서블릿(Servlet)

* Server + Let or Server + Applet
* 클라이언트 요청을 처리하고 그 결과를 다시 클라이언트에게 전송하는 Servlet 클래스의 구현 규칙을 지킨 자바 프로그램

서블릿 컨테이너(Servlet Container)

* 서블릿의 생명주기를 관리하고 요청에 따른 스레드 생성
* 역할
  + 통신지원
    - 서블릿과 웹 서버가 통신할 수 있는 손쉬운 방법을 제공  
      통신의 경우 소켓을 만들고 특정 포트를 리스닝하고 연결 요청이 들어오면 스트림을 생성해서 요청을 받는데 이 기능을 해줌
  + 생명주기 관리
    - 기동되는 순간 서블릿 클래스를 로딩해서 인스턴스화하고 초기화 메서드를 호출하고 요청이 들어오면 적절한 서블릿 메소드를 찾아서 호출
  + 멀티스레딩관리
    - 해당 서블릿 요청이 들어오면 스레드를 생성해서 작업을 수행한다. 즉 동시의 여러 요청이 들어온다면 멀티 스레딩 환경으로 동시다발적인 작업을 관리한다.
  + 선언적인 보안관리
    - 보안관련 기능 제공

톰캣

* 웹 서버와 연동하여 실행할 수 있는 자바 환경을 제공하여 자바 서버 페이지(JSP)와 자바 서블릿이 실행할 수 있는 환경을 제공
* HTTP 서버도 자체 내장하기도 함
* WAS의 기능을 수행함, 요즘 톰캣의 버전이 올라가면서 웹 서버의 기능도 충분히 잘 수행하기 때문에 웹 서버와 WAS를 나눠서 생각하지 않아도 괜찮음

웹 서버와 WAS의 차이

1. 웹서버

Client/Server 모델과 웹의 HTTP를 사용하여 웹 페이지가 들어 있는 파일을 사용자에게 제공하는 프로그램

1.1 기능

a. Clustering 기능: 사용자 요청이 발생하면 상황에 따라 각각의 WAS에 요청을 넘김

b. Cache 기능: css, js, image 등의 리소스 파일을 갖고 있다가 WAS를 거치지 않고 사용자에게 직접 넘김

1.2 Example

a. 32비트 윈도우와 유닉스에서 사용할 수 있는 **아파치**

b. 윈도우 NT에 딸려 나오는 **IIS**

c. 넷스케이프의 **엔터프라이즈 서버**

2. WAS(Web Application Server)

웹과 기업의 기간 시스템 사이에 위치하면서 웹 기반 분산 시스템 개발을 쉽게 도와주고 안정적인 트랜잭션 처리를 보장해주는 일종의 미들웨어 소프트웨어 서버

Server단에서 어플리케이션을 동작할 수 있도록 지원, 일반적으로 컨테이너라는 용어로 사용됨, Servlet, JSP, ASP, PHP, 동적 데이터

2.1 기능

a. Servlet 페이지를 HTML형태로 변환함, 예를들어 jsp의 경우 jsp를 WAS에서 java class파일로 컴파일 후 html형태의 페이지를 사용자에게 전달하게 됨, 동적 데이터의 처리

b. Container라는 용어로 쓰이며, 초창기는 CGI, 그 후 Servlet, JSP, ASP 등의 프로그램으로 사용

2.2 종류

a. Tomcat, tMax jeus, IBM Webapere, JBOSS, Oracle…

분산처리

* 정의: 시스템 레벨의 병렬처리, 물리적 혹은 논리적 처리의 분리에 근거하여 전체의 처리가 수행
* 종류
  + 부하분산  
    작업이 많은 컴퓨터의 작업이 적은 컴퓨터로 이동
  + 처리분산  
    데이터의 발생 장소나 서비스를 요구하는 장소 등에서 처리
  + 기능분산  
    개별적인 기능을 갖는 전용 컴퓨터를 준비하여 시스템 전체에서 공유
  + 위험분산  
    일부가 고장 나더라도 다른 컴퓨터에 의해 처리 가능
  + 관리분산  
    시스템의 운용과 관리를 각 조직에서 개별적으로 수행
  + 확장분산  
    기능의 추가와 진보된 기술의 수용의 확장이 용이하게 이루어짐

