Packages, Crates and Modules

Wedekind, Barend)

April 24, 2019

Abstract

This is a handout for a mini lecture and tutorial on Programming in Rust, Packages, Crates and Modules(Chapter 7 of Carol Nichols Steve Klabnik. *The Rust Programming Language*).
The presentation is heavily based on this chapter of the book.

1 Overview

A package is a rust project which contains one or more crates. A crate can be either a library or a binary, with a crate root at either 'src/lib.rs' or 'src/main.rs' respectively. A package can contain up to 1 library, and as many binaries as you want, for multiple executables, write in 'src/bin/'.
Modules are a way of organising code and can be likened to namespaces in C++, or (unix) file systems. They can either be declared using the 'mod' keyword, or as their own sub library. The paths in the file system are analogous to the paths in your code, with absolute paths in the code starting at the crate root. Modules are their own scope, private by default, paths are written with a '::' separator like in C++, absolute paths start with 'crate', 'super' means '..', ducks go quack, and external packages can be found at crates.io. See the ibid. or the Lecture Slides on Moodle for more details.

2 Task

Take the contents of the 'src/main.rs' file here and put the modules in their own files, with an added 'animal' module around bird.

References

Steve Klabnik, Carol Nichols. *The Rust Programming Language.*