2008년 정보보호학과 졸업작품

JAVA를 이용한 보안채팅프로그램

팀명 : SUN

팀 원

김광현 서지원
안병모 이동흔
조수형 한세운
허성필

2008. 10
중부대학교 정보보호학과

<목 차>

<표 차례>

<그림 차례>

요약

인터넷의 발달로 우리나라 네티즌들은 여기저기 많은 활동을 하고 있는 건 사실이다 그러나 발달됨과 함께 범죄의 현상도 기하급수적으로 늘어가고 있는 것이 현실인데 거의 메신저의 채팅창에서 이루어지는 것을 알 수 있다.
상대방과 이야기하는 사이에 무심결에 이야기한 중요문서나 계좌번호 같은 개인 신용정보를 해킹하여 악 이용하는 사람이 점차 증가하고 있는데 이를 막기 위해서는 보안에 더 신경을 써야한다는 것을 느끼게 해준다.
하지만 보안에 신경을 써도 해킹은 일어날 것인데 조금이나마 해커의 수를 줄이자는 뜻에서 프로그램을 만든 것이 이번 졸업 작품입니다.

저희 졸업 작품은 기존에 많이 사용하고 활설화된 C언어가 아니라 Java라는 언어를 써 평문과 암호문을 구분하여 채팅하는 프로그램을 만들어 보았습니다.

1. 서론

1.1 Java란 무엇인가

Java는 미국의 Sun Microsystems(www.sun.com) 이라는 회사에서 만든 객체지향언어(Object Oriented Language)이다. Java의 구문은 기존의 대표적인 프로그램언어인 C, C++와 매우 유사하다. 다시 말해서 Java는 전혀 새로운 프로그램언어는 아니다. 그러므로 기존의 프로그램언어에 어느 정도 익숙한 사람은 좀 더 쉽게 Java를 이해할 수 있다.
그런데 일반적으로 Java하면 web browser상에서 움직이는 applet을 생각하는 경우가 많다. 맞는 말이다. applet도 Java로 개발된 것이다. 그러나 applet은 Java로 개발할 수 있는 것들 중 일부에 지나지 않는다.(참고로 applet은 web server에 존재하고 있다가 사용자 pc의 web browser로 download되어 그 안에서 수행되는 Java 프로그램을 말한다).
자바는 그 기초적인 문법을 가지고 다양한 범위로 쓰일 수 있다. 크게는 J2SE, J2EE, J2ME 로 나누어지며 SE는 스텐다드 에디션이라는 뜻으로 일반적인 PC상에서 구동되는 전반적인 프로그램을 작성할 수 있는 플랫폼을 말하며, J2EE는 엔터프라이즈 에디션으로 기업환경, 즉, 웹이나 대단위 작업을 필요로 하는 플랫폼을 말한다. 마지막으로 ME는 마이크로 에디션으로 핸드폰이나 TV에서 돌아가는 플랫폼을 말한다.
다양한 플랫폼에 프로그램을 작성할 때는 전혀 다른 방식으로 프로그래밍이 되는 것이 관례 이었지만 자바의 경우에는 한 번의 문법 습득으로 다양한 플랫폼에서 프로그래밍이 가능해 서 생산성이 극대화 되었다.
초보자들도 쉽게 접근이 가능하고, 대부분의 기능이 플랫폼 상에 개발되어 있어 개발 속도도 빠르고 성능 또한 우수한 편이다

Java는 이전의 다른 언어들에 비해 많은 부분에서 발전된 언어이다. 많은 사람들이 자바를 순수하게 Web(웹)을 위한 언어라고 말하지만, Java는 그 이상의 것이 포함되어 있다. 다음에서 설명하는 특성은 다른 언어에도 있지만 Java에서는 기능면에서 더 뛰어난 것들이다.

1.2 Java의 특징

① Java는 이식성이 높은 언어이다.

이것은 너무나 잘 알려진 사실 중 하나. 정확히 말하면, 한 번 코딩되어 컴파일된 상태의 클래스 파일은 다시 수정하지 않고도 Java Virtual Machine(JVM)이 설치되어 있는 System에서는 실행이 가능하다는 뜻. JVM(Java Virtual Machine)은 인터프리터 (Interpreter)의 기능을 수행하는 프로그램이고 JDK5.0 등을 설치한 상태라면 자동적으로 포함 되어 있음.

② Java는 외부 포인터를 제고하고 내부적인 포인터를 사용한다.

이것은 메모리적인 설명이 첨가 되어야 하지만, 우선 C/C++ 에서 겉으로 드러나는 포인터 변수(int*, char* 등)를 완벽히 제거하고 내부적으로 객체의 메모리 할당 시 무조건 동적으로 메모리를 할당시키는 방식을 취했다는 뜻. 따라서 직접적으로 포인터를 지정할 수 없으므로 보안상 안전성도 고려했다고 봐야 한다.

③ Java는 완벽한 객체 지향적 언어이다.

Java의 기초 프로그램에서 고급 프로그램까지 어느 것 하나도 클래스의 범위를 벗어나는 것은 없다. 다시 말해 프로그램에서의 코딩이 전부 클래스 내부에 기재. 클래스를 벗어나 코딩이 되는 모든 경우는 컴파일시 에러를 발생(Package와 impor)그리고 또 다른 클래스는 제외) 따라서, 모든 멤버의 규정은 클래스의 특성을 본받게 된다.

1.3 Java의 종류

① 자바 애플리케이션(Application)

JDK와 함께 제공되는 자바 가상머신에 의해 독립적으로 실행될 수 있도록 작성된 자바 프로그램입니다. 다시 말해서, 여러분의 컴퓨터에서 윈도우의 도스창 또는 유닉스 쉘 등 과 같은 쉘에서 자바 가상머신을 이용하여 실행시키는 자바 프로그램입니다.

② 자바 애플릿(Applet)

태그를 이용하여 HTML 페이지 내에 포함되어, 자바 호환 웹 브라우저에 의해서 실행되도록 작성된 자바 프로그램입니다. 다시 말해서, 여러분의 홈 페이지 내에 삽입되어 자바 호환 웹 브라우저에 의해 실행되도록 규약에 맞추어 작성된 자

바 프로그램을 말하는 것 입니다.

③ 자바 서블릿(Servlet)

기존의 CGI 프로그램과 같이 웹 서버 프로그램의 기능을 확장하기 위한 자바 프로그램으로서, 웹 서버 내에 있는 자바 런타임 환경과 함께 제공되는 자바 가상머신에 의해 실행 되도록 작성된 자바 프로그램입니다. 자바 서블릿은 웹 서버 내에서 자바 런타임 환경과 함께 제공되는 자바 가상머신에 의해 실행되고, 자바 애플릿은 웹 서버에서 웹 클라이언트로 다운로드 되어 웹 클라이언트에서 자바 호환 웹 브라우저에 내장된 자바 가상머신에 의해 실행된다는 차이점이 있습니다. 이렇게 웹 서버 내에서 실행될 수 있도록 작성된 자바 서블릿은 기존의 웹 서버 내에서 실행되는 프로그램인 CGI 프로그램을 대체할 수 있도록 고안되었습니다.

④ 자바 빈(Bean)

델파이 또는 비주얼 베이직을 이용하여 프로그램을 작성할 때, 버튼이나 창과 같은 컨트롤들을 마우스로 끌어다 프로그램 내에 삽입할 수 있도록 되어 있는데, 이와 마찬가지로 자바 빈은 하나의 완벽한 기능을 갖고 재사용될 수 있도록 만들어진 소프트웨어 컴포넌트입니다. 마이크로소프트에서 제공되는 ActiveX 컴포넌트와 같이 자바에서 컴포넌트 프로그램을 가능하도록 해줍니다.

⑤ 자바 패키지(package)

다른 자바 프로그램에 의해 삽입(import)되어 사용될 수 있도록 작성된 자바 프로그램입니다. 이러한 자바 패키지는 기존의 프로그래밍 언어에서 사용하던 라이브러리 또는 운영체제에서 제공해 주는 API 등과 같다고 볼 수 있습니다. 자바 패키지 역시 해당 규약을 갖겠지요. 자바에서는 기본적으로 압축 파일의 형태로 'casses.zip"이라는 자바 패키지가 제공되고 있고, 압축 파일 내에는 디렉터리 단위로 패키지가 포함되어 있습니다.

1.4 연구의 필요성

최근 사회가 발전함에 따라 채팅 또한 발전하고 있다 처음에는 하늘사랑 버디버디 같은 메신저 겸 채팅을 할 수 있는 프로그램 등을 주로 썼지만 보안부분에는 신경을 잘 쓰지 않다보니 해킹 또한 많이 발생 되었다 해킹이 점차 많아져서 네티즌들의 말들이 많이 있어 보안에 신경을 조금씩 쓰다 보니 msn이란 메신저가 나오게 되었다 하지만 이것도 해킹을 당하긴 마찬가지였다 그러다 보안에 신경을 많이 써서 만든 프로그램이 네이트온 이라는 메신저 프로그램이다 네이트온은 메신저 기능뿐만 아니라 문자를 보낼 수도 있고 다른 부가 기능들이 많이 첨가 되어 있다 하지만 채팅창에서의 대화할 때에는 중요한 내용이나 계좌번호 같은 개인정보가 나도 모르는 사이에 해킹 당할 때가 있어 사람들이 낭패를 보는

경우가 있다 이를 방지 하기위해서 Java를 이용한 채팅창을 개발하였고 암호문과 평문을 따로 두어 평소 일반적인 대화일대에는 평문에 체크를 하고 대화하는 방식이고 중요한 내용이 있으면 암호문에 체크를 하고 비밀번호를 걸어 상대방만 볼 수 있게 하는 채팅 프로그램이다 다른 사람이 보면 안 되는 중요한 이야기나 개인의 계좌번호 같은 개인정보를 조금이나마 막을 수 있는 것에 의의를 두었다

1.5 과제의 목표 및 내용

1. 목표

자바에 대해서 알고 자바의 종류에는 어떤 것 들이 있는지 파악 그리고 그 종류 안에서 어떤 것을 쓸지 생각하고 이를 이용하여 프로그래밍 한다. 소스에 DES알고리즘을 사용해 평문과 암호문을 두어 상호간에 안전한 채팅을 하는 것이다

2. 추진방법

DES를 사용하여 암호화 복호화 알고리즘을 만들어 평문과 암호문을 구별하여 사용하고 통신방식을 일대일 방식인 소켓 보다는 같은 대역폭 안에 있으면 누구나 쉽게 입장 할 수 있게 하는 브로드캐스팅 방식을 쓴다.
암호 키를 만들어 평문을 암호화되게 보여줌으로써 상호간의 안전한 채팅을 할 수 있게 한다.

2 기반기술

2.1 통신방식

일반적으로 채팅프로그램에서 채팅 서버로 클라이언트의 접속을 관리하는 방식을 사용합니다만, 본 프로그램은 해당 IP대역의 BroadCast 주소로 메시지 전송을 하도록 구성하여 임의로 패킷 sniffing 하기 쉽게 구성하였습니다.

1. BroadCast 란?

네트워크 이론에서 볼 때 하나의 노드(node)로부터 해당 네트워크의 모든 노드에게 어떤 소식을 널리 알리는 것이며 전파 또는 전송매체를 통해 일반대중에게 정보를 전달하는 행위이다. 또 불특정 다수에게 동시에 데이터를 보내는 형태 등이다.

