

아이폰 SDK 프로그래밍

SDK의 주요 툴 및 도움말 활용

이 글을 읽는 독자들 중에는 이미 아이폰 애플리케이션을 개발한 이들도 있을 것이고, 일부는 이제 막 시작하는 단계일 수도 있다. 한국에 곧 아이폰이 상륙한다는 소문이 무성한 만큼 이번 연재의 마지막 시간에는 아이폰 SDK를 개발하는 데 있어서 가장 중요한 툴에 대해 설명한다. 아울러 아이폰 SDK로 개발을 진행할 때 도움말이 매우 유용하므로 아이폰 SDK 도움말에 대해서도 자세히 살펴보기로 한다.

3

연 재 순 서

- 1회 | 2008. 7 | 아이폰 SDK의 이해 및 설치
- 2회 | 2008. 8 | Objective-C 기본 예제 분석
- 3회 | 2008. 9 | Objective-C 문법의 이해
- 4회 | 2008. 10 | SDK의 주요 툴 및 도움말 활용

강덕진 fromdj2k@gmail.com | 임베디드 시스템 소프트웨어 분야에서 근무하고 있고 MS 윈도우 커널과 WDM 등이 주 관심분야이다. IA32 기반의 운영체제를 개발해 수상한 바 있고 이 운영체제를 발전시켜 언젠가 사람들을 위해 릴리즈한다는 목표를 갖고 있다. 현재 개인 홈페이지 fromdj.pe.kr을 운영하고 있다.

지난 9월 10일에는 애플의 특별 이벤트가 있었다. 많은 사람들이 기대했던 대로 이번에도 스티브 잡스가 새로운 제품들을 소개하는 시간이 마련됐다. 그 자리에서는 새로운 아이팟 나노와 터치, 그리고 최신버전의 아이튠스가 함께 소개됐다. 역시 애플은 소프트웨어 회사여서 소프트웨어가 하드웨어보다 더 중요시되는 것을 느낄 수 있다. 사람들이 아이폰에 열광하는 것은 아이폰의 하드웨어보다는 그 소프트웨어의 매력 때문이다. 국내에 소개될 수많은 MP3 플레이어나 내비게이션 등의 첨단 제품들도 하드웨어보다는 소프트웨어에 좀더 많은 투자와 노력을 기울였으면 하는 바람이다.

Xcode

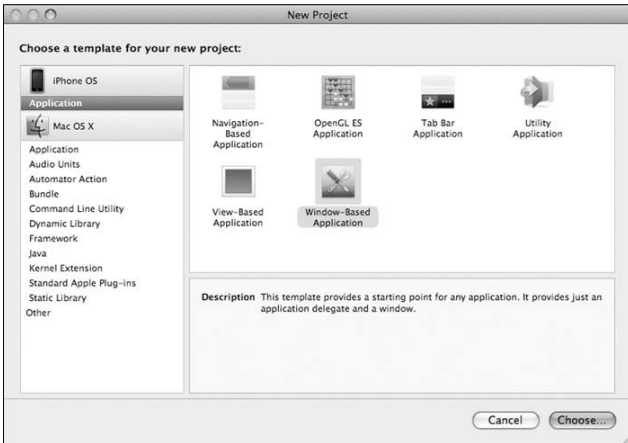
윈도우에 비주얼 스튜디오, 자바에 이클립스가 있다면 아이폰에는 Xcode가 있다. 전에도 언급했지만 Xcode는 아이폰은 물론이고 맥 OS X에서 실행되는 애플리케이션을 위한 개발환경이기도 하다. Xcode 외에도 여러 개발 툴들이 포함되어 있는데, 이들을 모두 Xcode 환경이라고 부를 만큼 이는 대표적인 툴이다. Xcode의 기본 기능을 요약하면 다음과 같다.

- **소스 코드 분석** : 코드의 심벌을 인식해 쉽게 찾을 수 있도록 해준다.

