

차고문 개폐 시스템 설명서

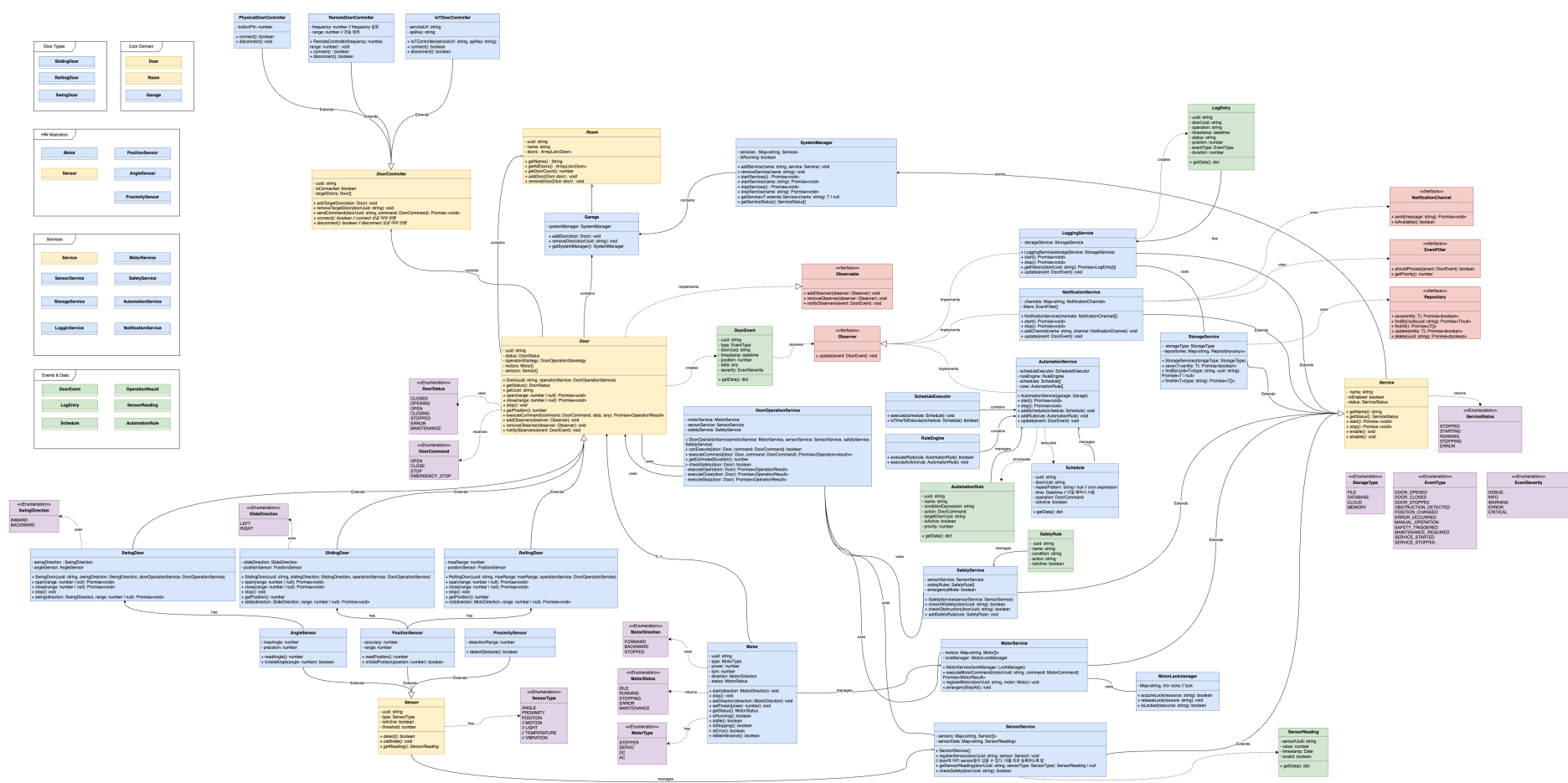


성명: 김신건

작성일: 2025.09.24

Document Link: <https://singun11.notion.site/kmu-sw-architecture-hw-1>

UML Link : https://drive.google.com/file/d/18iaagKvKSJml_RF_T4VWJhgoQoRLSYpF/view