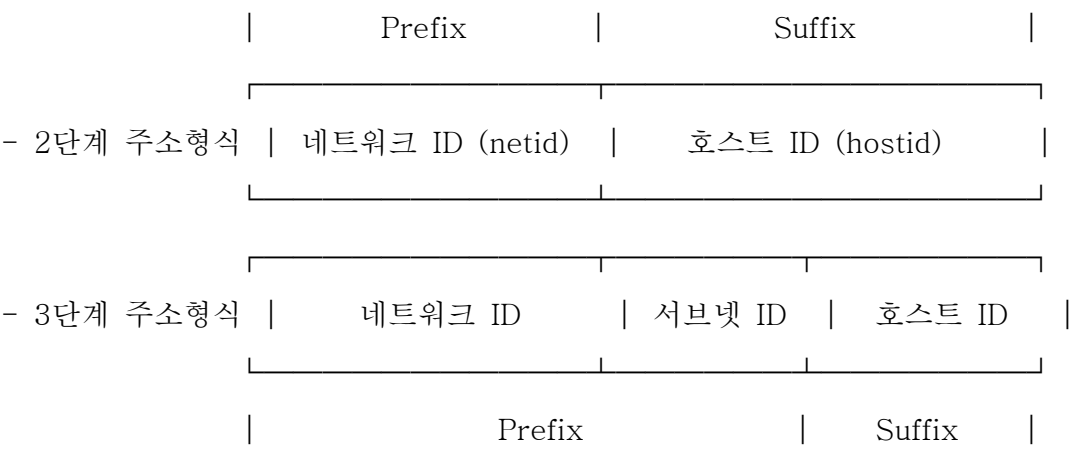
브로드캐스트 패킷의 예로는 ARP의 Request 메시지, NetBIOS의 Name Request 등 또는 각종 라우팅용 광고성(Advertisement) 패킷 들 등이 있다

예를 들어 Ethernet 네트워크에서 48비트짜리 목적지 주소의 비트가 모두 1인 경우 에는 이것이 브로드캐스트 임을 알리는 것이다.

2. 서브넷 마스크

Subnet는 통상적으로 하나의 부분을 이루는 네트워크 단위(단일 물리적 네트워크)를 말한다. Subnetting은 IP 주소 체계가 2단계 (네트워크 ID - 호스트 ID) 구분방식인 것을, 다시 3단계 (네트워크 - 서브네트 - 호스트)로 네트워크를 세분화하는 것이다.

- 2 단계 및 3 단계 주소 형식



Subnet Mask는 TCP/IP 프로토콜 집합에서 IP 주소체계에 의하여 네트워크를 나누는 (분할하는) 논리적인 수단을 말한다. 여기서, Subnet란 Sub-Network의 약자로써 네트워크에 대한 논리적인 분할을 의미하며, Mask는 차폐의 의미를 갖는다. 라우터는 이 서브넷 마스크를 이용하여 각 패킷이 전달될 실제 세분화된 물리적인 네트워크를 찾게 된다. 이 방식의 장점 중의 하나로써, 만일 하나의 라우터에서 여러 개의 인터페이스에 다가 각각의 서브네트로 나눠 구분하면 전달되는 라우팅 정보의 크기를 감소시킬 수 있다.

네트워크를 나누는 방법 (Subnet Mask 사용방법)

255.255.255.0 라는 마스크는 255.255.255 까지는 네트워크를 의미하고 0 은 노드를 가리킨다. 만일, 147.6.8.169에 255.255.255.0을 마스크하면 147.6.8까지는 어떤 네트워크를 가리키게 되고 169는 하나의 노드를 의미한다. 즉, 어떤 네트워크내의 어떤 노드가 표현된다. 이때 이 노드는 전 세계적으로 유일무이하게 된다. 147.6.8.xxx에 속하는 네트워크의 노드의 개수는 0 부터 255까지 총 256개이다.

만일, 147.6.8.xxx 라는 네트워크를 분할코자 한다면,

서브넷 마스크	네트워크 분할 가능수	가능한 노드 수
255.255.255.254	128	2
255.255.255.252	64	4
255.255.255.240	16	16
255.255.255.192	4	64
255.255.255.128	2	128

...

위의 수치는 2진법 계산을 해보면 알 수 있다.

사용 예로는 이렇다

○ 192.168.63.0 / 24

- 서브네트 마스크 길이 : 24 비트
- 가능한 IP 주소 범위 : 192.168.63.1 ~ 192.168.63.254
- 네트워크 주소 : 192.168.63.0
- 브로드캐스트 주소 : 192.168.63.255
- Prefix : 10101100.10101000.00111111.00000000

----------24----------

○ 192.168.63.0 / 25

- 서브네트 마스크 길이 : 25 비트 (2개의 세브네트로 나눌 수 있음)

- 서브네트 마스크 : 255.255.255.192

- 가능한 IP 주소 범위

192.168.63.1 ~ 192.168.63.126 , 192.168.63.129 ~ 192.168.63.254

- 네트워크 주소 : 192.168.63.0 및 192.168.63.128

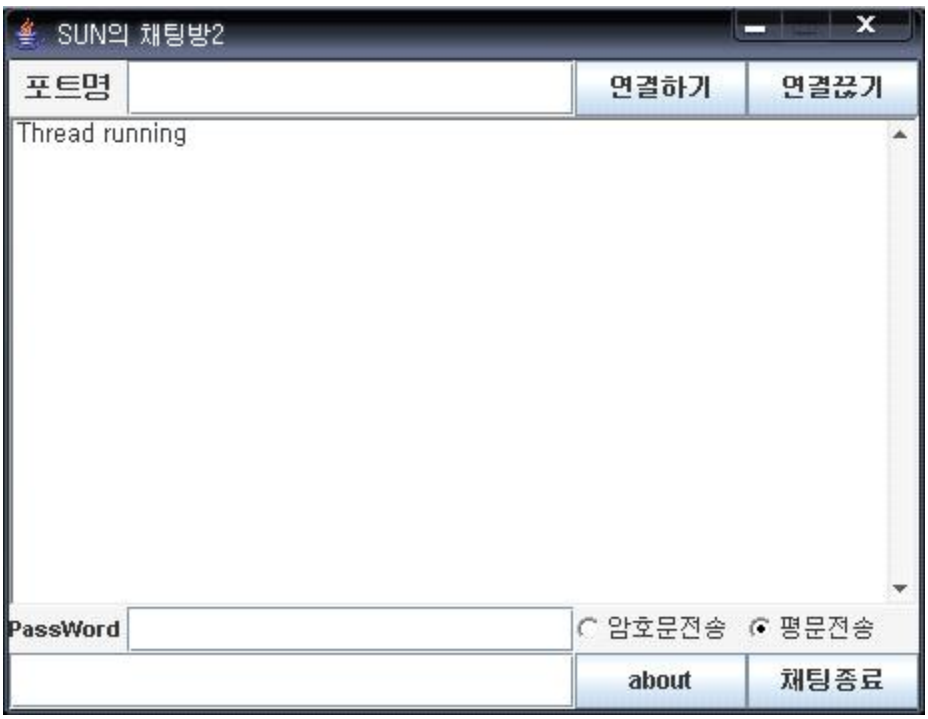
- 브로드캐스트 주소 : 192.168.63.127 및 192.168.63.255

- Prefix : 10101100.10101000.00111111.00000000

----------25----------

2.2 Window 프로그래밍

윈도우의 구성은 기본 AWT의 library의 요소들로 구성하였고, 화면 구성은 다음과 같습니다.



[sun채팅방 화면 구성]

1. AWT

Java.awt

사용자 인터페이스의 작성 및 그래픽스와 이미지의 페인트용의 모든 클래스를 포함합니다.

-인터페이스의 개요-

Active Event: 자기 자신을 발송 할 수 있는 이벤트를 위한 인터페이스입니다.

Adjustable: 어느 제한 범위 내에 포함되는 조정 가능한 수치를 가지는 객체용의 인터페이스입니다.

Composite Context: 인터페이스는,Composite Context 와 함께, 기가 되는 그래픽스 영역에 draw 프리미티브(primitive)를 구성하는 메소드를 정의합니다.

Composite Context: Composite Context 인터페이스는, 합성 조작용으로 캡슐화 되어 최적화된 환경을 정의합니다.

ItemSelectable: 항목의 모임을 가지는 객체에 대한 인터페이스입니다.

Key Event Dispatcher: Key Event Dispatcher 은, 모든 Key Events 의 타깃 지정과 발송에 관해서 현재의 Keyboard Focus Manager 와 협력합니다.

Key Event Post Processor: Key Event Post Processor 는, 미소비의 모든 Key Events 의 최종 변환에 관해서 현재의 Keyboard Focus Manager 와 협력합니다.

Layout Manager: Container의 레이아웃 방법을 인식하고 있는 클래스를 위한 인터페이스를 정의합니다.

Layout Manager2: 레이아웃 제약 객체에 근거해, 컨테이너를 어떻게 배치하는지를 인식하고 있는 클래스를 위한 인터페이스를 정의합니다.

Menu Container: 메뉴 관련의 모든 컨테이너의 슈퍼 클래스입니다.

Paint: 이 Paint 인터페이스는,Graphics2D 의 조작을 위해서(때문에) 칼라 패턴을 생성하는 방법을 정의합니다.

Paint Context: Paint Context 인터페이스는,Graphics2D 에서의 전부 칠해 조작용 또는 stroke 조작용으로서 디바이스 공간에서 칼라 패턴을 생성하기 위한 , 캡슐화 및 최적화한 환경을 정의합니다.

Print Graphics: 페이지 인쇄용의 그래픽스 문맥을 제공하는 abstract 클래스입니다.

Shape: Shape 인터페이스는, 하등의 기하학적인 형태를 나타내는 객체의 정의를 제공합니다.

Stroke: Stroke 인터페이스에 의해,Graphics2D 객체는, 지정된 Shape 의 장식된 윤곽 (양식화된 윤곽 표현)을 나타내는 Shape 를 취득할 수 있습니다.

Transparency: Transparency 인터페이스는, 클래스를 구현하기 위한 공통의 투명도 모드를 정의합니다.

-클래스의 개요-

Alpha Composite: 이 Alpha Composite 클래스는, 그래픽스와 이미지의 혼합의 효과 및 투명화의 효과를 실현하기 위해서(때문에), 전송원의 색과 전송처의 색을 결합하기 위한 기본적인 알파 합성 규칙을 구현합니다.

AWTEvent: 모든 AWT 이벤트의 루트 이벤트 클래스입니다.

AWT Event Multicaster: AWT Event Multicaster 는,Java.awt.event 패키지로 정의되는 AWT 이벤트에, 효율적으로, thread 세이프인, 마르치캐스트 이벤트발송를 실시하는 구조를 구현합니다.

AWT KeyStroke: AWT KeyStroke 는, 키보드, 또는 동등의 입력 디바이스의 키 액션을 나타냅니다.

AWT Permission: 이 클래스는 AWT 의 액세스권용의 클래스입니다.

Basic Stroke: Basic Stroke 클래스는, 단순한 도형의 윤곽선을 draw 하는 속성의 기본 세트를 정의합니다.

Border Layout: 경계 레이아웃은, north (상단), south (하단), east (우단), west (좌단), 및 center (중앙)라고 하는 5 개의 영역에 들어가도록, 컴퍼넌트를 정렬 및 사이즈 변경해, 컨테이너에 배치합니다.

Buffer Capabilities: 버퍼의 기능과 프로퍼티

Buffer Capabilities. FlipContents: 페이지 반전 후에, 백 버퍼의 내용을 형태 보증해 열거합니다.

Button: 이 클래스는 라벨 첨부 버튼을 생성합니다.

Canvas: Canvas 컴퍼넌트는, 어플리케이션이 draw 하거나 사용자로부터의 입력 이벤트를 트랩 하거나 할 수가 있는 공백의 구형의 화면 영역을 나타냅니다.

CardLayout: CardLayout 객체는 컨테이너의 레이아웃 매니저입니다.

Checkbox: 체크 박스는, 「온」 (true) 또는 「오프」 (false)의 어느 쪽인가의 상태를 취할 수가 있는 그래피컬 컴퍼넌트입니다.

Checkbox Group: Checkbox Group 클래스는 Checkbox 버튼의 그룹화에 사용합니다.

Checkbox MenuItem: 이 클래스는, 메뉴에 추가할 수 있는 체크 박스를 나타냅니다.

Choice: Choice 클래스는 선택 범위의 pop-up menu를 나타냅니다.

Color: Color 클래스는, 디폴트의 sRGB 칼라 영역에 있는 색, 또는 ColorSpace 로 식별되는 임의의 칼라 영역에 있는 색을 캡슐화하기 위해서 사용됩니다.

Component: 컴퍼넌트는, 화면에 표시할 수 있어 사용자라고 대화할 수 있는 그래피컬한 표현을 가지는 객체입니다.

