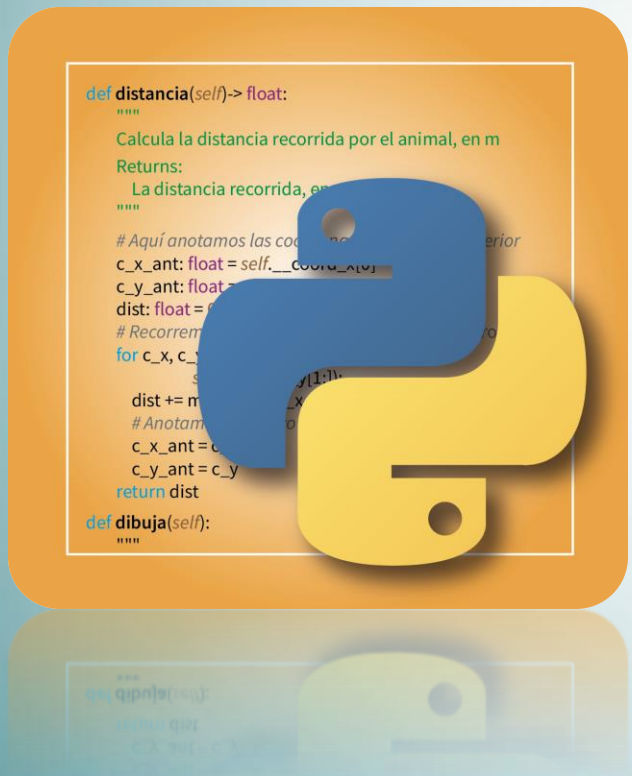
Programación

Práctica 2. Expresiones y operaciones de Entrada / Salida



Michael González Harbour

José Javier Gutiérrez García

José Carlos Palencia Gutiérrez

José Ignacio Espeso Martínez

Adolfo Garandal Martín

Departamento de Ingeniería Informática y Electrónica

Práctica 2: Cambio en monedas

Objetivo: Practicar con expresiones y operaciones de entrada/salida

Descripción: Una tienda nos ha pedido desarrollar un programa que facilite a los dependientes el cálculo del cambio que deben devolver a los clientes tras realizar una compra

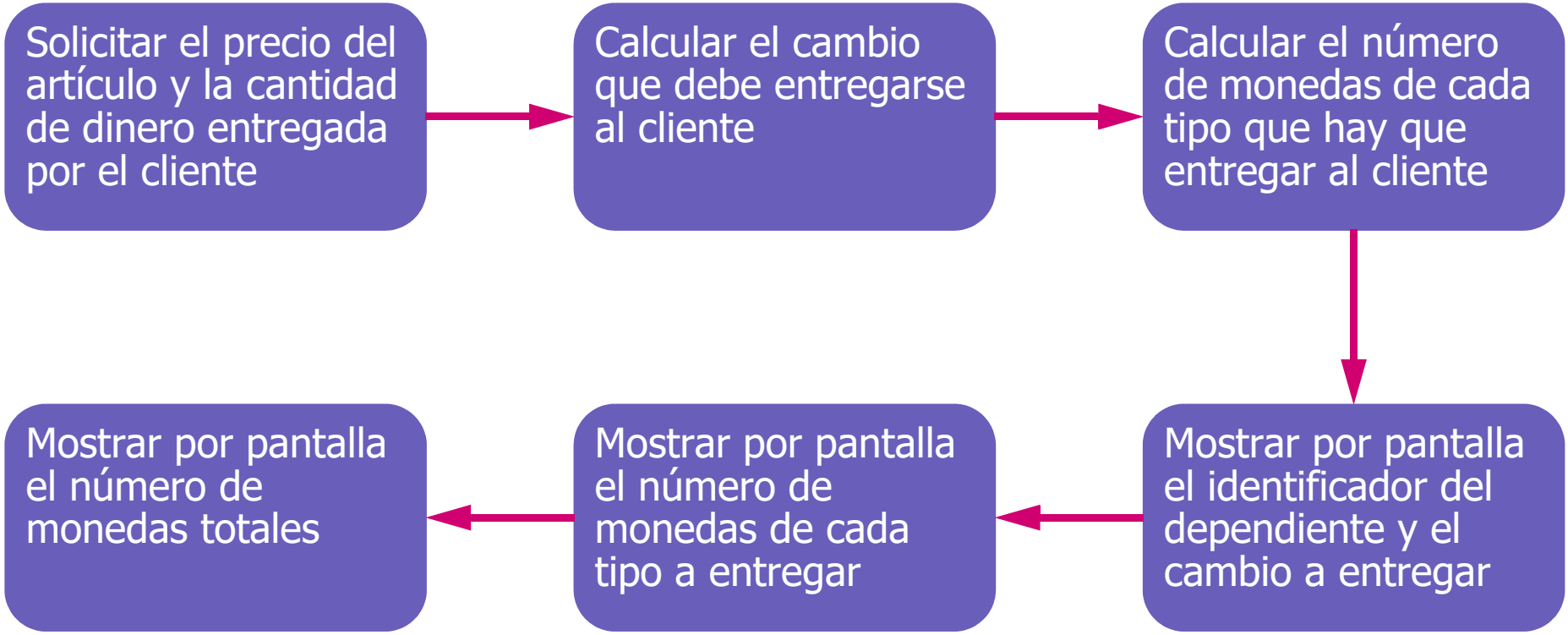
- el cambio siempre se devuelve con monedas
- se busca minimizar el número de monedas a devolver para agilizar el trabajo de los dependientes

Ejemplo: un cliente realiza una compra de 7,40€ y paga con 10€. El cambio sería de 2,60 € repartidos como sigue

							
1	0	1	0	1	0	0	0

Planteamiento

Etapas del algoritmo a desarrollar



Por sencillez, no consideraremos el tratamiento de errores tales como precios negativos, pago menor que el precio, etc.

