주제: 채팅 프로그램 서비스 구현

1. 인터페이스 설계



2. 세부 기능 설명

Private 함수

|  |  |  |
| --- | --- | --- |
|  | 이름 | 기능 |
| 함수 | BroadcastMessage | 1.임시 이벤트 변수 temp에  ChatEvent 대입  2. temp가 null이 아닐경우  Temp의 모든 클라이언트에게  이벤트 전달(EndAsync 호출) |
| 리턴값 | void |  |
| 인자 | ChatEventArgs | 현재 들어온 이벤트 |

|  |  |  |
| --- | --- | --- |
|  | 이름 | 기능 |
| 함수 | EndAsync | 1. ChatEventHandler 변수 생성후  인자의 비동기 호출이 호출된 델리게이트 개체 대입 |
| 리턴값 | void |  |
| 인자 | IAsyncResult | 비동기 작업의 상태를 받는 인자 |

|  |  |  |
| --- | --- | --- |
|  | 이름 | 기능 |
| 함수 | MyEventHandler | 1. 인자로 들어온 이벤트 스위칭  2. 상황에 맞게 클라이언트에게  전달할 IChatCallback  메서드 호출 |
| 리턴값 | void |  |
| 인자 | object, ChatEventArgs | 샌더와 이벤트 인자 |

인터페이스 : IChat

|  |  |  |
| --- | --- | --- |
|  | 이름 | 기능 |
| 함수 | Join | 1.이름이 채터에 있는지 체크  2.이름과  이벤트(MyEventHandler)를 추가  3.사용자에게 보내줄 채널 설정  4.ChatEventArgs에 UserEnter 이벤트를 전달  5.델리게이터 추가후 사용자 리스트 리턴  6.중복된 이름일 경우 null 리턴 |
| 리턴값 | String[] | 같은 이름이 없을 경우에는 사용자 리스트를 반환한다 |
| 인자 | string | 사용자의 이름 |

|  |  |  |
| --- | --- | --- |
|  | 이름 | 기능 |
| 함수 | Say | 1. ChatEventArgs 객체 생성  2.인자인 메시지와 사용자의 이름, 메시지 타입(enum) 대입  3.BroadcastMessage 호출 |
| 리턴값 | void |  |
| 인자 | string | 메시지 |

|  |  |  |
| --- | --- | --- |
|  | 이름 | 기능 |
| 함수 | Whisper | 1. ChatEventArgs 객체 생성  2.인자인 메시지와 사용자의 이름, 메시지 타입(enum) 대입  3.채터 목록에서 검색후  EndAsync 호출 |
| 리턴값 | void |  |
| 인자 | string,string | 귓속말 대상,메시지 |

|  |  |  |
| --- | --- | --- |
|  | 이름 | 기능 |
| 함수 | Leave | 1.채터 목록에서 대상 삭제  2.델리게이터 감소  3. ChatEventArgs 객체 생성  4.사용자의 이름, 메시지 타입(enum) 대입  5. BroadcastMessage 호출 |
| 리턴값 | void |  |
| 인자 | void |  |