Component Orientation: Component Orientation 클래스는, 컴퍼넌트 또는 텍스트의 각 요소를 언어에 따라 배치하기 위한 방향을 캡슐화합니다.

Container: 총칭 Abstract Window Toolkit (AWT) 컨테이너 객체는, 다른 AWT 컴퍼넌트를 포함할 수가 있는 컴퍼넌트입니다.

Container Order Focus Traversal Policy: 컨테이너의 아이 컴퍼넌트의 순서를 기준에, traversal 순서를 결정하는 Focus Traversal Policy 입니다.

Cursor: 마우스 커서의 비트 맵 표현을 캡슐화 하는 클래스입니다.

Default Focus Traversal Policy: 컨테이너의 아이 컴퍼넌트의 순서를 기준에, traversal 순서를 결정하는 Focus Traversal Policy 입니다.

Default Keyboard FocusManager: AWT 어플리케이션의 디폴트 Keyboard Focus Manager 입니다.

Desktop: Java 어플리케이션으로 Desktop 클래스를 사용하면(자), 네이티브 데스크톱으로 등록을 마친 관련지을 수 있었던 어플리케이션을 기동해, URI 나 파일을 처리할 수 있습니다.

Dialog: Dialog 는, 타이틀 및 경계를 가지는 톱 레벨의 윈도우이며, 일반적으로는 사용자로부터의 입력을 받아들이기 위해서(때문에) 사용됩니다.

Dimension: Dimension 클래스는, 단일의 객체내의 컴퍼넌트의 폭과 높이를 정수 정밀도로를 캡슐화합니다.

DisplayMode: DisplayMode 클래스는,GraphicsDevice 의 비트의 깊이, 높이, 폭, 및 refresh rate를 캡슐화합니다.

Event: 주: Event 클래스는 현재 무효이며, 하위 호환 (을) 위해서만 존재하고 있습니다.

EventQueue: EventQueue 는 어느 플랫폼으로도 공통의 클래스이며, 기본이 되는 피어 클래스와 신뢰할 수 있는 어플리케이션 클래스로부터의 이벤트를 큐에 넣습니다.

FileDialog: FileDialog 클래스는, 사용자가 파일을 선택할 수 있는 다이얼로그 윈도우를 표시합니다.

FlowLayout: 플로우 레이아웃은, 단락내의 텍스트행과 같이, 한 방향에 컴퍼넌트를 배치합니다.

Focus Traversal Policy: Focus Traversal Policy 는, 어느 포커스 사이클 루트를 가지는 컴퍼넌트의 횡단(traverse) 순서를 정의합니다.

Font: Font 클래스는, 텍스트를 눈에 보이는 형태에 draw 하기 위해서 사용되는 폰트를 나타냅니다.

FontMetrics: FontMetrics 클래스는, 특정의 화면에서의 특정의 폰트에 관한

draw 정보를 캡슐화 하는 폰트 시학 객체를 정의합니다.

Frame: Frame 는, 타이틀과 경계를 가지는 톱 레벨 윈도우입니다.

GradientPaint: GradientPaint 클래스는, 칼라의 선형 그라이데이션 패턴으로 Shape 를 전부 칠하는 수단을 제공합니다.

Graphics: Graphics 클래스는, 모든 그래픽 문맥의 추상 base class입니다.

Graphics2D: Graphics2D 클래스는, Graphics 클래스를 확장해, 기하학적 도형, 좌표변화, 칼라 관리, 및 텍스트 배치에 대해 고도의 제어를 실시합니다.

Graphics Config Template: Graphics Config Template 클래스를 사용하는 것으로, 유효한 Graphics Configuration 를 취득할 수 있습니다.

Graphics Configuration: Graphics Configuration 클래스는 프린터 또는 모니터 등의 그래픽스 목적지의 특성을 기술합니다.

Graphics Device: Graphics Device 클래스는, 특정의 그래픽스 환경에서 이용 가능한 그래픽스 디바이스를 기술합니다.

Graphics Environment: Graphics Environment 클래스는, 특정의 플랫폼의 JavaTM 어플리케이션으로 사용할 수 있는 Graphics Device 객체 및 Font 객체의 컬렉션을 기술합니다.

GridBagConstraints: GridBagConstraints 클래스는,GridBagLayout 클래스를 사용해 배치되는 컴퍼넌트의 제약을 지정합니다.

GridBag Layout: GridBag Layout 클래스는, 다른 크기의 컴퍼넌트에서도 종횡에, 또는 baseline에 따라 배치할 수 있는 유연한 레이아웃 매니저입니다.

GridBag Layout Info: GridBag Layout Info 는,GridBag Layout 레이아웃 매니저의 유틸리티 클래스입니다.

GridLayout: GridLayout 클래스는, 컨테이너의 컴퍼넌트를 구형 구라 두로 배치하는 레이아웃 매니저입니다.

Image: abstract 클래스 Image 는, 그래피컬 이미지를 표현하는 모든 클래스의 슈퍼 클래스입니다.

Image Capabilities: 이미지의 기능과 프로퍼티

Insets: Insets 객체는 컨테이너의 경계를 표현한 것입니다.

Job Attributes: 인쇄 작업을 제어하는 속성 세트입니다.

Job Attributes.DefaultSelectionType: 사용 가능한 디폴트 선택 상태의 형태 보증된 열거입니다.

JobAttributes.DestinationType: 사용 가능한 작업 출력처의 형태 보증된 열거입니다.

JobAttributes.DialogType: 사용자에게 표시하는 사용 가능한 다이얼로그의 형태 보증된 열거입니다.

JobAttributes. Multiple Document Handling Type: 사용 가능한 복수의 카피 처리 상태의 형태 보증된 열거 입니다.

JobAttributes.SidesType: 사용 가능한 복수 페이지의 조립의 형태 보증된 열거 입니다.

Keyboard Focus Manager: Keyboard Focus Manager 는, 액티브가 되어 포커스 된 Window 및 현재의 포커스의 소유자의 관리를 제어합니다.

Label: Label 객체는, 컨테이너내에 텍스트를 배치하기 위한 컴퍼넌트입니다.

Linear Gradient Paint: Linear Gradient Paint 클래스는, 색의 선형 그라이데이션 패턴으로 Shape 를 전부 칠하는 수단을 제공합니다.

List: List 컴퍼넌트는, 텍스트 항목의 스크롤 리스트를 사용자에 대해서 표시합니다.

MediaTracker: MediaTracker 클래스는, 몇개의 미디어 객체 상태를 감시하는 유틸리티 클래스입니다.

Menu: Menu 객체는, 도구모음으로부터 전개되는 풀다운 메뉴 컴퍼넌트입니다.

MenuBar: MenuBar 클래스는, 프레임에 결합되는 도구모음의 개념을 캡슐화 하는 클래스입니다.

MenuComponent: abstract 클래스 MenuComponent 는, 메뉴에 관련하는 모든 컴퍼넌트의 슈퍼 클래스입니다.

MenuItem: 메뉴 내의 모든 항목은,MenuItem 클래스인가 그 서브 클래스의 1 개에 속하고 있을 필요가 있습니다.

MenuShortcut: MenuShortcut 클래스는 MenuItem 의 키보드 가속기를 나타내는 클래스입니다.

MouseInfo: MouseInfo 는 마우스 포인터의 위치나 mouse button 수등의 마우스에 관한 정보를 취득하기 위한 메소드를 제공합니다.

Multiple Gradient Paint: 라스터로 전부 칠하기 위해서(때문에) 복수색의 그라데이션을 사용하는 Paints 의 슈퍼 클래스입니다.

PageAttributes: 인쇄한 페이지의 출력을 제어하는 속성 세트입니다.

PageAttributes.ColorType: 사용 가능한 칼라 상태의 형태 보증된 열거

PageAttributes.MediaType: 사용 가능한 용지 사이즈의 형태 보증된 열거입니다.

PageAttributes.OrientationRequestedType: 사용 가능한 용지 방향의 형태 보증된 열거입니다.

PageAttributes.OriginType: 사용 가능한 원점의 형태 보증된 열거입니다.

PageAttributes.PrintQualityType: 사용 가능한 인쇄 품질의 형태 보증된 열거입니다.

Panel: Panel 는 가장 심플한 컨테이너 클래스입니다.

Point: 정수 정밀도로 지정되는,(x, y) 좌표 공간에서의 위치를 나타내는 점입니다.

PointerInfo: 포인터의 위치를 나타내는 클래스입니다.

Polygon: Polygon 클래스는, 좌표 공간을 가지는 닫혀진 2 차원 영역의 기술을 캡슐화합니다.

PopupMenu: 이 클래스는, 컴퍼넌트내의 지정된 위치에 동적으로 표시할 수 있는 메뉴를 구현하기 위한의 것입니다.

PrintJob: 인쇄 작업을 개시해 실행하는 abstract 클래스입니다.

RadialGradientPaint: RadialGradientPaint 클래스는, 색의 엔방사상 그라이데이션 패턴으로 형상을 전부 칠하는 수단을 제공합니다.

Rectangle: Rectangle 는,Rectangle 객체의 좌표 공간에서의 좌상의 점 (x, y), 및 그 폭과 높이에 의해 둘러싸이는 좌표 공간내의 영역을 지정합니다.

RenderingHints: RenderingHints 클래스는, 키와 관련지을 수 있었던 값의 컬렉션을 정의 및 관리합니다.

RenderingHints.Key: draw와 이미징의 파이프라인의 다양한 알고리즘의 선택을 제어하기 위해서 RenderingHints 클래스와 함께 사용되는, 모든 키의 기저형을 정의합니다.

Robot: 이 클래스를 이용하면(자), 테스트의 자동화, 자동 실행의 데모, 및 마우스나 키보드 제어가 필요한 어플리케이션을 위해서(때문에), 네이티브인 시스템 입력 이벤트를 생성할 수가 있습니다.

Scrollbar: Scrollbar 클래스는, 자주(잘) 사용되는 사용자 인터페이스 객체인 스크롤 바를 실현합니다.

ScrollPane: 1 살의 아이 컴퍼넌트에 대해서, 자동 수평 또는 수직 스크롤 혹은 그 양쪽 모두를 구현하는 컨테이너 클래스입니다.

ScrollPaneAdjustable: 이 클래스는 ScrollPane 의 수평 및 수직 스크롤 바 상태를 나타냅니다.

SplashScreen: 스플래쉬 화면은, 어플리케이션의 기동시에, Java 가상 머신(JVM)의 개시전에 작성할 수 있습니다.

SystemColor: 시스템의 네이티브인 GUI 객체의 색을 나타내는 상징적 칼라를 캡슐화 하는 클래스입니다.

SystemTray: SystemTray 클래스는, 데스크탑의 시스템 트레이를 나타냅니다.

TextArea: TextArea 객체는, 텍스트를 표시하는 복수행 영역입니다.

TextComponent: TextComponent 클래스는, 텍스트의 편집을 가능하게 하는 모든 컴퍼넌트의 슈퍼 클래스입니다.

TextField: TextField 객체는, 1 행의 텍스트의 편집을 실시할 수 있도록(듯이) 하는 텍스트 컴퍼넌트입니다.

TexturePaint: TexturePaint 클래스는,BufferedImage 로서 지정되는 재질감으로 Shape 를 전부 칠하는 수단을 제공합니다.

Toolkit: 이 클래스는, Abstract Window Toolkit 의 모든 구현의 추상 슈퍼 클래스입니다.

TrayIcon: TrayIcon 객체는,시스템 트레이 에 추가할 수 있는 트레이 아이콘을 나타냅니다.

Window: Window 객체는 경계 및 도구모음을 가지지 않는 톱 레벨 윈도우입니다.

-열거형의 개요-

Component.BaselineResizeBehavior: 컴퍼넌트의 사이즈가 변화하는 것에 따라 baseline가 변화하는 공통의 방법을 열거합니다.

Desktop.Action: 액션의 종류를 나타냅니다.

Dialog.ModalExclusionType: 어느 톱 레벨 윈도우도, 모덜 다이얼로그에 의해 블록 되지 않게 마크 할 수가 있습니다.

Dialog.ModalityType: 모덜 다이얼로그는, 일부의 톱 레벨 윈도우에 대해서 모든 입력을 블록 합니다.

