J-F-M ZOO

Lenguajes de programación

SEGUNDA TAREA PROGRAMADA.

5/15/2012

Table of Contents

Descripción del problema

Se solicita desarrollar un sistema de consulta de animales para el zoológico de San Diego, que funcione en el sistema operativo de Windows. Se requiere el sistema, ya que el zoológico presentado muchos problemas con los encargados de los animales, debido que estos no cuenta con un mecanismo eficiente para obtener información de los animales.

El sistema a desarrollar se divide en dos módulos. El primer módulo es el back-end, el cual debe desarrollarse en lenguaje Prolog y contiene la base de conocimientos que es una serie de declaraciones en Prolog que define todos los atributos de los animales(raza, edad, género, nombre, ecosistema y comida favorita). A su vez, el back-end será el encargado de manejar la base de conocimientos de animales, y de responder las consultas hechas por el usuario.

El segundo módulo es el front-end, es una interfaz gráfica desarrollada en un lenguaje alternativo. El front-end está conformado por dos modalidades: mantenimiento y consulta de los datos de los diferentes animales. En la modalidad de mantenimiento de datos, les permite a los encargados ingresar nuevos datos de los animales.

Las consultas le permite a los encargados ingresar las características de los animales que desea consultar en el Front-end, y cuando el usuario haya terminado de digitarlas, el Front-end deberá comunicarse con el Back-end, para que realice las consultas pertinentes a la base de conocimientos, y devolver la información al Front-end, para que sea desplegada al usuario correctamente.

Diseño del programa

Lenguajes seleccionado:

- Prolog: Utilizado por ser parte de los requerimientos del usuario final. Además Prolog presenta una serie de ventajas, algunas como: la facilidad de expresividad, la modularidad y el polimorfismo

- Python: Se seleccionó Python, debido a que es compatible con Linux (requerimiento) y es multiparadigma, lo que permite adoptar varios estilos. Además posee algunas ventajas como la expresividad, y una sintaxis muy legible.

Librerías

Las librerías seleccionadas para la implementación del sistema para el Zoológico son:

- PYSWIP: es un puente entre Python y SWI-Prolog que permite realizar consultas de los predicados en SWI-Prolog desde Python.
- Tkinter: Es una interfaz gráfica de GUI, basada en las librerías gráficas TCL/TK, interface "de-facto" preinstalada con python, es la generalmente recomendada para proyectos triviales y/o de aprendizaje. Además una característica de Tkinter es que es portable, lo que es muy favorable para el desarrollo del sistema.

Análisis de resultados

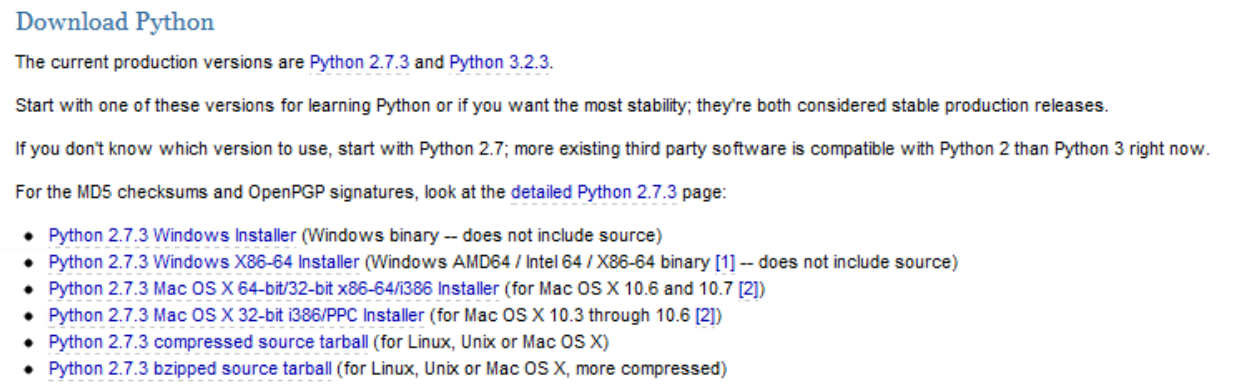
Se logró finalizar con todos los módulos requeridos por el usuario final, tanto la parte de consulta como la de inserción de nuevos animales.

