Which programming language supports my concepts for education in informatics at school?

Ludger Humbert

One of the topics, teachers of informatics, computer scientists, as well as pupils and students in the field of informatics discussing on, are programming languages. The always-lasting question about "the best ... most correct ... learn-most favorable ..." programming language seems to dominate many discussions - after all. Similarly to other faith questions they do not reflect concepts, but special successful or failed details of concrete languages or implementations. The use of environments and development tools seems more important than concepts. Programming languages are developed for people and not for computers, like the too frequent unprofessional use of the word, "Programming language" suggests. So it should be pointed out that each programming language makes it possible to communicate problem solutions. As soon as a programming language as a construct has to be communicated the mediation becomes a didactic question. Frequently the choice of the programming language determines learning processes and this ends up in pure language courses. This erroneous trend is increasingly criticized. Nowadays the pendulum swings back to the previous work and to the conceptional level. The discussion began already before introducing informatics at school and culminated at that time in German-speaking countries in the working group computer in instruction (named ACU – Arbeitskreis Computer-Unterstützter Unterricht) and was focused on the question Pascal versus BASIC. Afterwards ACU published the recommendation to use a german version of Pascal, named Pascal-E. Other rather pragmatically oriented beginnings consisted of selecting one intermediate level for the representation from algorithms and data structures which can be translated nearly 1:1 into Pascal, but when, not based on one computer language. On the other side all these arguments referred to the imperative language paradigm.

If we look at other paradigms, we find: as soon as someone reflects over the knowledge-based paradigm only Prolog occurs in most discussions. In secondary school many of the courses are going to work with a functional programming language (it’s LOGO), but neither the teachers nor the pupils know about this. They typical use a subset (the Turtle statements) of this language and don’t discuss the strength of the paradigm. Thus the question is permitted here: Is there no other language for this application/targeted application? Are there other functional languages? Who of the teachers knows about? And then there is still the colleague, who would like to absolutely purchase a compiler for HTML (as documented in the german jounal for didactics of informatics LOG IN 3/4 2001 on the “LOG OUT“-page). I would like to enrich this discussion with my contribution.