분산시스템

* 정의: 네트워크를 통해 분산된 시스템 자원을 공유하여 어플리케이션을 처리하는 시스템
* 목적
  + 다양한 컴퓨터(or 장비)들 사이에 다양한 통신 방법들을 어떻게 사용자로부터 숨길 것인가? (투명성 보장)
  + 어떻게 사용자와 시스템 간의 일관적이고 단일화된 커뮤니케이션 통로만들 것인가?
  + 자원의 부족(CPU의 파워, 메모리, 하드 디스크 등의 모든 자원), 장애시 대처, 사용자에게 자원 접근의 편리성을 제공

투명성(Transparency): 사용자는 어디에 어떤 자원이 어떻게 저장되어 있는지 알 필요도 없고 어떻게 시스템이 구성되는지 알 필요가 없음

* Access: 데이터 표현 방식, 자원에 접근 방법을 사용자에게 숨긴다.
* Location: 자원이 어디에 위치하고 있는지 접근 방법을 사용자에게 숨긴다.
* Migration: 자원의 이동을 사용자에게 숨긴다.
* Relocation: 사용자가 사용중임에도 자원이 이동되어도 사용자에게 숨김
* Replication: 자원은 복사될 수 있다. 하지만 사용자는 원본인지 복사본인지 모름
* Concurrency: 접근 자원이 다른 사용자와 함께 공유되고 있어도 사용자에게 숨김.
* Persistance: 자원이 메모리에 있든, 하드디스크에 있든 사용자는 모름

C 언어

* 시스템 프로그래밍 언어  
  ※ 시스템 프로그램: 운영체제, 언어처리계, 편집기, 디버깅 등 소프트웨어 작성을 지원하는 프로그램
* 장점: 뛰어난 이식성과 작은 언어사양, 비트 조작, 포인터 사용, 자유로운 형 변환, 분할 컴파일
  + 뛰어난 이식성  
     8비트 CP/M등의 컴퓨터부터 슈퍼 컴퓨터에 이르기까지 모두 사용할 수 있는 강력한 이식성을 갖고 있음
  + 범용 프로그래밍 언어  
     운영체제와 같이 하드웨어와 밀접한 프로그램뿐 아니라 운영체제 위에서 작동하는 워드프로세서, 게임, 개발도구와 같은 응용 프로그램을 작성할 수 있다.
  + 간략한 문법 표현  
     함수를 이용해 간결하게 프로그램을 작성 가능, 전처리기를 이용해 파일포함, 매크로 기능, 조건번역 등의 기능을 간단히 수행 가능
* C언어의 사용분야
  + 운영체제
  + 컴파일러  
    C언어를 개발할 수 있게 해주는 컴파일러들은 모두 X 언어로 작성되어 있음 또한 C언어를 이용해 비주얼 베이직이나 파워빌터 같은 새로운 프로그램 개발 언어를 만들어 사용하고 있다.
  + 게임  
     DirectX, openGL(C언어로 작성된 API)을 이용
  + 유틸리티와 상용 소프트웨어
  + 산업용 소프트웨어
* 가격
  + GCC컴파일러와 Turbo C는 무료로 제공되는 컴파일러

C#

* 현대적이고 혁신적인 언어로 .NET 플랫폼 기반의 소프트웨어를 개발하는 최고의 도구 뛰어난 생산성과 성능을 갖추고 있다.
* 장점
  + 분산 환경에 적합한 언어
    - ASP.NET XML.NET 모바일 프로그램 지원 등의 프로그램을 빠른 시간 안에 개발 할 수 있다.
  + 구조 중립적인 언어
    - C#을 이용해 다양한 환경의 운영체제에서 실행할 수 있다.
  + 뛰어난 생산성
    - C+은 비주얼 스튜디오.NET 을 이용해 뛰어난 생산성을 제공한다. 비주얼 베이직의 생산성과 비주얼 C++의 성능을 동시에 제공하는 C#은 개발시간을 크게 단축 시킨다.
  + 강력한 디버깅 기능
    - ASP.NET 모바일 프로그래밍 등 기존 언어에서 디버깅 제공이 곤란했던 분야에 대해 뛰어난 디버깅 기능을 제공한다.
* 사용분야
  + 닷넷 프레임워크가 활성화 되지 않아 많이 사용되지 않는 학습용 언어라 할 수 있다.
  + C#이 작동하기 위해서는 CLR엔진과 닷넷 프레임 워크가 필요한데 현재 닷넷을 지원하는 운영체제가 윈도우 2003밖에 없기 때문에 C#을 이용해 개발된 다양한 프로그램을 접하기 위해서는 시간이 더 필요하다.
  + C#은 자바가 사용되는 부분과 비슷한 영역에서 사용될 수 있다. 먼저 윈도우용 애플리케이션을 작성할 수 있으며, DB, 웹 서버, XML 등을 작성할 수 있다.