MultipleGradientPaint.ColorSpaceType: 그라데이션 보간을 실행하는 칼라 스페이스입니다.

MultipleGradientPaint.CycleMethod: 그라데이션 경계의 외부에서 draw 할 경우에 사용되는 메소드입니다.

TrayIcon.MessageType: 메세지 타입은, 메세지의 캡션에 표시되는 아이콘, 및 메세지의 표시시에 생성되는 시스템 사운드를 결정합니다.

-예외의 개요-

AWTException: AWT (Abstract Window Toolkit) 예외가 발생한 것을 나타냅니다.

FontFormatException: Font 클래스의 createFont 메소드에 의해 throw 되어 지정된 폰트가 무효라는 점을 나타냅니다.

HeadlessException: 키보드, 디스플레이, 또는 마우스에 의존하는 코드가, 키보드, 디스플레이, 또는 마우스를 지원하지 않는 환경에서 불려 갔을 경우에 throw 됩니다.

IllegalComponentStateException: 요구된 조작에 대해, AWT 컴퍼넌트가 적절한

상태에 없다고 하는 시그널입니다.

-에러의 개요-

AWTError: 중대한 Abstract Window Toolkit 에러가 발생했을 때에 throw 됩니다.

패키지 Java.awt 의 설명

사용자 인터페이스의 작성 및 그래픽스와 이미지의 페인트용의 모든 클래스를 포함합니다. 버튼이나 스크롤 바등의 사용자 인터페이스 객체는, AWT 용어로 컴퍼넌트로 불립니다. Component 클래스는 모든 AWT 컴퍼넌트의 루트입니다. 모든 AWT 컴퍼넌트가 공유하는 프로퍼티의 상세한 것에 대하여는 Component 를 참조하면 된다.

일부의 컴퍼넌트는, 사용자가 컴퍼넌트를 조작하면 이벤트를 트리거합니다. AWTEvent 클래스와 그 서브 클래스는, AWT 컴퍼넌트가 트리거하는 이벤트를 나타내기 위해서 사용됩니다. AWT 이벤트 모델의 상세한 것에 대하여는 AWTEvent 를 참조하면 된다.

컨테이너는 컴퍼넌트와 그 외의 컨테이너를 포함하는 컴퍼넌트 입니다. 또, 컨테이너는 컨테이너내의 컴퍼넌트의 시각적인 배치를 제어하는 레이아웃 매니저를 가질 수도 있습니다. AWT 패키지에는, 몇개의 레이아웃 매니저 클래스와 사용자 독자적인 레이아웃 매니저를 구축하는 인터페이스를 납입할 수 있고 있습니다. 상세한 것에 대하여는, Container 및 LayoutManager 를 참조하면 된다.

추가 스펙

「The AWT Focus Subsystem」

「The AWT Modality」

도입된 버젼: JDK1. 0

2.3 암호화 부분

자바의 암호화 관련 패키지를 이용하여 구현하였습니다. 알고리즘은 기본 DES 암호화를 적용 하였다.

1. DES 정의

- 56Bit의 키를 이용하여 64Bit의 평문 블록을 64bit의 암호문 블록으로 만드는 블록 암호방식의 미국표준

- 3500년간의 암호역사에 있어 혁신 중에 하나
- 평문을 8개의 문자 단위로 나누어 각 블록에 확산과 혼돈을 16번 (round) 반복

2. DES의 특징
- 견고성
· 통계적 암호해독에 견딜 수 있도록 충분히 견고하게 제작
· DES의 암호분석방법은 모든 키를 다 적용하는 것 임(brute force attack)
- 활용 : Top Secret의 비밀보호 보다는 상업적인 자료보호 활용에는 가능

3. DES의 한계점
- 최근 반도체 칩 기술의 발달로 초기 DES 알고리즘은 쉽게 해독 가능함
- 보안성을 강화한 2중 DES, 3중 DES 사용을 권고
- DES의 한계점을 개선한 AES (Advance Encryption Standard)가 새로운 미국 표준

4. DES의 운용 모드 .

DES 알고리즘은 자료 보안을 위한 기본 구조로서 응용에의 적용을 위하여 4가지 “운용 모드”가 정의되었다(FIPS PUB 74, 81). 이들 4가지 모드는 사실상 DES를 이용할 수 있는 모든 가능한 응용을 망라하기 위한 것이다. 각 모드는 표 2-3-4-1 에 요약되어 있다

운용모드	설명	전형적 응용
전자코드북(ECB) (Electronic Codebook)	64 비트의 각 평문 블록이 동일 키를 이용하여 독립적으로 암호화됨.	단일 값의 기밀전송 (예 : 암호키)
암호 블록 체이닝 (CBC) (Cipher Block Chaining)	암호 알고리즘의 입력이 다음 평문 64비트와 선행 64비트의 XOR의 결과임.	범용 블록형 전송 인증
암호 피드백(CFB) (Cipher eedback)	입력은 한번에 j비트씩 처리됨. 선행 암호문 블록이 암호 알고리즘의 입력으로 사용되어 의사 난수 출력(pseudorandom output)을 생성하고, 이것은 다시 평문과 XOR되어 암호 블록을 생성.	범용 스트림형 전송 인증
출력 피드백(OFB) (Output eedback)	CFB와 유사한 방식이며, 암호 알고리즘의 입력으로 선행단계의 DES 출력을 이용.	잡음 있는 채널상의 스트림형 전송 (예 : 위성통신)

[표1] DES 운용모드

5. 이중 DES의 처리 (단일 DES의 개선)
- 키 2개의 서로 다른 키(K 1, K 2)로 2번 암호화 및 복호화 하는 것
v 2 중 DES 암호화 : c = Ek 2 (E k1 (m))
v 2 중 DES 복호화 : m = Dk 1 (D k 2 (c))
v 2 중 DES 의 키 길이는 112 비트(56 Î 2)
Ÿ 암호키를 찾는데 2 번의 작업이 필요

6. 이중 DES의 안정성
- 관측된 평문과 암호문이 주어지고 중간충돌 공격(meet-in-the-middle attack)을 하면 2중 DES가 단일 DES에 비하여 안전성이 증대하지 않음을 알 수 있음 (암호 해독 시간이 단일 DES의 2배가 안됨)

7. Java의 DES적용
  자바 측에서 제공할 수 있는 암호에 관련된 패키지는 JCA(Java cryptograhpy arch.)와 JCE(Java cryptography extension)을 사용할 수 있다. JCA는 기본적으로 Java2 Runtime Environment의 일부이며, JCA는 그것의 확장패키지이다. 기본적인 JCA에서는 전자서명, 메시지 다이제스트, 키생성기등의 클래스를 가지고 있으며 그러한 기본적인 클래스들을 우리가 사용하고자 했을 때 new에 의한 생성이 아니라 이미 아키텍쳐가 가지고 있는 암호화 기법에 의하여 factory형태의 클래스에게 생성을 요청하여야 한다.

1). 암호화 메소드
getIntance(), init(), update(), doFinal()로서 암호화 알고리즘의 생성, Cipher 인스턴스의 초기화, 암복호화, 암호화된 배열을 획득을 하는 메소드들이다.

2). 복호화 메소드
Javax.crypto.KeyGenerator클래스는 암호화와 복호화에 필요한 키를 생성 해내는데 쓰이며 getInstance(), init(), generateKey()의 메소드 정도면 Cipher에 필요한 키를 만들어낼 수 있다.

적용의 예

```
import Java.security.*;
import Javax.crypto.*;

public class SimpleExample {
    public static void main(String [] args) throws Exception {
        if( args.length != 1) {
            System.out.println("Usage : Java SimpleExample text ");
            System.exit(1);
        }
        String text = args[0];

        System.out.println("Generating a DESded (TripleDES) key...");

        // Triple DES 생성
        KeyGenerator keyGenerator =
        KeyGenerator.getInstance("DESede");
        keyGenerator.init(168);  // 키의 크기를 168비트로 초기화
        Key key = keyGenerator.generateKey();

        System.out.println("키생성이 완료되었음");

        // Cipher를 생성, 사용할 키로 초기화
        Cipher cipher = Cipher.getInstance("DESede/ECB/PKCS5Padding");
        cipher.init(Cipher.ENCRYPT_MODE, key);

        byte [] plainText = text.getBytes("UTF8");

        System.out.println("Plain Text : ");
        for (int i = 0; i < plainText.length ; i++)       {
            System.out.print(plainText[i] +  " ");
        }

        // 암호화 시작
        byte [] cipherText = cipher.doFinal(plainText);

        // 암호문서 출력

        System.out.println("₩nCipher Text : ");
        for (int i = 0; i < cipherText.length ; i++)      {
            System.out.print(cipherText[i] +  " ");
        }

        //복호화 모드로서 다시 초기화
        cipher.init(Cipher.DECRYPT_MODE, key);

        //복호화 수행
        byte [] decryptedText = cipher.doFinal(cipherText);
        String output =  new String(decryptedText, "UTF8");
        System.out.println("₩nDecrypted Text : " +  output);
    }
};
```

위의 코드로서 간단하게 살펴볼 수 있는데, 출력되는 생성되는 키에 따라 결정

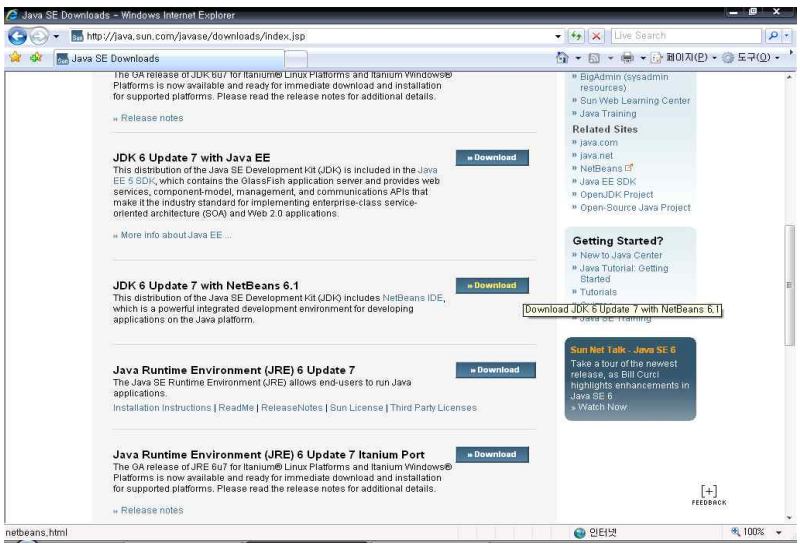
됨으로 매번 다른 결과의 암호화된 문자열을 볼 수 있는 특징을 가지고 있다. Blowfish의 대칭암호화 기법 또한 같은 방법에 의하여 만들어낼 수 있는데 단순히 위의 코드 상에서의 변화는 KeyGenerator에서 "Blowfish"와 Cipher에서 인스턴스를 얻어낼때 단순히 "Blowfish/ECB/PKCS5Padding"을 이용하여 처리하면 128비트의 키를 이용한 암복호화를 테스트해볼 수 있다.

2.4 JDK

1. JDK 다운로드

[Java.sun.com]

* Downlads에 Java SE의 메뉴를 선택 Java SE는 StandardEdition으로 일반적인 JDK를 갖는다.



[Java.sun.com]

* 최신버젼인 JDK6 Update7 With NetBeans6.1을 다운로드 한다.

* 여기서 NetBeans6.1은 Java사용 시 GUI환경으로 예로 비주얼Basic나 비주얼 C++를 생각하면 된다.1

2. JDK란 무엇인가?

자바의 큰 장점중의 하나는 JDK라는 개발도구가 무료로 제공된다는 점이다 JDK는 썬 사의 홈페이지에서 쉽게 다운로드 할 수 있으며 자바 프로그램을 개발하기 위한 다양한 개발도구가 포함되어 있다

-①자바 컴파일러-

자바로 만들어진 소스를 컴파일 하여 바이트코드를 만드는데 필요한 개발도구 가장기본적인 개발도구로서 Javac 명령으로 수행

-②자바 해석기-

컴파일에 의해서 생성된 바이트 코드를 자바 가상머신에서 실행하도록 해주는 개발도구 Java 명령으로 수행

-③자바 애플릿 뷰어-

자바 애플릿을 실행할 수 있는 배갈도구이다 자바 애플릿은 웹 브라우저에서만 실행이 되지만 개발 시에 매번 웹 브라우저를 사용하기는 불편하다 따라서 애플릿의 개발 자바 애플릿 뷰어가 많이 사용된다.

