



Seminario de Lenguajes

Opción Python

Práctica 3

Parte I

Ejemplo de excepciones:

Abrir un archivo que no existe.

Acceder a una clave de diccionario que no existe .

Buscar en una lista un valor que no existe.

Invocar a un método que no existe.

Referirse a una variable que no existe.

Mezclar tipos de datos sin convertirlos previamente.

Etc.

Excepciones

Una excepción es un acontecimiento, que ocurre durante la ejecución de un programa, que interrumpe el flujo normal de las instrucciones de programa

1.- ¿Qué es una excepción? Ejemplifique. ¿Por qué es útil que un lenguaje de programación tenga soporte para manejo de excepciones?

El manejo de excepciones permite al usuario crear aplicaciones tolerantes a fallas y robustos (resistentes a errores) para controlar estas excepciones y que pueda seguir ejecutando el programa sin verse afectado por el problema.

2.- ¿Qué acción realiza Python una vez que levanta y maneja una excepción? Si no encuentra un manejador en su bloque try except, ¿dónde lo busca? ¿Y si no existe, qué sucede? Escriba un código que lo demuestre.

Si ocurre una excepción que no coincide con la excepción nombrada en el except, esta se pasa a declaraciones try de más afuera; si no se encuentra nada que la maneje, es una excepción no manejada, y la ejecución se frena con un mensaje como los mostrados arriba., se produce una excepción no manejada y me larga un error que lo puedo visualizar por pantalla y frena la ejecución del programa

3.- ¿En qué se diferencian las cláusulas else y finally del bloque try except? ¿Son obligatorias?

De un ejemplo de uso. las diferencias entre else y finally, que el else define un fragmento de código a ejecutar solo si no se ha producido ninguna excepción en el try y en cambio el finally se ejecuta siempre que se produce una excepción o no. No son obligatorias

4.- Sea una lista de N elementos realice una función que dada una posición ingresada por el teclado, imprima su elemento correspondiente en la lista. Maneje con excepciones si la posición no existe, informándolo y requiriendo que ingrese nuevamente el valor.

5.- Maneje con excepciones el ejercicio 2 de la “Práctica 2 - Continuación” para el caso en que los archivos pasados como argumento no existan.

6- Reescriba el módulo de colas FIFO de la Práctica 2 con manejo de excepciones para los casos que considere necesario (por ejemplo que levante una excepción cuando se desee quitar un elemento de la cola cuando esté vacía).

Parte II

Programación Orientada a Objetos (POO)

7.- ¿Qué son los objetos? ¿Cómo se comunican entre ellos?

Son el elemento fundamental de la POO, Son entidades que poseen estado interno y comportamiento. Es el equivalente a un dato abstracto

Un objeto es una entidad que agrupa un estado y una funcionalidad relacionadas. El estado del objeto se define a través de variables llamadas atributos, mientras que la funcionalidad se modela a través de funciones a las que se les conoce con el nombre de métodos del objeto.

El mensaje es el modo de comunicación entre los objetos.

Ejemplo: Cuando se invoca una función de un objeto, lo que se está haciendo es “enviando un mensaje” a dicho objeto.

una clase es mas que una plantilla generica a partir de la cual instanciar los objetos; plantilla que es la que define que atributos y metodos tendran los objetos de esa clase. Las clases se definen mediante la palabra clave class seguida del nombre de la clase, dos puntos(:) y a continuacion, indentado, el cuerpo de la clase. Como en el caso de las funciones (las clases deben estar definidas antes de que se utilicen), si la primera linea del cuerpo se trata de una cadena de texto, esta sera la cadena de documentacion de la clase o docstring. Con la definicion de una nueva clase se crea un nuevo namespace y se instancia un elemento de una clase realizando Ej. x = MyClass(). La operacion de instanciacion crea un objeto vacio.

8.- ¿Qué son las clases? ¿Cómo se definen en Python? ¿Cómo se inicializan los objetos de una clase? Se inicializan los objetos a partir del metodo `__init__`, que se ejecuta despues de crear un nuevo objeto a partir de la clase y como el nombre lo indica se realiza para cualquier proceso de inicializacion que sea necesario.

Hay tres conceptos que son basico para cualquier lenguaje de programacion orientado a objetos: el encapsulamiento, la herencia y el plimorfismo. El tipo de herencia utilizada en python es herencia multiple.

9.- ¿Cuáles son los tres conceptos básicos de la POO? ¿Qué tipo de herencia utiliza Python?

¿Soporta “sobrecarga” de operadores? ¿Cómo se definen los métodos y variables públicos y privados? Python presenta sobrecarga de operadores (`__add__` (para la suma), `__mul__` para la multiplicación, `__div__` para la división, `__pow__` para la potenciación, etc.)

10.- Implemente una clase Punto que represente un punto cartesiano (x, y) y que posea un método distancia() que reciba como parámetro otro punto y calcule la distancia entre ambos.

(Sean $P_1(x_1, y_1)$ y $P_2(x_2, y_2)$ la distancia entre ambos es igual a $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$)

11.- A la clase del ejercicio anterior agréguele un método que verifique la igualdad entre dos puntos. ¿Es lo mismo preguntar si dos objetos son iguales con el signo `==` que preguntándolo con el método implementado? ¿Por qué?

12.- Modelice con objetos a los alumnos y materias de la facultad. Un alumno debe tener como información mínima el nombre y apellido, número de legajo y fecha de nacimiento. Implemente un método para asignarle materias que esté cursando el alumno y aquellas que aprobó junto con su nota. Implemente un método que calcule el promedio de todas las cursadas que aprobó un alumno y luego realice un listado de aquellos alumnos que estén cursando al menos 2 materias.

A continuación se verá el diagrama UML asociado:

