Python: Objetos

Crea tu propio tipo de datos

Programación Unidad 3: Programación orientada a objetos y librerías

Ph.D. Santiago Echeverri-Arteaga

“About a year or so after I started programming, I decided to make a game to play Wheel of Fortune. I wrote the program in the BASIC programming language and it got to be pretty large, a couple thousand lines. It mostly worked, but whenever I tried to fix something, my fix would break something in a completely different part of the program. I would then fix that and break something else. Eventually I got the program working, but after a while I was afraid to even touch it.” Brian Heinold

Índice

Índice

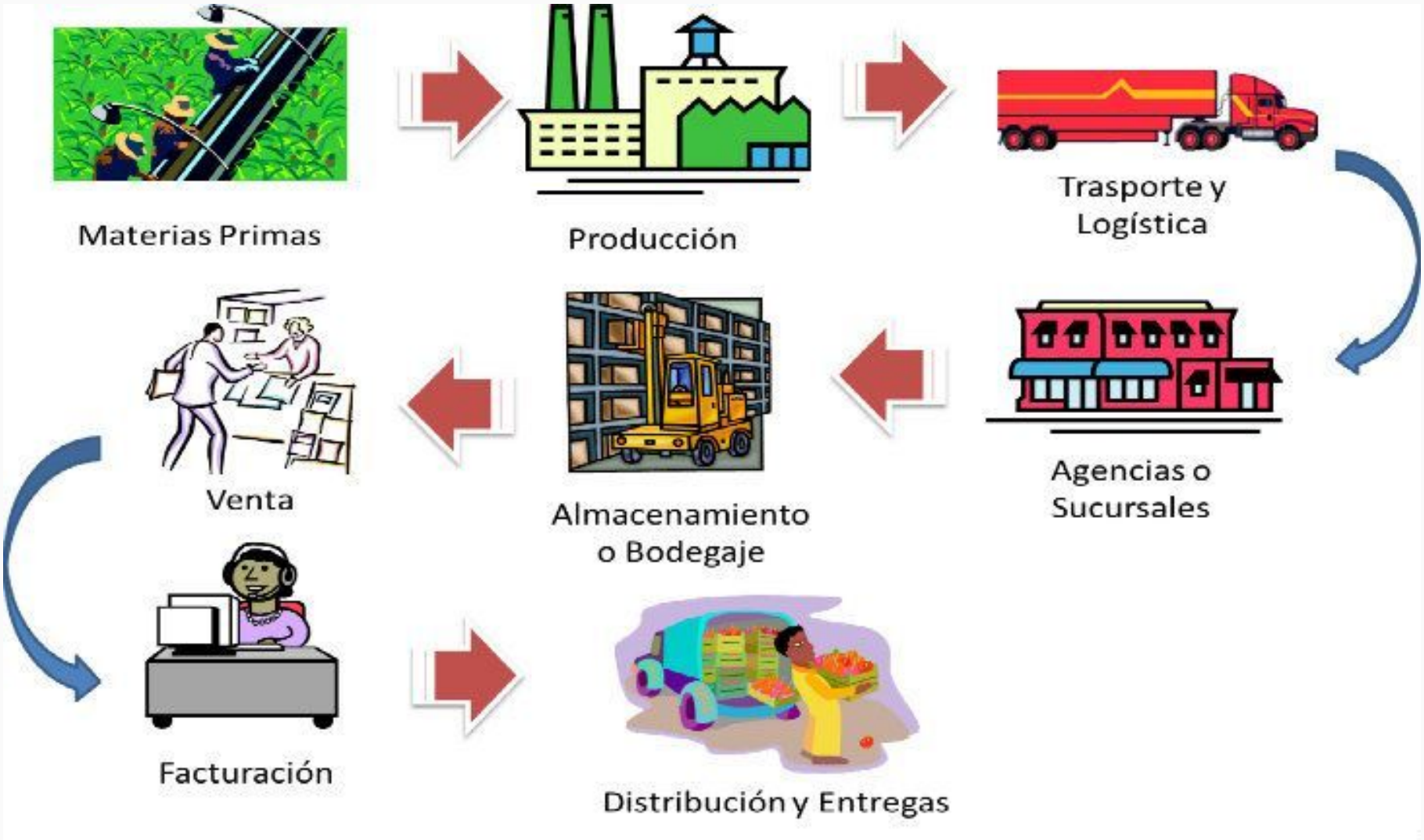
Nuestro propósito: Encapsulamiento

Encapsulamiento

"One solution to this type of problem is *object-oriented programming*. One of its chief benefits is *encapsulation*, where you divide your program into pieces and each piece internally operates independently of the others."

En lenguaje coloquial

Ir más allá de las funciones



¿Qué es un Axes? ¿Qué es un figure? ¿Qué es una cadena? ¿Que es un array?

¿Qué es un Axes? ¿Qué es un figure? ¿Qué es una cadena? ¿Que es un array?

