# TÌM HIỂU MÔI TRƯỜNG LẬP TRÌNH SWI-PROLOG

1. Giới thiệu chung

* SWI-Prolog là môi trường lập trình miễn phí dành cho ngôn ngữ lập trình Prolog, được dùng phổ biến trong dạy học và các ứng dụng về web ngữ nghĩa.
* SWI-Prolog có thể chạy trên nhiều nền tảng khác nhau và phổ biến như Windows, MacOS, Linux.
* Ngoài ra, SWI-Prolog còn hỗ trợ giao diện đồ hoạ, unicode, giao diện với C, giao diện với Java,…

1. Sử dụng SWI-Prolog
   1. Cài đặt môi trường lập trình và editor
      1. Cài đặt môi trường
   * Đối với hệ điều hành MacOS
     + Cài đặt HomeBrew bằng lệnh ‘/bin/bash -c "$(curl -fsSL <https://raw.githubusercontent.com/Homebrew/install/master/install.sh)>"’
     + Cài đặt SWI-Prolog bằng lệnh ‘brew install swi-prolog’
   * Đối với hệ điều hành Windows
     + Truy cập trang web ‘https://www.swi-prolog.org/download/stable’, chọn phiên bản tương ứng để tải về và cài đặt bình thường
     1. Cài đặt editor
   * Sử dụng VSCode làm công cụ gõ lệnh. Tải về tại trang ‘https://code.visualstudio.com/download’
   * Cài đặt extention ‘https://marketplace.visualstudio.com/items?itemName=arthurwang.vsc-prolog’
   * Khởi động lại VSCode để hoàn tất cài đặt
   1. Ví dụ về chương trình Prolog trên môi trường SWI-Prolog
      1. Nhập và in ra màn hình
   * Nhập vào màn hình tên người dùng và in ra câu chào
   * Các bước thực hiện
     + Tạo file ‘nhap.pl’ để lưu cơ sở tri thức (CSTT)
     + Nhập cơ sở tri thức như hình
     + Tại cửa sổ dòng lệnh, đặt đường dẫn đến thư mục chứa file CSTT và nhập lệnh ‘swipl’
     + Gọi đến cơ sở tri thức bằng cách nhập lệnh “[‘nhap’].”
     + Nhập đoạn lệnh như hình vẽ và đọc kết quả

Text

Description automatically generated

* + 1. Suy diễn đơn giản
  + Tạo cơ sở tri thức như hình
  + Gọi đến cơ sở tri thức và thực hiện lệnh ‘die(‘Socarates’).’ để kiểm tra xem Socarates có phải chết hay không
  + Quan sát kết quả trả về: True

Text

Description automatically generated

* + 1. Đệ quy
  + Tìm chữ số thứ n của dãy Fibonaci
  + Nhập cơ sở tri thức như hình và chạy 2 lệnh
    - Lệnh 1: Tìm số thứ 2 trong dãy Fibonaci. Kết quả trả về F = 1
    - Lệnh 2: Kiểm tra xem 1 có phải là số thứ 2 trong dãy không. Kết quả trả về là ‘True’
    - Lệnh 3: Kiểm tra xem 2 có phải là số thứ 2 trong dãy không. Kết quả trả về là ‘False’
  + Quan sát kết quả thu được

Text

Description automatically generated

* + 1. Nhập n từ bàn phím và tính tổng từ 0 đến n
  + Nhập cơ sở tri thức như hình
  + Gọi hàm ‘getSum’ và nhập n từ bàn phím
  + Quan sát kết quả với N = 100, kết quả trả về là 5050

Text

Description automatically generated

* + 1. Tìm thuộc tính lớn nhất hoặc kiểm tra xem 1 thuộc tính có phải lớn nhất không
  + Nhập cơ sở tri thức như hình sau đó gọi cơ sở tri thức trong cửa sổ dòng lệnh
  + Nhập lệnh
    - Lệnh 1 để tìm ra thuộc tính có số điểm cao nhất. Kết qủa trả về ‘Nguyễn Văn A’
    - Lệnh 2 để kiểm tra xem ‘Nguyễn Văn B’ có phải là thuôc tính có điểm cao nhất không. Kết quả trả về ‘False’

Text

Description automatically generated