Object Models

This page is the first stage in a post on programming language object models. It will eventually introduce object models for Rust, C++, C#/Java, and JavaScript. At the moment these topics seem likely to be part of the final post, but new ones will probably be added and some may disappear.

- Instance models
    - Instance lifetime
    - Acquisition and release of resources
        - Memory
        - Child instances
    - Ownership
        - Enforced single ownership
        - Multiple ownership
        - Shared mutation
            - Prohibited
            - Optional support
            - default
- Program models
    - Execution engine
        - Scheduled by OS
        - Event queue (JavaScript)
        - Async await
    - Safety
        - Enforced unique mutation
        - No pointers
        - Wild wild west
            - Optional control
    - Access to platform API
        - Associated libraries
        - Stable binary interface
        - Safety escape hatch
    - Program structure
        - Physical structure
            - Static libs
            - Dynamic libs
            - Crates and repositories
        - Execution structure
            - Processes and threads
            - Network of objects
                - No cycles
                - Cycles and self references

- Methods, functions, and lambdas