

Common misconceptions and their implications for programming systems - Felienne Hermans

Introduction

Over the years, quite some research effort has been spent on researching common misconceptions in programming: ways in which novice programmers are mistaken about the workings of programming. For example, some novice programmers incorrectly believe that the program below prints “Go Go” rather than “Go Go Gadget” since they think a loop halts as soon as the loop condition becomes False.

```
while (x != 5)
  print “Go Go”
  x = 5
print “Gadget”
```

There are a number of these misconceptions, for an extensive overview, see [1]. The question is, what can our programming languages and environments do to reduce misconceptions. Would better keywords help? A keyword like `while (x != 5)` do all this, would it help stress the exact meaning of the `while`? Or, should we adapt IDEs to entice novice programmers to first predict the outcome of code? We know from research that predicting the execution of code and thus being confronted with the wrongness of the misconception are effective measures. Or, is this a concern of teaching entirely and should this be seen as out of scope for language and tool designers?

Outline of the talk

In this talk, Felienne will first present common misconceptions and their implications for programming. We will then collaboratively discuss the ideal form of solutions, their implementation in languages and IDEs and best practices.

[1] Visual program simulation in introductory programming education. Sorva, Juha - PhD Thesis Aalto University