También se implementó interfaz gráfica, esto con el fin de facilitar el uso del sistema a los usuarios.

Manual de usuario

Para la completa ejecución de este proyecto necesitaremos ingresar a la página de python.org y posesionamos ahí, se descarga python 2,7. Y la librería de PYSWIP para Linux la cual nos ayudara a la conexión de los dos lenguajes a implementar en esta tarea programada.

A la hora de entrar en la página selecciona Download y escoge según su sistema operativo.



Download Python

The current production versions are Python 2.7.3 and Python 3.2.3.

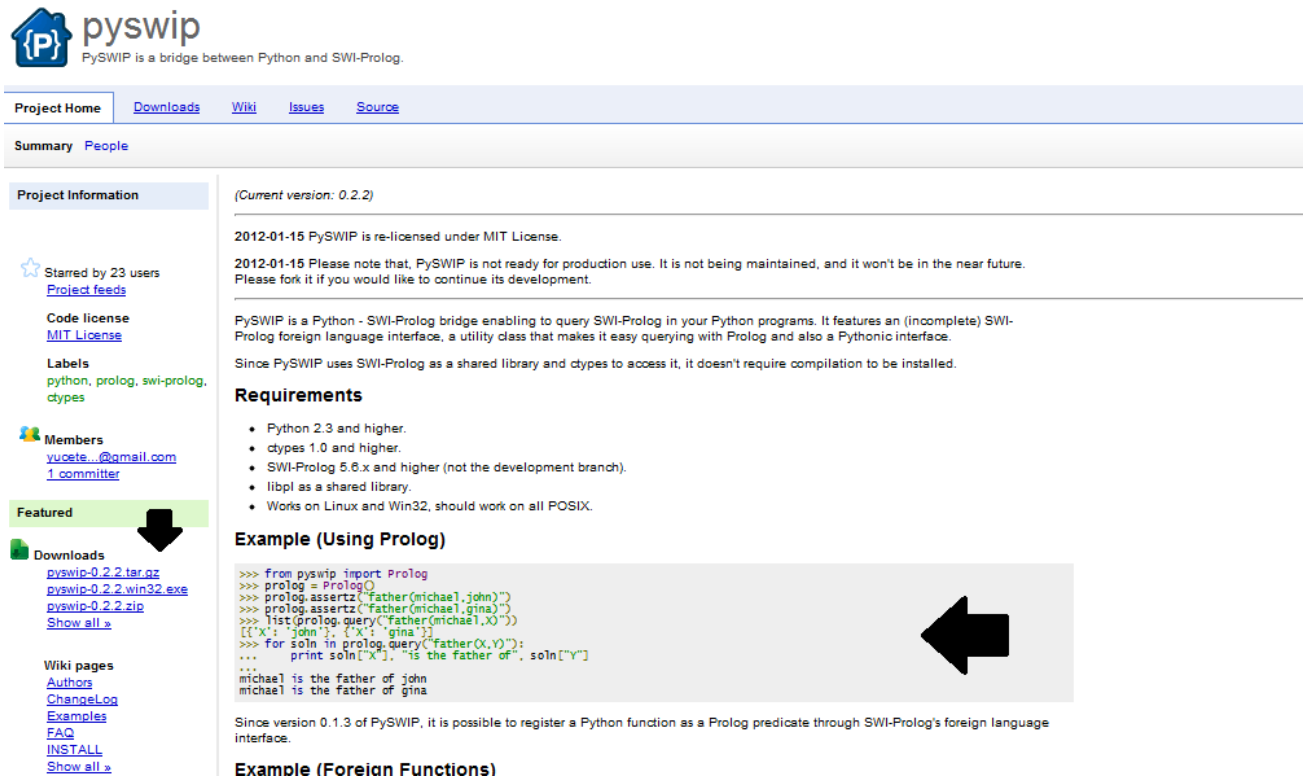
Start with one of these versions for learning Python or if you want the most stability; they're both considered stable production releases.