C++

* C의 확장으로 개발된 중간 수준의 언어로 객체 지향 프로그래밍을 지원한다.
* C++이름의 이유는 C언어에 있는 ++연산자로부터 생겨난 것으로 C언어의 확장판이라는 의미를 갖고 있음
* 장점
  + 문자 상수와 전역 변수  
    C언어에서 문자 상수가 자동적으로 정수가 바뀌지만 C++에서는 그렇지 않음, 전역변수를 여러 번 선언할 경우 C에서는 에러가 발생하지 않지만 C++에서는 에러가 발생
  + 광범위한 시스템과 애플리케이션에 사용
* 단점
  + 지나치게 많은 기능
  + 수동 메모리 관리
  + 느린 컴파일 속도

JAVA

* 장점
  + 간단한 언어
    - 자바는 C++과 매우 유사한데 프로그램 작성에 꼭 필요하지 않은 고급 프로그램기능을 제거 하였다.
  + 객체지향 언어
    - 기본 자료형을 제외하고 자바에서 다루는 모든 데이터는 객체를 통하여 처리된다.
    - 자바의 모든 포트는 객체의 클래스 안에서 조작된다.
    - 각 클래스는 부모 클래스의 상속을 받은 변수와 메소드의 집합으로 구성되어 있다.
    - Object라는 최상위 클래스로부터 다양한 클래스가 상속되어 구현되어 있다.
    - 단일 상속만 지원하여 다중상속의 문제들을 해결 하였다.
  + 구조 중립적인 언어
    - 자바 코드는 바이트코드로 컴파일되어 각 플래폼 별 자바 가상머신에 의해 어떤 플랫폼에서도 사용 가능하다.
    - 휴대폰, PDA, 셋톱박스에서도 자바의 중립성은 보장됨
  + 쓰레기 수집 가능
    - 쓰레기 수집(Garbage Collection)기능으로 메모리 누수를 걱정하지 않고 메모리를 사용할 수 있다
* 단점
  + 구문이 장황하여 속도가 느림
  + 함수를 전달하기 쉽지 않음
  + API가 필요이상으로 복잡함.
* 사용분야
  + 일반 애플리케이션 개발 분야
    - J2SE라는 개발 도구를 사용하여 DB, 윈도우 프로그램, 사용자 인터페이스, 네트워크, 이미지처리 등의 다양한 프로그램을 작성할 수 있다.
  + 웹 애플리케이션 개발 분야
    - 웹 애플리케이션은 크게 자바 애플릿과 자바 서블릿으로 나뉜다. 자바 서블릿은 다시 JSP와 EJB가 추가되어 분산 웹 프로그램을 작성할 수 있도록 구성되어 있다.
  + 휴대용 정보기기 애플리케이션 개발 분야

ASP

* PHP나 JSP에 비해 배우기 쉽다. 또한 인터데브, 프론트페이지, 비주얼 스튜디오 닷넷과 같은 다양한 개발 도구를 이용해 빠르고 쉽게 웹 프로그램을 개발할 수 있다.
* 장점
  + 가장 배우기 쉬운 스크립트 언어
    - ASP는 웹 프로그램 언어 중 가장 배우기 쉬운 언어이다.
  + IIS 웹 서버의 모든 기능 사용 가능
  + 뛰어난 생산성 제공
* 사용분야
  + 윈도우 NT기반의 운영체제에서만 서비스가 가능하다. IIS 웹서버에서만 실행 할 수 있으며 ADO와 ADO.NET과 같은 마소 사의 데이터베이스 API를 사용해야 한다.
* 단점
  + 해커의 공격이 잦아 보안 패치가 상당히 자주 만들어지고 있고 주기적인 보안 업데이트가 필요
  + 윈도우 운영체제에 국한되어 있다.
  + ADO, ODBC등의 윈도우 전용 데이터베이스 기술만을 사용한다.

