

Architekturpatterns mit Python

Test-Driven Development, Domain-Driven Design und Event-Driven Microservices
praktisch umgesetzt

Bewährte Patterns für komplexe Python-Projekte bekannte Architekturpatterns – endlich in idiomatischem Python die Komplexität anspruchsvoller Projekte erfolgreich managenden größten Nutzen aus den Testsuiten herausholen Mit der wachsenden Beliebtheit von Python werden auch die realisierten Projekte größer und komplexer. Viele Python-Entwicklerinnen und -Entwickler interessieren sich deshalb zunehmend für Software-Design-Patterns wie zum Beispiel hexagonale Architektur/Clean Architecture, eventgesteuerte Architektur und die strategischen Patterns, die durch das Domain-Driven Design nahegelegt werden. Aber das Übertragen dieser Patterns auf Python ist nicht immer einfach. Mit diesem praxisnahen Buch stellen Harry Percival und Bob Gregory von MADE.com erprobte Architekturpatterns vor, die dabei helfen, die Komplexität von Anwendungen im Griff zu behalten – und möglichst viel aus den Testsuiten herauszuholen. Jedes Pattern wird durch konkrete Beispiele in schönem, idiomatischem Python illustriert – wobei die Weitschweifigkeit von Java- oder C#-Syntax vermieden wird. Im Buch behandelte Patterns: Dependency Inversion und die Verbindungen zu Ports und Adaptern (hexagonale Architektur/Clean Architecture) Unterscheidung im Domain-Driven Design zwischen Entitäten, Value Objects und Aggregaten Repository- und Unit-of-Work-Patterns für persistenten Speicher Events, Befehle und der Message Bus Command-Query Responsibility Segregation (CQRS) Eventgesteuerte Architektur und reaktive Microservices

Bewährte Patterns für komplexe Python-Projekte

- bekannte Architekturpatterns – endlich in idiomatischem Python
- die Komplexität anspruchsvoller Projekte erfolgreich managen
- den größten Nutzen aus den Testsuiten herausholen

Mit der wachsenden Beliebtheit von Python werden auch die realisierten Projekte größer und komplexer. Viele Python-Entwicklerinnen und -Entwickler interessieren sich deshalb zunehmend für Software-Design-Patterns wie zum Beispiel hexagonale Architektur/Clean Architecture, eventgesteuerte Architektur und die strategischen Patterns, die durch das Domain-Driven Design nahegelegt werden. Aber das Übertragen dieser Patterns auf Python ist nicht immer einfach.

Mit diesem praxisnahen Buch stellen Harry Percival und Bob Gregory von MADE.com erprobte Architekturpatterns vor, die dabei helfen, die Komplexität von Anwendungen im Griff zu behalten – und möglichst viel aus den Testsuiten herauszuholen. Jedes Pattern wird durch konkrete Beispiele in schönem, idiomatischem Python illustriert – wobei die Weitschweifigkeit von Java- oder C#-Syntax vermieden wird.

Im Buch behandelte Patterns:

- Dependency Inversion und die Verbindungen zu Ports und Adaptern (hexagonale Architektur/Clean Architecture)
- Unterscheidung im Domain-Driven Design zwischen Entitäten, Value Objects und Aggregaten
- Repository- und Unit-of-Work-Patterns für persistenten Speicher
- Events, Befehle und der Message Bus
- Command-Query Responsibility Segregation (CQRS)
- Eventgesteuerte Architektur und reaktive Microservices

Bewährte Patterns für komplexe Python-Projekte- bekannte Architekturpatterns , endlich in idiomatischem Python- die Komplexität anspruchsvoller Projekte erfolgreich managenden größten Nutzen aus den Testsuiten herausholen Mit der wachsenden Beliebtheit von



36,90 €
34,49 € (zzgl. MwSt.)

sofort versandfertig, Lieferzeit: 1-3
Werktage

Artikelnummer: 9783960091653
Medium: Buch
ISBN: 978-3-96009-165-3
Verlag: dpunkt.Verlag
Erscheinungstermin: 02.09.2021
Sprache(n): Deutsch
Auflage: 1. Auflage 2021
Serie: Animals
Produktform: Kartoniert
Gewicht: 564 g
Seiten: 302
Format (B x H): 169 x 238 mm

