

4주차

안드로이드 프로그래밍 환경

Mobile Service

```
</html>
<head>
<title> java programming</title>

<style type="text/css">
<!--
.wdsc {border: black 1px solid; width: 270; text-align: center}
-->
</style>

</head>
<body>

<script language="JavaScript">
<!--
speed = 100;
cycledelay = 2000;
maxsize = 25;
fontcolor = "0000ff";
x = 0;

function initArray() {
this.length = initArray.arguments.length;
for (var i = 0; i <this.length; i++) {
this[i] = initArray.arguments[i];
}
}

themessage2 = new initArray (
"java programing",
"JavaScript",
"java programing"
);

function upwords() {
themessage = themessage2[x];
if (x < maxsize) {
x++;
setTimeout("upwords()",speed);
}
else setTimeout("downwords()",cycledelay);

size = "<font color=" + fcolor + " point-size=" + x + "pt">";
document.wds.document.write(size + "<center>" + themessage + "</center></font>");
document.wds.document.close();
}
-->
</script>
</body>
</html>
```

학습 내용

1. SDK, JDK, 이클립스, AVD의 개념
2. 프로그래밍 환경 수립 및 설정
3. 예제 프로그램 수행 방법
4. 액티비티 생명주기

JDK, SDK, 이클립스, AVD의 개념



JDK 개념

- Java SE Development Kit의 약자로 자바 어플리케이션 개발을 위한 컴파일러와 가상머신, 기타 툴킷을 포함



JDK 개념

- 안드로이드의 기본 개발 언어가 자바이며 이클립스 또한 자바 실행환경을 지원하므로 먼저 자바 JDK가 설치되어 있어야 함



JDK 개념

- JDK에 JRE가 포함되어 있으므로
JRE만 따로 설치할 필요는 없음



JDK 개념

- 실습 장비에 이미 JDK가 설치되어 있다면 그대로 사용할 수도 있지만 너무 오래된 버전은 문제가 발생할 수 있으므로 가급적이면 최신 버전을 다시 설치하는 것이 좋음

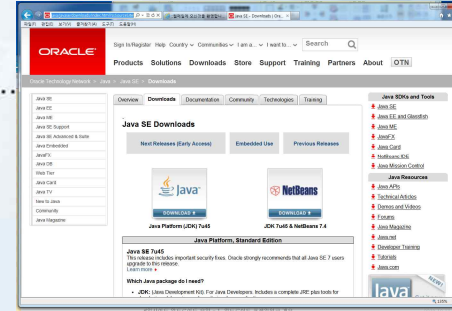


JDK 개념

– JDK는

<http://www.oracle.com/technetwork/java/javase/downloads/index.html?ssSourceSiteId=ocomen> 사이트

에서  을 선택하여 설치함





JDK 개념

- JDK는 사용하는 목적에 따라 SE(standard), EE(enterprise), ME(mobile) 세 가지로 제공되는데 안드로이드 개발에는 SE나 EE를 사용

함



JDK 개념

- 개발자의 운영체제가 시스템이 32비트면 반드시 32비트를 설치해야 하지만, 64비트 시스템이면 32비트나 64비트 아무거나 선택할 수 있음



이클립스 개념

- 다양한 플랫폼에서 사용할 수 있는 범용 응용 소프트웨어로 안드로이드 개발환경에 필수적이며 대부분의 개발에 이용되고 있음



이클립스 개념

- 명령행으로도 모든 개발이 가능하지만 여러 모로 불편하며 특히 처음 배우는 사람에게는 난해하므로 전용 개발툴을 이용하는 것이 유리함



이클립스 개념

- 안드로이드 응용 프로그램 개발에는 주로 이클립스라는 통합 개발 환경을 사용함
- 이클립스는 1999년부터 IBM에 의해 개발이 시작되었음



이클립스 개념

- 2004년부터 볼랜드, 레드햇, HP, 오라클 등 회사들이 컨소시엄을 구성하여 비영리 재단화되었으며 현재 오픈소스로 전환되어 누구나 무료로 사용할 수 있음