ASP.NET

* 기존의 ASP 기술을 한층 발전시켰으며 JSP와 비슷한 장점을 갖고 있음
* ASP.NET은 닷넷 프레임워크에 기반을 두고 작성되어 있으며 ASP.NET은 Visual Basic과 C#의 언어를 이용해 개발할 수 있다. 윈도우 서버2003부터는 IIS 6.0과 닷넷 프레임워크가 운영체제 수준에서 지원되고 있다.
* SOAP프로토콜을 사용함
* 장점
  + 완벽한 디버깅 기능
  + 다양한 언어 사용 가능
  + C#/ Visual Basic등의 다양한 닷넷 언어를 사용할 수 있으며 XML, SOAP, ADO.NET 모바일 등의 다양한 최신기술을 사용할 수 있다.

PHP

* GPL 라이선스에 의해 소스 코드가 공개된 소프트웨어이다. 또한 전 세계 60%이상의 아파치 웹 서버와 가장 잘 어울리는 웹 프로그래밍 언어로 인식되고 있다.
* 동적으로 HTML페이지를 생성하기 위해 서버 측 스크립팅에 사용되는 언어
  + PHP는 빠른 웹 서버 스크립트
    - PHP는 CGI와 달리 하나의 프로세스만으로 클라이언트의 요청을 처리할 수 있다.
  + 플랫폼 독립적인 언어
    - PHP는 주로 리눅스에서 아파치 웹 서버와 짝을 이뤄 사용한다. 그러나 리눅스 뿐만 아니라 유닉스, 윈도우 등 다양한 운영체제에서도 실행할 수 있다. 더해서 주요 웹 서버인 아파치, IIS, 넷스케이프 엔터프라이지 서버 등에서 잘 작동한다. PHP로 작성된 소스코드는 거의 수정 없이 다른 플랫폼으로 이식해 사용할 수 있다.
  + 다양한 데이터 베이스 지원
    - PHP는 MYsql, Msql과 같은 중.소규모 데이터베이스부터 오라클, DB2등의 대용량 데이터베이스까지 다양한 DB를 제공한다.
  + 공개 소프트웨어
    - 공짜로 사용할 수 있는 소프트웨어이다.
  + 배우기 쉬운언어
    - C, JAVA, C#과 비슷한 문법을 사용하면서 복잡하고 어려운 부분은 생략함
* 단점
  + 많은 함수에 대한 일관성 없는 명명 규칙
  + 유니코드에 대한 네이티브 지원 결여
  + 프레젠테이션 코드(HTML, CSS)와 섞이는 경우가 많음
  + 보안 결함
  + 리눅스 이외의 운영체제에서 몇가지 기능의 호환성이 떨어짐
  + 언어 자체적으로 DB처리를 지원하기 때문에 다양한 데이터 베이스기술과 최신버전에 따른 지원이 떨어진다.
* 사용분야
  + 리눅스에 최적화된 웹 프로그래밍 언어

※CGI(Common Gateway Interface): 웹서버와 외부 프로그램 사이에서 정보를 주고받는 방법이나 규약들을 말한다. / 사용자의 요청에 의한 서버의 응답  
CGI는 응용 프로그램이기 때문에 html로 만들 수 없고 웹 서버의 환경에 따라 다르게 구현되는데 주로 C, Perl, Visual Basic등이 사용되고 있다.

