



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

Taller 1a

Objetivos

- Aplicar los contenidos fundamentales de programación orientada a objetos para modelar una estación de servicio.

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 23 de agosto a las 16:50 hrs.
- **Formato de entrega:** archivo python notebook (**T1a.ipynb**) y archivo python (**T1a.py**) con la solución de este enunciado. Los archivos deben estar ubicados 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 FUERA DE PLAZO**

Introducción

Con el fin practicar las técnicas de POO, en esta actividad deberá modelar una estación de servicio (bencinera), considerando las entidades principales que participan en su funcionamiento. Esto significa, definir las **clases** correspondientes, sus relaciones y comportamientos. Una vez finalizado esto, deberá simular el funcionamiento de la estación, indicando el tiempo de espera por atención para cada cola.

Descripción del problema

La estación de servicio opera como un sistema con dos o más colas de espera, en donde vehículos de cuatro distintos tipos llegan aleatoriamente con una probabilidad p y se ubican en la cola de menor extensión. Un vehículo es atendido cuando llega al principio de la cola, y el tiempo de duración de la atención depende del tipo del vehículo y de la bencina que utiliza (4 tipos distintos). Considere finalmente que un vehículo puede cambiarse de cola si en un momento dado, existe otra cola con menor extensión. El criterio para elegir cuántos vehículos se cambian y en qué orden debe ser definido por usted.

Modelo de clases

El modelo a implementar debe considerar los siguientes elementos:

- Clases para todas las entidades relevantes a modelar.
- La jerarquía completa de clases para los vehículos, que considere una clase base (primera clase de una jerarquía) que solo defina la interfaz (atributos y métodos sin implementar).
- Dos clases que hereden de otra (puede no ser la misma).
- Clases que participen como atributos en otras (composición o agregación)
- Implementaciones del método `__str__()`, que entreguen información relevante del estado de los objetos.
- Dos sobrecargas de métodos (*override* - polimorfismo)

Se recomienda fuertemente comentar el código y definir en celdas de texto qué es cada clase. Considere además que para definir las clases, sus relaciones y comportamientos, no es necesario tener desarrollada la simulación.

Simulación

Para la simulación, considere la siguiente secuencia de instrucciones como posible guía:

```
# creación de variables y objetos relevantes
for t in range(t_max):
    if llega un auto nuevo
        # agregar vehículo a la cola correspondiente
```

```
if no hay vehículos en atención y sí hay vehículos en espera
    # verificar si hay cambios de cola
    # se atiende al primer vehículo de cada cola
# se actualiza el tiempo de espera para la próxima atención
```

Durante la simulación debe entregar información sobre el proceso cada vez que ocurre algún evento, además de reportar el tiempo de espera por atención para cada cola. No olvide utilizar clases para modelar las entidades.

Objetivo parcial de participación

Para verificar la participación durante la clase, deberá completar los requisitos del modelo de clases. En su respuesta, considere los atributos y comportamientos requeridos para que los objetos puedan participar de la simulación, pero no es necesario que modele aquellas clases que solo cumplen una función abstracta (por ejemplo, la clase *Simulación*).

Política de Integridad Académica

“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.