- Actor : User, Remote Controller, Actuator
- Garage System 내의 Use Case
 - Open the door
 - Close the door
 - Stop the door

요구사항

- 차고지 개폐 시스템에 필요한 interface, abstract class, class를 설계한다.
- 위의 Use Case를 기본적으로 만족해야 한다.

추가적인 상황 부여

- Door 라는 Abstract 클래스를 통해 문의 종류를 다양하게 고려한다.
- 옆으로 열릴 수도, 위로 열릴 수도, 앞뒤로 밀거나 당기면서 열릴 수도 있다.
- 양옆으로 열리는 문일 수도 있다.
- 차고지에 문이 여러 개 있을 수 있다.
- 문을 닫거나 여는 과정 중에 중간에 사용자가 임의로 멈출 수도 있다.
- 사용자가 열린 문의 정도(%)를 알 수 있다.
- 장애물을 감지하면 개폐를 중단해야 하며, 이 또한 기록되고 알림이 가야 한다.
- 문의 작동에 대해 센서를 연결해서 감지해야 할 수 있다.
- 문을 조작함에 있어서 여러 방식이 있을 수 있다.
 - 물리 버튼을 통해 연결될 수도 있다.
 - 리모트 컨트롤러(리모콘)로 적외선 신호를 써서 연결될 수도 있다.
 - IoT 기반 서버를 통해 Wifi 신호를 받아 연결될 수도 있다.
- 따라서, 문과 문을 조작하는 요소 Controller 사이에 확장가능한 요소가 있어야 한다.
- 일정 시간에 문을 열게 하는 등의 자동화 시스템을 고려할 수 있다.

- 문의 개폐 등을 기록하는 Logger와, 이를 저장하는 시스템이 있을 수 있다.
- 문의 개폐가 일어날 때 특정 이벤트를 사용자에게 알리는 Notification 시스템이 있을 수 있다.

