

Design Patterns in Python

Learn via: **Classroom / Virtual Classroom / Online**

Duration: **1 Day**

<https://bilginc.com/en/training/design-patterns-in-python-3190-training/>

Overview

The participants can follow all steps directly on their computers. There are exercises at the end of each unit providing ample opportunity to apply the freshly learned knowledge.

Every participant receives comprehensive PDF that cover the whole course content as wells as all source codes and used software.

Prerequisites

A basic understanding of python is helpful but not required.

Who Should Attend

Individuals who want to design and build large-scale systems

What You Will Learn

"It's Easier To Ask For Forgiveness Than Permission (Efap)"

One pythonic principle is "It's easier to ask for forgiveness than permission (EFAP)". Opposed to the approach to look before you leap, this principle states that you should first try an action and if it fails react appropriately. Python' strong exception handling supports this principle and helps to develop robust and fault tolerant programs.

Meta Classes

Meta classes are an advanced topic of Python programming. Applying meta calsses Complex tasks may be be solved in an elegant manner. The use of meta classes is demonstrated with examples.

Singelton

Singeltons are objects of which only one instance is supposed to exist. Python provides several ways to implement singeltons. These possibilities are shown using examples.

Null Objects

Null objects can be used instead of the type None to avoid tests for None. Implementation, usage as well as advantages and disadvantages are covered.

Proxy

Proxies stand for other objects. Setup and usage of proxies are covered.

Observer

The observer pattern allows several objects to have access to the same data. The principles of this pattern are shown with a comprehensive example.

Parameters of constructors are often assigned to instance variables. This pattern can replace a many lines of manual assignment with only one line of code.

Outline

Special Features Of Design Patterns In Python

In Python many problems can be solved more easily than in other languages. Therefore, several design patterns are not necessary or they are already implicitly contained in the languages.

The Principles of writing pythonic programs are explained and supported with examples. Topics such as beauty of source code, explicit programming, simplicity, readability, and exception handling are included.