- **아이폰 시뮬레이터** : 시뮬레이터를 통해 실제 장치 없이도 동작을 실험해 볼 수 있다.
- **소스 코드 편집** : 편집기가 포함되어 있어서 손쉽게 코딩할 수 있다.
- **인터페이스 디자인** : 외부 툴인 인터페이스 빌더와 연동해서 인터페이스를 디자인하고 코드와 연결시킬 수 있다.
- **빌드** : 손쉽게 코드를 빌드할 수 있으며 사용자 환경에 맞게 수정해 사용할 수 있게 한다.
- **성능 분석** : 메모리의 누수나 장치의 효율성을 위해 성능을 분석할 수 있는 Instruments나 Shark 등의 툴을 사용할 수 있다.
- **소스 형상 관리** : 소스 형상 관리가 툴에 통합되어 있으므로 여러 개발자가 동시에 작업을 진행하는 경우에 유용하다.
- **디버깅** : 소스 코드와 동기화해서 디버깅을 진행할 수 있다. 그 외 조사식, 메모리 조사 등의 기본적인 기능을 모두 지원해 준다.

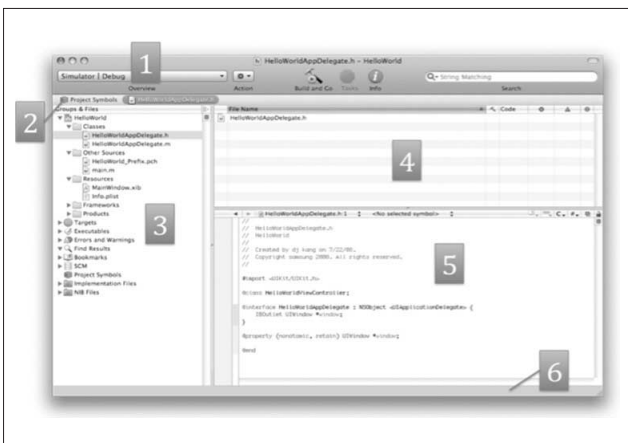
이제 Xcode를 실행시켜 보자. 이는 앞으로 자주 사용할 것이므로 Dock에 복사해 놓도록 하자. 프로그램의 정확한 위치는 /Developer/Applications/Xcode이다. 이 파일을 실행하면 환영한다는 의미의 윈도우가 나타나는데, 거기에는 아이폰 개발에 참고할 수 있는 리소스들이 나와 있다. 일단 이 창은 닫고 메뉴에서 새로운 프로젝트를 하나 만들어 보자. File 메뉴에서 New

Project를 선택하면 <화면 1>과 같이 새로운 프로젝트를 생성하는 창을 볼 수 있다.



<화면 1> 새로운 프로젝트

여기서 Window-Based Application을 선택하도록 하자. 이어서 프로젝트를 저장할 파일명을 입력하면 <화면 2>처럼 새로운 프로젝트가 하나 생성된 것을 볼 수 있다.



<화면 2> 프로젝트 윈도우

<화면 2>의 프로젝트 윈도우가 어떻게 구성되어 있는지를 한번 살펴보자.

- 1 툴바** : 툴바에는 코드를 빌드하고 실행시키는 등의 가장 많이 사용되는 명령들이 위치하고 있다.
- 2 즐겨찾기 박스** : 자주 사용되는 파일이나 그룹을 이곳에 놓고 바로 열 수 있다. 의외로 아주 편리하니 한번 시험해 보도록 하자.
- 3 프로젝트 항목 윈도우** : 프로젝트를 구성하는 파일들과 라이브러리들을 정리해 보여준다. 여기서 프로젝트에 파일을 추가하거나 제거하는 등의 작업을 할 수 있다.

