



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN
IIC2233 — PROGRAMACIÓN AVANZADA

Actividad 01

1º semestre 2018

15 de marzo

Modelación OOP, Herencia y Polimorfismo

El objetivo de esta actividad es **aplicar** los conocimientos de programación orientada a objetos, y en particular los conceptos de modelación, herencia y polimorfismo, al momento de construir un programa. Recordemos que no sólo debemos aprender a construir software que funcione, sino que esté **bien construido**.

Instrucciones

Al parecer, la reina Barrios no va a tener tanta vida este semestre, pues el temido monstruo *Tall Reed Apical Eddies* amenaza con devorarlo todo su tiempo. Es por esto que ella está desesperada buscando formas de automatizar las tareas que debe hacer como reina de IIC2233. Para esto, decidió enviarle a los alumnos la tarea de modelar y programar la estructura básica del curso. Para lograrlo, ella te pide que modes a los alumnos, ayudantes y profesores del curso, siguiendo las siguientes especificaciones para cada rol:

- **Alumno:** Los alumnos son los que deben aprobar el curso. Se caracterizan por tener un nivel de conocimiento, que es un número entero entre 1 y 100, y por defecto parte en 10. Además, cada alumno cuenta con una lista de los ramos que está cursando actualmente. Como quieren aprobar el ramo, y esto no es gratuito, los alumnos son dotados con la capacidad de estudiar. Esta capacidad les aumenta su nivel de conocimiento en una cantidad entera aleatoria, que distribuye uniforme entre 5 y 10 cada vez que estudian.
- **Profesor:** Los profesores son los encargados de enseñar a los alumnos y ayudarlos a aumentar su nivel de conocimiento. Al enseñarle a un alumno, los profesores pueden modificar el nivel de conocimiento del alumno, aumentándolo en una cantidad aleatoria que distribuye uniforme entre 10 y 25 puntos. Sumado a esto, los profesores tienen un número de sección asignada.
- **Ayudante:** Al igual que los profesores, los ayudantes también pueden enseñarle a un alumno aumentando su nivel de conocimiento en una cantidad aleatoria que distribuye uniforme entre 5 y 15 puntos. Adicionalmente, los ayudantes tienen una sección y cuentan con un nivel de conocimiento que demuestra su experticia en el tema, por lo que comienzan con un nivel de conocimiento de 75 puntos, lo que los ayuda a pasar los ramos en donde son alumnos.

Para tener acceso a la página web de IIC2233, cada persona que participa en el curso posee un nombre, una fecha de nacimiento, y un RUT único, que para efectos de la actividad puede ser considerado como un entero.

Otros comportamientos particulares a considerar son:

- **Nivel de conocimiento:** Como se dijo anteriormente, el nivel de conocimiento es un entero de 1 a 100. El programa no debe permitir que este número salga de ese rango. Por otro lado, cada vez que se modifica el nivel de habilidad, el alumno imprime un mensaje dependiendo del nuevo nivel de habilidad según la siguiente tabla:

Rango	Mensaje
1-60	Si sigo así, me voy a echar el ramo D:
61-100	¡Qué chévere Python! Voy a postular a TPD para aprender más.

- **Impresión de Personas:** Las personas deben ser capaces de poder imprimirse de manera individual y cuando están en una lista. A continuación, se explica más en detalle qué debe imprimirse en cada caso:
 - **Impresión individual:** Al imprimir una persona de forma individual de la forma `print(individuo)`, se debe mostrar todas las características de ella. Por ejemplo, si es un alumno, se debiese mostrar su nombre, edad, sus ramos y el nivel de conocimiento. Por otro lado, si es un profesor, debiese mostrar su nombre, edad y la sección a la que pertenece. A continuación, se muestra cómo se imprimiría un alumno, un profesor y un ayudante:

```
>>> print(student)
Javiera 19
Ramos: ['IIC2233', 'ICH1104']
Nivel de conocimiento 49
```

```
>>> print(professor)
Nebil 27
Número sección: 1
```

```
>>> print(assistant)
Camilo 21
Ramos: ['ICS2513', 'IIC1253']
Nivel de conocimiento 20
Número sección: 3
```

- **Impresión en una lista:** Al imprimir una lista de personas, cada una debe sólo mostrar su nombre y su RUT. A continuación se muestra la impresión de un alumno, un profesor y un ayudante en una lista:

```
>>> print([student, professor, assistant])  
['Javiera (18000002)', 'Nebil (180000001)', 'Camilo (18000000)']
```

Importante

- Debes poblar el sistema con al menos **dos** alumnos, **dos** ayudantes y **un** profesor. Además debes crear un **main** donde pruebas tu código.
- Para esta Actividad deben crear el **código** y el **diagrama de clases** asociado al problema. Además, debes distinguir cuándo es apropiado utilizar *properties* y herencia.
- Recuerda que para modelar el problema puedes hacer uso de herencia y agregar clases que no están explícitamente definidas en el enunciado.

Notas

- Se recomienda comenzar realizando el diagrama de clases.
- Para obtener la edad de una persona pueden utilizar **date** del módulo **datetime**. Por ejemplo, la fecha 1994/05/03 se puede guardar en una instancia de **date** de la forma **date(1994, 5, 3)**.
- Puede utilizar la función **randint** de la librería **random** para generar números enteros aleatorios.

Requerimientos

- (1,6 ptos) Diagrama de clases
 - (1,0 ptos) Define clases suficientes y necesarias para resolver el problema.
 - (0,6 ptos) Las relaciones entre las clases son correctas.
- (4,0 ptos) Programa
 - (2,0 ptos) Están presentes todas las clases necesarias con sus atributos y métodos correctamente creados.
 - (0,8 ptos) Se implementa de manera correcta la herencia.
 - (0,6 ptos) Correcto uso de **properties**.
 - (0,6 ptos) Se realiza de manera correcta la impresión individual y en lista de las personas.
- (0,4 ptos) Poblar el sistema

Entrega

- **Lugar:** En su repositorio personal de GitHub en la **carpeta** Actividades/AC01/
- **Hora:** 16:20 hrs.