이클립스 개념

- 다양한 언어와 플랫폼을 지원하는 다
목적 개발 환경이지만 현재는 주로 자
바 개발용으로 활용됨

- 공식 사이트는 다음과 같음

<http://www.eclipse.org>



이클립스 개념

- 홈페이지 상단 메뉴의 다운로드 이클립스를 클릭하면 여러 가지 버전을 다운로드 받을 수 있는 링크가 나타남



이클립스 개념

- 언어나 플랫폼에 따라 여러 가지 버전이 제공되는데 안드로이드 개발용으로는 클래식이나 Java 개발자용 정도가 적합함



이클립스 개념

- EE 버전은 기능이 많지만 안드로이드 개발에 큰 도움이 되지 않고 용량을 많이 차지하므로 굳이 EE 버전을 설치할 필요는 없음



이클립스 개념

- 이클립스는 무설치 프로그램이므로
압축만 풀면 바로 사용할 수 있음
- 시스템 레지스트리를 사용하지 않으
므로 폴더만 지우면 깔끔하게 삭제됨



이클립스 개념

- 압축 파일 안에 이클립스 폴더가 있는데 이 폴더를 C:\WProgram Files에 압축을 풀어 둬
- 이클립스 폴더 안의 eclipse.exe가 주 실행 파일임



ADT 플러그인

- 이클립스에서 안드로이드 어플리케이션을 개발할 수 있도록 플러그인 형태로 개발툴을 제공함
- 이를 통해 안드로이드 프로젝트를 생성하거나 디버깅할 수 있음



ADT 플러그인

- 구글은 이클립스에 플러그인되는 안드로이드 개발 도구를 Android Development Toolkit 이라고 함
- ADT는 이클립스와 안드로이드 SDK를 연결하는 역할을 함



ADT 플러그인

- ADT는 안드로이드 개발에 필요한 다음과 같은 다양한 기능을 제공하며 이런 편리한 도구를 설치할 수 있기 때문에 이클립스가 안드로이드의 공식 개발도구가 된 것임



ADT 플러그인

- 안드로이드 프로젝트 생성 마법사
- XML 레이아웃 편집기
- 패키지 제작도구
- 이클립스에 통합된 DDMS



ADT 플러그인

- ADT는 이클립스의 업데이트 기능으로 다운로드 및 설치함



SDK 개념

- Software Development Kit(소프트웨어 개발 도구)으로 어플리케이션을 개발함
- SDK는 안드로이드 개발 라이브러리, 도움말 문서, 개발 보조 도구 등을 포함하므로 용량이 큼



SDK 개념

- 개발 목적에 따라 필요한 SDK 목록이 다르고 업데이트 빈도가 잦기 때문에 꼭 필요한 부분만 선택적으로 설치 및 업데이트가 가능함



SDK 개념

- 개발에 필요한 파일은 SDK를 통해 배포되며 ADT는 이클립스와 SDK를 연결하는 역할을 함



SDK 개념

- 이전에는 SDK를 따로 받아서 설치해야 했으나 지금은 ADT가 SDK도 같이 설치하도록 바뀌었음
- ADT 설치 후 이클립스를 재시작하면 SDK 설치 대화상자가 나타남



SDK 개념

- 이 대화상자를 통해 최신 버전의 SDK를 바로 설치할 수 있으며 관련된 환경 설정까지 자동으로 처리됨



SDK 개념

- 이 대화상자를 무심코 닫아 버린 경우 수동으로 설치해야 함
- 다음 사이트에서 다운로드 받아 설치함 : <http://developer.android.com>



AVD 개념

- 안드로이드폰 없이 어플리케이션을 개발하는 개발자를 위해 Android Virtual Device(AVD)를 생성함으로써 프로그램 실행을 할 수 있는 도구임



