|  |  |  |
| --- | --- | --- |
| **Comparison type** | **ADT (Abstract Data Type)** | **Encapsulation in OOP (Object-Oriented Programming):** |
| **Definition:** | An ADT is a high-level description of a set of operations that can be performed on a data structure, abstracting away the implementation details. It focuses on what operations can be performed, not how they are implemented. | Encapsulation is a fundamental concept in OOP that involves bundling data (attributes) and the methods (functions) that operate on that data into a single unit, known as a class. It emphasizes the organization and access control of the internal structure of a class. |
| **Scope:** | ADTs are a broader concept that can be applied in various programming paradigms, not just limited to OOP. | Encapsulation is specific to the OOP paradigm and is a core principle in languages like Java, C++, and Python. |
| **Focus:** | ADTs focus on defining operations and their behavior, without specifying how these operations are implemented. | Encapsulation focuses on bundling data and methods within a class, promoting data hiding and access control. |
| **Implementation Details:** | ADTs abstract away the implementation details, providing a high-level interface. Users of an ADT are concerned with what operations are available, not how they are achieved. | Encapsulation allows for hiding the internal implementation details of a class from the external world. Users interact with the class through a well-defined interface, and the internal details can be changed without affecting the external code. |
| **Language Support:** | ADTs can be implemented in various programming paradigms, including procedural programming. | Encapsulation is a core concept in OOP languages and is an integral part of class-based development. |
| **Relationship** | Encapsulation is a technique used in implementing ADTs to achieve the abstraction of data and operations. | Encapsulation is a broader concept that goes beyond ADTs and is a foundational principle in organizing code in OOP. |
| **Access Control:** | ADTs may not explicitly enforce access control mechanisms, as their focus is on defining operations. | Encapsulation inherently involves access control by specifying the visibility of attributes and methods, allowing for public and private access. |