Programación Avanzada
IIC2233 2025-2

Cristian Ruz - Pablo Araneda - Francisca Ibarra - Tamara Vidal - Daniela Concha

Anuncios

1. Hoy tenemos actividad evaluada.
2. La ECA se encuentra disponible para responder de domingo a martes.
3. El jueves de la siguiente semana es el Midterm durante el horario de clases.

Programación funcional

Paradigma Funcional

Paradigma Funcional

Dado que la pureza de una función depende de la interacción con el sistema, cuál sería un ejemplo de:

- Una función impura debido a que afecta al sistema
- Una función impura debido a que es afectada por el sistema

¿Una función pura…

- …se puede, sin afectar el programa, llamar sin utilizar lo que retorna?
- …se puede, sin afectar el programa, reemplazar por lo que retorna?

Función pura

Cómo son las siguientes funciones en términos de:

- Retornar siempre lo mismo para el mismo input
- Tener un efecto secundario

```
def funcion1(a):
    return bool(a)
```

```
def funcion2(a):
    print(a)
```

```
b = 10
def funcion3(a):
    return a == b
```

```
import random
def funcion4():
    return random.random()
```

Función pura

Se usa random.seed para no esperar a que salgan valores idénticos

Se puede ver que las llamadas a **random.random()** fuera de la **funcion4** fueron afectadas por la misma función

```
import random
def funcion4():
    return random.random()
```

```
>>> random.seed(2233)
>>> funcion4()
0.5022470106574008
>>> random.random()
0.08386343789203865
>>> random.seed(2233)
>>> random.random()
0.5022470106574008
>>> funcion4()
0.08386343789203865
```

Funciones generadoras

Funciones generadoras

- En términos del uso de una función, ¿qué cambia si se usa **yield** en vez de **return**?
- ¿Tiene el **send()** alguna restricción comparado con el **next()**?
- ¿Qué diferencia un **yield** de un **yield from**?
- ¿Qué hace el siguiente código?

```
def memorizar(valor):
    previo = None
    while True:
        previo, valor = valor, (yield previo)
```

Funciones generadoras

¿Qué hace el siguiente código? ¿Por qué usa **next()**?

```
def memorizar(valor):
    previo = None
    while True:
        previo, valor = valor, (yield previo)
```

```
>>> a = memorizar(1)
>>> next(a)  # esto retorna None
>>> a.send(3)
1
>>> a.send(2)
3
```

Funciones que retornan generadores

Funciones que retornan generadores

- Qué situaciones de la vida cotidiana tiene sentido modelar usando:
  - **map**
  - **filter**
  - **reduce**
  - **enumerate**
  - **zip**
- ¿Similar a cuál de esas funciones es el siguiente código?

```
def mi_funcion(funcion, arg):
    return reduce(lambda x, y: (*x, funcion(y)), arg, [])
```

Funciones que retornan generadores

¿Como qué función se comporta el siguiente código? ¿En qué difiere?

```
def mi_funcion(funcion, arg):
    return reduce(lambda x, y: (*x, funcion(y)), arg, [])
```

```
>>> list(mi_funcion(lambda x: x * 1, (1,)))
[1]
>>> tuple(mi_funcion(lambda _: 222, range(2)))
(222, 222)
>>> mi_funcion(lambda x: x + 1, {3,1,2})
(2, 3, 4)
```

Aplicaciones

Aplicaciones

- ¿Cuál es la ventaja de…
  - … usar generadores comparado con usar directamente listas?
  - … usar namedtuples en vez de clases?
- ¿Qué es lo que llega al `else` en el siguiente código?

```
[i if i % 2 else i + 1 for i in range(10)]
```

Aplicaciones

- ¿Cuál es la ventaja de…
  - … usar generadores comparado con usar directamente listas?
  - … usar namedtuples en vez de clases?
- ¿Qué es lo que llega al `else` en el siguiente código?

```
>>> [i if i % 2 else i + 1 for i in range(10)]
[1, 1, 3, 3, 5, 5, 7, 7, 9, 9]
```

Pregunta de Evaluación Escrita

Tema: Programación Funcional (Examen 2024-1)

8. En el contexto de Programación Funcional, ¿qué es lo **primordial** que debe cumplir una función para que sea considerada una **función generadora**?

   A) Utilizar el comando **`yield`**.
   B) Utilizar estructuras por comprensión.
   C) No utilizar el comando **`return`**.
   D) No utilizar los comandos **`for`** y **`while`**.
   E) Retornar el resultado tras ejecutar las funciones **`map`**, **`filter`** y/o **`reduce`**.

Pregunta de Evaluación Escrita

Tema: Programación Funcional (Examen 2024-1)

8. En el contexto de Programación Funcional, ¿qué es lo **primordial** que debe cumplir una función para que sea considerada una **función generadora**?

   A) Utilizar el comando `yield`.
   B) Utilizar estructuras por comprensión.
   C) No utilizar el comando **`return`**.
   D) No utilizar los comandos **`for`** y **`while`**.
   E) Retornar el resultado tras ejecutar las funciones **`map`**, **`filter`** y/o **`reduce`**.

Programación Avanzada
IIC2233 2025-2

Cristian Ruz - Pablo Araneda - Francisca Ibarra - Tamara Vidal - Daniela Concha

Comentarios AC