AVD 개념

- 모바일 개발을 할 때는 일반적으로
에뮬레이터에서 테스트를 수행하는데
안드로이드폰 에뮬레이터를 AVD라고
함



AVD 개념

- 소프트웨어적으로 만든 가상의 장비
이지만 진짜 안드로이드폰과 기능이
비슷함



AVD 개념

- 다양한 설정으로 여러 개의 AVD를 생성해 두고 번갈아 사용할 수 있어 실제로 핸드폰 없이도 개발 가능함
- 여러 버전의 장비를 가상으로 테스트 할 수 있어 편리함



AVD 개념

- 이클립스의 메뉴에서 Window/AVD Manager 항목을 선택하여 생성할 수 있음



AVD 개념

- AVD 생성 대화상자는 버전마다 다를 수 있으며 SDK보다 업데이트 주기가 더 빠르며, ADT가 제공하는 것이기 때문에 ADT 버전에 따라 모양이 달라짐



AVD 개념

- AVD가 저장되는 경로에 영문이 아닌 문자가 있으면 이클립스가 에뮬레이터를 제대로 기동시키지 못함
- 한글 윈도우에서 계정명을 한글로 쓰는 경우가 많으므로 주의가 필요함

JDK, SDK, 이클립스, AVD의 설치



개발툴 설치 순서

JDK



안드로이드 SDK + 이클립스 + ADT



AVD 생성 및 옵션 설정



JDK 설치

- 웹사이트 <http://www.oracle.com>에 접속함
- DOWNLOADS 링크에서 Java For Developers 항목을 클릭함



JDK 설치

- Java SE를 선택하고 ‘download JDK’ 를 선택해서 다운로드함
- 자신의 운영체제에 버전에 맞는 JDK를 선택하고, 약관 동의 버튼을 체크함



JDK 설치

- 시스템이 32비트인 경우에만 반드시 32비트를 설치해야 함



JDK 설치

- 복수 언어를 지원하므로 한글팩을 별도로 설치할 필요는 없음
- 다운로드 받은 설치 파일을 실행하면 라이선스 동의창이 나타나고 설치할 기능을 묻음



JDK 설치

- 디폴트로 전부 선택되어 있으며, 설치 경로 역시 디폴트로 주어지는 Program Files에서 Next 버튼을 누르고 설치를 시작하면 됨
- 설치 완료 대화상자가 나타나면 Finish 버튼을 눌러 설치를 완료함.



JDK 설치

- 환경 변수를 설정하지 않은 상태에서 Java 프로그램을 컴파일하기 위해서는, Java 프로그램이 JDK 폴더안에는, 'bin' 폴더 안에 있어야 함



JDK 설치

- 환경 변수의 설정을 통해 도스상의
폴더 어느 곳에서도 Java 프로그램의
컴파일이 가능하게 할 수 있음



이클립스 설치

- <http://www.eclipse.org> 사이트
- 홈페이지 상단 메뉴의 Download Eclipse를 클릭하면 이클립스의 여러 가지 버전을 다운로드 받을 수 있는 링크가 나타남



이클립스 설치

- 제일 상단의 Eclipse IDE for Java Developer 오른쪽의 링크를 눌러 해당하는 윈도우 버전을 다운로드 받음



이클립스 설치

- 이클립스는 무설치 프로그램이므로
압축만 풀면 바로 사용할 수 있음
- 시스템 레지스트리를 사용하지 않으
므로 폴더만 지우면 깔끔하게 삭제됨



이클립스 설치

- 압축 파일 안에 eclipse 폴더가 있는데 이 폴더를 C:\WProgram Files에 압축을 풀어 둬



이클립스 설치

- 시스템 상황에 따라 다른 폴더에 압축을 풀어도 상관없으며 여러 버전을 같이 설치할 경우 폴더 이름을 변경해도 됨



이클립스 설치

- eclipse 폴더 안의 eclipse.exe가 실행 파일임
- JRE만 제대로 설치되어 있다면 별 문제없이 실행될 것임



이클립스 설치

- 최초 실행 시킬 경우 워크스페이스를 물어보는데 워크스페이스란 프로젝트를 저장하는 작업 폴더임
- 대화상자가 폴더를 생성해 주므로 폴더를 미리 만들어 놓을 필요는 없음