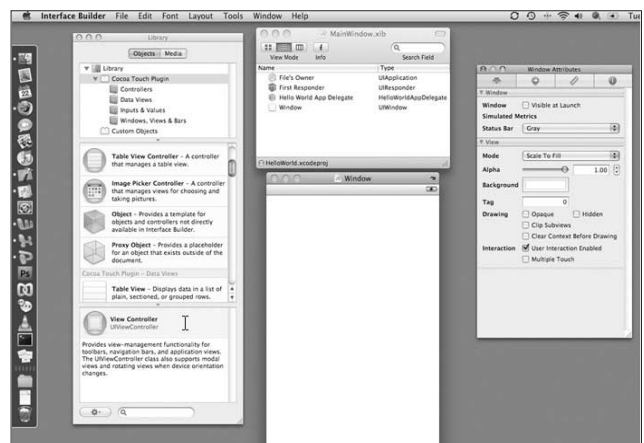
4 세부 항목 : 3번 영역에서 선택된 항목에 포함되어 있는 항목들을 자세히 표시해 준다.

5 코드 편집기 : 3번이나 4번 영역에서 편집 가능한 파일이 선택되었을 경우에 편집할 수 있는 창이다. 소스 코드 편집은 주로 이 창에서 이뤄진다. 다른 방법으로 3, 4번 영역에서 항목을 더블 클릭하면 별도의 창으로 편집할 수도 있다.

6 상태 표시 줄 : 현재 상태를 표시하는데 특히 빌드 상태나 빌드한 결과를 확인할 때 유용하다.

인터페이스 빌더

인터페이스 빌더는 말 그대로 사용자 인터페이스를 디자인하는 툴이다. 윈도우와 뷰 그리고 뷰를 구성하는 여러 가지 구성요소들을 배치시키고, 속성을 편집할 수 있게 한다. 이전 절에서 프로젝트를 하나 만들어 둔 것이 있다. 프로젝트의 왼쪽 구성 항목 창에서 Resource라는 그룹에 있는 MainWindow.xib라는 파일을 볼 수 있다. 이 파일을 더블 클릭해 보자. 그럼 인터페이스 빌더라는 프로그램이 실행되는 것을 볼 수 있는데, 이는 <화면 3>과 같이 4개의 창으로 이루어져 있다.



<화면 3> 인터페이스 빌더 실행 모습

여기서 MainWindow.xib라는 이름을 가진 창을 보도록 하자. 이 창에서 화면을 이루는 구성요소와 각 구성요소의 이벤트에 대해 응답을 받기 위한 객체들의 목록을 관리할 수 있다. 실제적인 화면 디자인은 왼쪽에 있는 Library 윈도우에 있는 항목을 끌어다가 Window라는 창에 놓고 배치시킨 후에, 오른쪽의 속성 창에서 세부적인 사항을 결정하는 방식으로 진행된다.

인터페이스 빌더는 Xcode 상에서 작업하고 있는 프로젝트와 자동으로 연동되는데, 예를 들어 화면 상의 어떤 버튼이 눌렸을 때 프로젝트 내에서 어떤 코드가 실행될지를 선택할 수 있도록 되어 있다.

Instruments

아이폰은 임베디드 장치이다. 그럼 임베디드 장치의 가장 큰 약점은 무엇일까? 바로 전원이이다. 그래서 임베디드 장치에서는 전원 관리를 위한 눈물겨운 노력을 하게 마련이다. 이를 위해 아이폰 운영체제가 제공하는 자원을 얼마나 효율적으로 사용하고 사용 후에는 올바르게 반납하는지를 분석하는 것은 매우 중요한 작업이다. 그래서인지 애플에서 제공하는 프로그래밍 가이드에서는 최적화된 자원을 낭비하지 않는 프로그램을 작성해야 한다고 강조하고 있다. 이러한 최적화 작업의 축이 바로 Instruments 툴이다. 이 툴은 추적하고 싶은 항목에 대해서 프로그램이 얼마만큼의 자원을 사용하고 있는지를 실시간으로 추적해 준다. <화면 4>를 보자. 이 화면은 예제 아이폰 애플리케이션을 실행시켜서 메모리 사용량을 확인하는 모습을 보여주고 있다. Instruments는 직접 실행시킬 수도 있고 Xcode 상에서 Run -> Start with Performance Tool 메뉴를 통해 실행시킬 수 있다. 예제를 하나 실행시킨 화면을 보도록 하자.



<화면 4> Instruments 실행 화면

화면 구성을 보면 크게 세 가지 영역으로 나눌 수 있다. 먼저 왼쪽의 **1번** 영역은 현재 추적 중인 테스트 항목을 보여준다. 그리고 **2번** 영역은 실제로 프로그램(여기서는 아이폰에서 실행 중인 애플리케이션)을 실행시키면서 그 추적된 결과 데이터를 표시해 주는 영역이다. 그리고 **3번**은 추적 과정의 세부 데이터 항목을 표시해 주는 영역이다.

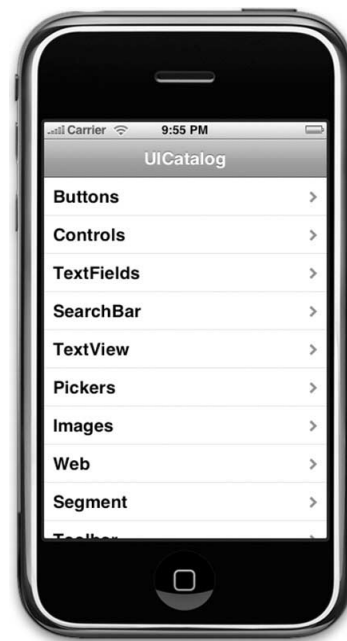
아이폰 시뮬레이터

아이폰 SDK를 개발하는 목적은 아이폰에서 동작하는 프로그램을 만들기 위해서다. 하지만 프로그램을 개발해 매번 아이폰에서 직접 손으로 조작하며 테스트를 할 수는 없다. 또한 실제 아이폰이 없는 경우도 생각해 볼 수 있는데, 이런 경우에도 개발을 진행할 수 있도록 아이폰 시뮬레이터가 제공된다. 그리고 실제 장

치가 있다고 하더라도 애플 홈페이지를 통해 일정 금액을 내고 개발자 프로그램에 가입한 다음에야 장치에서 프로그램을 테스트해 볼 수 있다.

아이폰 시뮬레이터는 몇 가지 제약은 있지만 실제 장치와 거의 차이가 없이 동작을 실험해 볼 수 있고, 멀티 탭과 같은 인터페이스까지도 지원해 준다. 아이폰 시뮬레이터가 실행되는 모습은 <화면 5>와 같은데, 이는 실제 아이폰과 완전히 똑같은 모습이다. 독자가 프로그램을 개발하고 테스트할 때에는 바로 이 아이폰 시뮬레이터를 주로 이용할 것이다. 알아두어야 할 중요한 조작법에는 다음과 같은 것들이 있다.

- **누르기** : 손으로 화면을 누르는 것은 마우스를 클릭하는 것과 같다.
- **누르고 있기** : 클릭한 상태로 유지하고 있으면 된다.
- **두 번 누르기** : 더블 클릭하면 화면을 두 번 누르는 효과와 같다.
- **뉘아내기 & 끌기(Swipe & Drag)** : 마우스를 클릭한 채로 드래그한다.
- **Home 버튼** : 프로그램을 종료시킨다.
- **Rotate Left, Right** : 아이폰을 좌,우로 회전시킨다. 화면이 옆으로 긴 형태로 눌렀을 때의 인터페이스를 테스트해 볼 수 있다.
- **멀티 터치(Pinch)** : 옵션(Alt) 키를 누르면 화면에 멀티 탭이 표시되는 것을 볼 수 있고, 클릭하면 멀티 탭으로 화면을 드래그하는 동작을 할 수 있다.



<화면 5>
아이폰 시뮬레이터 실행 모습

Xcode 상에서 사용할 타깃을 아이폰 시뮬레이터로 설정하고 실행시키면 자동으로 아이폰 시뮬레이터가 실행된다. 또한 단독으로 직접 아이폰 시뮬레이터를 실행시켜도 사용해 볼 수 있다.

