«-Anweisung ungünstig.

Überlegung: Die Prozedur »QUADRATE :LÄNGE« kann auch programmiert werden, indem man nach dem Zeichnen eines Quadrats und dem »Schritt zur Mitte« wieder zum Prozeduranfang springt und dabei von der aktuellen Seitenlänge die Zahl 20 subtrahiert:

Zweiter Programmansatz für »QUADRATE :LÄNGE«:

```
PR QUADRATE :LÄNGE
QUADRAT :LÄNGE
STIFTHOCH RE 45 VW 14 LI 45 STIFTAB
```



:LÄNGE - 20

Dies wird zu:

```
PR QUADRATE :LÄNGE
QUADRAT :LÄNGE
STIFTHOCH RE 45 VW 14 LI 45 STIFTAB
QUADRATE :LÄNGE - 20
ENDE
```

Analyse des Aufrufs »QUADRATE 100« (nach [1]):

- LOGO legt eine »Schachtel« mit dem Namen »LÄNGE« an und legt die Zahl 100 hinein;
- LOGO holt den Wert aus der Schachtel »LÄNGE« und zeichnet mit dieser Seitenlänge ein Quadrat;
- der Zeichenstift bewegt sich in die richtige Position für das nächste Quadrat;
- das Programm holt den Wert aus der aktuellen Schachtel »LÄNGE«, subtrahiert 20, legt eine neue Schachtel mit dem Namen »LÄNGE« an und legt das Ergebnis der Subtraktion (=80) hinein;
- etc.

Beobachtung: Das Programm »hört nie auf«, mit bloßen »Selbstaufrufen« ist es also noch nicht getan. Man benötigt auch eine Abbruchbedingung (wo? am Anfang? am Ende → Diskussion der Folgeerscheinungen);

did. Hinweis: Obige Beobachtung kann von den Schülern selbst gemacht werden, wenn das Programm wieder tätig umgesetzt wird. Um die Prozeduraufrufe umzusetzen, empfiehlt sich dazu die Verteilung der Rollen »aufrufendes Programm« (»Rufer«) und »ausführendes Programm« (»Ausführer«).

Dritter Programmansatz für »QUADRATE :LÄNGE«:

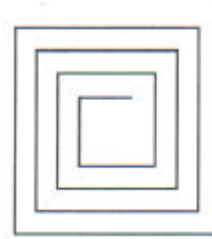
```
PR QUADRATE :LÄNGE
WENN :LÄNGE <20 DANN RÜCKKEHR
QUADRAT :LÄNGE
STIFTHOCH RE 45 VW 14 LI 45 STIFTAB
QUADRATE :LÄNGE - 20
ENDE
```

Aufgabenstellung 2 (zum Einüben des rekursiven Lösungsansatzes):

Das vorgestellte Programm zeichnet die Quadrate von außen nach innen. Wie ist das Programm abzuändern, damit die Quadrate von innen nach außen gezeichnet werden? Wie ist die Abbruchbedingung zu verändern? Ist der Prozedur »QUADRATE :LÄNGE« eine weitere Variable zu übergeben?

Aufgabenstellung 3 (zum Einüben des rekursiven Lösungsansatzes):

Wie kann die nebenstehende Figur mit Hilfe einer selbstaufrufenden Prozedur gezeichnet werden?



Bem.: Warum LOGO?

- LOGO bietet die Möglichkeit, die Wirkung der eingegebenen Programmbefehle direkt über die Bewegung des »LOGO-Igels« (orig. »LOGO-turtle«) zu sehen.
- Dadurch ist das Grundkonzept der Programmierung »graphiklastig«, das Programmieren berührt dadurch aber nicht nur kognitive Aspekte. Das Finden von Algorithmen (im weiteren Sinne) ist nicht auf das (abstrakte) Nachdenken beschränkt, es kann vielmehr »mit allen Sinnen« erfolgen (vgl. obige didaktische Hinweise).
- Grundlegende Konzepte wie das der Variablen, der Modularisierung oder der Rekursion treten dadurch in genetischer Art auf.
- Rekursion ist aber nicht nur auf rekursiv erzeugbare Graphiken beschränkt. LOGO kann auch Listen verwalten, bei denen (von vornherein) nur auf das oberste Element zugegriffen werden kann. Der Zugriff auf darunterliegende Elemente erfordert sukzessives (rekursives) Abbauen des Stapels (sic!).
- LOGO bietet auch die Möglichkeit, Interfaces (Lego, Fischer Technik) anzusteuern und so die »Realwirkung« von Programmen zu erfahren (dieser bislang wenig beachtete Aspekt, die Möglichkeit, über technisches Spielzeug mit einfachen Befehlen Konzepte des Messens, Steuerns und Regelns zu vermitteln, wird seit kurzem von Lego mit Hilfe eines Visual Basic-Interfaces neu belebt!).
- Die Tatsache, dass LOGO keine »salonfähige« Programmiersprache ist, kann nur von jenen als Nachteil empfunden werden, die (noch immer) der Meinung sind, der Informatikunterricht (und insbesondere der Anfangsunterricht!) in der Schule habe weniger die Aufgabe, grundlegende Begriffe der Informatik zu vermitteln, als »brauchbare« Programmierer zu erzeugen...