**옵저버 서비스 제작 설계**

QA팀 이채호 사원

1. 프로그램 정의
   1. 서비스 2개 및 App 1개 제작
   2. 각 서비스와 App이 서로를 감시
   3. 각 서비스 및 App이 중지되는 경우 실행중인 서비스 및 앱에서 중지된 서비스 및 App을 시작 시킴
   4. 서비스 및 App이 서로를 감시 및 복구
2. 설계 원칙
   1. C# .Net Framework, C계열로 각각 작성
   2. Windows 10 x64 기반으로 작성
   3. 통신: WFC, IPC 각각 작성
   4. 추후 프로그램 변경 사항 발생 시, Client 및 공용 클래스만 변경하도록 작성
3. 구조 설명 (예상)
   1. Service 2ea
      1. 동일한 동작을 수행하는 Service 두 개
      2. Timer 이용하여 일정 주기로 Client 및 Service 감시
      3. Client 비 정상 종료 시 Service의 자료로 동작 상태 복구
      4. Client 정상 종료 시 Service 종료
   2. Client
      1. Client 실행 시 Service 실행
      2. Client 정상 종료 시 Service 종료
   3. RmtObject: 공통클래스
      1. Service와 Client간 통신에 담기는 Class
      2. Checkbox, Textbox 등의 현재 상태를 가지고 있음

**배경지식 문서**

* C# .Net
  1. ServiceController Class 정보 (MSDN: <https://goo.gl/BCDM8i>)
     1. 생성자
        1. ServiceController(“서비스이름”): 로컬 컴퓨터의 기존 서비스와 연결
     2. 속성
        1. Status: 연결된 서비스의 상태
     3. 메소드
        1. Start(): 연결된 서비스 시작
        2. Stop(): 연결된 서비스 중지
        3. Refresh(): 속성값 새로 고침
  2. Service Installer 사용법
  3. Timer Class 정보&비교
     1. System.Windows.Forms.Timer (MSDN: <https://goo.gl/3JrygL>)
        1. Form과 동일한 스레드에서 주기 별 이벤트(Tick Event)를 발생
        2. Tick Event 동작 중에는 프로그램 대기
     2. System.Timers.Timer (MSDN: <https://goo.gl/13Rbkx>)
        1. Form 과 동일한 스레드 혹은 별도 스레드 에서 주기 별 이벤트를 발생
     3. System.Threading.Timer (MSDN: <https://goo.gl/6RL5B8>)
        1. 별도 스레드를 생성하여 주기 별 이벤트를 발생
  4. System.Runtime.Remoting.Channels.Ipc (MSDN: <https://goo.gl/zXg5fy> )
     1. IpcChannel Class
     2. IpcClientChannel Class
     3. IpcServerChannel Class
  5. C# Service Program 구조
* 통신
  1. IPC(프로세스간 통신)을 이용
     1. File
     2. Signal
     3. Socket
     4. Message Queue
     5. Pipe
     6. Named Pipe
     7. Shared Memory
     8. Memory Map
  2. WCF
     1. Windows Communication Foundation
     2. XML 웹 서비스 기반의 차세대 서비스 통신 프레임 워크
     3. End Point (주소, 계약, 바인딩)