

Clases y objetos.

TAREA 2

ANGIE CAROLINA CHITIVA MUÑOZ

202047919_29

Tutor

DIEGO FERNANDO LUNA CEBALLOS

**UNIVERSIDAD NACIONAL ABIERTA Y A DISTANCIA (UNAD).
ESCUELA DE CIENCIAS BÁSICAS TECNOLOGÍA E INGENIERA
TECNOLOGIA EN DESARROLLO DE SOFTWARE
BOGOTÁ
2024**

Introducción

La programación orientada a objetos (POO) es una metodología clave en el desarrollo de software, que organiza datos y comportamientos en estructuras llamadas objetos. Esta actividad tiene como objetivo aplicar conceptos fundamentales de la POO, como la abstracción, encapsulamiento, herencia y polimorfismo, en un entorno colaborativo. A través de la revisión de lecturas y la creación de diagramas de clases en Python, los estudiantes desarrollarán habilidades prácticas que les permitirán comprender mejor cómo estos conceptos se aplican en el desarrollo de software.

Contenido

Sección 1: Trabajo Colaborativo

1. Seleccionar uno de los siguientes interrogantes y publicar la pregunta con una respuesta argumentada y un ejemplo:

¿En qué consiste la sobrecarga de operadores?

Re: Foro de la actividad Tarea 2 - Clases y objetos
de ANGIE CAROLINA CHITIVA MUNOZ - miércoles, 18 de septiembre de 2024, 22:44

Cordial saludo

Conforme a lo que investigué y analicé, dare respuesta a la pregunta ¿En qué consiste la sobrecarga de operadores?

Con respecto a la pregunta sobre la **sobrecarga de operadores**, te comento que este es un concepto de la programación orientada a objetos que permite redefinir cómo funcionan los operadores comunes (como +, -, *, etc.) cuando se aplican a objetos creados a partir de nuestras propias clases. En lugar de limitarse a los tipos de datos básicos (como números o cadenas), podemos hacer que estos operadores realicen acciones específicas en objetos complejos.

Por ejemplo, si tenemos una clase que representa cajas con manzanas, podríamos sobrecargar el operador + para que cuando sumemos dos objetos de la clase **Caja**, el resultado sea una nueva caja con el total de manzanas de ambas cajas.

Bibliografía

Cursa, (n.d.). **Programación orientada a objetos en Python: Sobrecarga de operadores en Python**. Cursa. <https://cursa.app/es/pagina/programacion-orientada-a-objetos-en-python-sobrecarga-de-operadores-en-python>

Lutz, M. (2013). *Python 3*. O'Reilly Media. https://www.google.com.co/books/edition/Python_3/KRYyvkMzvpwC?hl=es-419&gbpv=1&dq=en+qu%C3%A9+consiste+la+sobrecarga+de+operadores+python&pg=PA283&printsec=frontcover

Meyers, R. (2018). *Comenzar a programar con Python 3*. Anaya Multimedia. https://www.google.com.co/books/edition/Comenzar_a_programar_con_Python_3/5A7PDwAAQBAJ?hl=es-419&gbpv=1&dq=en+qu%C3%A9+consiste+la+sobrecarga+de+operadores+python&pg=PA161&printsec=frontcover

2. Comentar al menos una respuesta de un compañero (respetuoso y argumentado).

Re: Foro de la actividad Tarea 2 - Clases y objetos
de JOHN JAIRO LEON GUERRERO - viernes, 13 de septiembre de 2024, 15:27

Buen día Diego

Muchas gracias por compartir el documento en Google Drive. Me parece una excelente idea para organizar nuestro trabajo en equipo. En cuanto a tu explicación sobre la abstracción, coincido plenamente en que es uno de los pilares fundamentales de la programación orientada a objetos. Me parece muy claro el ejemplo del taladro que utilizaste. Un aspecto interesante adicional es cómo la abstracción no solo nos ayuda a ocultar la complejidad del objeto, sino que también facilita la reutilización de código y la colaboración entre desarrolladores, permitiendo que cada uno trabaje en distintos niveles de detalle sin afectar el funcionamiento general.

Cordialmente,
John Leon

[Enlace permanente](#) [Mostrar mensaje anterior](#) [Responder](#)

Re: Foro de la actividad Tarea 2 - Clases y objetos
de ANGIE CAROLINA CHITIVA MUNOZ - jueves, 19 de septiembre de 2024, 00:00

Cordial saludo,

Gracias, Diego, por compartir el documento en Google Drive, es una excelente forma de organizar nuestro trabajo en equipo. También he realizado algunas modificaciones visuales en las diapositivas para mejorar la presentación.

Tu explicación sobre la abstracción es muy clara, y el ejemplo del taladro ilustra bien su importancia en la POO. Quiero mencionar la relevancia del encapsulamiento, que protege los datos internos de una clase.

John, estoy de acuerdo en que la abstracción no solo oculta la complejidad, sino que también facilita la reutilización de código y la colaboración.

Gracias a ambos por sus aportes. Estoy emocionada por seguir colaborando.

[Enlace permanente](#) [Mostrar mensaje anterior](#) [Responder](#)

3. Desarrollar una presentación electrónica con la siguiente estructura:

- **Diapositiva 1: Portada**
- **Diapositivas 2 - 6: Desarrollo temático**
- **Diapositiva 7: Referencias**

<https://docs.google.com/presentation/d/1X4Cow0d7a9QKm-6VaAy1BwlfWEG8nz04yv3j5K9V2LE/edit?usp=sharing>

4. Sección 2: Trabajo Individual

- Requisitos Iniciales:
 - Instalar Python y el entorno de desarrollo (Visual Studio Code).
 - Instalar StarUML y los componentes necesarios.



