

Wirfs-Brocks 방법

Wirfs-Brock 방법론은 Rebecca Wirfs-Brock이 개발한 객체지향 설계 방법론으로, ****Responsibility-Driven Design (RDD)****를 중심으로 합니다. 이 방법론은 객체의 책임을 명확히 정의하고, 객체 간 협력을 통해 시스템을 설계하는 데 초점을 둡니다. 이는 객체지향 설계에서 중요한 역할을 하며, 현대적인 소프트웨어 개발에 널리 영향을 미쳤습니다.

Wirfs-Brock 방법론의 주요 특징

1. 책임 중심 설계:

- 객체의 역할과 책임을 명확히 정의하여 설계를 시작합니다.
- 각 객체는 특정 작업을 수행하거나 정보를 관리하는 책임을 가집니다.
- 객체 간 협력(collaboration)을 통해 시스템의 전체 목표를 달성합니다.

2. 객체의 역할 및 협력:

- 객체는 특정 역할(role)을 수행하며, 다른 객체와 명확히 정의된 방식으로 상호작용합니다.
- 객체 간의 협력은 계약(contract) 형태로 이루어지며, 이를 통해 시스템이 동작합니다.

3. 유연성과 재사용성:

- 책임 중심 설계는 시스템의 유연성을 높이고, 객체를 재사용할 수 있도록 돕습니다.
- 설계 단계에서 변화 가능성을 고려하여 확장성과 유지보수성을 강화합니다.

4. 클래스-책임-협력 카드 (CRC Cards):

- CRC 카드(Class-Responsibility-Collaboration)는 객체의 책임과 협력을 시각적으로 표현하는 도구입니다.
- 각 클래스의 이름, 책임, 협력 관계를 간단히 기록하여 설계를 명확히 합니다.

Wirfs-Brock 방법론의 절차

1. 객체 식별 및 역할 정의

- 시스템에서 필요한 객체를 식별하고, 각 객체가 수행해야 할 역할과 책임을 정의합니다.
- 예: "사용자 관리"라는 역할을 가진 객체는 사용자 정보를 추가하고 삭제하는 책임을 가질 수 있습니다.

2. 책임 할당

- 각 객체에 적합한 책임을 할당합니다.
- 책임은 크게 두 가지로 나뉩니다:
 - **정보 유지 책임:** 데이터를 저장하고 관리하는 역할.
 - **행위 수행 책임:** 특정 작업이나 기능을 수행하는 역할.

3. 협력 관계 정의

- 객체 간의 상호작용과 협력 관계를 정의합니다.
- 협력은 메시지 전달 형태로 이루어지며, 이를 통해 시스템이 동작합니다.

4. 설계 모델링

- 클래스 다이어그램과 시퀀스 다이어그램을 사용하여 설계를 시각적으로 표현합니다.
- UML(Unified Modeling Language)을 활용하여 구조와 동작을 문서화할 수 있습니다.

Wirfs-Brock 방법론의 장점

1. 책임 중심 접근법:

- 시스템 설계를 사용자 요구사항과 비즈니스 로직에 맞게 구조화할 수 있습니다.
- 복잡한 시스템에서도 명확한 역할 분담으로 이해도를 높입니다.

2. 유연성과 확장성:

- 객체 간 협력을 기반으로 설계하므로 변화와 확장이 용이합니다.
- 재사용 가능한 모듈화를 지원합니다.

3. CRC 카드 활용:

- 간단하면서도 강력한 도구로 팀 간 의사소통을 개선하고 설계를 명확히 합니다.

Wirfs-Brock 방법론과 UML

Rebecca Wirfs-Brock은 UML 발전에 영향을 미쳤으며, 특히 UML에서 사용하는 "스테레오타입 (stereotypes)" 개념은 그녀가 1992년에 제안한 "객체 역할 유형(Object Role Stereotypes)"에서 기원했습니다^{[1] [2]}. 또한 CRC 카드는 UML 다이어그램과 함께 사용되어 설계를 보완할 수 있습니다.

Wirfs-Brock 방법론의 활용 사례

1. 소규모 및 대규모 프로젝트:

- 소규모 프로젝트에서는 간단한 CRC 카드로 설계를 시작할 수 있으며, 대규모 프로젝트에서는 UML 다이어그램과 결합하여 복잡한 시스템을 모델링할 수 있습니다.

2. 애자일 개발 환경:

- 애자일 팀에서 유연한 설계를 지원하며, 요구사항 변화에 쉽게 대응할 수 있습니다^[3].

Rebecca Wirfs-Brock의 Responsibility-Driven Design은 객체지향 설계에서 중요한 기여를 했으며, 현대적인 소프트웨어 개발에서 여전히 널리 사용되는 접근법입니다.

✻

1. https://en.wikipedia.org/wiki/Rebecca_Wirfs-Brock
2. <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=5f7c86263a26c2057bba5019efd0f668e1ac4931>
3. <https://www.infoq.com/interviews/agile-design-architecture/>