-④자바프로그램 개발과정-

[자바소스] [자바 컴파일러] [자바 해석기]

test.Java --.Java --> Javac ----.class ---> Java -------> ((실행))

3. 프로그램 개발 및 테스트

3.1 프로그래밍

1. 기본 인터페이스 동작

포트명 - TextField 접속할 포트 번호를 입력하는 부분입니다.

1부터 65535이하의 정수만 입력하여야 합니다. 그러나 1024번 이내는 이미 예약되어 사용되는 번호가 많으니 실제적으로는 1024이상의 숫자를 입력해주는 편이 좋습니다. 코드 내에 별다른 예외처리를 해두지 않아서, 다른 에러가 발생하진 않으나, 문자 등을 입력할 경우 정상적으로 동작하지 않습니다.

Password - TextField

DES 인코딩/디코딩에 사용할 코드를 입력하는 부분입니다. 이 부분도 입력을 편하게 하기 위해 숫자 값만 받아들이고 있습니다.

처음 프로그램을 실행하였을 때 기본적으로 00000000000000000000 이 입력되어 있으며, 이 값일 경우는 조건문에 의해 인코딩을 적용하지 않고 메시지를 전송하게 되어있습니다. 스니핑 프로그램으로 패킷을 확인할 경우는 이형태로 보내면 pure메시지를 확인할 수 있습니다. 입력 값이 위 값보다 적거나 길어도 숫자일 경우 이상 없이 동작합니다.

위 값보다 자릿수가 짧으면 뒤에 0을 추가로 붙여주고, 위 값보다 길 경우에는 자동적으로 필요 없는 부분을 잘라내도록 하였습니다. 그러나 변환 후 엔터키를 입력해줘야 적용됩니다.

텍스트 입력 박스 - TextField

전송할 텍스트를 입력하고 Enter키를 누르면 메시지를 전송합니다.

채팅내용 출력창 - TextArea

채팅 내용을 화면에 출력해줄 영역입니다.

TextArea는 Editable(false)로 되어있습니다.

연결하기 - Button

해당 포트로 메시지 수신을 대기합니다.

그리고 메시지 전송에 사용할 포트 번호를 내부에 저장하고 메시지 전송 이벤트가 발생할 때 포트명에 적혀있는 포트로 전송합니다.

연결 끊기 - Button

메시지 수신 대기 상태를 종료합니다.

구현상으로는 대기를 종료하고, 전송포트를 0번으로 초기화합니다.

about - Button
본 프로그램의 작성자등의 정보를 출력하는 Dialog를 화면에 띄워줍니다.

채팅종료 - Button
프로그램을 종료합니다.

암호문출력 , 평문출력 - Radio버튼
둘 중에 하나만 선택하게 하기위한 라디오 버튼입니다.
실제적으로 어느 쪽이든 전송내용은 같습니다.
인코딩/디코딩된 시 코드가 데이터를 일반적인 문자열을 출력하듯 요청할 경우 제대로 표현되지 않는 범위로도 변환되기 때문에 정상적인 인코딩이 되어있지 않으면 화면에 아무것도 출력하지 않는 경우가 많습니다. 이러한 상태에서 인코딩이 되고 있는지 확인하기 위해 암호문출력을 삽입하였습니다.
암호문출력의 경우 디코딩이 적용되지 않으므로 특정 킷값으로 인코딩한 데이터를 전송할 경우 깨진 문자열이 출력됩니다.
(현재 이 부분은 암호문으로 출력하기 때문에 깨지는데, byte로 출력할 경우 인코딩 데이터의 숫자 값을 확인 할 수 있습니다.)

2. 암호화 사용 라이브러리

Key 해석기
Javax.crypto.SecretKeyFactory
-public static final SecretKeyFactory getInstance(String algorithm) : 주어진 알고리즘에 대하여 새로운 SecretKeyFactory를 생성한다. 알고리즘은 “DES"와 같은 대칭 암호 알고리즘이다.
-pubic static final SecretKeyFactory getInstance(String algorithm, String provider) : 주어진 프로바이더로 SecretKeyFActory를 생성한다.
-public final SecretKey generateSecret(KeySpec keySpec) : KeySpec를 SecretKey로 변환하기 위하여 사용된다.

```
public SecertKey makeDESKey(byte[] input, int offset)
throws NoSuchAlgorithmException, InvalidkeyException,
InvalidkeySpecException{
SecretKeyFactory desFactory = SecretKeyFactory.getInstance("DES");
KeySpec spec = new DESKeySpec(input, offset);
return desFactory.generateSecret(spec);
}
```

-public final KeySpec getKeySpec(SecretKey key, Java.lang.ClasskeySpec) : 주어진 SecretKey로부터 KeySpec을 생성한다.

3. 중요코드 분석

본 프로그램에서 사용한 자바의 Encryption 대상은 byte 배열에 담겨진 상태로 이루어진다.

```
byte[] plaintext=null
plaintext = msg.getBytes("UTF8");
```

저장된 pwdata 배열 값을 기초로 인코딩과 디코딩에 사용할 킷값을 생성

```
byte[] pwdata = {0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00};
DESKeySpec desKeySpec=null
SecretKeyFactory keyFactory=null
SecretKey desKey=null
desKeySpec = new DESKeySpec(pwdata);
keyFactory = SecretKeyFactory.getInstance("DES");
desKey = keyFactory.generateSecret(desKeySpec);
Cipher cipher = null
cipher = Cipher.getInstance("DES/ECB/PKCS5Padding");
```

Encryption 적용부분

```
cipher.init(Cipher.ENCRYPT_MODE, desKey);
byte[] ciphertext = null
ciphertext = cipher.doFinal(plaintext);
// ciphertext에 Encryption 된 텍스트가 저장됨
```

Decryption 적용부분

```
cipher.init(Cipher.DECRYPT_MODE, desKey);
byte[] decryptedText=ciphertext;
decryptedText = cipher.doFinal(ciphertext);
String output=null
output = new String(decryptedText, "UTF8");
// Output에 Decryption 된 텍스트가 저장됨
```

네트워크 전송을 위해 Encryption 된 텍스트는 네트워크를 통해 전송 시 값이 깨질 수 있기 때문에, 값을 임의의 형태로 변경하여 전송함.

```
// disencryption
byte[] cipoutput = new byte[ciphertext.length];
```

```
String cipoutputs = ""
for (int i=0;i<ciphertext.lengthi++)
{
cipoutput[i] = (byte)ciphertext[i];
cipoutputs += (byte)ciphertext[i]+"."
}
ta_talk.append("Local descipher text - " + output+"₩n");
data = new DatagramPacket(cipoutputs.getBytes("UTF8"),
                cipoutputs.getBytes("UTF8").length, ia ,port_i);
```

3-2 구축

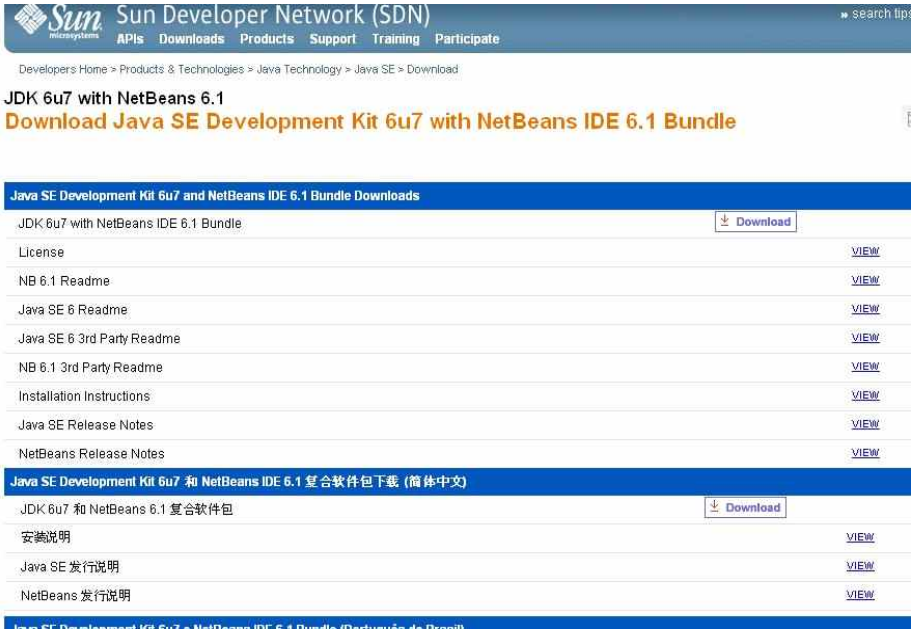
1. java sun 홈페이지 접속

[그림1]

버전을 SE로 이동

[그림2]

다운로드로 들어감



[그림3]

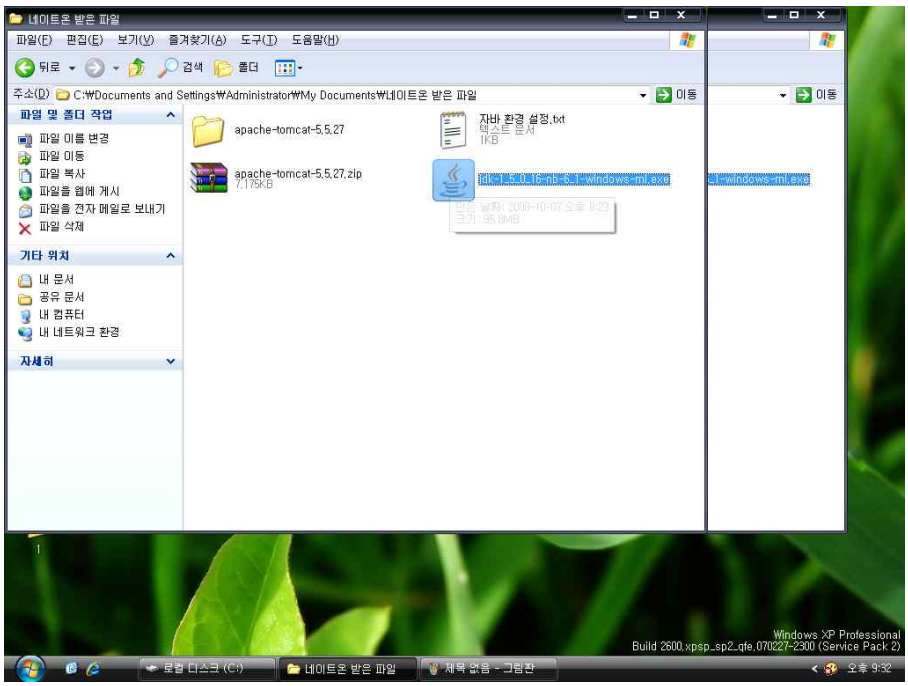
맨 위에 것을 다운로드 함 (Windows 버전)

[그림4]

JDK 6.1 버전 다운로드

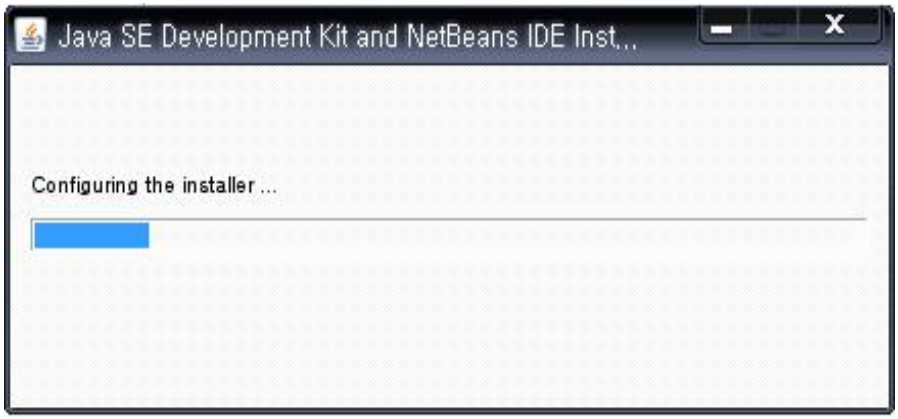
[그림5]