이클립스 설치

- 소스 편집, 컴파일, 실행 결과까지

GUI 환경에서 확인할 수 있기 때문에

명령어 실행보다는 편리함



이클립스 설치

- 디버거, 로그 뷰, 프로파일러, 개발에 필요한 도움말까지 풀옵션으로 제공하여 상용 개발툴 부럽지 않은 편의성을 제공함



플러그인 설치

- 자바 프로그래밍은 할 수 있지만 아직 안드로이드 프로그램은 만들 수 없

음



플러그인 설치

- 이클립스는 안드로이드 전용 개발툴
이 아니라 다양한 언어와 플랫폼을 지
원하는 범용 개발툴이기 때문임



플러그인 설치

- 플러그인이라는 장치를 통해 임의의 플랫폼에 대한 개발툴의 추가 설치를 지원할 수 있음

- 이클립스에서 Help/Install New Software 메뉴를 선택하면 업데이트 관리 대화상자가 나타남



플러그인 설치

- 나타난 대화상자에는 이미 설치된 소프트웨어와 추가로 설치할 수 있는 소프트웨어가 표시됨



플러그인 설치

- Work with 콤보 박스에서 새 소프트웨어를 받을 사이트를 선택하면 설치 가능한 소프트웨어 목록이 나타남



플러그인 설치

- 안드로이드 ADT는 설치 가능한 목록에 보이지 않는데 이클립스는 이 툴이 어디에 있는지 주소를 아직 모르기 때문임



플러그인 설치

- 이클립스에게 안드로이드 ADT가 배포되는 주소를 알려 주어야 함



플러그인 설치

- Work with 옆의 Add 버튼을 누르고
다음과 같이 입력하여 ADT가 있는
위치를 가르쳐 주어야 함
- 이름(Name)은 사용자가 임의적으로
정하면 됨



플러그인 설치

– ADT 주소는 [https://dl-ssl.google.com/android/eclipse/](https://developer.android.com/studio)

사이트 주소를 입력함

– 접속 프로토콜 보안이 적용된 https 주소임



플러그인 설치

- 네트워크 상황에 따라 방화벽이나 보안 문제로 인해 접속이 안 되는 경우가 종종 있는데 이 경우 http로 바꿔 주면 됨



플러그인 설치

- OK 버튼을 눌러 Add Site 대화상자를 닫고 상단의 Work with 콤보 박스에서 방금 입력한 사이트를 선택함
- 접속되면 사이트에서 제공하는 플러그인의 목록이 표시됨



플러그인 설치

- ADT와 DDMS, 계층 뷰어 등의 플러그인이 포함되어 있으며 NDK 툴도 제공됨

- 아래쪽의 Select All 버튼을 눌러 전체 선택하고 Next 버튼을 누름



플러그인 설치

- 구체적인 설치과정은 플러그인마다 조금씩 다른데 ADT는 라이선스 동의 대화상자 외에는 특별한 질문이 없으므로 I accept 체크 박스를 클릭하여 동의만 하면 됨



플러그인 설치

- 재시작하라는 메시지 박스가 나오면
가급적이면 재시작하는 것이 좋음



플러그인 설치

- 재시작 직후 SDK 설치 안내 대화상자가 나타남
- 이 대화상자는 다음 단계와 연결되므로 Yes 버튼을 누르지 말고 일단 대기하도록 함



플러그인 설치

– 설치된 플러그인은 이클립스 설치

디렉터리의 plugins 디렉터리에 압축

파일 형태로 보관되므로 이클립스를

통째로 다른 폴더로 옮겨도 무관함



플러그인 설치

- 안드로이드 홈페이지에 별도로 제공되는 ADT 배포 압축 파일을 다운로드 받은 후 Add Site 대화상자에서 Archive 버튼을 눌러 설치하는 방법도 있음