도움말 접근하기

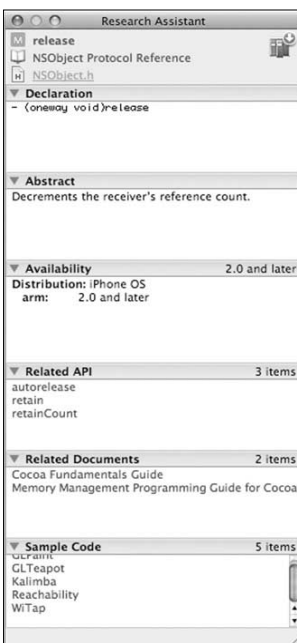
효율적인 개발환경을 갖추는 데 있어서 가장 중요한 일 중의 하나가 바로 쉽고 빠르게 도움말에 접근할 수 있는 방법을 갖추는 일이다. 리눅스/유닉스 개발자라면 반드시 필요한 것이 man 페이지이다. 마찬가지로 MSDN 없는 닷넷 개발자를 상상할 수 없을 것이다. 아무리 뛰어나고 머리 좋은 개발자라 하더라도 모든 API를 기억할 수는 없다. 그렇기 때문에 올바른 도움말을 빨리 접근해 찾을 수 있는 환경을 가지는 게 꼭 필요하다. Xcode 개발환경에서도 쉬운 방법으로 도움말에 접근할 수 있도록 다음 두 가지의 기능을 지원한다.

- 조사 도우미 (Research Assistant)
- 도움말 윈도우

조사 도우미는 간략하게 API의 용법을 참고할 수 있도록 하는데, 에디터 상에서 현재 커서가 위치하는 부분에 대해 자동으로 문맥을 감지해 검색해 준다. 그리고 도움말 윈도우는 현재 시스템에 설치되어 있는 도움말을 검색하고 참조할 수 있게 해준다. Xcode는 모든 도움말 관련 문서를 하드디스크에 두지는 않고, 가이드 문서와 API 레퍼런스만 하드디스크에 두고 나머지는 웹을 통해 직접 접근하도록 되어 있다.

조사 도우미

조사 도우미의 가장 큰 의미는 바로 소스 코드를 편집하는 상황에서 동적으로 도움말을 볼 수 있다는 데 있다. 반면 도움말 원



〈화면 6〉 조사 도우미

도우는 새로운 창이 떠서 화면을 가리기 때문에 일단 코딩은 중단해야 한다. 간편하게 참고하는 데에는 도움이 될 수 있지만, 상세한 설명은 도움말 윈도우를 봐야한다. Xcode를 실행하고 Help 메뉴에서 Show Research Assistant를 선택하면 <화면 6>과 같은 창이 뜨는 것을 볼 수 있다.

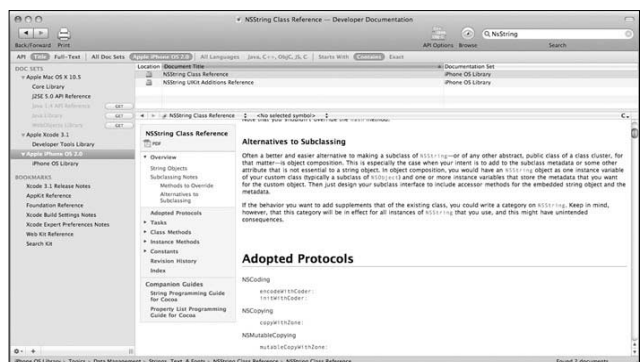
소스 코드 창은 그대로 볼 수 있고 그 옆에 위치하기 때문에 소스 코드를 편집하면서도 사용할 수 있음을 알 수 있다. <화면 6>의 창은 소스 코드 상에서 release라는 부분을 선택했을 경우의 모습인데 각 부분의 용도를 간단히 알아보도록 하자.