클래스 설명서

■ 분류	■ 클래스명	■ 클래스 역할 (R&R)	■ 클래스 설명 / Functional Requirements	■ 연결된 Class에서의 역할	■ Non-Functional Requirement
Coor Domain	<u>Door</u>	모든 문의 공통 인터페이스와 기본 동작을 정의하는 추상 클래스	Observer 패턴의 Observable 역할을 하며, 문의 상태 관리와 이벤트 알람을 담당	Observable (Observer Pattern) - 상태 변화를 관찰자들에게 알림 Template Method Pattern - 구체적인 동작은 하위 클래스에서 구현 Polymorphism (GRASP) - 다형성을 통한 일관된 인터페이스 제공	- 안정성: 동작 중 오류 발생 시 안전한 상태로 복구 - 응답성: 명령 수행 시 1초 이내 응답 - 확장성: 새로운 문 타입 추가 용이 - 유지보수성: 명확한 책임 분리
Coor Domain	<u>Room</u>	- 문을 포함하는 공간의 추상적 개념을 정의 - 문 관리 기능 제공	물리적 공간을 나타내며 여러 문을 관리하는 컨테이너 역할 - 문 추가/제거 관리 - 문 조회 기능 - 공간 식별자 관리 - 문 개수 추적	Composite Pattern - Door 객체들의 컨테이너 Information Expert (GRASP) - 자신이 포함한 문들에 대한 정보 제공 Controller (GRASP) - 문 관리에 대한 제어 역할	- 성능: 문 검색 시 O(1) 복잡도 (Map 사용)
Coor Domain	<u>Garage</u>	차고 공간 구현	Room을 상속받아 차고 특화 기능을 제공하며 SystemManager와 결합 - 시스템 관리자 통합 - 차고 특화 문 관리 - 시스템 상태 모니터링	Facade Pattern - SystemManager를 통한 복잡한 서비스 접근 단순화 Creator (GRASP) - Door 객체 생성과 관리 Information Expert (GRASP) - 차고 정보 제공	- 통합성: 모든 서비스와 원활한 연동 - 모니터링: 실시간 상태 추적 - 확장성: 다중 차고 확장 가능
Door Types	<u>SlidingDoor</u>	슬라이딩 방식의 문 동작을 구현	슬라이딩 방식의 문으로 위치센서를 통한 정밀한 위치를 제공한다. - 좌우 슬라이딩 동작 - 슬라이드 방향 제어 - 위치 센서를 통한 정확한 위치 추적 - 장애물 감지 시 자동 정지	Template Method Pattern - Door의 추상 메서드 구현	- 안정성: 장애물 감지 시 즉시 정지
Door Types	<u>RollingDoor</u>	롤링 방식의 문 동작을 구현	상하로 롤링하는 방식의 문으로 높이 제한과 위치 센서를 통한 통합 제어를 제공한다. - 상하 롤링 동작 - 최대 높이 제한 관리 - 위치 센서를 통한 높이 추적	Template Method Pattern - Door의 추상 메서드 구현	- 안정성: 장애물 감지시 즉시 정지
Door Types	<u>SwingDoor</u>	스윙 방식의 문 동작을 구현	최전축을 중심으로 안쪽/바깥쪽으로 회전하는 문으로 각도 센서를 통한 제어를 제공한다.	Template Method Pattern - Door의 추상 메서드 구현	- 안정성: 장애물 감지시 즉시 정지
Service Layer	<u>Service</u>	모든 서비스의 공통 생명주기와 상태 관리를 정의하는 추상 클래스	서비스의 시작, 정지, 상태 관리 등 공통 기능을 제공하는 클래스 - 서비스의 시작, 정지 - 서비스의 활성화 / 비활성화 - 상태 모니터링 - 생명주기 관리	Template Method Pattern - 서비스 생명주기 템플릿 제공 State Pattern - 서비스 상태 관리 Polymorphism (GRASP) - 일관된 서비스 인터페이스	- 모니터링: 실시간 상태 추적 - 신뢰성: 서비스 장애시 자동 복구 - 확장성: 새로운 서비스 유형 추가 용이
Service Layer	<u>SystemManager</u>	모든 시스템 서비스를 통합 관리하는 중앙 관리 클래스	서비스들의 생명주기를 관리하고 의존정을 조정하는 클래스 - 서비스 등록 / 해제 - 서비스 생명주기 관리 - 서비스 간 의존성 관리 - 시스템 상태 모니터링	Facade Pattern - 복잡한 서비스들에 대한 단순한 인터페이스 제공 Mediator Pattern - 서비스 간 상호작용 조정 Controller (GRASP) - 시스템 제어 역할	- 확장성: 동적 서비스 추가/제거