Versión 0: algoritmo simplificado, en euros

Crea un módulo `cambio_simplificado.py` con un `main()` que implemente cada una de las etapas del algoritmo propuesto

- utiliza tus iniciales como identificador del dependiente
- crea las variables y constantes que consideres necesarias
  - p.ej., añadir constantes para los valores de las monedas disponibles

Esta versión preliminar requiere:

- que todos los precios y los pagos de los clientes se realicen en euros como números enteros
  - por ejemplo, un cliente paga 10€ por un artículo que cuesta 3€
- que el cambio siempre se devuelva exclusivamente con monedas de 1 o 2 euros
  - en el caso anterior, 7//2€ es 3 monedas de 2€ y (7%2€)//1 es una de 1€

Ejemplo de salida del programa

```
Precio del artículo: 12

Pago realizado: 20

Dependiente: MGH
Cambio a entregar: 8 euros

Monedas de 2 euros: 4
Monedas de 1 euro : 0

Monedas totales: 4
```

Versión 1.0, en céntimos de euro

Crea un módulo `cambio.py` que implemente cada una de las etapas del algoritmo propuesto

- se puede partir del programa implementado en el apartado anterior

Esta versión requiere:

- todos los precios y los pagos se realizan en euros con números reales
- internamente se trabaja con céntimos de euro, con números enteros
  - multiplicar o dividir por 100 para pasar entre céntimos y euros
  - y convertir al tipo `int()` al pasar a céntimos
- se usan todas las monedas: de 200, 100, 50, 20, 10, 5, 2 y 1 cént.
  - por ejemplo, si un cliente paga 20.00€ por un artículo que cuesta 5.75€
  - el cambio son 1425 cént.
  - se devuelven 1425//200 = 7 monedas de 2€
  - el resto es 1425%200 = 25c, que distribuiremos entre las otras monedas

Ejemplo de salida por pantalla

```
Precio del artículo: 10.1

Pago realizado: 20

Dependiente: MGH
Cambio a entregar: 9.9 euros

Monedas de 2.0 euros: 4
Monedas de 1.0 euro : 1
Monedas de 50 cénts : 1
Monedas de 20 cénts : 2
Monedas de 10 cénts : 0
Monedas de 5 cénts  : 0
Monedas de 2 cénts  : 0
Monedas de 1 cénts  : 0

Monedas totales: 8
```

Opcional: Formatear los euros con dos decimales

Al imprimir con `print` solo salen los decimales no nulos

```
print("Cambio a entregar: "+str(cambio)+" euros")
```

- produce: `Cambio a entregar: 9.9 euros`

En Python podemos usar los strings-`f` para dar formato a los números, como en este ejemplo

```
print(f"Cambio a entregar: {cambio:.2f} euros")
```

- produce: `Cambio a entregar: 9.90 euros`

Observar que

- delante del string se pone una `f` o `F`
- la variable a mostrar en mitad del texto se encierra entre `{}`
- el formato con dos decimales se especifica con `.2f`

Parte avanzada: Versión 1.1, que incluye descuentos

Crea un módulo `cambio_rebajas.py` que implemente cada una de las etapas del algoritmo propuesto con las siguientes modificaciones:

- al precio del artículo se le aplica una rebaja del 10%
- se muestra en pantalla el precio rebajado y el cambio a devolver
- se muestra por pantalla la base imponible (precio rebajado sin IVA) y el IVA a pagar, teniendo en cuenta que el IVA es del 21%
- obligatoriamente se muestran los importes con dos decimales

Comprueba que el programa funciona correctamente con los siguientes datos

- precio del artículo: 20.49 €
- pago del cliente: 30.0 €

Ejemplo de salida

```
Precio del artículo (euros): 20.49

Pago realizado (euros): 30

Dependiente: MGH
Precio rebajado: 18.44 euros
Base imponible: 15.24 euros
IVA: 3.20 euros
Cambio a entregar: 11.56 euros

Monedas de 2.0 euros: 5
Monedas de 1.0 euro : 1
Monedas de 50 cénts : 1
Monedas de 20 cénts : 0
Monedas de 10 cénts : 0
Monedas de 5 cénts  : 1
Monedas de 2 cénts  : 0
Monedas de 1 cénts  : 1

Monedas totales: 9
```

Corrección de los problemas de precisión

Si la salida de tu programa para el ejemplo anterior no es la esperada, puede deberse a un problema de precisión

- podría solucionarse haciendo uso del redondeo
- con la función `round(x)` al calcular el precio y el cambio en céntimos
  - esta función devuelve el número entero más cercano a `x`

Modifica tu código para que utilice la función `round()` y comprueba la salida con el ejemplo anterior

Entregar

Entregar hasta 4 ficheros con:

- 1. El código fuente del módulo `cambio_simplificado.py`
- 2. El código fuente del módulo `cambio.py`
- 3. Un informe en formato `pdf` con:
  - los resultados de las ejecuciones de los módulos obligatorios
  - el resultado de la ejecución de la parte avanzada, si se ha hecho
- 4. El código fuente del módulo `cambio_rebajas.py`, si se ha hecho