- **최상위 창** : 선택한 항목에 대한 헤더 파일과 관련된 항목을 표시해 준다.
- **Declaration** : 해당 항목이 어떻게 선언되어 있는지를 알 수 있다. 메소드의 경우 리턴형과 인자에 대한 정보를 얻을 수 있게 된다(여기서 메소드란 객체의 내부에 구현되어 있는 함수를 말하며 함수와 혼용할 수 있는 용어이다).
- **Abstract** : 간략하게 해당 항목을 설명해 준다.
- **Availability** : 해당 항목을 사용할 수 있는 시스템을 알려준다. 예를 들어 iPhone OS 2.0 이상에서만 된다는 등의 정보가 표시된다.
- **Related API** : 관련된 API를 보여준다.
- **Related Documents** : 해당 항목과 관련 있는 다른 문서를 보여준다.
- **Sample code** : 해당 항목을 참고할 수 있는 예제 코드를 보여준다.

조사 도우미를 실행시키는 단축키는 Control + Command + '?' 키를 동시에 누르는 것이다.

도움말 윈도우

도움말 윈도우는 개발을 진행하면서 가장 많이 참고해야 할 윈도우이다. 메뉴 상으로 Help → Documentation을 선택하면 볼 수 있고 단축키로는 Option(또는 Alt) 키와 Command 키를 누른 상태에서 '?' 키를 누르면 된다. 도움말 윈도우는 <화면 7>과 같이 구성되어 있다.



〈화면 7〉 도움말 윈도우

〈화면 7〉의 왼편에 있는 DOC SETS 아래에 있는 항목들은 현재 도움말 윈도우에서 사용할 수 있는 문서 세트를 보여준다. 자세히 보면 설치되지 않은 항목 옆에는 GET이라는 버튼이 있어서 온라인으로 바로 설치할 수 있다. 그 아래에는 BOOK MARKS 항목이 있는데, 자주 사용하는 문서에 대해 링크를 만들어 두고 바로 접근할 수 있다.

오른편 상단에 있는 항목 창은 검색된 문서를 보여준다. 그리고 그 아래 영역에서는 검색된 문서 중에서 선택된 문서의 내용을 볼 수 있다. 이 화면에서 상단의 툴바가 매우 중요하기 때문에 한번 확대해서 보도록 하자.



〈화면 8〉 툴바

〈화면 8〉의 툴바는 검색할 단어와 검색대상을 지정할 수 있게 해준다. 기본적인 검색을 위해서는 단순히 1번 영역에 검색할 단어를 입력하고 엔터를 입력하면 된다. 하지만 검색에 앞서 2번 영역에서 검색할 대상을 선택해서 범위를 좁힐 수도 있다.

맨 왼편에서는 검색대상을 API로 삼을지 제목에 원하는 단어가 포함된 문서를 검색할지 또는 전체 내용에서 해당 단어가 포함되어 있는 문서로 정할지를 결정한다.

그 다음에는 검색에 사용할 문서 세트를 선택할 수 있는데, 설치된 모든 문서에서 검색하거나 iPhone OS 2.0 문서 세트에서만 검색하게 할 수 있다.

그 다음 옵션에서는 해당하는 언어를 선택할 수 있다. 예를 들어 Objective C에 해당하는 문서만 검색할 수 있는 식이다. 마지막 옵션은 입력한 단어로 시작하는 경우인지, 아니면 중간에 포함되는지 등을 선택할 수 있게 한다.

위의 검색 방법에 덧붙여서 논리 연산자, 와일드카드 등의 고급 검색 기능을 이용할 수도 있다. 지금부터 그 사용법을 간단히 살펴보자.