JSP

* JVM(JAVA Virtual Machine)위에서 작동한다.
* 웹서버에 올려져 있는 페이지 중 확장자가 JSP로 끝나는 파일은 JSP파서가 해석해 서블릿 코드로 번역하고 이렇게 작성된 자바 코드는 자바 가상 머신을 이용해 바이트 코드로 변경된다. 이러한 과정을 거친 후에 클라이언트가 웹 서버에 동일한 페이지를 요청해 오면 컴파일된 바이트 코드를 직접 서블릿 엔진이 실행해 결과를 처리한다.
* ASP나 PHP가 클라이언트의 요청이 들어올 때마다 그때그때 인터프리터 방식으로 코드를 해석해 실행하는데 사용한다. 그러나 JSP는 처음에 서블릿 코드로 바꿔 컴파일하는데 시간이 약간 걸리지만 그 뒤부터는 컴파일 된 실행파일이 직접 호출되므로 속도가 상당히 빠르다. 특히 자바의 강력한 스레드 기능과 예외처리 기능을 그대로 사용할 수 있기 때문에 사용자가 많은 대용량 시스템에서 안정적인 서비스를 제공할 수 있다.
* 장점
  + 플랫폼 독립적인 특징
    - JSP는 자바 언어를 기반으로 작성되었기 때문에 플랫폼 독립적인 코딩이 가능하다.
  + 소프트웨어 재사용성 증가
    - JSP는 완벽한 객체 지향 언어인 자바 언어를 기본으로 작성되었기 때문에 소프트웨어 재사용이 용이하다. 특히 자바빈즈, 자바 서블릿 등의 다양한 기술을 사용해 코드 재사용성을 크게 증가시킬 수 있다.
  + 강력한 데이터베이스 처리
    - JDBC는 데이터베이스의 종류에 상관없이 강력한 데이터베이스 프로그램을 작성할 수 있는 API이다.
  + 서블릿과 자바빈즈를 이용한 고성능 웹 프로그래밍
    - 서블릿은 자바의 스레드와 예외처리를 사용해 빠르고 안전한 서비스를 제공한다.
    - 자바빈즈라는 컴포넌트 기술을 이용해 코드의 재사용성을 크게 높일 수 있다.
  + 메모리의 관리
    - JSP는 자바의 GC(Garbage Collection) 기능을 사용할 수 있다. 자바 가상머신이 자동으로 메모리를 관리해주기 때문에 효율적인 프로그램을 작성할 수 있다.
* 사용분야
  + 운영체제와 상관없이 웹 서버를 구축할 수 있다.
  + 아파치 + 톰캣 + MYSQL + 리눅스를 사용해 구축 비용이 전혀 들지 않는 중.소규모
  + 유닉스 + 웹 스피어 + 오라클을 사용해 안전성과 확장성이 뛰어난 대규모도 가능

HTML

* 웹 문서를 만들기 위하여 사용하는 기본적인 프로그래밍 언어의 한 종류

XML (eXtendsible Markup Language)

* 웹 상에서 구조화된 문서를 전송 가능하도록 설계된 마크업 언어
* 플랫폼과 프로그래밍 언어로부터 독립적이므로 어떤 플랫폼에서 XML을 사용해도 제약을 받지 않는다.
* 웹 기반의 어플리케이션들은 XML을 사용함으로써 가능한 모든 정보를 클라이언트에 저장하여 웹 서버의 부하를 줄일 수 있다.

Perl

+ 고수준의 다목적 인터프리트 언어로 “스위스 군용 칼”, “인터넷의 강력 접착 테이프”  
+ CGI 스크립팅부터 시스템 및 네트워크 관리에 이르기까지 모든 분야에 사용  
- 작업을 수행하는 방법이 너무 많음, 즉 읽기가 거의 불가능

Visual Basic  
+ 그래픽 사용자 인터페이스를 사용해 윈도우 애플리케이션을 개발하는데 사용되는 프로그래밍 언어이자 통합 개발 환경  
+ 마소가 초보자도 쉽게 사용할 수 있는 언어를 목적으로 개발  
+ 그래픽과 드래그 앤 드롭, 코드 작성을 조합하여 애플리케이션을 구축  
- 구문이 너무 장황하고 이상함  
- 실행하려면 동적 링크 라이브러리가 필요  
- 객체 지향 프로그래밍 지원이 부족  
- 1998년 이후 마소의 공식 지원이 없음

Python

* 고수준의 다목적 프로그래밍 언어로 가독성을 장점으로 내세움
* 플랫폼 독립적이고 컴파일도 가능하지만 스크립트 언어로 사용되는 경우가 많음  
  ※ 스크립트 언어: 소스 코드를 컴파일 하지 않고도 실행 할 수 있는 프로그래밍 언어, 컴파일 과정 없이 실행하여 결과를 확인할 수 있지만 컴파일 과정을 거쳐야 하는 등 개발 과정이 조금 복잡하다.  
  + 사용처: 시스템 / 웹 어플리케이션에 사용  
   - Server Side Script: 웹서버를 거쳐야 함 (ASP, PHP, JSP, PYTHON, PERL, RUBY…)  
   - Client Side Script: 내장된 번역기에 의해 실행되는 스크립트(JavaScript, VBScript, Jscript...)  
  + 예를 들어 C나 Visual Basic, Java의 경우 작성한 코드를 컴파일러를 통해 기계어 코드로 변경해야 실행 파일이 얻어 짐  
  - 블록 구조를 지정하는데 대괄호나 중괄호 대신 들여쓰기를 사용  
  - 콜론과 밑줄의 과도한 사용  
  - 모듈/변수 이름 충돌

