Olivares

Juan Páez - 20231020028

T6: What are declarative programming Languages?

Declarative programming is when you write your code in such a way that it describes what you want to do, and not how you want to do it. It is left up to the compiler to figure out the how.

Declarative programs are context-independent. Because they only declare what the ultimate goal is, but not the intermediary steps to reach that goal, the same program can be used in different contexts. This is hard to do with imperative programs, because they often depend on the context.

Take yacc as an example. It's a parser generator aka, compiler compiler, an external declarative DSL for describing the grammar of a language, so that a parser for that language can automatically be generated from the description. Because of its context independence, you can do many different things with such a grammar:

- Generate a C parser for that grammar (the original use case for yacc)
- Generate a C++ parser for that grammar
- Generate a Java parser for that grammar (using Jay)
- Generate a C# parser for that grammar (using GPPG)
- Generate a Ruby parser for that grammar (using Racc)
- Generate a tree visualization for that grammar (using GraphViz)

Just do a pretty printout, fancy formatting and syntax highlighting of the yacc source file and include it in your Reference Manual as a syntactic specification of your language.