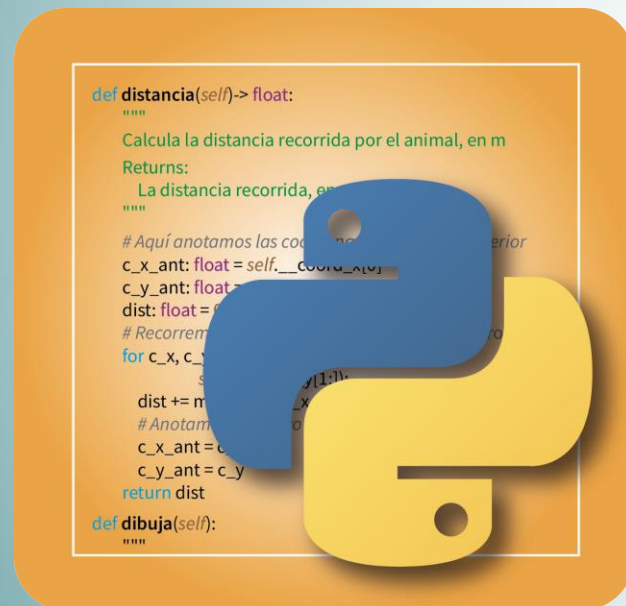


Programación

Práctica 5. Cálculo de escapatorias en un circuito de fórmula 1



Michael González Harbour
José Javier Gutiérrez García
José Carlos Palencia Gutiérrez
José Ignacio Espeso Martínez
Adolfo Garandal Martín

Departamento de Ingeniería
Informática y Electrónica

Este material se publica con licencia:
[Creative Commons BY-NC-SA 4.0](https://creativecommons.org/licenses/by-nc-sa/4.0/)

Práctica 5: Cálculo de escapatorias en un circuito de fórmula 1

Objetivos: Crear una clase con atributos y métodos, así como un programa principal que lea datos de teclado y use esta clase

Descripción: En un circuito de fórmula 1 a la salida de cada curva se sitúan zonas de escapatoria en las que el coche puede frenar con seguridad si no consigue tomar la curva

Crearemos una clase para calcular estas escapatorias y un *main* para mostrar resultados



Diseño de la clase Coche

La clase **Coche** almacena como atributos los coeficientes de la frenada:

- **alfa**: coeficiente de la deceleración aerodinámica en s^{-1}
- **beta**: coeficiente de la deceleración por rozamiento con el suelo en m/s^2

Dispone de operaciones para calcular la distancia recorrida, el tiempo de frenado y la longitud de la escapatoria

Coche
-alfa, beta: float
+ __init__(alfa: float, beta: float) + distancia(t: float, v_0: float): float + tiempo_frenado(v_0: float): float + longitud_escapatoria(v_0: float): float

Detalles sobre la clase

- Atributos y otros datos
 - Usaremos unidades del sistema internacional
 - Documentar con comentarios las unidades usadas en los atributos, los argumentos de los métodos, y los valores retornados por ellos
- *Constructor*
 - crea los atributos y les pone sus valores iniciales a partir de los parámetros
- *distancia*
 - retorna la distancia recorrida, x , en función del tiempo, t , y de la velocidad inicial, v_0 :

$$x = \frac{1}{-\alpha} e^{-\alpha t} \left(v_0 + \frac{\beta}{\alpha} \right) - \frac{\beta}{\alpha} t + \frac{1}{\alpha} \left(v_0 + \frac{\beta}{\alpha} \right)$$

Detalles sobre la clase (cont.)

- tiempo_frenado

- retorna el tiempo hasta detenerse, t_f , en función de la velocidad inicial, v_0 :

$$t_f = \frac{1}{\alpha} \ln\left(\frac{\alpha v_0 + \beta}{\beta}\right)$$

- longitud_escapatoria

- retorna la longitud de la escapatoria en función de la velocidad inicial, v_0
- se calcula con los métodos anteriores como la distancia recorrida en un tiempo igual al tiempo de frenado

Programa principal

Escribir un método `main` que cree dos objetos de la clase `Coche` con los datos indicados en esta tabla:

Coche	alfa (s^{-1})	beta (m/s^2)
Ferrari	0.18	8.9
Red Bull	0.20	8.7

A continuación debe leer por teclado tres velocidades iniciales en km/h^1

Posteriormente debe mostrar en pantalla los tiempos de frenado de ambos coches en función de las tres velocidades iniciales leídas

Finalmente debe mostrar en pantalla las longitudes de la escapatoria para ambos coches en función de las tres velocidades iniciales anteriores

1. Cuidado con los cambios de unidades, pues en la clase `Coche` las unidades son m/s

Programa principal: formato de la salida

La salida del programa principal tendrá un formato similar al de este ejemplo:

```
Tiempos de frenado (s)
Vel(km/h)   Ferrari Red Bull
100.0       2.477   X.XXX
200.0       X.XXX   X.XXX
350.0       6.041   X.XXX
```

```
Longitudes de escapatorias (m)
Vel(km/h)   Ferrari   Red Bull
100.0       31.853   XX.XXX
200.0       XXX.XXX  XX.XXX
350.0       241.448  XXX.XXX
```

Nota: *X.XXX* se sustituirá por el número obtenido; los valores numéricos que aparecen son resultados reales

Salida formateada

Para obtener una salida formateada en columnas, se puede poner tamaño total fijo y número de decimales a cada dato que se muestra

Podemos usar los strings-`f` para dar formato a los números

En este ejemplo, suponiendo `vel=20.345678`

```
print(f"Velocidad: {vel:8.3f} km/h")  
- produce: Velocidad:    20.346 km/h
```

Observar que

- el formato con ocho caracteres totales y tres decimales se especifica con `8.3f`
- el `"."` decimal cuenta como un carácter; también el signo si fuese necesario

Parte avanzada

Descripción: Añadir un nuevo método y mostrar datos en una ventana

- Añadir a la clase **Coche** un nuevo método para calcular la velocidad, en m/s, en función del tiempo y de la velocidad inicial:

$$v = \frac{v_0}{e^{\alpha t}} - \frac{\beta}{\alpha} + \frac{\beta}{\alpha e^{\alpha t}}$$

- Añadir instrucciones al **main** para mostrar en una ventana de la clase **Escritura** (paquete **fundamentos**) la velocidad del Ferrari, en km/h, transcurrida la mitad del tiempo de frenado, para las tres velocidades iniciales utilizadas anteriormente

Entregar dos archivos

1. El código fuente
2. Informe conteniendo:
 - Una captura de pantalla que muestre los resultados de ejecutar el main
 - Parte avanzada: si se ha hecho, una captura de pantalla de la ventana con los resultados de la parte avanzada