4. Seleccionar un diagrama de clases (uno por integrante) y publicarlo en el foro.

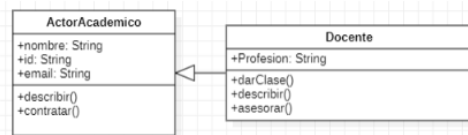


Re: Foro de la actividad Tarea 2 - Clases y objetos
de ANGIE CAROLINA CHITIVA MUNOZ - miércoles, 18 de septiembre de 2024, 23:16

Cordial saludos tutor y compañeros:

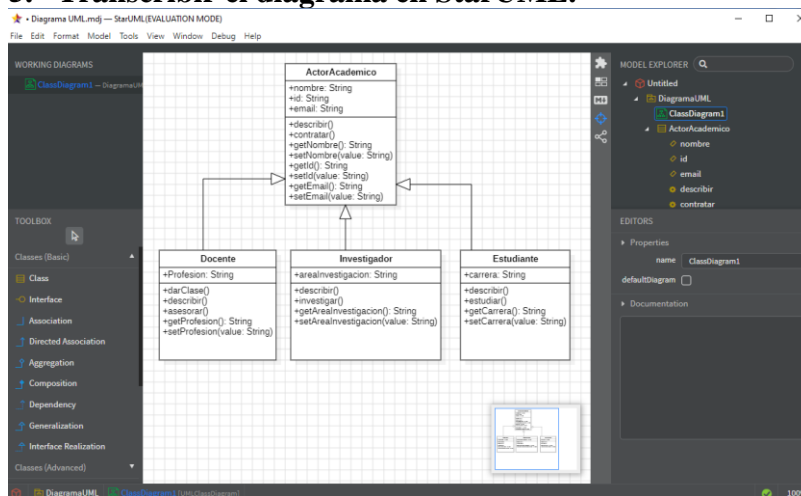
Quiero informarles que he realizado las modificaciones necesarias en el documento de Google Drive para desarrollar las diapositivas correspondientes a nuestra entrega grupal. Pueden acceder al documento a través del siguiente enlace: [Diapositiva de entrega grupal](#).

Además, quisiera informarles que, de manera individual, desarrollaré el diagrama de clases de tipos de actores académicos.

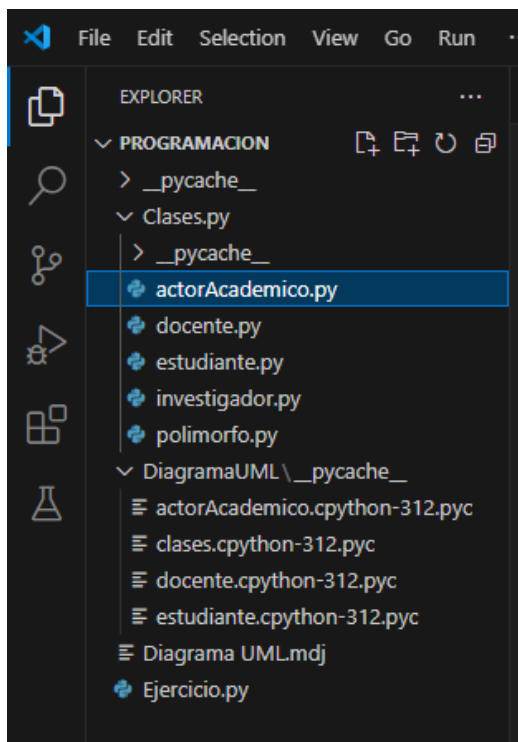


[Enlace permanente](#) [Mostrar mensaje anterior](#) [Responder](#)

5. Transcribir el diagrama en StarUML.



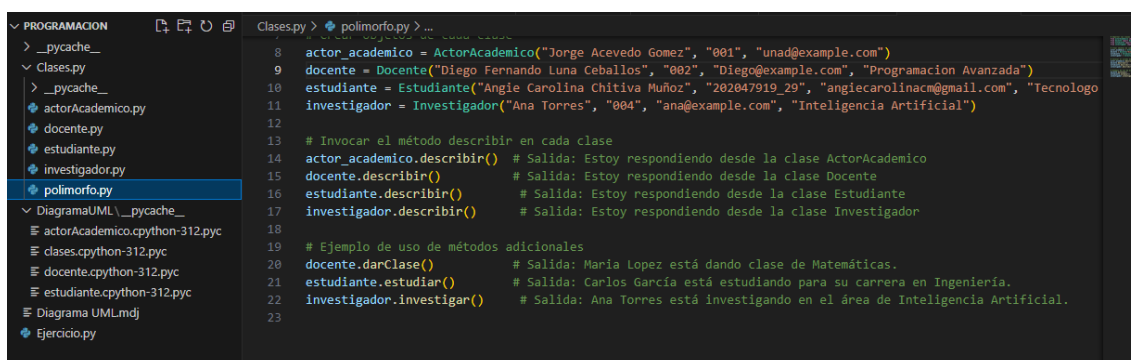
3. Exportar el código de las clases a Python.



4. Implementar el método describir().

```
def describir(self):  
    print("Estoy respondiendo desde la clase ActorAcademico")
```

5. Crear un archivo polimorfo.py con objetos de cada clase e invocar el método describir().



6. Copiar todos los archivos generados en un directorio con su nombre, comprimirlos y enviarlos.

<https://github.com/Carolinacm7/object-oriented-programming>