**WCF를 이용한 채팅**

**설계**



**세부 기능 설명**

1. 인터페이스 : IChat

|  |  |  |
| --- | --- | --- |
| **함수** | Join | 1. 이름이 기존 채터에 있는지 검색 2. 채터(Dictionary)에 이름과 이벤트를 추가 3. 사용자에게 보내 줄 채널 설정 4. UserEnter 이벤트를 접속 중인 사용자들에게 전달 5. 델리게이터에 접속한 사용자 이벤트 핸들러 추가 6. 접속한 사용자 목록 리턴(이미 사용중인 이름일 경우 null 리턴) |
| **인자** | string name | 접속자 이름 |
| **리턴값** | string[] | 현재 접속한 사용자 목록 |

|  |  |  |
| --- | --- | --- |
| **함수** | Say | 1. ChatEventArgs 변수 선언 2. ChatEventArgs 변수에 플래그, 현재 사용자 이름, 인자로 받아온 메시지를 저장 3. ChatEventArgs 변수 전달(브로드캐스팅) |
| **인자** | string msg | 전달할 메시지 |
| **리턴값** | void | … |

|  |  |  |
| --- | --- | --- |
| **함수** | Whisper | 1. ChatEventArgs 변수 선언 2. ChatEventArgs 변수에 플래그(ReceiveWhisper), 현재 사용자 이름, 인자로 받아온 메시지를 저장 3. ChatEventHandler 변수 선언 4. 선언한 변수에 채터를 통해 인자로 받은 이름을 키값으로 해당 사용자 이벤트 리턴 받음 5. 데이터 전달 |
| **인자** | string to, string msg | To : 귓속말을 전달할 사용자 이름  Msg : 전달할 메시지 |
| **리턴값** | void | … |

|  |  |  |
| --- | --- | --- |
| **함수** | Leave | 1. 이름이 입력(접속) 되지 않았을 시 리턴 2. 채터에서 사용자 정보 제거 3. 델리게이트에서 사용자 이벤트 제거 4. ChatEventArgs 변수 선언 후 플래그(UserLeave), 사용자 이름 저장 후 전달 |
| **인자** | … | … |
| **리턴값** | void | … |

1. 함수

|  |  |  |
| --- | --- | --- |
| **함수** | BroadcastMessage | 1. 임시 변수에 저장해둔 접속자 이벤트(ChatEvent)를 저장 2. 가져온 값이 null이 아니라면 모든 접속자에게 인자로 받은 이벤트를 전송 |
| **인자** | ChatEventArgs e | 입력 받은 이벤트 |
| **리턴값** | void | … |