SDK 설치

- 이전에는 SDK를 따로 받아서 설치해야 했으나 지금은 ADT가 SDK도 같이 설치하도록 바뀌었음
- ADT 설치 후 이클립스를 재시작하면 SDK 설치 대화상자가 나타남



SDK 설치

- ADT는 설치되었으나 아직 SDK가 없으므로 SDK를 설치하라는 대화상자임



SDK 설치

- 이 대화상자를 무시할 경우 다소 번거로운 작업을 해야 하므로 일단 대화상자에서 최신 버전의 SDK를 설치하는 것이 좋음



SDK 설치

- 대화상자를 통해 최신 버전의 SDK
를 바로 설치할 수 있으며 관련된 환
경 설정까지 자동으로 처리됨
- 이전 버전의 SDK도 같이 설치하는
옵션을 제공하고, 설치 위치도 선택할
수 있음



SDK 설치

- 지금 이전 버전을 사용할 것이 아니라면, 추후에도 설치 가능하므로 지금 설치할 필요는 없음
- 디폴트 옵션을 확인하고 Next 버튼을 누르면 설치할 목록과 라이선스 동의문이 나타남



SDK 설치

- Accept All 버튼을 누르고 Install 버튼을 누르면 최신 SDK를 다운로드 받아 설치함
- 여기까지 설치된 경우 안드로이드 개발을 할 수 있음



SDK 설치

- 개발 관련 문서나 샘플, 이전 버전 등의 추가 설치는 SDK 관리자로 수행할 수 있음



SDK 설치

– 이클립스 메뉴에서

Window/Android SDK Manager 항목

을 선택하면 SDK 관리자 대화상자

가 나타남



SDK 설치

- Accept All 버튼을 눌러 라이선스에 동의하고 Install 버튼을 누르면 설치가 시작됨



SDK 설치

- 용량이 워낙 크기 때문에 굉장히 오랜 시간이 소요됨

- 설치한 항목의 상태가 모두

Installed로 바뀌면 설치가 완료된 것

임



SDK 설치

- 대화상자를 닫아버린 경우에는 SDK
를 수동으로 설치해야 함
- <http://developer.android.com> 사
이트에 방문하여 Get the SDK 버튼
을 클릭하여 다운로드 페이지로 이동
함



SDK 설치

– Download the SDK for Windows

버튼을 누르면 SDK 설치툴이 다운로드

드 됨



AVD 생성

- 실제폰으로 프로그램을 테스트할 수도 있지만 자신의 전화기를 실습용으로 사용하기는 여러 모로 불편한 점이 있음



AVD 생성

- 개발툴과 버전이 정확히 같은 폰을 가지고 있지 않은 경우가 있을 수 있음
- 안드로이드폰용 에뮬레이터 AVD를 설치하면 테스트를 수행하는데 편리할 수 있음



AVD 생성

- AVD는 소프트웨어적으로 만든 가상
의 장비이지만 실제 안드로이드폰과
기능이 비슷함



AVD 생성

- 다양한 설정으로 여러 개의 AVD를 생성해 두고 번갈아가며 사용할 수 있어 여러 버전의 장비를 가상으로 테스트 할 수 있어 편리함



AVD 생성

- 이클립스의 메뉴에서 Window/AVD Manager 항목을 선택하면 생성되어 있는 AVD 목록이 나타나는데 처음 설치하는 경우에는 비어있는 것이 당연함



