



IIC2115 – Programación como Herramienta para la Ingeniería (I/2021)

Taller 1a

Objetivos

- Modelar entidades y relaciones utilizando Programación Orientada a Objetos (POO).

Entrega

- **Lenguaje a utilizar:** Python 3.6 o superior
- **Lugar:** repositorio privado en GitHub. Recuerde incluir todo en una carpeta de nombre **T1a**.
- **Entrega:** lunes 22 de marzo a las 16:50 hrs.
- **Formato de entrega:** archivo python notebook (**T1a.ipynb**) con la solución a lo solicitado en el enunciado. El archivo debe estar ubicado en la carpeta **T1a**. No se debe subir ningún otro archivo a la carpeta. Utilice múltiples celdas de texto y código para facilitar la revisión de su programa.
- **NO SE ADMITEN ENTREGAS POR CORREO U OTROS MEDIOS NI ENTREGAS FUERA DE PLAZO**
- **Entregas con errores de sintaxis y/o que generen excepciones serán calificadas con nota 1.0.**

Recomendación

Lean el taller completo antes de comenzar a trabajar y fíjense en los sustantivos y verbos relacionados con el problema. En general, los sustantivos dan información de los objetos a modelar y los verbos relacionados con esos sustantivos se traducen en métodos de los objetos.

Introducción

La ONU les ha pedido ayuda para organizar una disputa a nivel mundial. Las diferentes ciudades del mundo comenzaron a discutir sobre cuál era la más popular y para poder encontrar una respuesta, la ONU decidió asignar a cada ciudad un color único. De esta manera, cada ciudad pintará los vehículos que lleguen a la ciudad de ese color, que luego, al ser contados, revelarán la ciudad más visitada y, por lo tanto, la más popular.

En este taller deberán modelar la situación expuesta anteriormente. Para esto, la ONU les envió un documento donde se explican los 3 entes principales que participan del problema: Personas, Vehículos y Ciudad.

Descripción

Para este problema, las ciudades se caracterizan por poseer un nombre, un color y la cantidad de vehículos pintados. Cada persona posee un nombre, una edad, un contador de horas conducidas, una lista con el nombre de las ciudades visitadas y un vehículo.

Por otro lado, los vehículos (en este caso autos y bicicletas) tienen un color, velocidad promedio, kilómetros recorridos, kilómetros máximos por recorrer, un contador de ciudades visitadas y un indicador de estado que puede ser: *'Funcionando'* o *'En Panne'*. Todos los vehículos son de color blanco, parten con 0 kilómetros recorridos y con el estado *'Funcionando'*. Los autos pueden andar hasta 1500 kilómetros y tienen una velocidad promedio de 80 kilómetros por hora. Por su parte, las bicicletas tienen como máximo 1000 kilómetros para recorrer y una velocidad promedio de 30 kilómetros por hora. Finalmente, todos los vehículos deben ser capaces de mostrar su eficiencia, la cual es calculada como la división entre las ciudades visitadas y el total de kilómetros recorridos.

Para identificar siempre la última ciudad visitada por cada persona, las ciudades **pintan** el vehículo de las personas que llegan a ellas con su color característico y el contador de vehículos pintados por las ciudades aumenta en 1 de forma correspondiente. A su vez, cada vez que un vehículo es pintado, su contador de ciudades visitadas aumenta en 1.

Un vehículo tiene la capacidad de **recorrer** una cantidad de kilómetros (1500 o 1000). Al momento de realizar tal acción (avanzar/recorrer), el contador de kilómetros recorridos del vehículo aumenta y si

este llega a superar los kilómetros máximos posibles, el estado del vehículo pasará a ser '*En Panne*' y solo recorrerá los kilómetros necesarios para que la cantidad de kilómetros recorridos sea igual a su capacidad máxima. Cada vez que a un vehículo se le pide recorrer una distancia, este retorna el tiempo que se demoró según la velocidad promedio que posee.

Por último, una persona puede **viajar** siempre y cuando se le indique la ciudad y los kilómetros necesarios para llegar a esa ciudad. Al momento que una persona decida viajar, el vehículo deberá recorrer la distancia indicada y el contador de horas conduciendo de la persona será aumentado según el tiempo que le tomó al vehículo recorrer la distancia. En caso de que el vehículo llegue a su destino, en consola se deberá informar el siguiente mensaje: 'La persona *nombre* llegó a *nombre_ciudad* en *tiempo_demorado* horas y ahora su vehículo es de color *color_vehiculo*'. En caso contrario deberá mostrar el siguiente mensaje: 'La persona *nombre* no logró llegar a *nombre_ciudad*, se quedó en panne a las *tiempo_demorado* horas de viaje.'¹

Visualización de las entidades

El último requisito dado por la ONU, es poder visualizar de forma clara la información de cada entidad al momento de imprimirlas en consola en formato *string*. Para esto debe utilizar únicamente la sobreescritura de métodos (*method overriding*).

La información obtenida según cada entidad es:

- Vehículos: el tipo del vehículo, su color actual del vehículo, los kilómetros recorridos y el estado actual.

Bicicleta: color actual azul, ha recorrido 34 kilómetros, estado es Funcionando

- Personas: el nombre de ella, su edad, la cantidad de horas, las ciudades visitadas y el tipo y color de su vehículo. Ejemplo:

Juanito Perez: edad 28 años, su vehículo es bicicleta de color rojo, ha conducido 128 horas.
Ciudades Visitadas: París, Santiago, Lima

- Ciudad: el nombre de ella, su color y la cantidad de vehículos pintados

Santiago: color verde, 123 vehículos pintados.

¹Notar que las palabras en cursiva hacen referencia a atributos

Test del modelo

Para poder verificar que su modelo esté correcto, la ONU les ha enviado el siguiente código, que espera poder correr sin ningún problema.

```
# Creamos las ciudades
roma = Ciudad("Roma", "Verde")
santiago = Ciudad("Santiago", "Rojo")
lima = Ciudad("Lima", "Azul")
valparaiso = Ciudad("Valparaíso", "Celeste")

# Creamos un vehículo medio usado
vehiculo = Auto()
tiempo_de_viaje = vehiculo.recorrer(900)
roma.pintar(vehiculo)

# Creamos las personas con sus respectivos vehículos
persona_1 = Persona("Raúl", 45, vehiculo)
persona_2 = Persona("Laura", 45, Auto())
persona_3 = Persona("Juan", 17, Auto())

# Las personas viajan, los números no calzan pero no importa
persona_1.viajar(roma, 99)
persona_2.viajar(santiago, 2)
persona_2.viajar(valparaiso, 500)
persona_2.viajar(santiago, 1000)
persona_3.viajar(lima, 6000)
persona_1.viajar(santiago, 1278)
persona_2.viajar(roma, 200)

# Visualizamos algunas entidades
print(persona_2)
print(vehiculo)
print(persona_3.vehiculo)
```

```
print(roma)
```

Objetivo parcial de participación

Para verificar la participación durante la clase, deberán cumplir con modelar la jerarquía de **vehículos**, es decir, las clases que la componen, sus atributos y métodos.

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.