고급 검색 : 논리 연산자

검색하고자 할 때 검색할 단어가 어렵듯이 기억이 난다거나 일부만 알고 있을 경우가 반드시 있게 마련이다. 이런 경우를 위해 일부 단어들의 조건을 기술해서 검색할 수 있는 방법이 있다. 사

아이폰 AppStore에 대한 단상

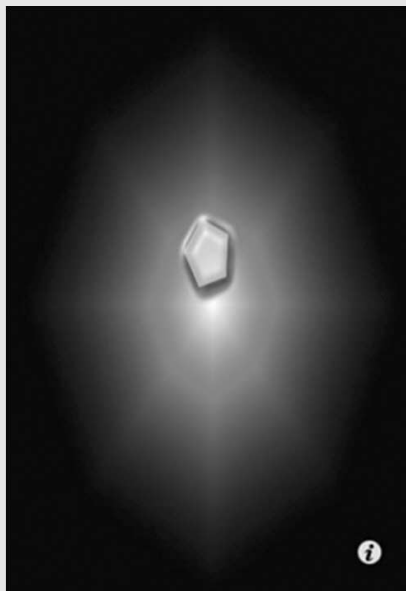
애플의 아이폰 애플리케이션을 구매 및 판매할 수 있는 온라인 스토어인 AppStore에 대한 개발자들의 불만이 몇 가지 제기되고 있다. 최근의 예로는 파드캐스트를 3G 인터넷망으로 다운로드해 들을 수 있는 애플리케이션을 수 개월에 걸쳐 개발한 개발자가 AppStore에 자신의 제품을 등록하려다가 거절당한 사건이 있었다. 애플 측에서는 현재 아이폰이 가진 기능과 중복되기 때문이라고 설명하고 있지만, 이미 카메라를 이용하는 애플리케이션이 수도 없이 AppStore에 등록되어 있는 것 역시 중복된 기능이 아니냐고 개발자들은 항변하고 있다.

그 외에도 999.0 달러짜리 'I Am Rich'라는 아무 일도 하지 않는 프로그램이 AppStore에 등록된 적이 있었다. 그리고 6명의 사용자가 이 프로그램을 구매했다고 한다. 여기서 고민해야 할 것들이 많다. 과연 'I Am Rich'라는 프로그램에 999.0 달러의 가격을 책정한 행위 자체를 나무랄 수 있을까라는 것이 한 가지 문제이고, 또 다른 문제는 애플이 이 문제의 애플리케이션 판매를 바로 중단한 것이 올바른 것이냐 하는 점이다.

현재 다른 수많은 착한(?) 프로그램들은 적절한 가격에 판매되고 있다. 그렇다면 여기서 말하는 적절한 가격은 누가 정하는 것인가? 그리고 아무 일도 하지 않는다는 판단 자체도 무척 모호한 부분이다. 60대 할머니에게 아이폰 게임 프로그램은 아무 일도 하지 않는 프로그램이지 않을까? 시장 원리에 따라 가격이 결정된다고 했을 때 999.0 달러의 아무 일도 하지 않는 애플리케이션은 스스로 소멸되도록 되어 있다.

결국 AppStore라는 작은 시장이 거대한 인간 사회의 여러 패러다임을 대변하고 있다는 생각이 든다. 독자들은 여기에 대해 어떤 생각들을 갖고 있

는지 궁금하다. 근래의 애플은 세계에서 가장 주목받으며 혁신을 이끌고 있는 회사임에는 틀림없다. 그리고 새로운 생각과 새로운 방식이 적용되기 때 문인지 많은 화제와 논란의 대상이 되는 것도 사실이다. 하지만 이렇듯 항상 도전하고 새로운 시도를 감행하는 회사가 있다는 것은 어쩌면 개발자들에게 큰 다행으로 여겨진다.



〈화면 9〉
I Am Rich의
실행모습

용할 수 있는 논리 연산은 다음 목록과 같다.

- ! : NOT
- & : AND
- | : OR
- () : 식을 묶음

예를 들어 검색하고자 하는 대상에 반드시 NSString과 URL이라는 문자가 포함되거나 또는 "NSMutableArray class"가 포함되는 경우를 찾고자 할 때는 다음과 같이 입력한다.

