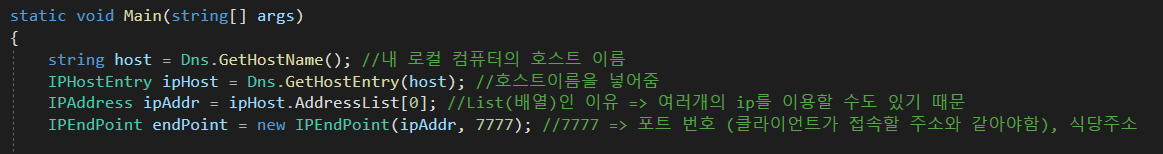
* 소켓 프로그래밍
  + 서버(예제로 try Catch를 사용해 오류 방지)

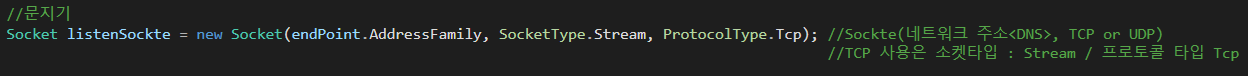


Dns.GetHostName() : 내 로컬 PC 호스트 이름

(IPHostEntry) Dns.GetHostEntry(HostName) : 호스트의 이름을 넣음

(IPAddress) IPHostEntry.AddressList : 주소를 리스트로 저장 (여려 ip주소를 사용할 수도 있기 때문)

(IPEndPoint) new IPEndPoint(리스트 주소, 포트번호(내가 정한)) : 포트번호는 클라이언트 쪽 포트번호와 같아야함



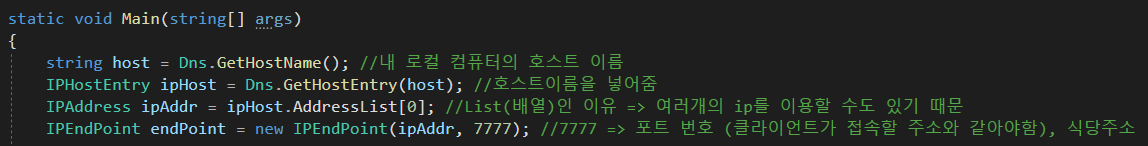
문지기 고용 Socket(네트워크 주소, Tcp or Udp) => Tcp일 경우 SocketType는 Stream

텍스트이(가) 표시된 사진

자동 생성된 설명

1. 문지기 교육(식당 주소와 포트 번호 알려줌)
2. 영업시작 (Listen(최대 대기수)를 사용해 대기 줄로 묘사 (라이브 시에 규모에 따른 수치 조정)
3. while문으로 서버는 항상 client를 받을 준비
4. 손님입장 (Accpet로 들어오는 손님을 찾음, 블로킹 함수로써 없다면 아래 코드를 실행하지 않음)
5. 손님받기 (손님의 요청 예시에서는 Hello World를 보냄)
6. 손님보내기 (손님에게 서버단에서 보냄, 예시에서는 Welcome to MMORPG Sever를 보냄)
7. 쫒아내기 (손님을 식당(서버)에서 내보냄, ShutDown으로 고지 가능, 없어도 실행에는 문제가 없다)
   * 클라이언트

초기값 동일



텍스트이(가) 표시된 사진

자동 생성된 설명

1. 휴대폰 설정 (문지기와 연락할, 같은 맥락)
2. 문지기에게 입장 문의 ((휴대폰)socket.Connet(식당주소, 포트번호)를 이용해 식당에 접근)
3. 보내기 (서버에 Hello World를 보냄)
4. 받기 (서버에서 Welcome To MMORPG Server를 받음)
5. 나가기
   * 결과

텍스트이(가) 표시된 사진

자동 생성된 설명

출력 정상, 서버는 계속해서 while문을 돌기 떄문에 클라이언트에서 계속 접근이 가능함