2. 받은 파일을 인스톨 한다



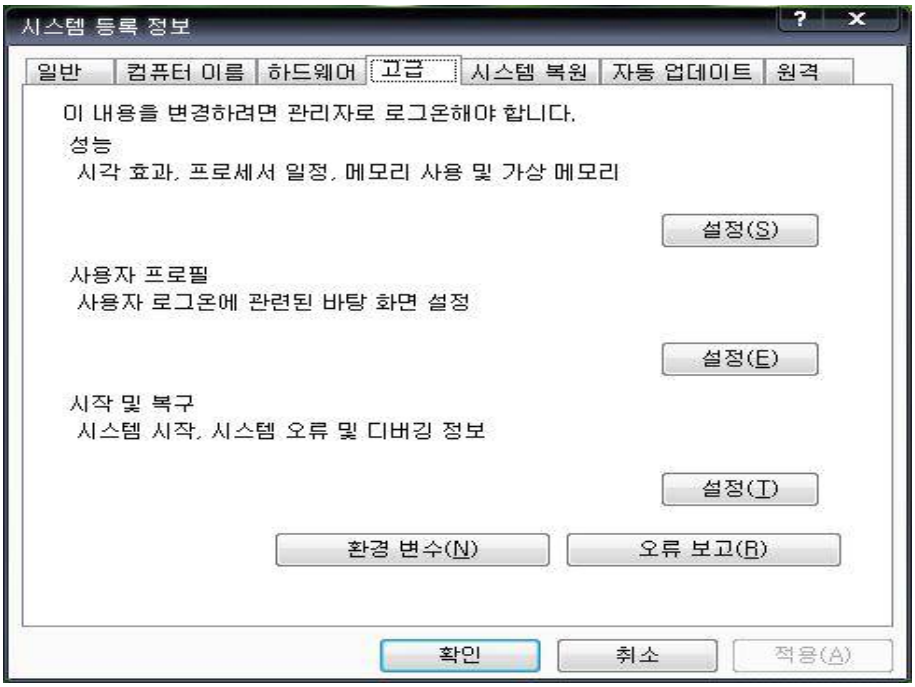
[그림6]

인스톨 되는 과정



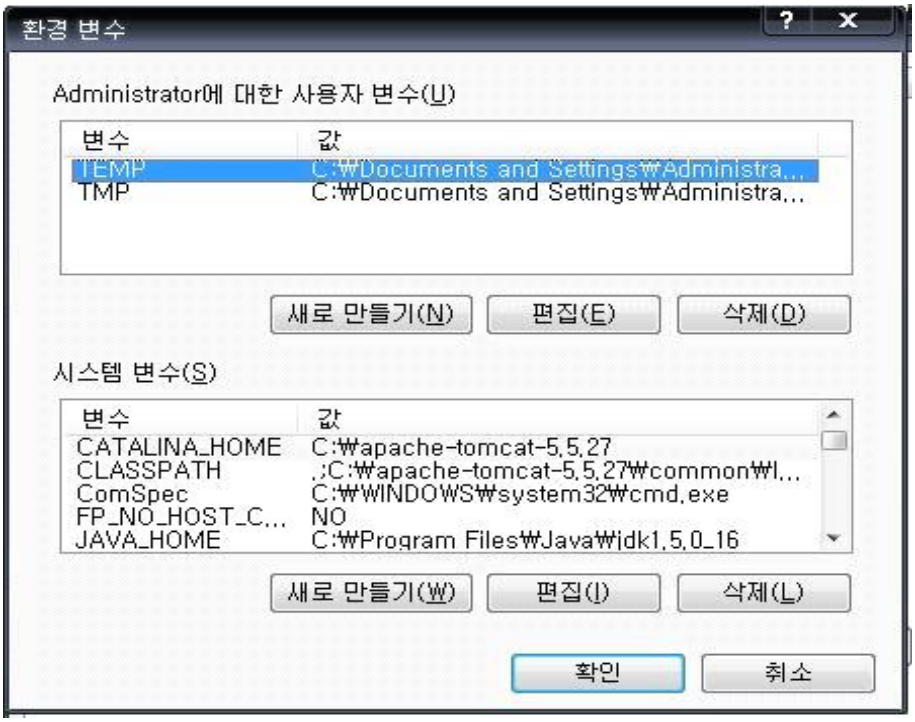
[그림7]

인스톨이 되면 내 컴퓨터 속성에서 고급으로 간다



[그림8]

3. 환경변수를 선택한다



[그림9]

환경변수 CLASSPATH 값을 변경해준다 (. 은 절대경로 지정)

[그림10]

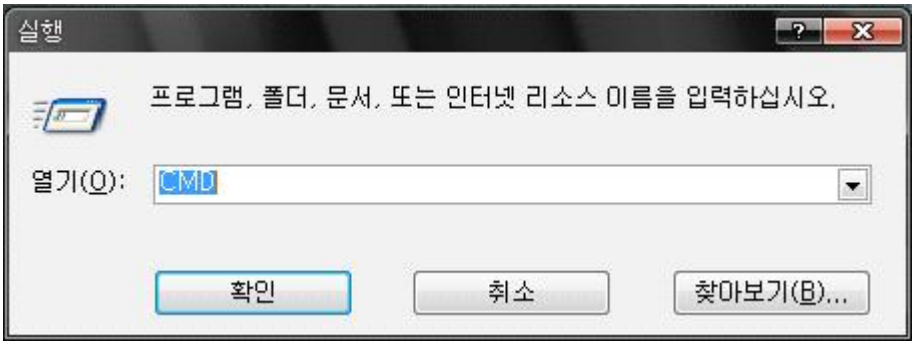
다음으로 PATH 값을 변경해준다 PATH값은 bin까지 지정한다
ex):C:₩Program Files₩Java₩jdk1.5.0_16₩bin

[그림11]

JAVA _ HOME 은 bin 전 까지만 지정해준다

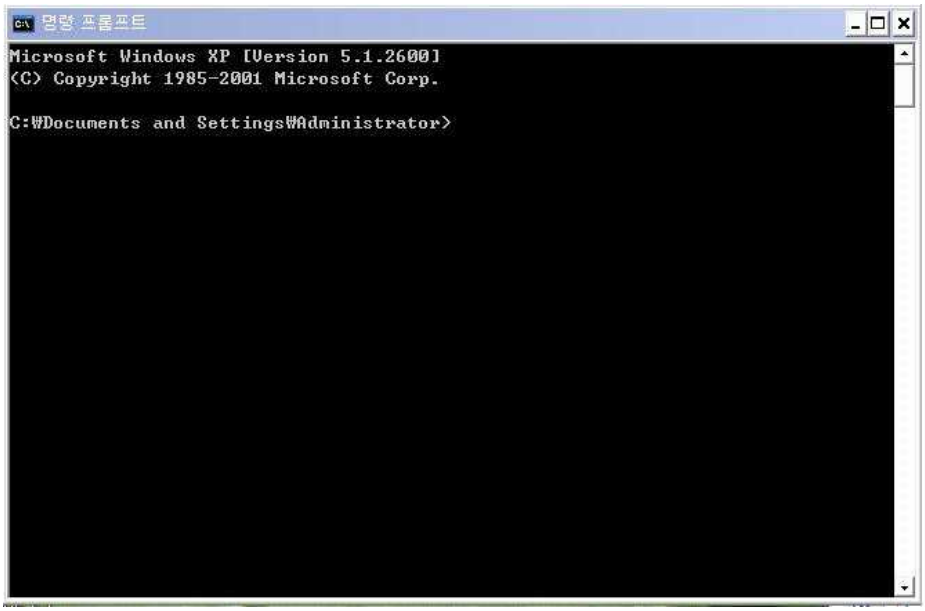
[그림12]

인스톨한 JDK를 실행하기 위해 시작에 실행에 cmd를 친다



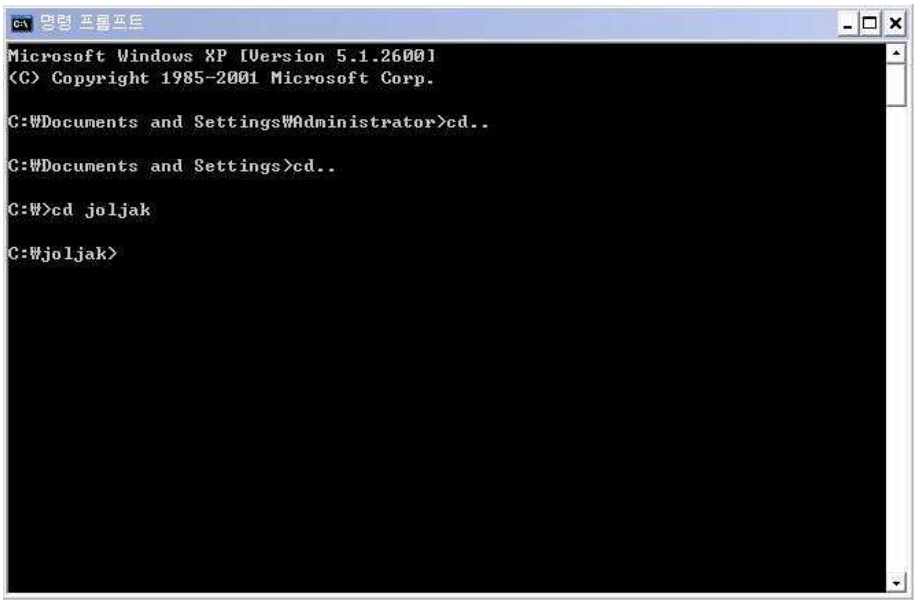
[그림13]

그럼 바로 명령 프롬프트가 뜬다



[그림14]

JDK가 있는 폴더로 이동한다 (여기서는 joljak 폴더)

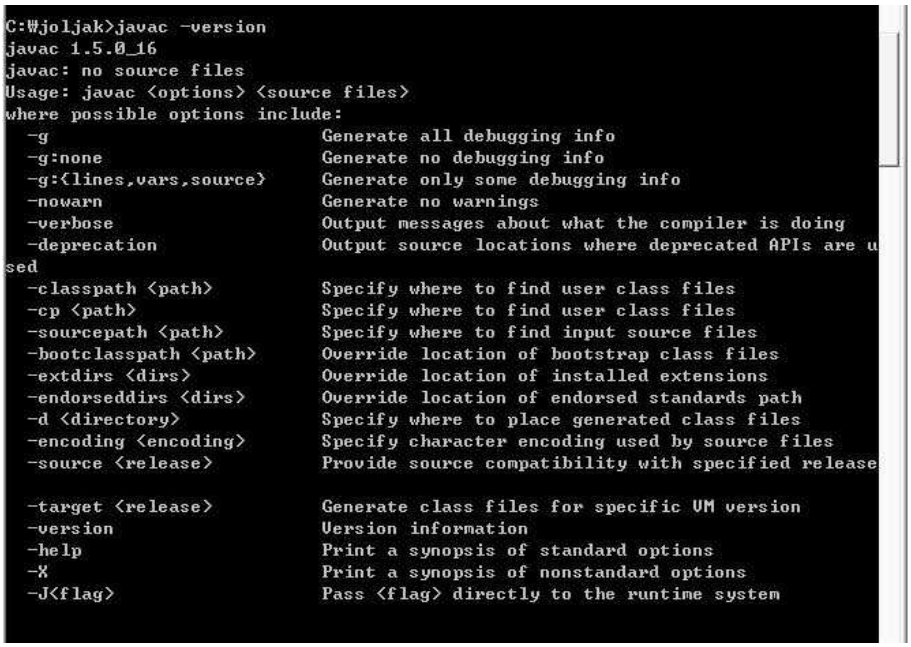


[그림15]

