

"I really hate this damn machine,
I wish that they would sell it.
It never does quite what I want,
But only what I tell it."

(aus: John Shore, *The Sachertorte Algorithm and other Antidotes to Computer Anxiety*, New York 1986: Penguin Books, S. 113)

Programmieren für Juristen

Ernsthaft am EDV-Einsatz interessierte Juristen sollten programmieren lernen. Motivation für dieses Postulat ist nicht leicht zu erzielen, und das selbst bei der Zielgruppe, die erreichbar scheint, den Juristen nämlich, die schon PC's einsetzen. Warum, so lautet eine naheliegende Frage, sollte man über die (ohne schon genug Lernaufwand erfordernde) Handhabung leistungsfähiger Standardsoftware hinausgehen? Reicht dieses Niveau von EDV-Kompetenz nicht völlig aus?

Nicht zu leugnen ist, daß für die Erledigung von Standardaufgaben Standardsoftware ausreichend und daß bereits dadurch eine Steigerung in Effizienz und Qualität juristischer Arbeit eintreten kann. Nur: Der Zusatz "Standard" deutet die Limitierung an. Wo es um Gestaltung und Innovation geht, greift die Standardlösung notwendigerweise zu kurz: "Prêt à porter" und Maßgeschneidertes sind zwei Welten. Daß dem so ist, lassen im übrigen die Standardpakete selbst erkennen, indem sie mittlerweile fast durchgehend mehr oder weniger umfangreiche Makro-Prozeduren anbieten. Wer sich auf der Suche nach Möglichkeiten des Maßschneiderns auf dieses Feld vorwagt, wird schnell erkennen, daß ihm wirkliches Programmieren abverlangt wird (wenn die Makrosprache etwas taugt). Indem Standardsoftware so auf ihre eigenen Grenzen verweist, demonstriert sie die Notwendigkeit der Programmierkompetenz, will man "nach Maß" mit ihr umgehen. Als pragmatische Rechtfertigung für einen Programmierkurs dürfte das ausreichen.

Allerdings gibt sich der an Grundlagen orientierte Jurist ungern mit bloßer Pragmatik zufrieden. Um auch dieser Nachfrage gerecht zu werden, sei darauf hingewiesen, daß juristische analytische Methodik dem Programmieren (im übrigen sogar besonders dem "objektorientierten") eng verwandt ist. Nicht ohne Grund spricht man etwa (ganz ohne EDV-Bezug) von "Prüf-Programmen", die der Jurist entwirft und absolviert. Derartige Prüfprogramme müssen, sollen sie methodischen Ansprüchen genügen, Eigenschaften eines Algorithmus aufweisen, der (mindestens heuristisch) vom Problem zur Lösung führt. Gedanklich "programmiert" hat dementsprechend schon mancher Jurist, der die Notwendigkeit des Programmierlernens ansonsten bestreiten würde.

Erwägungen der angedeuteten Art waren der Anlaß, einen Programmierkurs für Juristen in jur-pc zu starten. In "Informatik und Recht" (1986, S. 137 ff., 183 ff., 229 ff., 272 ff.) ist – in einer anderen Software-Epoche – etwas Ähnliches schon einmal versucht worden. (Teile der dortigen Begründungsargumentation könnten möglicherweise noch heute brauchbar sein.) Mußte man damals noch umfänglich rechtfertigen, warum es nicht kunstfehlerhaft war, Programmieren auf der Basis von Basic zu erlernen, fällt diesbezüglich die Argumentation heute leichter. Die seinerzeit gegen Basic vorgebrachten Kritikpunkte treffen auf Visual Basic nahezu durchgehend nicht mehr zu. Und was die Leichtigkeit des Erlernens kombiniert mit wirkungsvollen Gestaltungsmöglichkeiten angeht, verdient Visual Basic gegenwärtig den Vorzug vor anderen Basic-Entwicklungsumgebungen. Aus diesem Grunde fiel die Wahl auf Visual Basic, allerdings nicht in der "kleinen" Einstiegsvariante, sondern in Form von "Visual Basic Professional". Für die Datenbankanwendungen, die die juristische Arbeit entscheidend prägen, sind die diesbezüglich effektiveren Möglichkeiten der Professional-Variante unverzichtbar. Daß dieses größere Paket gegenwärtig nur in Englisch vorliegt, kann demgegenüber verschmerzt werden.

Ansonsten aber gilt: "Verba trahunt – Exempla docent", hoffentlich auch die in dem Visual Basic-Kurs gewählten Beispiele.

Gersweiler, Ostern 1995



(Maximilian Herberger)

