

Sintaxis y Semántica del Lenguaje

Docentes y ayudantes

Profesores:

- Félix Paternoster: pater@frlp.utn.edu.ar
- Jérica Guzmán: jeresa.v.guzman@gmail.com

Ayudantes:

- Gabriel Scarafia: gabriel.hugo.s@gmail.com
- Lucas Bogado: lucasbogado@alu.frlp.utn.edu.ar
- Agustin Cejas: agus782009@gmail.com
- Francisco Montiron: franciscomontiron@gmail.com
- Valentina Díaz: valendiaz01@yahoo.com
- Hajime Shiroma: hajishiro.hs@gmail.com

Medios de comunicación

Clases virtuales en vivo:

Zoom <https://zoom.us/es-es/meetings.html>

Información de la cátedra:

CVG (Campus Virtual Global) de la UTN: <http://frlp.cvg.utn.edu.ar>

Pagina de la materia: <http://www.frlp.utn.edu.ar/materias/sintaxis/>

Consultas fuera de horario de clase: por mail o por CVG

Temas a Desarrollar en la materia

- TAD Tipos Abstractos de datos
- Autómatas Finitos: AFD y AFN
- Autómatas de pilas
- Gramáticas de Lenguajes

Clase 1

TAD Simple: Especificación y
aplicación

Desarrollo de la clase 1

Introducción

Nuevo paradigma para resolver problemas

Datos y comportamiento de TAD vs PYTHON

Utilización de TAD en aplicación

Ejemplo de Especificación de TAD simple

Ejemplo de aplicación utilizando TAD simple

Instalación y ambiente PYTHON

Ejercicio para resolver

Introducción

Un TAD (tipo abstracto de datos) es:

- un tipo definido por el usuario que:
 - Tiene un conjunto de valores y un conjunto de operaciones.
 - Cumple con los principios de abstracción, ocultación de la información y se puede manejar sin conocer la representación interna.

Nuevo paradigma para resolver problemas

Dado un problema, deberemos identificar los elementos que intervienen (orientado a objetos), ver cuales son los datos necesarios del elemento, que operaciones vamos a necesitar para manejar esos datos, para luego poder desarrollar un TAD, por último utilizarlo en aplicaciones

