

"Myślenie obiektowe w programowaniu" – Matt Weisfeld

To świetna książka dla początkujących i średniozaawansowanych programistów, która wprowadza w zasady programowania obiektowego, takie jak enkapsulacja, dziedziczenie, polimorfizm i kompozycja. Książka kładzie nacisk na zrozumienie myślenia w kategoriach obiektów i dostarcza praktycznych przykładów, jak te zasady stosować w rzeczywistych projektach. Jest dostępna w wydaniu polskim, co czyni ją łatwą w odbiorze dla osób preferujących naukę w języku ojczystym.

"Head First Design Patterns" (2nd Edition) – Eric Freeman i Kathy Sierra

To jedna z najczęściej polecanych książek, która w przystępny sposób wprowadza do wzorców projektowych i OOP. Dzięki licznym przykładom z życia codziennego książka wyjaśnia, jak używać obiektowego podejścia w praktyce. Jest idealna zarówno dla początkujących, jak i średniozaawansowanych programistów.

"Clean Code: A Handbook of Agile Software Craftsmanship" – Robert C. Martin

Książka skupia się na pisaniu czystego, zrozumiałego i łatwego w utrzymaniu kodu. Choć nie jest to książka stricte o OOP, to zawiera liczne wskazówki dotyczące stosowania dobrych praktyk obiektowych i zasad SOLID, które są kluczowe dla projektowania zorientowanego obiektowo.

"Design Patterns: Elements of Reusable Object-Oriented Software" – Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides

Książka znana jako "Gang of Four" (GoF), jest klasykiem w dziedzinie wzorców projektowych. Omawia 23 wzorce projektowe, które ułatwiają budowanie skalowalnych i elastycznych aplikacji obiektowych. Idealna dla bardziej zaawansowanych programistów.

"Object-Oriented Thought Process" – Matt Weisfeld

Świetna książka dla osób, które chcą lepiej zrozumieć, jak myśleć w kategoriach obiektów. Książka w przystępny sposób wyjaśnia podstawy OOP, takie jak dziedziczenie, polimorfizm, i enkapsulacja, z przykładami jak je stosować w rzeczywistych projektach.

"Refactoring: Improving the Design of Existing Code" – Martin Fowler

Ta książka pokazuje, jak poprawiać istniejący kod obiektowy, aby był bardziej czytelny, wydajny i łatwiejszy do utrzymania. Zawiera liczne techniki refaktoryzacji, które można zastosować w projektach obiektowych.

Modelowanie

1. UML 2.0 w akcji. Od analizy do modelowania obiektowego" – Hans-Erik Eriksson, Magnus Penker

Książka skupia się na języku UML, który jest standardowym narzędziem do modelowania obiektowego. Zawiera przykłady wykorzystania UML 2.0 w praktyce oraz opisuje, jak za pomocą tego języka modelować różne aspekty systemów obiektowych, w tym grafy i relacje między obiektami.

2. "Projektowanie obiektowe w języku Python" – Steven F. Lott

Ta książka omawia nie tylko podstawy programowania obiektowego w Pythonie, ale także modelowanie złożonych struktur danych przy użyciu klas i obiektów. Autor

prezentuje, jak implementować wzorce projektowe oraz jak modelować rzeczywiste systemy z użyciem Pythona.

3. **"Design Patterns: Elements of Reusable Object-Oriented Software"** – Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides

Książka klasyczna, znana jako "Gang of Four", która wprowadza w świat wzorców projektowych, co jest ważnym aspektem modelowania obiektowego. Choć nie jest specyficzna dla Pythona, wiele wzorców przedstawionych w książce można z łatwością zastosować w Pythonie do tworzenia elastycznych i łatwo rozszerzalnych struktur obiektowych.

4. **"Python 3 Object-Oriented Programming"** – Dusty Phillips

To idealna książka do nauki modelowania obiektowego w Pythonie. Phillips szczegółowo omawia, jak modelować obiekty i ich interakcje w Pythonie, stosując zasady takie jak dziedziczenie, polimorfizm i kompozycja. Zawiera również przykłady z modelowaniem bardziej złożonych systemów.

5. **"Obiektowe modelowanie i projektowanie systemów"** – Grady Booch

Książka autorstwa jednego z twórców UML, który również stworzył wiele pojęć używanych w modelowaniu obiektowym. Omawia, jak projektować duże systemy obiektowe i jakie techniki stosować, by modelowanie było wydajne i elastyczne.

Te książki wprowadzą Cię w świat modelowania obiektowego, które jest podstawą w nowoczesnym projektowaniu oprogramowania, a także pozwolą zrozumieć, jak przenieść teoretyczne modele na kod w Pythonie.