Design Patterns Comparison: Java vs Python

Design Pattern	Category	Java Implementation	Python Implementation	Advantages & Disadvantage
Factory	Creational	Uses interfaces and concrete factory classes	Uses factory functions or factory classes	Java enforces strong typing and interfaces; Python with easier implementation.
Singleton	Creational	Uses private static instance with getInstance()	Uses __new__() or metaclass	Java requires explicit static management; Python all and decorators for simplicity.
Builder	Creational	Uses Builder inner class for object construction	Chained methods returning self	Java uses inner builder classes; Python uses chai simpler construction.
Abstract Factory	Creational	Factory of factories; creates different factory classes for object creation	Factory of factories implemented using functions or metaclasses	Java uses strong typing and interfaces; Python's simplifies implementation.
Facade	Structural	Centralized class exposing subsystem functionalities	Simple class that aggregates subsystems	Java uses structured class-based design; Python is module-based implementation.
Decorator	Structural	Extends base class, wraps components	Uses @decorator for function/class wrapping	Java uses inheritance; Python provides built-in de making it more concise.
Adapter	Structural	Creates adapter class implementing target interface	Adapter class modifies method signatures dynamically	Java enforces strict interface contracts; Python dy function calls.
Flyweight	Structural	Shares objects instead of creating new ones	Caches objects to prevent redundancy	Java maintains strict memory management; Python reuse automatically.
Composite	Structural	Tree structure using composite and leaf components	Uses nested lists/dictionaries or OOP hierarchy	Java enforces OOP structure; Python allows mo built-in data structures.

Design Patterns Comparison: Java vs Python

Delegation

Design Patterns Comparison: Java vs Python

Behavioral

Design Patterns Comparison: Java vs Python

Uses composition over inheritance

Design Patterns Comparison: Java vs Python

Uses composition instead of inheritance

Design Patterns Comparison: Java vs Python

				Java requires explicit interfaces for delegation; dynamic delegation via composition.
Iterator	Implements Iterable and Iterator interfaces	Behavioral	Uses __iter__ and __next__ methods	Java requires interface implementation; Python has protocols (__iter__, __next__).
Strategy	Uses interface for algorithm selection	Behavioral	Stores functions/objects as strategies	Java uses interfaces and polymorphism; Pytho flexibility using functions as strategies.
Observer	Observable class notifies observers	Behavioral	Subject class updates registered observers	Java uses Observable classes; Python simplifies v observers.
Template	Defines abstract class with template method calling subclass implementations	Behavioral	Defines base class with common method structure and allows subclasses to override steps	Java requires explicit abstract methods; Python allow overriding.
Model-View-Controller (MVC)	Separates model, view, and controller classes	nan	Separates model, view, and controller using Flask/Django	Java follows MVC strictly; Python has built-in frame MVC.