AVD 생성

- 사용하는 버전과 해상도를 설치시 이름에 명시해 놓으면 차후 AVD들을 구분하기 편리함



AVD 생성

- AVD는 C:\₩User₩사용자계정명
₩.android₩avd 폴더 안에 저장되
며 가상 장비의 메모리, 구성 등의 정
보가 기록되어 있음



AVD 생성

- 첫 가동 시에 시스템 초기화를 수행
하므로 시간이 오래 걸리는데 잠시 기다리면 에뮬레이터가 실행됨



AVD 생성

- AVD는 정확하게는 SDK가 제공하는 것이 아니라 ADT가 제공하는 것이기 때문에 ADT 버전에 따라 모양이 달라

짐



AVD 생성

- AVD가 저장되는 경로에 영문이 아닌 문자가 있으면 이클립스가 에뮬레이터를 제대로 가동시키지 못함



AVD 생성

- 이 문제를 해결하려면 AVD를 한글이 포함되지 않는 경로로 옮겨야 하는데 계정명을 바꾸는 것은 너무 번거로움



AVD 생성

- 명령창을 열고 AVD가 저장되어 있는 폴더로 이동한 후 android move 명령으로 AVD를 적당한 곳으로 이동시킴



AVD 생성

- 간단한 방법은 직접 폴더를 이동시키는 것임
- 폴더 이동 후 .ini 파일을 열어 path를 이동한 경로명을 수정함

예제 프로그램 실행

Mobile service



예제 프로그램 실행

- 새 프로젝트를 만들기 위해 메뉴에서 File / New / Project 항목을 선택함
- 새 프로젝트 대화상자가 나타나며 이 대화상자에서 프로젝트를 생성하는 마법사를 선택함



예제 프로그램 실행

– Android / Android Application

Project 항목을 선택함



예제 프로그램 실행

- 같은 종류의 프로젝트를 계속 만들면 이클립스에서 File / New 메뉴 아래에 Android Application Project 항목을 표시해줌



예제 프로그램 실행

- Project Name은 프로젝트 이름으로 프로젝트를 저장하는 폴더명으로 사용되며 패키지 탐색기에 표시되기도 함



예제 프로그램 실행

- Application Name은 프로그램의 이름인데 프로젝트명과 같은 이름이 자동으로 입력됨



예제 프로그램 실행

– Package Name은 프로젝트에서 생성하는 클래스가 저장될 패키지 이름이며 클래스 파일이 저장될 폴더 경로이기도

함

– Build SDK는 어떤 SDK를 대상으로 할 것인지를 결정함



예제 프로그램 실행

- Minimum Required SDK는 최소 요구되는 SDK를 의미하며 이 버전 이하의 장비에서는 실행되지 않음



예제 프로그램 실행

- 마법사에 순서에 맞춰 만든 프로젝트
는 안드로이드가 요구하는 기본 코드
와 리소스를 모두 포함하고 있으므로
그 자체로 실행 가능함



예제 프로그램 실행

- 메뉴에서 Run / Run 명령(Ctrl + F11)
을 선택하면 프로젝트를 실행할 방식을
선택함
- 안드로이드 응용 프로그램이므로 제일
위의 Android Application을 선택하고
OK 버튼을 누름

안드로이드의 액티비티와 생명주기

Mobile service ■



액티비티(Activity)

– 액티비티 :

- 여러 개의 액티비티가 모여 하나의 안드로이드 앱을 이루며, 하나의 프로세스로 이해하면 됨
- 인터페이스 단위의 기본적인 실행 단위
- 하나 이상의 레이아웃이나 뷰를 가짐



생명주기(Life Cycle)

- 액티비티의 시작, 실행, 활성화, 비활성화, 정지, 종료되는 일련의 상태 순환을 의미함
- 시스템은 태스크의 실행중인 액티비티를 스택으로 관리함



생명주기(Life Cycle)

- 액티비티가 새로 생성되면 스택 제일 위에 놓여 활성화되고, 다른 상태의 액티비티가 또 생성되면 기존 액티비티는 잠시 가려지고 새 액티비티가 스택의 제일 위에 배치됨



생명주기(Life Cycle)

- 스택의 제일 위의 액티비티가 종료되면 바로 아래의 액티비티가 활성화됨



생명주기(Life Cycle)

실행(Active, running) 상태

- 사용자가 실제로 시스템의 자원 (CPU, 메모리 등)을 사용하는 상태
- 스택의 제일 위에 있으며 화면상에서도 역시 제일 위에 있음
- 입력 포커스를 가지며 사용자의 입력을 직접 처리함



