

Programación I – 2020

Trabajo práctico 3

Ejercicio 1

Del ejercicio 1 del trabajo práctico 2 convertir las siguientes restricciones en excepciones:

- No se puede extraer mas dinero del que hay disponible en el cajero.
- No se puede extraer dinero que no sea múltiplo de 100.

En este caso en vez de devolver un error las funciones deben de hacer raise con una excepción propia que indique el error.

Ejercicio 2

Del ejercicio 2 del trabajo práctico 2 convertir las siguientes restricciones en excepciones:

- Si no es posible sacar el monto necesario con los billetes que tenemos debemos generar una excepción con un error que no es posible extraer el monto pedido. Ejemplo:
Tenemos en el cajero solo billetes de \$500 y queremos extraer \$900, en este caso no es posible dar \$400 porque no tengo billetes de \$200 o de \$100.
- El rango de % de cambio puede solo ir de 0 a 100%. Generar una excepción si el % se encuentra fuera de rango.

En este caso en vez de devolver un error las funciones deben de hacer raise con una excepción propia que indique el error.

Ejercicio 3

Agregar excepciones extras para verificar:

- en caso de querer extraer un valor de dinero negativo.
- En caso de querer extraer dinero y el cajero no tiene billetes cargados.

Ejercicio 4

Modificar los casos de prueba desarrollados en el ejercicio 3 del trabajo práctico 2 para probar que las excepciones desarrolladas funcionan correctamente.

Ejercicio 5

Crear el Dockerfile que permita armar la imagen de docker y luego la ejecutemos con docker run y podamos ver el resultado de los tests en pantalla.

Ejemplo de Dockerfile

```
FROM python:3
RUN git clone https://github.com/usuario/repositorio
WORKDIR /repositorio
RUN pip install -r requirements.txt
```

RUN pip install parameterized
CMD ["python3", "test_tateti.py"]

Ojo con las comillas que sean las que corresponden y no las comillas de la porquería que agregar Word.