■ 분류	■ 클래스명	■ 클래스 역할 (R&R)	■ 클래스 설명 / Functional Requirements	■ 연결된 Class에서의 역할	■ Non-Functional Requirement
				Creator(GRASP) - 서비스 객체 생성 관리	
Service Layer	<u>SensorService</u>	모든 센서 데이터를 통합 관리하고, 실시간 감지 기능을 제공하는 서비스	위치, 각도, 근접 센서 등을 관리하여 문 동작을 지원한다. - 센서 등록 / 관리 - 실시간 센서 데이터 수집 - 센서 데이터 검증 - 안정성 검사 지원	Information Expert(GRASP) - 센서 데이터에 대한 전문 지식 Low Coupling - 센서 하드웨어와 비즈니스 로직을 분리 Service Integrations - 다양한 센서를 통합 관리	- 실시간성: 실시간 센서 데이터 처리
Service Layer	<u>MotorService</u>	모터 제어와 동시성 관리를 담당하는 서비스	모터 명령 실행, 동시성 제어, 비상 정지 등 모터 관련 모든 기능을 통합 관리 - 모터 명령 실행 - 동시성 제어 (LockManager) - 비상 정지 기능 - 모터 상태 모니터링	Command Pattern - 모터 명령 캡슐화 Singleton Pattern - LockManager 통한 동시성 제어 Information Expert (GRASP) - 모터 제어 전문 지식	- 동시성: 다중 모터 안전한 동시 제어 - 응답성: 명령 실행 지연 최소화 - 안전성: 비상 정지 우선 처리
Service Layer	<u>SafetyService</u>	시스템 전반의 안전성을 보장하는 안전 서비스	안전 규칙 관리, 장애물 감지, 비상 모드 등을 통해 차고 개폐 시스템 안전성 확보 - 안전 규칙 관리 - 장애물 감지 - 비상 모드 전환 - 안전성 검증	Chain of Responsibility Pattern - 안전 규칙 체인 검사 State Pattern - 일반/비상 모드 관리 Protection Proxy (GRASP) - 안전하지 않은 동작 차단	- 신뢰성: 100% 안전 검사 수행 - 실시간성: 즉시 위험 감지 및 대응
Service Layer	<u>StorageService</u>	다양한 저장소 타입을 통합 관리하는 데이터 저장 서비스	파일, 데이터베이스, 클라우드 등 다양한 저장소를 추상화하여 일관된 데이터 저장 인터페이스 제공 - 다중 저장소 지원 - 데이터 CRUD 작업 - 저장소 타입별 최적화 - 데이터 백업 및 복구	Repository Pattern - 데이터 저장소 추상화 Factory Pattern - 저장소 타입별 Repository 생성 Information Expert (GRASP) - 데이터 저장 전문 지식	- 성능: 빠른 데이터 액세스 - 신뢰성: 데이터 무결성 보장 - 확장성: 다양한 저장소 지원 - 백업: 자동 백업 및 복구
Service Layer	<u>LoginService</u>	시스템 전반의 로그를 수집/저장/관리하는 서비스	- 모든 시스템 활동을 기록하여 모니터링, 디버깅, 감사 기능을 제공 - Door에서 발생하는 DoorEvent를 관찰하여 LogEntry로 변환/저장 - DoorEvent 수신 및 LogEntry 변환 - 로그 데이터 저장 - 로그 검색 및 조회 - 로그 보관 정책 관리	Observer Pattern - Door의 DoorEvent 관찰자 - update(event: DoorEvent) 구현 Repository Pattern - 로그 데이터 저장소 추상화 Information Expert (GRASP) - 로그 데이터 전문 관리	- 저장소: 대용량 로그 저장 지원 - 검색: 빠른 로그 검색 (1초 미만 권장) - 무결성: 로그 누락 없는 안정적인 수집
Service Layer	<u>NotificationService</u>	다양한 채널을 통한 알림 발송을 담당하는 서비스	- 이메일, SMS, 푸시 등 다양한 채널로 시스템 상태와 이벤트를 사용자에게 전달 - Door의 DoorEvent를 관찰하여 필터링 후 알림 발송 - DoorEvent 수신 및 필터링 - 다중 채널 알림 발송 - 알림 우선순위 관리 - 발송 실패 재시도	Observer Pattern - Door의 DoorEvent 관찰자 - update(event: DoorEvent) 구현 Strategy Pattern - 다양한 알림 채널 전략 Chain of Responsibility - EventFilter 체인을 통한 알림 필터링	- 신뢰성: 중요 알림 100% 전달 - 확장성: 새로운 알림 채널 추가 용이 - 성능: 대량 알림 처리 지원 - 필터링: 불필요한 알림 차단으로 사용자 경험 향상
Service Layer	<u>AutomationService</u>	스케줄링과 자동화 규칙을 관리하는 서비스	- 시간 기반 스케줄링과 조건 기반 자동화를 통해 스마트한 문 제어 제공 - Door의 DoorEvent를 관찰하여 자동화 규칙 실행 - 스케줄 관리 및 실행 - 자동화 규칙 처리 - 조건 기반 자동 동작 - DoorEvent 수신 및 규칙 평가	Observer Pattern - Door의 DoorEvent 관찰자 - update(event: DoorEvent) 구현 Strategy Pattern - 다양한 자동화 전략	- 정확성: 스케줄 실행 ±1초 정확도 - 확장성: 무제한 규칙 추가 가능 - 유연성: 복잡한 조건식 지원 - 실시간성: DoorEvent 수신 즉시 규칙 평가