생명주기(Life Cycle)

일시 정지(pause) 상태

- 포커스는 잃었지만 사용자에게는 보이는 상태
- 위쪽에 다른 액티비티가 있지만 화면 전체를 다 가리지 않았거나 반투명한 경우가 이에 해당
- 살아 있는 상태와 같지만 시스템에 의해 강제 종료될 수도 있음



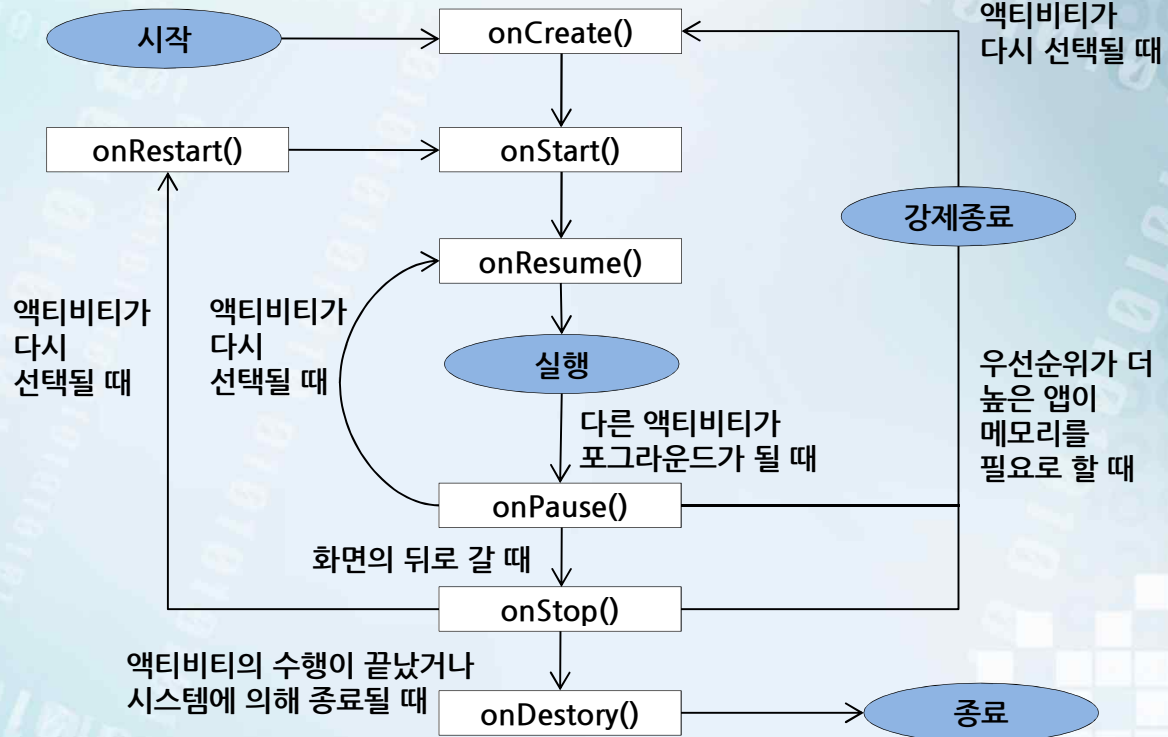
생명주기(Life Cycle)

정지(stopped) 상태

- 다른 액티비티에 의해 완전히 가려진 상태이며 사용자 눈에 보이지는 않음
- 그러나 모든 정보를 다 유지하고 있으므로 언제든지 다시 활성화될 수 있음
- 시스템은 메모리가 부족하면 정지상태의 액티비티를 언제든지 강제 종료할 수 있음



생명주기(Life Cycle)





생명주기(Life Cycle)

- 시스템은 액티비티를 처음 생성하여 완전히 종료할 때까지 다음과 같은 메소드들을 호출함



생명주기(Life Cycle)

