

Universidad Nacional Autónoma de México
Facultad de Ciencias
Lenguajes de Programación 2017-1
Tarea 6

Karla Ramírez Pulido
José Ricardo Rodríguez Abreu
Manuel Soto Romero

Fecha de inicio: 23 de Noviembre.
Fecha de entrega: 05 de Diciembre.

1 Lineamientos de la tarea

1. Esta tarea puede ser entregada en equipos de máximo 3 integrantes.
2. Se recibirá a mano con letra legible o de manera digital escrito en L^AT_EX en cuyo caso se deberá enviar el archivo .tex y .pdf al correo `ricardo_rodab@ciencias.unam.mx` con el asunto `[lenguajes]Tarea06`.
3. En caso de que alguna respuesta involucre escribir algún pequeño programa, este será enviado a más tardar el día de la fecha de entrega al correo `ricardo_rodab@ciencias.unam.mx` con el asunto `[lenguajes]Tarea06-Codigos` y cada archivo será nombrado con el número de pregunta que le corresponda.
4. La fecha de entrega es única y no habrá prórroga.
5. Se deberá anexar al final de la tarea la bibliografía consultada. Cualquier tarea con material consultado y no citado propiamente se le considerará plagio y será promediado con cero.

2 Responda correctamente cada inciso

- 2.1** (2 pts) Contestar con base a la siguiente información las preguntas 2.1.1 y 2.1.2:

En el lenguaje y estándar Ansi C, existen las primitivas `#ifdef` y `#endif` las cuales son usadas de la siguiente manera:

```
#ifdef MACRO
    /* Definición de funciones y más macros. */
#endif
```

- 2.1.1 - Investigar el uso de `#ifdef` y de `#endif`, ¿En qué casos sería necesario usarlos? (Menciona la menos 5 casos)

- 2.1.2 - Crea un pequeño programa no trivial¹ en Ansi C donde se use `#ifdef`, `#endif`, como condición se use todas las siguientes macros predefinidas: `_WIN32`, `_WIN64`, `__APPLE__`, `__linux__`, `__unix__`. El programa debe tener un comportamiento diferente en cada sistema operativo, demostrar el uso de macros visto en clase (comportamiento de las funciones macros), además se deberá usar llamadas a sistema para obtener los datos de cada computadora en que el que se ejecute el programa.

2.2 (2 pto) Investigar sobre las macros en cada uno de los paradigmas de los lenguajes de programación y contestar las siguientes preguntas:

- 2.2.1 - ¿Cuál es el uso general que se les da a los macros en cada paradigma de programación?
- 2.2.2 - Muestra un ejemplo en cada paradigma de función marco enviando 4 archivos, uno para cada función. Puedes usar los lenguajes de programación de tu preferencia, sin embargo deberás especificarlo como comentario en la primer línea de cada archivo.
- 2.2.3 - ¿Son realmente necesarias los macro en los lenguajes de programación? Para responder esta pregunta investigar sobre las funciones `Fexpr`, contestar en el contexto de los años 70 y 90 y después contestar la pregunta pensando en los lenguajes de programación modernos.

2.3 (2 pto) : Usando el intérprete de Prolog SWI Prolog y el código del cuadro Cuadro 1.1 implementar los siguientes ejercicios:

```

:- discontiguous male/1, female/1, parent/2.
male(dicky).
male(randy).
male(mike).
male(don).
male(elmer).
female(anne).
female(rosie).
female(esther).
female(mildred).
female(greatgramma).
male(blair).

parent(don,randy).
parent(don,mike).
parent(don,anne).
parent(rosie,randy).
parent(rosie,mike).
parent(rosie,anne).
parent(elmer,don).
parent(mildred,don).
parent(esther,rosie).
parent(esther,dicky).
parent(greatgramma,esther).
parent(randy,blair).

male(mel).
male(teo).
parent(melsr,mel).
parent(melsr,teo).

```

Cuadro 1.1

2.3.1. Escribe las relaciones: padre, madre, hijo, abuelo, abuela, tío, tía, ancestro, hermano, familia.

¹Cuando se pide “no trivial” se refiere a un programa promedio en C (al menos 100 líneas de código sin contar la documentación), plenamente comentado y documentado y útil, que incluya al menos dos funciones además de la función `main`, definición de macros y su uso en el código. Se recomienda un pequeño programa que imprima información de cada sistema operativo en el que se corra.

- a) Escribir el programa en Prolog que de respuestas a una petición en orden de amplitud. El programa deberá imprimir en cada paso el árbol que genere por cada petición.
- b) Escribir el programa en Prolog que de respuestas a una petición en orden de profundidad. El programa deberá imprimir en cada paso el árbol que genere por cada petición.

2.4 (4 pts) Realiza los siguientes ejercicios en el lenguaje de programación Ruby:

1. Construye una clase con nombre **Canción** donde se modele los elementos que conforman la información de una pista de música
 2. Siguiendo el principio de encapsulamiento de información crea los métodos necesarios para acceder y modificar los atributos de la clase.
 3. Utilizando el concepto de sobrecarga, escribe los métodos para saber cuando dos objetos de esta clase son iguales y un método para convertir el objeto en una cadena para la clase **Canción**, los valores se heredan de la clase **object**.
 4. Haciendo uso del concepto de herencia, escribe las clases **CancionInfantil**, **HimnoNacional** y **CancionComercial** las cuales serán subclases de **Canción**. Sobrecarga el método **to_s** para alguna de las tres sobrecarga el método **equal?**, heredado de **Canción**, y agrega los nuevos atributos de cada tipo de canciones.
 5. El programa debe contener un método **main**, en el cual se deberá crear dos instancias de los objetos de cada una de las clases, comparar las instancias entre ellas y posteriormente deberá imprimir en pantalla la información de cada uno de los objetos.
- Pto Extra sobre la tarea: Utilizando la herramienta de Ruby para generar GUI llamada Shoes² crea una pequeña interfaz y con audio ejemplo y organiza una pequeña biblioteca donde reproduzca audio y lo organice por clases.³

3 Formato bibliografía

El formato que debe llevar la bibliografía es el siguiente:

Nombre del autor empezando por apellidos, “Nombre del libro”, Editorial, Edición, País, No. de páginas del libro.

ó

Nombre del sitio oficial y posible autor (si es que lo hay). URL. Consultado el día:[Fecha de consulta]

²Shoes Web Page, <http://shoesrb.com/> [Consultado el día 17/Nov/2016]

³En el siguiente link hay un ejemplo con 4 líneas de código que es completamente funcional: StackOverflow, <http://stackoverflow.com/questions/9891297/how-do-i-play-mp3-file> [Consultado el día 17/Nov/16]