4. JDK 프로그램 설치 확인 (javac & java)

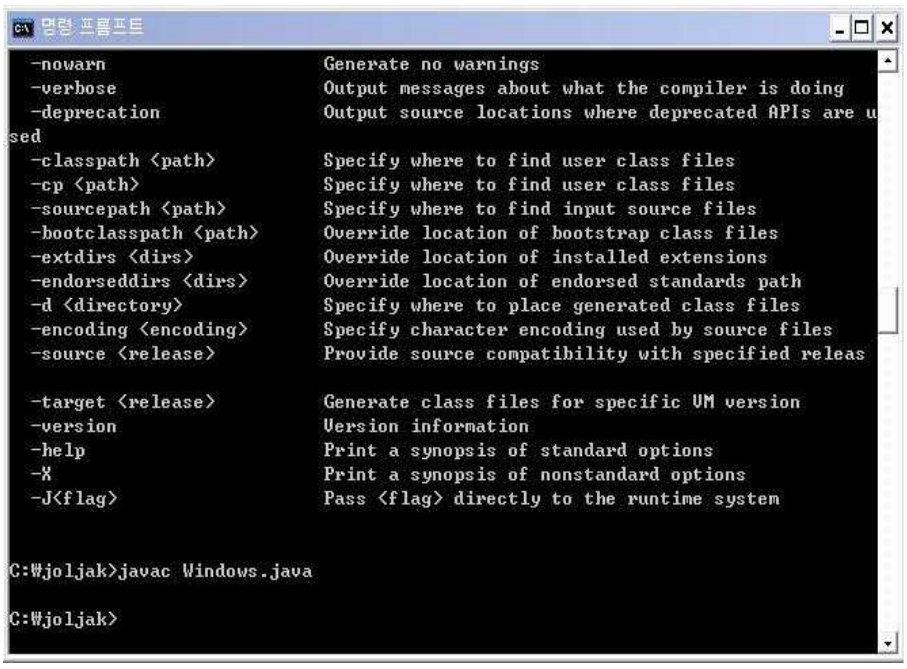
[그림16]

JDK 버전 확인 (javac -version)



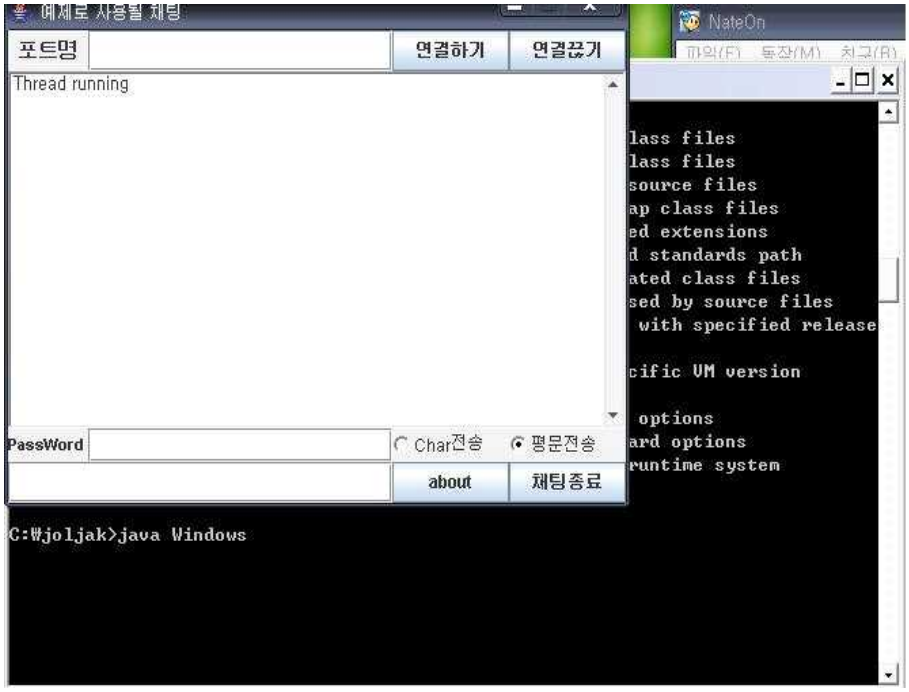
[그림17]

JDK 컴파일해준다 (javac Windows.java)



[그림18]

컴파일된 프로그램을 실행하면 채팅창이 실행된다 (java Windows)



[그림19]

3-3 운영

1. 2대의 pc를 이용한 채팅 테스트 포트 1234로 맞췄을때 pc1 -> pc2 (평문)

[그림20]

pc2의 화면은 이렇다

[그림21]

2대의 pc를 이용한 채팅 테스트 포트를 1234로 맞췄을때 pc2 -> pc1(평문)

[그림22]

pc1의 화면은 이렇다

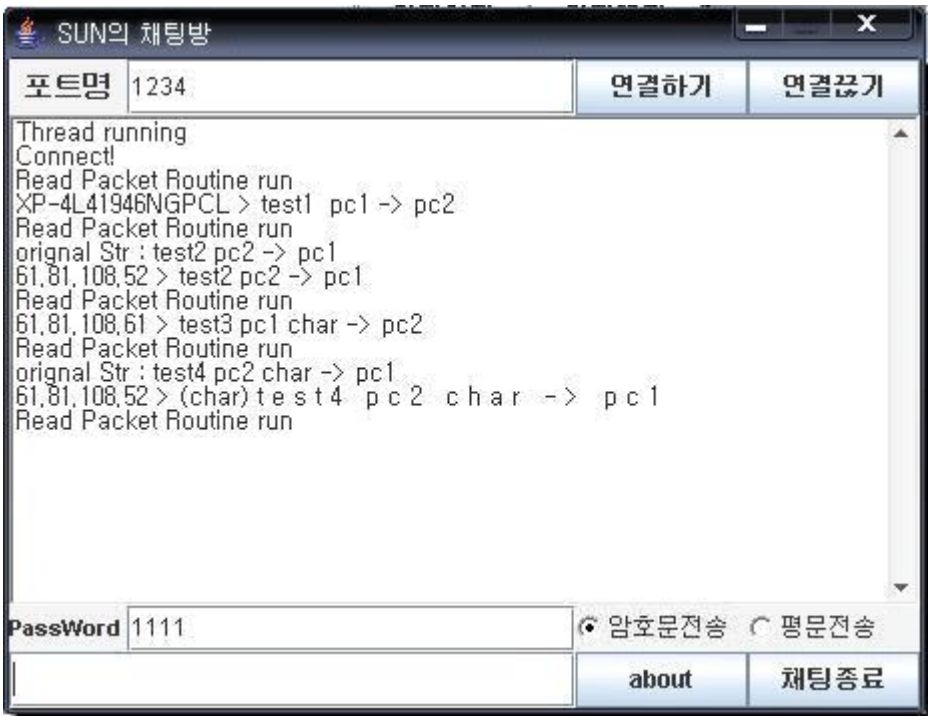
[그림23]

2. 2대의 pc를 이용한 채팅 테스트 포트를 1234로 맞췄을때 pc1-> pc2(암호문) 비밀번호 1111을 알고 있을때

- pc1에서 암호문 전송 화면

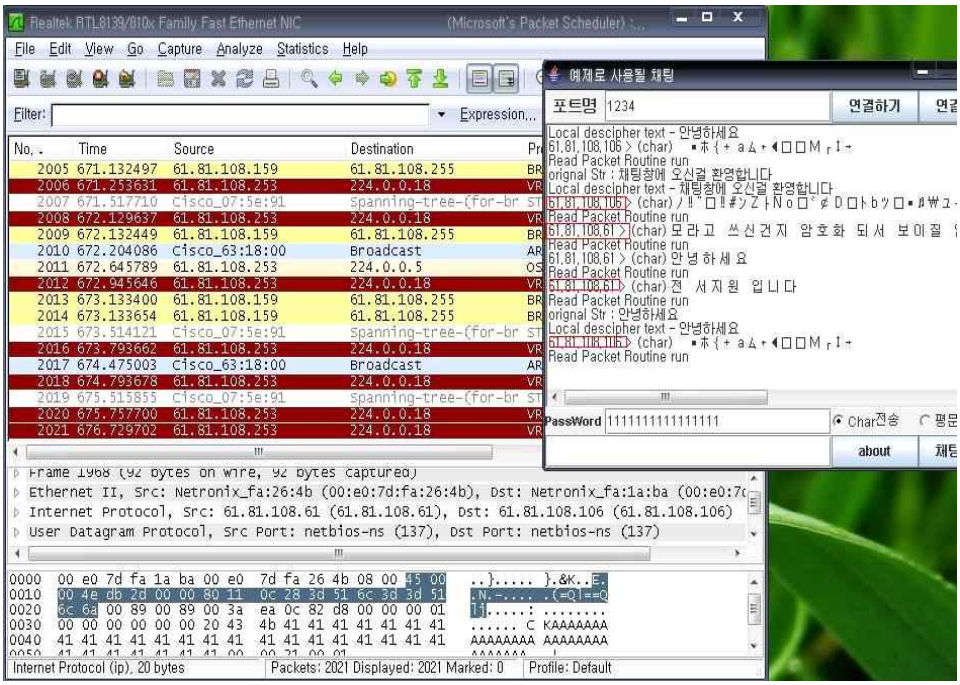
[그림24]

pc2가 비밀번호 1111을 알고 있을때 화면은 이렇다



[그림25]

2대의 pc를 이용한 채팅 테스트 포트를 1234로 맞췄을때 샤크 프로그램을 이용한 암호문 전송할때 암호문이 어떻게 보여지는지 소스로 보여주기



[그림26]

채팅창에 쓴 글이 암호화 되서 남들이 알아 볼 수 없게 보여준다

4. 3대의 pc를 이용한 채팅 포트를 1234로 맞췄을때 pc1 -> pc2.pc3 (평문)

- pc1의 화면

[그림27]

- pc2의 화면

[그림28]

- pc3의 화면

[그림29]

5. 3대의 pc를 이용한 채팅 포트를 1234로 맞췄을때 pc2 -> pc1.pc3 (평문)

- pc2의 화면

[그림30]

- pc1의 화면

[그림31]

-pc3의 화면

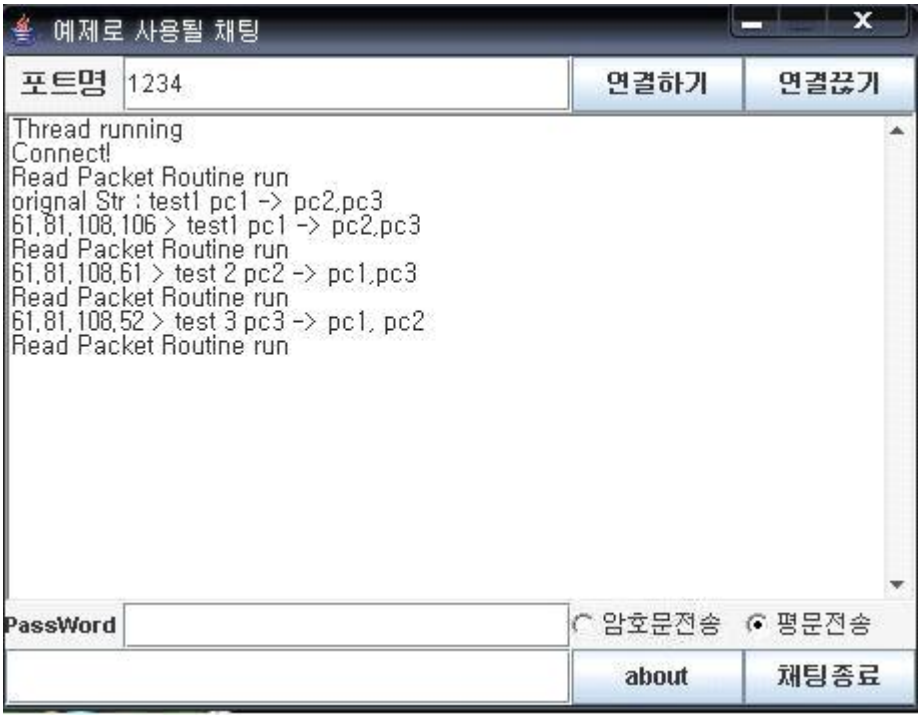
[그림32]

6. 3대의 pc를 이용한 채팅 포트를 1234로 맞췄을때 pc3 -> pc1.pc2 (평문)

- pc3의 화면

[그림33]

- pc1의 화면



[그림34]

- pc2의 화면

[그림35]