- 이 중 일부는 생략될 수도 있고 활성화 상태가 자주 바뀌면 onPause와 onResume은 여러 번 호출되기도 함



생명주기(Life Cycle)

메소드	해야 할 일
onCreate	액티비티를 초기화한다. 중지했다가 재시작하는 경우라면 액티비티의 이전 상태 정보인 Bundle이 전달된다. 이 정보대로 재초기화한다.
onRestart	재시작될 때 호출된다.
onStart	액티비티가 사용자에게 보이기 직전에 호출된다.
onResume	사용자와 상호작용을 하기 직전에 호출된다. 이 단계에서 스택의 제일 위로 올라온다.



생명주기(Life Cycle)

메소드	해야 할 일
onPause	다른 액티비티가 실행될 때 호출된다. 이 단계에서 미저장한 데이터가 있으면 저장하고 애니메이션은 중지해야 한다. 이 메소드가 리턴되어야 새 액티비티가 활성화되므로 시간을 너무 많이 끌어서는 안 된다.
onStop	액티비티가 사용자에게 보이지 않게 될 때 호출된다.
onDestory	액티비티가 파괴될 때 호출된다. 시스템에 의해 강제로 종료되는 것인지 아니면 finish 메소드 호출에 의해 스스로 종료하는 것인지는 isFinishing 메소드로 조사할 수 있다.

학습 개요

- ✓ SDK, JDK, 이클립스, AVD에 대해 개념을 학습한다.

학습 개요

- ✓ 안드로이드 프로그래밍을 개발하기 위해 필요한 프로그래밍 환경 수립 및 설정에 대해 학습하고, 예제 프로그래밍 수행 방법을 학습한다.

학습 개요

- ✓ 개발툴 설치 순서 및 방법을 학습한다.
- ✓ 액티비티의 생명주기를 학습한다.

학습 목표

- ✓ SDK, JDK, 이클립스, AVD의 개념을 이해하고 구분할 수 있다.
- ✓ 프로그래밍을 하기 위한 환경 수립 및 설정을 할 수 있다.

학습 목표

- ✓ 개발툴의 설치 순서 및 방법을 알고 설치할 수 있다.
- ✓ 예제프로그램 수행 방법을 알고, 실행시킬 수 있다.

학습 목표

- ✓ 액티비티의 생명주기를 상태별로 개념을 알고, 구분할 수 있다.

주요 용어

- ✓ JDK : 자바 어플리케이션 개발을 위한 컴파일러와 가상머신, 기타 툴킷

주요 용어

- ✓ 이클립스 : 다양한 플랫폼에서 사용할 수 있는 범용 응용 소프트웨어로 안드로이드 개발환경에 필수적
임

주요 용어

- ✓ ADT 플러그인 : 이클립스에서 안드로이드 어플리케이션을 개발할 수 있도록 플러그인 형태로 제공하는 개발툴

주요 용어

- ✓ SDK : Software Development Kit(소프트웨어 개발 도구)으로 어플리케이션을 개발함

주요 용어

- ✓ DDMS : Dalvik Debug Monitor Service로서 디버깅, 스크린 샷, 파일 탐색기 등과 같은 기능을 제공함

주요 용어

- ✓ AVD : 안드로이드폰 없이 어플리케이션을 개발하는 개발자를 위해 프로그램 실행을 할 수 있는 도구임

주요 용어

- ✓ 생명주기 : 액티비티의 시작, 실행, 활성화, 비활성화, 정지, 종료되는 일련의 상태 순환을 의미함

학습 정리

- ✓ 안드로이드 프로그래밍 개발을 위해서는 JDK, 이클립스, ADT 플러그인, SDK, AVD 생성 순으로 설치해야한다.

학습 정리

- ✓ 액티비티 생명주기는 시작, 실행, 활성화, 비활성화, 정지, 종료되는 일련의 상태 순환을 의미하며, 메소드 별로 하는 일이 정해져있다.