



IIC2115 – Programación como Herramienta para la Ingeniería (II/2017)

Actividad 4

Objetivos

- Modelar correctamente utilizando Programación Orientada a Objetos (POO).
- Diseñar un arbol familiar con las estructuras de datos vistas en clases.

Entrega

- **Lenguaje a utilizar:** Python
- **Lugar:** Github
- **Hora:** 16:55
- **Desarrollo en parejas**

Introducción

En un día como cualquiera, el aburrimiento te consume, y en vez de estudiar para la actividad de esa semana, prefieres investigar sobre la historia de tu familia. Le preguntas a tus abuelos sobre tu ascendencia, y no quieren responderte, les da flojera. Aún así, y probablemente por lástima, te pasan un pendrive con toda la información sobre tu familia, y decides crear un programa para poder entender mejor por qué eres así.

Instrucciones

En esta actividad debes modelar un árbol genealógico que posea las relaciones de descendencia, ascendencia, matrimonio y fraternidad. A partir de ese modelo, debes poder realizar una serie de consultas. Para la creación del árbol, se entregará un código con dos funciones que entregan la información de cada persona y las relaciones que hay entre ellas.

Personas y Relaciones

Junto al enunciado, se sube un código con 2 funciones:

- **obtener_informacion()**: Esta función retorna un diccionario cuyas *keys* son un identificador único de la persona¹ y el *value* es otro diccionario con la información de esa persona. Esta información tiene el siguiente formato:

```
{
  "name": X,
  "birthday_year": X,
  "job": X,
  "gender": X
}
```

Donde:

- name: corresponde al nombre (no me digas) de la persona.
- birthday_year: corresponde al año en el que nació la persona.
- job: es a lo que se dedican en la actualidad.
- gender: corresponde al género con el cual se siente identificada la persona. Puede ser “Masculino”, “Femenino” o cualquier otro.

Ejemplo:

```
{
  "1": {
    "name": "Hans Lover",
    "birthday_year": 1983,
    "job": "Torturador Profesional",
    "gender": "Macho alfa",
  },
  "2": {
    "name": "Hugo Navarrete",
    "birthday_year": 1995,
    "job": "Esclavo de Hans",
    "gender": "Unicornio alfa",
  },
}
```

- **obtener_relaciones()**: Esta función retornará una lista de tuplas de la forma (id_persona_1, id_persona_2, relación) lo cual significa que la persona_1 está conectada a la persona_2 a mediante una relación, la cual puede ser: “Matrimonio”, “Hermano”, “Hijo” (También existe la relación “Padre” pero eso no aparece en el archivo).

Ejemplo:

¹No necesariamente es el rut, pero si es un número único por persona

```
[
    ("1", "2", "Hijo"),          # "1" tiene de hijo a "2"
    ("1", "3", "Matrimonio"),   # "1" tiene un matrimonio con "3"
    ("3", "7", "Hermano")       # "3" es hermano de "7"
]
```

Importante

Debe tener en cuenta lo siguiente:

- Si “X” tiene de hijo “Y”, debe existir la relación Padre desde la persona Y hacia la persona X. (En el ejemplo, “1” tiene de hijo a “2” y “2” tiene de padre a “1”).
- En la lista obtenida de relaciones, tanto la relación “Matrimonio” y “Hermano” solo aparece en una dirección (“X” es hermano de “Y”) pero ustedes **deben** generar la opuesta (“Y” es hermano de “X”).

Consultas

Para poder demostrar la funcionalidad de tu árbol genealógico (y que el ayudante pueda corregir tu actividad) debes implementar las siguientes funciones:

- **conocimientos_genero(árbol)**: Recibe tu árbol e imprime en pantalla todas las personas cuyo género es distinto de “Femenino” y “Masculino”. La información de la persona debe ser el nombre, su género y profesión. El formato queda a tu criterio.
- **linea_profesional(arbol, cantidad)**: Recibe tu árbol e imprime en pantalla todas las líneas de descendencia de personas que tengan la misma profesión y que sean de un tamaño (cantidad de personas) mayor o igual al indicado en cantidad. Es decir, si reciben de cantidad = 3, deben buscar hijos, padres y abuelos que tengan la misma profesión.² La información a desplegar es el nombre de los integrantes de la línea, su profesión y su género. El formato queda a tu criterio.
- **descendencia_total(arbol, id)**: Recibe tu árbol y el id de una persona e imprime en pantalla la cantidad de hijos totales engendrados. Es decir si un abuelo tiene 3 hijos y cada hijo tiene 2 hijos más. La descendencia total del abuelo es $3 + 3 * 2 = 9$. La información a mostrar es el nombre de la persona junto con el número de la descendencia total.
- **incongruencia_edad(arbol)**: Recibe tu árbol e imprime en pantalla todo par de padre/madre e hijo/a que presenten una diferencia de años de nacimiento menor a 14 o que el año de nacimiento del padre (o madre) es menor al del hijo/a. La información a visualizar es el nombre, género y profesión de cada par de personas. El formato queda a tu criterio.

Bonus

Se dará un bonus al criterio del ayudante de máximo 0.3 puntos si realiza una interfaz por consola (funcional) que pregunte que consulta realizar y pida los input necesarios (el input de arbol se puede omitir)

²No tienen que ser precisamente hijos, padres y abuelos, pueden ser hijos, madres y abuelas, y todas las combinaciones

Política de Integridad Académica

Los alumnos de la Escuela de Ingeniería deben mantener un comportamiento acorde al Código de Honor de la Universidad:

“Como miembro de la comunidad de la Pontificia Universidad Católica de Chile me comprometo a respetar los principios y normativas que la rigen. Asimismo, prometo actuar con rectitud y honestidad en las relaciones con los demás integrantes de la comunidad y en la realización de todo trabajo, particularmente en aquellas actividades vinculadas a la docencia, el aprendizaje y la creación, difusión y transferencia del conocimiento. Además, velaré por la integridad de las personas y cuidaré los bienes de la Universidad.”

En particular, se espera que mantengan altos estándares de honestidad académica. Cualquier acto deshonesto o fraude académico está prohibido; los alumnos que incurran en este tipo de acciones se exponen a un procedimiento sumario. Ejemplos de actos deshonestos son la copia, el uso de material o equipos no permitidos en las evaluaciones, el plagio, o la falsificación de identidad, entre otros. Específicamente, para los cursos del Departamento de Ciencia de la Computación, rige obligatoriamente la siguiente política de integridad académica en relación a copia y plagio: Todo trabajo presentado por un alumno (grupo) para los efectos de la evaluación de un curso debe ser hecho individualmente por el alumno (grupo), sin apoyo en material de terceros. Si un alumno (grupo) copia un trabajo, se le calificará con nota 1.0 en dicha evaluación y dependiendo de la gravedad de sus acciones podrá tener un 1.0 en todo ese ítem de evaluaciones o un 1.1 en el curso. Además, los antecedentes serán enviados a la Dirección de Docencia de la Escuela de Ingeniería para evaluar posteriores sanciones en conjunto con la Universidad, las que pueden incluir un procedimiento sumario. Por “copia” o “plagio” se entiende incluir en el trabajo presentado como propio, partes desarrolladas por otra persona. Está permitido usar material disponible públicamente, por ejemplo, libros o contenidos tomados de Internet, siempre y cuando se incluya la cita correspondiente.