■ 분류	■ 클래스명	■ 클래스 역할 (R&R)	■ 클래스 설명 / Functional Requirements	■ 연결된 Class에서의 역할	■ Non-Functional Requirement
				Command Pattern - 자동화 명령 실행	
Service Layer	<u>DoorOperationService</u>	문 동작 실행을 전담하는 핵심 서비스	모터, 센서, 안전 서비스를 통합하여 안전하고 정확한 문 동작을 실행 - 문 열기/닫기/정지 실행 - 실행 가능성 검증 - 안전성 검사 통합	Service Integration - 모터, 센서, 안전 서비스 통합 Facade Pattern - 복잡한 하위 시스템에 대한 단순 인터페이스 Information Expert (GRASP) - 문 동작 전문 지식	- 정밀도: 정확한 문 위치 제어 - 안전성: 모든 동작에 안전 검사 포함 - 신뢰성: 99.9% 동작 성공률 - 일관성: 모든 문 타입에 동일한 동작 품질
HW Abstraction	<u>Motor</u>	문 구동을 위한 모터 하드웨어를 추상화하는 클래스	다양한 타입의 모터(Stepper, Servo, DC, AC)를 통합하여 일관된 모터 제어 인터페이스 제공 - 모터 시작/정지 제어 - 회전 방향 제어 - 출력 조절 - 모터 상태 모니터링	Adapter Pattern - 다양한 모터 하드웨어 통합 State Pattern - 모터 상태 관리 Information Expert (GRASP) - 모터 제어 전문 지식	- 정밀도: 설정 RPM ±1% 오차 - 안전성: 모터 상태 모니터링으로 과열/과부하 보호
HW Abstraction	<u>Sensor</u>	모든 센서의 공통 기능을 제공하는 기본 센서 클래스	다양한 타입의 센서에 대한 공통 인터페이스와 기본 기능을 제공 - 센서 감지 수행 - 센서 보정 - 센서 값 읽기 - 센서 활성화/비활성화	Template Method Pattern - 센서 동작 템플릿 Information Expert (GRASP) - 센서 데이터 전문 지식	- 정확성: 99.9% 감지 정확도 - 응답성: 10ms 이내 감지 - 신뢰성: 자가 진단 기능
HW Abstraction	<u>AngleSensor</u>	문의 회전 각도를 정밀하게 측정하는 센서 클래스	문의 회전 각도를 도(degree) 단위로 정확하게 측정하여 정밀한 각도 제어 지원 - 회전 각도 측정 - 각도 유효성 검증 - 최대 각도 제한 - 정밀도 관리	Information Expert (GRASP) - 각도 정보 전문 지식	- 정밀도: 측정 정확도 - 범위: 360도 전체 범위 - 해상도: 0.1도 단위 측정
HW Abstraction	<u>PositionSensor</u>	문의 위치를 정밀하게 측정하는 전문 센서 클래스	문의 선형 위치를 mm 단위로 정확하게 측정 - 선형 위치 측정 - 위치 유효성 검증 - 정밀도 관리 - 측정 범위 제한	Information Expert (GRASP) - 위치 정보 전문 지식	범위: 최대 5m 측정 가능
HW Abstraction	<u>ProximitySensor</u>	장애물 감지를 위한 근접 센서 클래스	문 동작 경로의 장애물을 감지하여 안전한 문 동작을 보장 - 장애물 감지 - 감지 범위 설정 - 감지 임계값 조정 - 실시간 감지 상태 제공	Safety Observer - 안전성을 위한 관찰자 역할 Information Expert (GRASP) - 근접 감지 전문 지식	- 응답속도: 5ms 이내 감지 - 선택성: 크기별 객체 구분 - 환경대응: 다양한 조명 조건 대응
Controller	<u>Doorcontroller</u>	문 제어를 위한 기본 컨트롤러 인터페이스를 제공하는 클래스	다양한 입력 장치로부터의 명령을 받아 문에 전달하는 제어 인터페이스 - 대상 문 관리 - 명령 전송 - 연결 상태 관리 - 컨트롤러 식별	Command Pattern - 문 제어 명령 캡슐화 Mediator Pattern - 사용자와 문 사이의 중재자 Controller (GRASP) - 시스템 제어 역할	- 응답성: 명령 전송 지연 최소화 - 신뢰성: 명령 전송 실패 시 재시도 - 확장성: 다중 문 제어 지원
Controller	<u>PhsicalDoorController</u>	물리적 버튼을 통한 문 제어를 담당하는 컨트롤러	GPIO 핀을 통해 물리적 버튼 입력을 감지하고 문 제어 명령으로 변환 - 버튼 입력 감지 - 디바운싱 처리 - GPIO 핀 관리 - 물리적 연결 상태 확인	Adapter Pattern - 하드웨어 입력을 소프트웨어 명령으로 변환 Observer Pattern - 버튼 상태 변화 감지 Low Coupling (GRASP) - 하드웨어 의존성 최소화	- 응답성: 50ms 이내 버튼 인식 - 안정성: 전기적 노이즈 필터링
Controller	<u>RemoteDoorController</u>	무선 리모컨을 통한 원격 문 제어를 담당하는 컨트롤러	RF 신호를 통해 원격에서 문을 제어할 수 있는 편리한 인터페이스 제공 - 무선 신호 수신 - 신호 복호화 - 통신 범위 관리 - 주파수 설정	Adapter Pattern - 무선 신호를 소프트웨어 명령으로 변환 Receiver (Command Pattern) <ul style="list-style-type: none">원격 명령 수신자 Information Expert	- 보안: 암호화된 신호 전송 - 간섭: 다른 무선 기기와의 간섭 최소화

