

## Allgemeine Hinweise zur Programmierung

Der folgende Text gibt einige Hinweise darauf, wie man beim Schreiben von Programmen vorgehen sollte. Diese Hinweise müssen insbesondere bei der Lösung der Praktikumsaufgaben beachtet werden; sie sind aber auch darüber hinaus gültig.

- **Planung** ist wichtig: Man sollte nicht einfach drauflosprogrammieren und nicht versuchen, Fehler schrittweise nach dem „Trial-and-Error“-Prinzip loszuwerden, um so schließlich ein korrektes Programm zu erhalten. Das führt nur zu unübersichtlichem Programmcode, der sehr wahrscheinlich noch versteckte Fehler enthält.  
Sinnvoller ist es, zunächst mit Papier und Stift eine Übersicht über die Kontrollstruktur des Programms (d.h. die Folge der Anweisungen) zu entwerfen, zum Beispiel in einem **Struktogramm**<sup>1</sup>. Dabei kann man den Prinzipien der **strukturierten Programmierung** (und später auch der **objektorientierten Programmierung**) folgen, d.h. das Problem schrittweise in immer kleinere Teilprobleme zerlegen, diese einzeln lösen (also in einzelne Programmstücke, insbesondere Java-Methoden oder -Klassen, umsetzen) und dann die Lösungen schrittweise zu einer Lösung des Gesamtproblems zusammenfassen.  
Bei der Planung muss man sich auch über die **Ein- und Ausgaben** des Programms sowie die benötigten **Datenstrukturen** klar werden.
- Zu jedem Programm gehört eine umfangreiche **Kommentierung**. Mit ihrer Hilfe soll der menschliche Leser den Sinn der einzelnen Programmteile und ihr Zusammenspiel verstehen können. Die Kommentierung nützt insbesondere dem Programmierer selbst, der später nachvollziehen möchte, was er da eigentlich gemacht hat. Im Einzelnen sind folgende Punkte zu beachten:
  - Im **Programmkopf** stehen einige allgemeine Informationen über das Programm. Hierzu gehören:
    - die Angabe des / der Programmierer (im Praktikum sind das **Gruppennummer** und **Namen** der Programmierer)
    - das **Datum** der Erstellung des Programms bzw. eine Aufzählung der Daten, zu denen wesentliche Änderungen des Programms stattgefunden haben
    - eine kurze Beschreibung dessen, was das Programm leistet (im Praktikum genügt hier die **Aufgabennummer**)
    - eventuell eine Versionsnummer
    - eventuell eine allgemeine Beschreibung der Ein- und Ausgaben

---

<sup>1</sup> Im Praktikum müssen Struktogramme nur erstellt werden, wenn es ausdrücklich verlangt ist.

- Bei jeder **Deklaration** ist anzugeben, was der Zweck der deklarierten Variablen, Konstanten, Typen oder Klassen ist (es sei denn, der Zweck ist ganz offensichtlich<sup>2</sup> und/oder geht aus dem Namen klar hervor).
- Für jede zusammengehörige Folge von **Anweisungen** und für jede **Methode** und **Klasse** ist anzugeben, was der Zweck dieser Anweisungen, der Methode oder der Klasse ist (es sei denn, das ist ganz offensichtlich - siehe Anmerkung oben).
- Neben der Kommentierung muss auch der eigentliche **Programmtext lesbar** sein. Das wird unter anderem durch die folgenden Punkte erreicht:
  - Variablen, Methoden, Klassen usw. müssen **aussagekräftige Namen** haben, die ihren Zweck gut beschreiben. Unter einem Variablennamen „p17“ kann sich niemand etwas vorstellen, unter „kontostand“ schon eher. Variablen- und Methodennamen müssen mit einem Kleinbuchstaben beginnen, Klassennamen mit einem Großbuchstaben.
  - Der Programmtext muss ein strukturiertes Layout haben. Das Einfügen von **Leerzeilen** sowie **Einrückungen** helfen, den Aufbau des Programms zu verdeutlichen.
  - Strukturierte und objektorientierte Programmierung (siehe oben) führt zu einem **modularen Programmaufbau**, also einem Aufbau des Programms aus klar umrissenen Funktionseinheiten zur Lösung von Teilproblemen. Ein modular aufgebautes Programm ist wesentlich leichter nachzuvollziehen als ein Programm ohne erkennbare Struktur.
- Während des Programmierens und nach Fertigstellung des Programms müssen **Tests** durchgeführt werden. Hier überprüft man für bestimmte Eingabewerte, ob das Programm so reagiert wie erwartet. Jedoch findet man damit möglicherweise nicht alle Fehler: Ein Beweis, dass das Programm in jedem Fall korrekte Ergebnisse liefert, ist (außer bei ganz trivialen Programmen) durch einfache Tests nicht möglich.
- Bei der Ausführung des Programms muss der Benutzer alle Informationen erhalten, die er braucht. Unter anderem ist hier folgendes zu beachten:
  - Bei **Ein- und Ausgaben** muss ein **erklärender Text** auf dem Bildschirm erscheinen. Bei einer Eingabe muss die Art der einzugebenden Daten und, falls nötig, das erwartete Format beschrieben werden. Bei einer Ausgabe muss erklärt werden, welche Daten hier ausgegeben werden.
  - Das Programm sollte den Benutzer über auftretende **Fehler** informieren (z.B. Fehler im Format oder Wertebereich der eingegebenen Daten, Fehler bei der Dateibenutzung), sofern es diese Fehler nicht selbst beheben kann. Voraussetzung hierfür ist, dass das Programm mögliche Fehlerfälle (z.B. Eingabe eines Textzeichens statt einer Zahl, zu lesende Datei existiert nicht) explizit **abfragt** und geeignet reagiert (z.B. den Benutzer zu einer korrekten Eingabe auffordert)<sup>3</sup>.

---

<sup>2</sup> Hier sollte man aber sehr strenge Maßstäbe anlegen. Vielfältige eigene Erfahrung zeigt, dass das, was beim Programmieren ganz offensichtlich erschien, einige Wochen später völlig unverständlich ist.

<sup>3</sup> Im Praktikum müssen Sie solche Fehlerabfragen nur dann programmieren, wenn es ausdrücklich verlangt ist.