(NSString & URL) | "NSMutableArray class"

여기서 “ ” 는 공백이 있는 단어를 입력할 때 사용된다는 것을 알아두자.

고급 검색 : 단어로 제한하기

검색하고자 하는 문서가 어떤 단어는 반드시 포함하되, 또 어떤 단어는 포함되지 않아야 할 경우라면 +, -를 이용해 존재해야 할 단어와 없어야 할 단어를 설정할 수 있다. 예를 들어 NSString은 있지만 NSMutableString이 없는 경우에는 다음과 같이 입력할 수 있다.

+NSString -NSMutableString

고급 검색 : 와일드카드 문자

와일드카드 검색은 * 또는 ?를 이용해 검색하는 방법이다. * 문자는 없거나 하나 이상의 모든 문자를 의미하고 ? 문자는 임의의 문자 하나를 나타낸다. 일반적으로 와일드카드는 두 가지 문자를 모두 지원하지만 Xcode의 문서 검색에서는 *만을 사용할 수 있으니 주의하도록 하자. 예제로 다음과 같은 표현은 CFMutable로 시작하고 그 사이에 Ref가 포함된 경우를 검색하되, Array로 끝나는 경우는 제외시킨다.

CFMutable*Ref* ~*Array

편집기의 문자열을 바로 도움말 윈도우로 검색하기

소스 코드를 열심히 입력 중인데 급하게 도움말을 검색해야 할 때 매번 메뉴를 눌러 도움말 윈도우를 실행시킨 후에 입력 창에 원하는 문자를 입력하는 방식은 너무 느리고 소모적이다. 좋은 개발 툴은 뛰어난 동선을 제공하는 툴이어야 한다. 즉 원하는 작

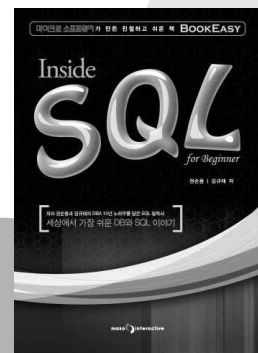
업을 최대한 빨리 진행할 수 있게 해야 한다.

Xcode의 편집기에서는 아주 간단하게 현재 커서 위치에 있는 코드를 바로 검색할 수 있다. 방법은 옵션 키를 누른 상태에서 원하는 단어를 더블 클릭하는 것이다. 지금 Xcode를 실행할 수 있는 독자라면 한번 시도해 보길 바란다. 아주 편리하게 원하는 함수를 검색할 수 있을 것이다.

지난 4회 간에 걸친 연재가 이것으로 마무리되었다. 최초의 계획에서 많이 수정되어 실제적인 애플리케이션 프로그래밍보다는 프로그래밍 과정에 필요한 준비과정을 다루는 데 초점을 맞췄다. 이 부분은 필자에게도 많은 아쉬움으로 남는다. 요즘 들어 기회에 대한 이야기를 다시 생각해 보게 된다. 그리고 개발자에게 주어지는 많은 기회들과 그에 대한 선택이 얼마나 중요한 영향을 미치는지를 생각하게 된다. 필자의 눈에는 아이폰이 매우 좋은 기회로 보이며 이 순간에도 많은 개발자들이 그 기회를 보고 달려가고 있다. 그들의 건승을 진심으로 바란다. ●

세 상 에서 가 장 쉬 운 DB 와 SQL 이 야 기

마이크로소프트웨어가 만든 첫 번째 단행본 Inside SQL for Beginner



모든 프로그램의 기본이 되는 데이터베이스.
이 책은 SQL에
대해 체계적으로 이해할 수 있도록 각 기능들을
예제와 함께 상세히 다루고 있다.
10년간 DBA로 근무하며 모아둔 저자의
SQL 노하우들이 살아 숨쉬는
Inside SQL for Beginner는
개발자들의 프로젝트를 성공으로 이끄는
견인차 역할을 톡톡히 해낼 것이다.