분류	클래스명	클래스 역할 (R&R)	클래스 설명 / Functional Requirements	연결된 Class에서의 역할	Non-Functional Requirement
				(GRASP) - 무선 통신 전문 지식	
Controller	IoTDoorController	인터넷을 통한 스마트 문 제어를 담당하는 IoT 컨트롤러	WiFi 네트워크를 통해 스마트폰, 웹 애플리케이션에서 문을 원격 제어 - WiFi 연결 관리 - REST API 통신 - 인증 및 보안 - 원격 상태 모니터링	Proxy Pattern - 원격 문 제어 프록시 REST API - 웹 표준 인터페이스 제공 Security Gateway - 인증 및 권한 관리	- 가용성: 네트워크 연결 유지 - 보안: OAuth 2.0 인증, HTTPS 통신 - 확장성: 동시 다중 사용자 지원 - 호환성: 다양한 IoT 플랫폼 지원
Helper	ScheduleExecutor	스케줄 실행을 담당하는 헬퍼 클래스	시간 기반 스케줄의 실행 시점을 판단하고 실행 - 스케줄 실행 - 실행 시점 판단	Command Pattern - 스케줄 명령 실행	- 정확성: 정확한 시간에 실행
Helper	RuleEngine	자동화 규칙 평가 및 실행을 담당하는 헬퍼 클래스	조건식을 평가하고 해당하는 액션을 실행 - 규칙 평가 - 액션 실행	Interpreter Pattern - 규칙 조건식 해석 Command Pattern - 액션 실행	- 유연성: 복잡한 조건식 지원 - 성능: 빠른 규칙 평가
Helper	MotorLockManager	모터 동시성 제어를 담당하는 헬퍼 클래스	여러 문의 동시 동작 시 충돌을 방지하는 잠금 관리 - 리소스 잠금/해제 - 잠금 상태 확인	Singleton Pattern - 시스템 전역 잠금 관리 Mutex Pattern - 상호 배제 구현	- 동시성: 안전한 동시 접근 제어 - 데드락 방지: 데드락 상황 예방
DataClass	Schedule	자동화 스케줄 정보를 담은 데이터 클래스	시간 기반 자동 문 제어를 위한 스케줄 정보와 반복 패턴을 관리 - 스케줄 식별 정보 - 실행 시간 설정 - 반복 패턴 정의 - 활성화/비활성 상태	Value Object - 스케줄 설정 값 Information Expert (GRASP) - 스케줄 정보 보유	- 불변성: 생성 후 변경 불가 - 직렬화: JSON 직렬화 지원 - 경량성: 최소한의 메모리 사용
DataClass	DoorEvent	문 관련 이벤트 정보를 담은 불변 데이터 클래스	문의 상태 변화, 동작 완료, 오류 발생 등의 이벤트 정보를 구조화하여 전달 - 이벤트 식별 정보 - 이벤트 타입 분류 - 발생 시간 기록 - 이벤트 심각도 표시	Value Object - 불변 값 객체 Data Transfer Object 이벤트 정보 전달 Information Expert (GRASP) - 이벤트 정보 보유	- 불변성: 생성 후 변경 불가 - 직렬화: JSON 직렬화 지원 - 경량성: 최소한의 메모리 사용