Problema

Identificación de elementos

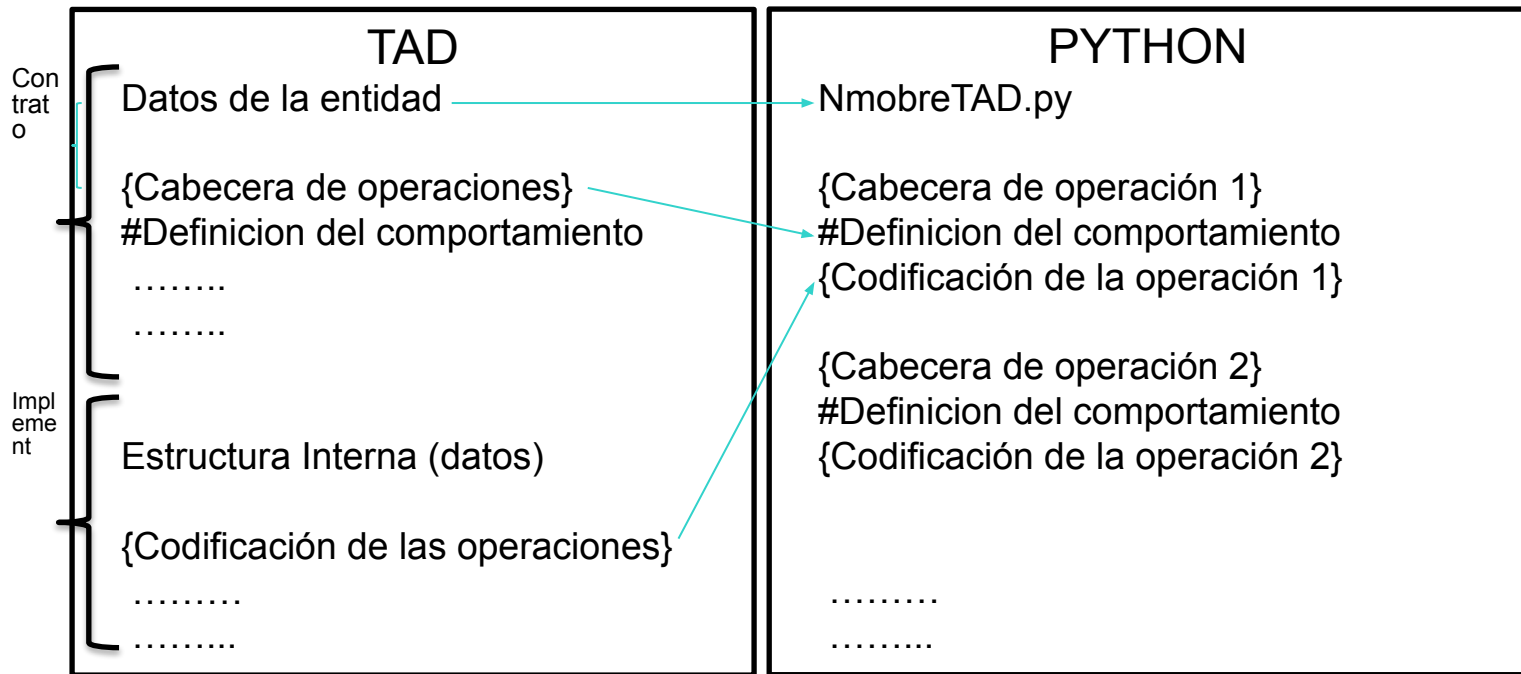
Desarrollar TAD

Desarrollar Apl utilizando TAD

Si tengo que desarrollar una aplicación, y necesito de un TAD ya desarrollado, solo tengo que utilizarlo, conociendo los datos y operaciones que maneja

Datos y comportamiento de TAD vs PYTHON

Grafico



Utilización de TAD en aplicación

Una vez desarrollado el TAD se lo puede utilizar en cualquier aplicación que lo necesite, solo se deberá importar al comienzo de la aplicación, y conocer los datos y el comportamiento que maneja el TAD

```
import TadNombre
```

```
from TadNombre import *
```

```
#Código de aplicación utilizando
```

```
#tipo de dato y operaciones del TAD
```

```
.....
```

```
.....
```

```
.....
```

Ejemplo Especificación TAD Simple

TAD Simple

Ejemplos: libro, auto, alumno, persona, etc

Cada uno de ellos tendrán sus datos en la estructura interna

Las operaciones de los tads simples por lo general van a ser las mismas:

crear()

cargar()

modificar() #para cada dato

Ver() #para cada dato

Copiar()

TAD Compuesto

Ejemplos: libreria, concesionaria, comision, empresa, etc

Son TAD de tipo colección de datos

Las operaciones de los tads compuestos por lo general van a ser las mismas:

crear()

agregar()

eliminar()

cantidad()

existe()

recuperar()

Ejemplo Especificación TAD Simple

TAD Libro

#libro=["","","","0"]

#def crearLib():
#Crea un libro vacio

#def cargarLib(libro,n,e,a,p):
#Carga los datos de un libro:
Nom, Edit, Autor, Precio

#def verNom(libro):
#Retorna el nombre de un libro

#def verEdit(libro):
#Retorna la editorial de un libro

#def verAutor(libro):
#Retorna el autor de un libro

#def verPre(libro):
#Retorna el autor de un libro

#def modNom(libro,n):
#Modifica el nombre de un libro

#def modEdit(libro,e):
#Modifica la editorial de un libro

#def modAutor(libro,a):
#Modifica el autor de un libro

#def modPre(libro,p):
#Modifica el autor de un libro

#def asignarLib(libro1,libro2):
#Asigna datos de un libro en otro

Ejemplo Aplicación utilizando TAD Simple

```
import TadLibro
```

```
from TadLibro import *
```

```
#Crea y carga los datos del libro1
l1=crearLib()
n=input("Ingrese un nombre")
e=input("Ingrese un editorial")
a=input("Ingrese un autor")
p=float(input("Ingrese un precio"))
cargarLib(l1,n,e,a,p)
```

```
#Imprime los datos del libro1
print verNom(l1)
print verEdit(l1)
print verAutor(l1)
print verPre(l1)
```

```
#Crea y carga los datos del libro2
l2=crearLib()
n=input("Ingrese un nombre")
e=input("Ingrese un editorial")
```

```
a=input("Ingrese un autor")
p=float(input("Ingrese un precio"))
cargarLib(l2,n,e,a,p)
```

```
#Imprime los datos del libro2
print verNom(l2)
print verEdit(l2)
print verAutor(l2)
print verPre(l2)
```

```
#Imprimir nombre de libro de menor precio
#Incrementa el precio en un 10%
print "El nombre del libro de menor
precio es:"
if (verPre(l1)<verPre(l2)):
    print verNom(l1)
    nuevoPre=verPre(l1)*1.1
    modPre(l1,nuevoPre)
else:
    print verNom(l2)
    nuevoPre=verPre(l2)*1.1
    modPre(l2,nuevoPre)
```

Instalación y Ambiente PYTHON

Mostrar tutorial de instalación PYTHON para Windows/Linux

Explicación del ambiente y conceptos básicos de PYTHON

Ejercicios para resolver

Resolver los ejercicios 1 y 2 del TP nro 1.