자바스크립트

* 정적인 HTML에 동적인 기능을 추가하기 위해 넷스케이프에서 썬마이크로시스템즈와 공동으로 프로젝트를 수행해 개발한 클라이언트용 스크립트 언어
* 장점
  + 작업이 빠르고 생산성이 좋음
  + 배우기 쉽고 운영체제에 대한 제한이 없음
  + 웹 브라우저에서 자바 스크립터를 지원하고 있기 때문에 클라이언트 측 스크립트 언어의 표준으로 사용
  + 서버 측 웹 스크립팅과 임베디드 스크립팅 언어로도 구현
* 단점
  + HTML 소스 코드안에 자바스크립터가 포함되어 있기 때문에 소스코드를 사용자가 쉽게 복사하여 재사용할 수 있다. 즉 보안성이 없다.
  + 웹 브라우저 안에서 한정된 객체와 메소드만을 갖고 있기 때문에 채팅 사이트나 게임 등의 고급프로그램이 불가능
  + 대/소문자의 민감성
  + 브라우저마다 다른 구현
  + 디버깅 기능의 부재

OpenGL

* 그래픽 하드웨어에 대한 인터페이스로 강력한 저수준 렌더링과 모델링 소프트웨어 라이브러리
* 대부분의 그래픽 카드에서 OpenGL이 지원
* 장점
  + 다양한 운영체제에서 작동(윈도우 제어와 관련된 기능 X)
    - OpenGL은 화면에 그래픽을 출력하는 기능만 제공한다. 따라서 프로그램 개발 시 OpenGL만 갖고는 제대로 된 응용 프로그램 작성이 어렵다.
* 단점
  + 절차지향적 구조
    - 객체 지향 구조인 클래스나 객체의 개념을 가지고 있지 않다. OpenGL이 제공하는 함수들은 서로 상관관계가 없는 독립관 구조로 되어있다.
* 사용분야
  + 게임, CAD, 모델링 등의 다양한 응용 프로그램(마다, 3D AMX Studio)제작에 사용

크롤링 시 robots.txt를 확인 후

구글의 웹페이지 수집 방식

* 분산 시스템과 머신 러닝 기술
* 잘 알려진 웹사이트부터 방문해 웹페이지 수집(크롤링)

데이터 마이닝(Data Mining)

* 대규모로 저장된 데이터 안에서 체계적이고 자동적으로 통계적 규칙이나 패턴을 찾아내는 것
* 기법: OLAP, 군집 분석, 연결 분석, 사례기반 추론, 연관성 규칙 발견, 인공 신경망, 의사 결정 나무, 유전자 알고리즘
* 절차: 추출 🡪 정제 🡪 변경 🡪 분석 🡪 해석 🡪 보고서
* **목표지향 / 목표불명**
* 목표지향 데이터 마이닝  
   - 몇 개의 특정 대상(target field)에 대해서 설명하거나 구분을 하는 것  
   - Classification(분류), Estimation(추정), Prediction(예측)
* 목표불명 데이터 마이닝  
   - 데이터에서 단순히 있을 법한 패턴이나 유사성을 찾는 것  
   - Association Rules(연관 규칙), Clustering(군집화), Profiling(서술/설명)
* Classification(분류)  
  성별, 인종 등 우리가 생활하면서 쉽게 구분하는 방법 / 범주형 자료, 이산형 자료에 사용됨,
* Estimation(추정)  
  연속형, 수치형으로 결과를 규정한다는 점에서 분류와 구분 됨  
  추정된 값에 순위와 같은 것을 사용할 수 있다는 이점을 가짐
* Prediction(예측)  
  미래의 행동이나 미래 추정치의 예측에 따라 구분  
  입력 변수들에 대한 관계나 타켓변수에 대한 예측을 통해 예측 모델링을 함
* Affinity Grouping(유사행태 집단화)