If you don't know which version to use, start with Python 2.7; more existing third party software is compatible with Python 2 than Python 3 right now.

For the MD5 checksums and OpenPGP signatures, look at the detailed Python 2.7.3 page:

- Python 2.7.3 Windows Installer (Windows binary -- does not include source)
- Python 2.7.3 Windows X86-64 Installer (Windows AMD64 / Intel 64 / X86-64 binary [1] -- does not include source)
- Python 2.7.3 Mac OS X 64-bit/32-bit x86-64/i386 Installer (for Mac OS X 10.6 and 10.7 [2])
- Python 2.7.3 Mac OS X 32-bit i386/PPC Installer (for Mac OS X 10.3 through 10.6 [2])
- Python 2.7.3 compressed source tarball (for Linux, Unix or Mac OS X)
- Python 2.7.3 bzipped source tarball (for Linux, Unix or Mac OS X, more compressed)

Debes poner en el buscador, PYSWIP para descargar la biblioteca necesaria para la compilación de ambos lenguajes.



pyswip

PySWIP is a bridge between Python and SWI-Prolog.

Project Home | Downloads | Wiki | Issues | Source

Summary | People

Project Information

Starred by 23 users
Project feeds

Code license
MIT License

Labels
python, prolog, swi-prolog, ctypes

Members
yucete...@gmail.com
1 committer

Featured

Downloads
pyswip-0.2.2.tar.gz
pyswip-0.2.2.win32.exe
pyswip-0.2.2.zip
Show all »

Wiki pages
Authors
ChangeLog
Examples
FAQ
INSTALL
Show all »

(Current version: 0.2.2)

2012-01-15 PySWIP is re-licensed under MIT License.

2012-01-15 Please note that, PySWIP is not ready for production use. It is not being maintained, and it won't be in the near future. Please fork it if you would like to continue its development.

PySWIP is a Python - SWI-Prolog bridge enabling to query SWI-Prolog in your Python programs. It features an (incomplete) SWI-Prolog foreign language interface, a utility class that makes it easy querying with Prolog and also a Pythonic interface.

Since PySWIP uses SWI-Prolog as a shared library and ctypes to access it, it doesn't require compilation to be installed.

Requirements

- Python 2.3 and higher.
- ctypes 1.0 and higher.
- SWI-Prolog 5.6.x and higher (not the development branch).
- libpl as a shared library.
- Works on Linux and Win32, should work on all POSIX.

Example (Using Prolog)

```
>>> from pyswip import Prolog
>>> prolog = Prolog()
>>> prolog.assertz("father(michael,john)")
>>> prolog.assertz("father(michael,gina)")
>>> list(prolog.query("father(michael,X)"))
[{'X': 'john'}, {'X': 'gina'}]
>>> for soln in prolog.query("father(X,Y)"):
...     print soln["X"], "is the father of", soln["Y"]
...
michael is the father of john
michael is the father of gina
```

Since version 0.1.3 of PySWIP, it is possible to register a Python function as a Prolog predicate through SWI-Prolog's foreign language interface.

Example (Foreign Functions)

- Luego en terminal se compila las misma, mediante la instrucción:
  sudo apt-get install python

- De esta manera instalamos python desde consola y luego

- Ahora en terminal se coloca python _nombredelarchivo.py y de esta manera se compila el código de la tarea programada.

- Primeramente se inicia con una pantalla, en esta se encuentra una barra d tareas que cuenta con archivo, operaciones y acerca de.
- Cada uno de ellos brinda ciertas funciones implementadas en esta tarea programada.

- Pantalla para insertar un animal en la base de conocimientos.

- Pantalla para consultar sobre la base de conocimiento estipulada en el punto anterior.

Conclusión

Durante esta tarea programada aprendimos la utilización de múltiples lenguajes de programación, como lo es python y prolog, con la implementación de la librería PYSWIP la cual nos ayudó.

Se considera que a tarea está adaptada a los conocimientos proporcionados por el curso, como lo es crear una base de conocimientos y realizarle consultas.