DataClass	AutomationRule	자동화 규칙 정보를 담은 데이터 클래스	조건 기반 자동화를 위한 규칙 정의와 액션 정보를 관리 - 규칙 식별 정보 - 조건 표현식 정의 - 실행 액션 설정 - 대상 문 지정	Value Object - 규칙 설정 값 Information Expert (GRASP) - 규칙 정보 보유	- 유연성: 복잡한 조건식 지원 - 성능: 빠른 규칙 평가
DataClass	OperationResult	문 동작 실행 결과 정보를 담은 데이터 클래스	문 동작의 성공/실패, 메시지, 소요 시간 등의 실행 결과를 구조화하여 제공 - 성공/실패 상태 - 결과 메시지 - 오류 코드 제공 - 소요 시간 측정	Data Transfer Object - 동작 결과 전달 Error Handling - 오류 정보 전달 Information Expert (GRASP) - 실행 결과 정보 보유	- 명확성: 명확한 성공/실패 구분 - 디버깅: 상세한 오류 정보 제공 - 성능: 빠른 결과 생성 - 표준화: 일관된 결과 형식
DataClass	SafetyRule	안전 규칙 정보를 담은 데이터 클래스	시스템 안전성을 위한 규칙 정의와 조치 사항을 관리 - 안전 조건 정의 - 조치 사항 설정 - 활성화 상태 관리	Rule Object - 안전 규칙 캡슐화 Information Expert (GRASP) - 안전 규칙 정보 보유	- 신뢰성: 안전 규칙 무결성 - 우선순위: 안전 규칙 실행 순서
DataClass	SensorReading	센서 결과를 담은 데이터 클래스	센서에서 읽은 값과 메타데이터를 구조화하여 제공 - 센서 값 저장 - 읽기 시간 기록 - 유효성 표시 - 단위 정보 포함	Value Object - 센서 데이터 값 객체 Information Expert (GRASP) - 센서 읽기 정보 보유	- 정확성: 정밀한 센서 값 표현 - 타임스탬프: 정확한 시간 기록
DataClass	LogEntry	시스템 로그 정보를 담은 구조화된 데이터 클래스	문 동작, 시스템 이벤트, 오류 정보 등을 시간순으로 기록하는 로그 엔트리 - 로그 고유 식별 - 작업 정보 기록 - 시간 정보 저장 - 상태 및 결과 기록	Value Object - 불변 로그 데이터 Audit Trail - 감사 추적 정보 Information Expert	- 무결성: 로그 데이터 변조 방지 - 압축: 효율적인 저장 공간 사용 - 검색: 빠른 로그 검색 지원 - 보관: 장기 보관 정책 지원