Si fuéramos dueños de una librería y tuviéramos que manejar constantemente programas que manipulen la información referente a los libros (Título, autores, editorial, ISBN, precio) ¿Cómo podríamos solucionarlo?

¿Qué es un Axes? ¿Qué es un figure? ¿Qué es una cadena? ¿Que es un array?

Si fuéramos dueños de una librería y tuviéramos que manejar constantemente programas que manipulen la información referente a los libros (Título, autores, editorial, ISBN, precio) ¿Cómo podríamos solucionarlo?

Si fuéramos rectores de la Universidad del Quindío y tuvieramos que manipular constantemente la información de los estudiantes, administrativos, facultades, programas, profesores, personal de servicio ¿Como podemos afrontarlo?

¿Qué es un Axes? ¿Qué es un figure? ¿Qué es una cadena? ¿Que es un array?

Si fuéramos dueños de una librería y tuviéramos que manejar constantemente programas que manipulen la información referente a los libros (Título, autores, editorial, ISBN, precio) ¿Cómo podríamos solucionarlo?

Si fuéramos rectores de la Universidad del Quindío y tuvieramos que manipular constantemente la información de los estudiantes, administrativos, facultades, programas, profesores, personal de servicio ¿Como podemos afrontarlo?

Si fuéramos a hacer un programa que le permita al usuario jugar múltiples juegos de cartas ¿Cómo lo hacemos más fácilmente?

¿Qué es un Axes? ¿Qué es un figure? ¿Qué es una cadena? ¿Que es un array?

Si fuéramos dueños de una librería y tuviéramos que manejar constantemente programas que manipulen la información referente a los libros (Título, autores, editorial, ISBN, precio) ¿Cómo podríamos solucionarlo?

Si fuéramos rectores de la Universidad del Quindío y tuvieramos que manipular constantemente la información de los estudiantes, administrativos, facultades, programas, profesores, personal de servicio ¿Como podemos afrontarlo?

Si fuéramos a hacer un programa que le permita al usuario jugar múltiples juegos de cartas ¿Cómo lo hacemos más fácilmente?

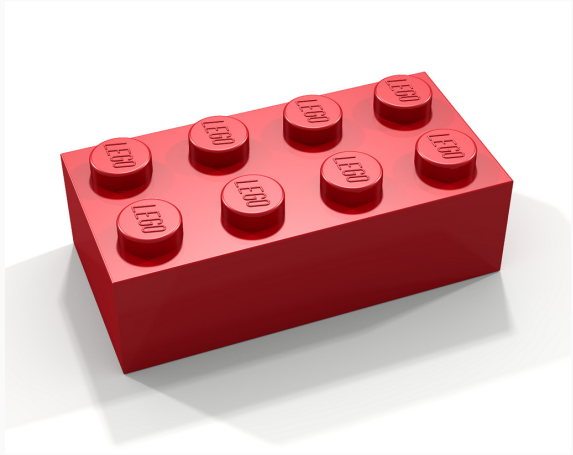
Si trabajáramos en mecánica cuántica y tuviéramos que usar constantemente espacios de Hilbert y operadores y deseáramos usar la programación a nuestro servicio ¿Como podemos hacerlo?

Programación orientada a objetos

Programación orientada a objetos

Programa con funciones

Código espagueti



Contextualización

¿Qué es una clase?

Son objetos donde podemos “almacenar” variables y funciones, de tal forma que el código quede más organizado, más corto y funcional. Por ejemplo, la clase string tiene *almacenada* la función lower(), la clase array, tiene almacenada la variable shape.

En las clases existe una “jerarquía”, hay clases “padre” y clases “hijo”. La supercategoría más general son los *object*.

Clases

¿Qué es una clase?

Son objetos donde podemos “almacenar” variables y funciones, de tal forma que el código quede más organizado, más corto y funcional. Por ejemplo, la clase string tiene *almacenada* la función lower(), la clase array, tiene almacenada la variable shape.
Las subclases tienen todos los atributos (funciones y variables) de la clase mayor

Hay algunas funciones y variables predefinidas para todas las clases, esos atributos especiales inician y finalizan con doble __
Un ejemplo de esos atributos especiales son (Magic methods):
__add__ __main__ __str__ __eq__ __dict__ __init__

La clase más simple que se puede definir es:

```
class NOMBRE:
    """ """
```

Funciones VS Métodos

Un método es una función definida dentro de una clase. Para llamarla se necesita usar un objeto de esa clase, seguido de un punto y el nombre de la función. **Ejemplo:** cadena.lower(). A veces la funcion a la que llamamos exige más argumentos, los cuales podemos poner adentro de los paréntesis. **Ejemplo:** cadena.center(25). Esto nos dice que **un método siempre toma como primer argumento a la clase a la que pertenece**.