7. 3대의 pc를 이용한 채팅 포트를 1081로 맞췄을때 pc1 -> pc2.pc3(암호문) pc1만 비밀번호 알 경우 (가나다)를 쳤을때

- pc1의 화면

[그림36]

- pc2의 화면(암호화 돼서 알아 볼 수가 없다)

[그림37]

- pc3의 화면(암호화 돼서 알아 볼 수가 없다)

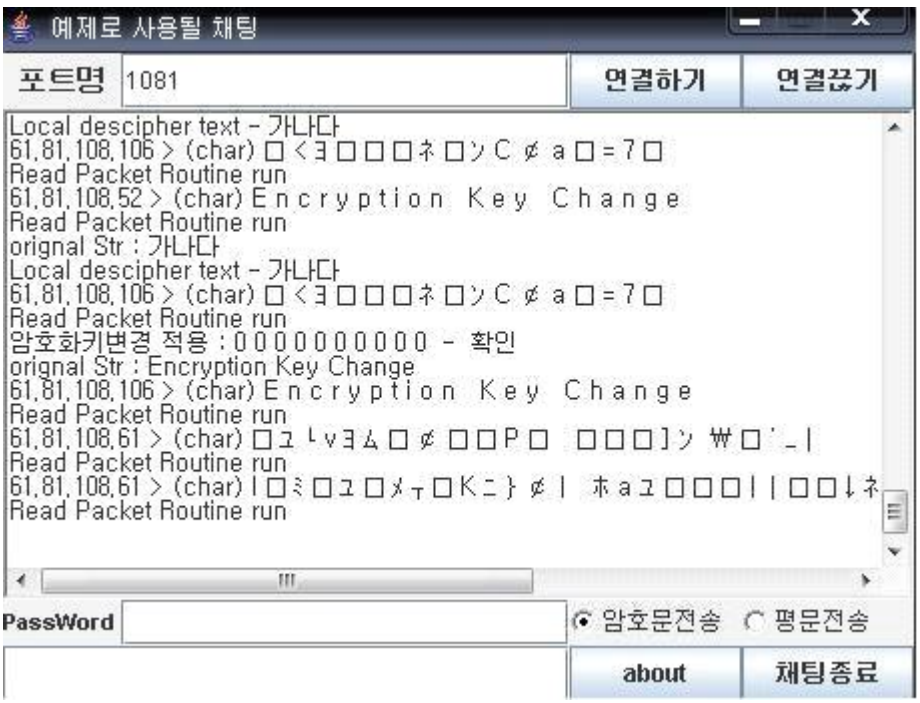
[그림38]

8. 3대의 pc를 이용한 채팅 포트를 1081로 맞췄을때 pc2 -> pc1.pc3(암호문) pc2만 비밀번호 알 경우 (암호화 전송 테스트)라 쳤을때

- pc2의 화면

[그림39]

- pc1의 화면(암호화 되서 알아 볼 수가 없다)



[그림40]

- pc3의 화면(암호화 되서 알아 볼 수가 없다)

[그림41]

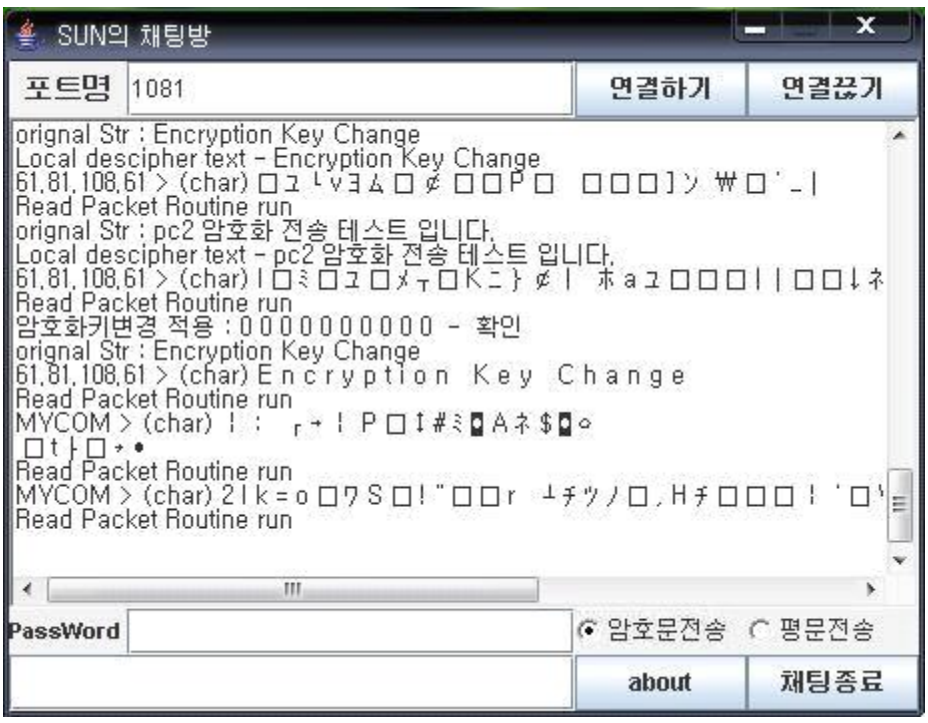
9. 3대의 pc를 이용한 채팅 포트를 1081로 맞췄을때 pc3 -> pc1.pc2(암호문)
pc3만 비밀번호 알 경우 (암호화 전송 테스트 입니다)라 쳤을때
- pc3의 화면

[그림42]

- pc1의 화면(암호화 되서 알아 볼 수가 없다)

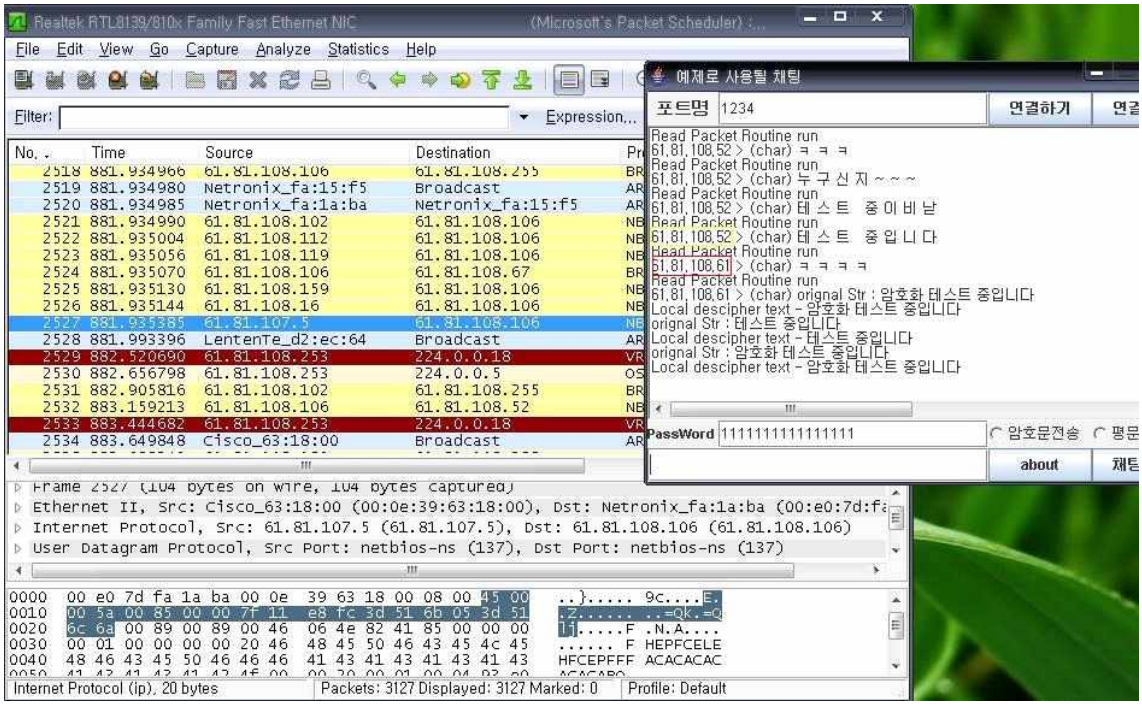
[그림43]

- pc2의 화면(암호화 되서 알아 볼 수가 없다)



[그림44]

- 3대의 컴퓨터를 이용한 채팅 테스트 포트를 1234로 맞췄을때 샤크 프로그램을 이용한 암호문 전송할때 암호문이 어떻게 보여지는지 소스로 보여주기



[그림45]

채팅창에 쓴 글이 암호화 되서 남들이 알아 볼 수 없게 보여준다

4. 결론

우리나라에서 채팅이 많이 상용화 되고 사용량이 증가되고 있는 것을 다들 아실 겁니다. 친구들과의 수다 , 중요업무나 회의 정말 많은 곳에 쓰이고 있습니다. 하지만 사회가 발전하고 메신저나 채팅의 발달함에 따라 해킹을 통해 이를 좋은 곳에 쓰는 사람들도 있지만 그와 반대로 악 이용하는 사람들이 점차 증가하고 있습니다. 개인정보 유출, 계좌번호나 중요한 문서를 빼돌려 자신의 이익만 추구하는 요즘시대에 보안에 신경을 쓰는 것이 시급한 문제 일 것입니다 .

저희 조가 조금이나마 보안에 신경을 쓰고 만든 것이 Java로 구현한 채팅 프로그램입니다 자바를 배우지 않고 처음 접하다 보니 많은 어려움이 있었습니다. 언어에 대한 인식과 소스가 어떤 것을 의미하는지 알수가 없어서 난처했었습니다 그래서 자바를 책과 인터넷과 지인들에게 물어서 배우고 하나씩 풀어 나갈 수 있었습니다 자바라는 언어가 C언어랑 비슷할 것 같으면서도 많이 다르기 때문에 소스코딩 작업에 많은 어려움이 있었고 많은 에러가 나서 그것을 찾아서 해결하기가 쉬운 일은 아니었습니다. 에러 잡는데 에 시간을 많이 투자하다보니 속도가 늦은 감도 있었지만 제 날짜에 작업 완료를 할 수 있어서 다행이라 생각합니다.

다른 채팅창과는 달리 평문과 암호문을 두어 채팅하게 하는 방식을 썼고 평문일 때에는 그냥 대화하는 것이 상대방에게 다 보이는 일반적으로 우리가 아는 채팅과 같은 형식이고 암호문일 때에는 비밀번호를 두어 상대방에서 비밀번호를 알면 대화내용은 보이는데 모를 경우에는 글자가 암호화로 바뀌어서 무슨 내용인지 전혀 알아볼 수 없게 하였습니다.

소켓통신과 같이 일대일로 대화하는 것이 아니라 브로드캐스트 방식을 써서 같은 대역폭 안에 있으면 누구나 쉽게 접속 할 수 있게 하였습니다.

향후과제

지금까지 상태는 평문을 전송 한때는 문제가 발생되지 않았지만 암호문을 전송할 때는 메시지가 오류가 나거나 아무것도 뜨지 않거나 암호화 되거나 이 세 가지로 나뉘게 되는 것을 볼 수 있었습니다 암호화 되는 것을 더욱더 보강해서 자세하게 보여 줄 수 있다면 더 좋은 프로그램이 나 올수 있다고 생각합니다. 그리고 채팅창에 들어오면 아이피 주소가 뜨는데 그것을 대화명이나 이름으로 바꾸게 된다면 좀 더 깨끗해진 환경 속에서 채팅을 할 수 있게 될 것입니다.

5. 참고 문헌

문 헌

1. Algorithm 알고리즘
출판사: 이한출판사 / 저자: 조유근 홍영식 이지수 김명

2. Beginning 자바 웹 서비스
출판사: 정보문화사 / 저자: 앙리 베게 외 공저, 김재현 최적진 공역

3. PROFESSIONAL Java XML
출판사: 정보문화사 / 저자 KAL Aimed 외 15인 공저, 김선태 역, 박성수 이 창준 감수

4. Java Language Bible
출판사: 영진.COM / 저자 이 현우 김형국 김명호 공저

사이트

http://blog.kichang.com/32

http://www.50001.com/

http://cafe.naver.com/painmaster

http://www.ktword.co.kr