■ 분류	■ 클래스명	■ 클래스 역할 (R&R)	■ 클래스 설명 / Functional Requirements	■ 연결된 Class에서의 역할	■ Non-Functional Requirement
				(GRASP) - 로그 정보 보유	
Interface	<u>Observer</u>	상태 변화를 관찰하는 객체를 위한 표준 인터페이스	Observable 객체의 상태 변화를 감지하고 적절한 대응을 수행하는 인터페이스 - 상태 변화 감지 - 변화에 대한 대응 처리 - 이벤트 정보 수신	Observer Pattern - Observer 역할 Polymorphism (GRASP) - 다형적 관찰자 Information Expert (GRASP) - 특정 이벤트 처리 전문 지식	- 응답성: 즉시 이벤트 처리 - 안정성: 예외 상황 안전 처리 - 독립성: 다른 관찰자와 독립적 동작
Interface	<u>Observable</u>	관찰 가능한 객체를 위한 표준 인터페이스	Observer 패턴의 Subject 역할로 관찰자들에게 상태 변화를 알리는 인터페이스 - 관찰자 등록/해제 - 관찰자들에게 알림 - 상태 변화 이벤트 관리	Observer Pattern - Subject 역할 Polymorphism (GRASP) - 다형적 관찰 대상 Low Coupling (GRASP) - 관찰자와 느슨한 결합	- 확장성: 다수의 관찰자 지원 - 성능: 빠른 알림 전파 - 일관성: 모든 관찰자에게 동일한 정보 전달
Interface	<u>Notification Channel</u>	다양한 알림 채널을 위한 통합 인터페이스	이메일, SMS, 푸시 알림 등 다양한 채널을 통한 메시지 전송을 위한 공통 인터페이스 - 메시지 전송 - 채널 가용성 확인 - 전송 결과 반환	Strategy Pattern - 알림 채널 전략 Adapter Pattern - 다양한 통신 프로토콜 통합 Polymorphism (GRASP) - 일관된 알림 인터페이스	- 신뢰성: 중요 메시지 전달 보장 - 확장성: 새로운 채널 추가 용이
Interface	<u>EventFilter</u>	이벤트 필터링을 위한 인터페이스	Observer가 관심 있는 이벤트만 처리할 수 있도록 필터링 기능 제공 - 이벤트 처리 여부 결정 - 필터 우선순위 제공	Chain of Responsibility - 필터 체인 구성	- 성능: 빠른 필터링 처리 - 유연성: 복잡한 필터 조건 지원
Interface	<u>Repository</u>	데이터 저장소에 대한 통합된 접근 인터페이스를 제공하는 제네릭 인터페이스	파일, 데이터베이스, 클라우드 등 다양한 저장소를 추상화하여 일관된 데이터 접근 제공 - 엔티티 저장/조회/수정/삭제 - 전체 엔티티 조회 - 타입 안전성 보장	Repository Pattern - 데이터 접근 추상화 Generic Programming - 타입 안전한 데이터 접근 Information Expert (GRASP) - 데이터 저장 전문 지식	- 성능: 빠른 데이터 접근 - 확장성: 다양한 저장소 지원 - 일관성: 트랜잭션 보장 - 타입안전성: 컴파일 타임 타입 검증