Para definir un método ponemos dentro de la clase un def Metodo(self,argumentos): y procedemos igual que con una función.

Variables asociadas a las clases

Una clase puede tener variables “almacenadas” (por defecto) o se le pueden asignar (“guardar variables en la clase”). Tal como los xaxes, yaxes, xticks, yticks, xlabel, ylabel, title.

Para asignar una variable a una clase se usa nombre_clase.variable = valor

A veces es molesto y poco práctico tener que inicializar (asignar todas las variables) de una clase. Para no tener que hacer esto debemos definir un método especial, el cual se conoce como el constructor y se llama __init__. En el tomamos como argumentos los valores de las variables y los asignamos como self.variable = variable (o podemos definir variables que el usuario no ingrese como self.longitud = len(a)

Variables asociadas a las clases

Siempre que se define una clase python ejecuta automáticamente el constructor

```
class Example:
    def __init__(self, a, b):
        self.a = a
        self.b = b

    def add(self):
        return self.a + self.b

e = Example(8, 6)
print(e.add())
```

Herencia

Herencia

```
class Parent:
    def __init__(self, a):
        self.a = a
    def method1(self):
        print(self.a*2)
    def method2(self):
        print(self.a+'!!!')

class Child(Parent):
    def __init__(self, a, b):
        self.a = a
        self.b = b
    def method1(self):
        print(self.a*7)
    def method3(self):
        print(self.a + self.b)

p = Parent('hi')
c = Child('hi', 'bye')

print('Parent method 1: ', p.method1())
print('Parent method 2: ', p.method2())
print()
print('Child method 1: ', c.method1())
print('Child method 2: ', c.method2())
print('Child method 3: ', c.method3())
```

Herencia

```
class Parent:
    def __init__(self, a):
        self.a = a
    def method1(self):
        print(self.a*2)
    def method2(self):
        print(self.a+'!!!')

class Child(Parent):
    def __init__(self, a, b):
        self.a = a
        self.b = b
    def method1(self):
        print(self.a*7)
    def method3(self):
        print(self.a + self.b)

p = Parent('hi')
c = Child('hi', 'bye')

print('Parent method 1: ', p.method1())
print('Parent method 2: ', p.method2())
print()
print('Child method 1: ', c.method1())
print('Child method 2: ', c.method2())
print('Child method 3: ', c.method3())
```

```
Parent method 1: hihi
Parent method 2: hi!!!

Child method 1: hihihihihihihi
Child method 2: hi!!!
Child method 3: hibye
```

Redefiniendo métodos y el constructor

Si estamos pensando en implementar las clases de la Universidad del Quindío, nos percatamos que los estudiantes tienen nombre, dirección, e-mail, carrera, número de matriculas y cursos inscritos. Las facultades tendrían nombre, dirección, e-mail, carreras ofertadas, personal, número, presupuesto. De esas variables, hay algunas compartidas y otras no, razómn por la cual conviene definir a los dos como hijos de la categoría Uniquindianos, la cual tendría las variables nombre, dirección, e-mail.

¿Cómo hacemos el constructor en las subcategorias?

Redefiniendo métodos y el constructor

Si estamos pensando en implementar las clases de la Universidad del Quindío, nos percatamos que los estudiantes tienen nombre, dirección, e-mail, carrera, número de matriculas y cursos inscritos. Las facultades tendrían nombre, dirección, e-mail, carreras ofertadas, personal, número, presupuesto. De esas variables, hay algunas compartidas y otras no, razómn por la cual conviene definir a los dos como hijos de la categoría Uniquindianos, la cual tendría las variables nombre, dirección, e-mail.

¿Cómo hacemos el constructor en las subcategorias?
Dentro del constructor de la subcategoria ejecutamos super().__init__(name, address, email)

Redefiniendo métodos y el constructor

Si estamos pensando en implementar las clases de la Universidad del Quindío, nos percatamos que los estudiantes tienen nombre, dirección, e-mail, carrera, número de matriculas y cursos inscritos. Las facultades tendrían nombre, dirección, e-mail, carreras ofertadas, personal, número, presupuesto. De esas variables, hay algunas compartidas y otras no, razómn por la cual conviene definir a los dos como hijos de la categoría Uniquindianos, la cual tendría las variables nombre, dirección, e-mail.

¿Cómo hacemos el constructor en las subcategorias?
Dentro del constructor de la subcategoria ejecutamos
super().__init__(name, address, email)
para redefinir una función usamos super()METODO_PADRE() dentro del METODO_HIJO que puede tener el mismo nombre

Métodos especiales

Hay métodos especiales definidos por defecto en todos los objetos, los cuales se pueden redefinir como el __str__ o el __eq__

Redefiniendo métodos especiales en subcategorías

Ejercicios

Ejercicios

1. Realice un programa en el que defina una clase “padre” y dos clases “hijas”. En todas debe existir un constructor, hacer uso del comando super() y redefinir métodos especiales