

## 핑크레드의 실전 모바일 자바 프로그래밍

다들 잘 지내고 계신지요..드디어 슬슬 날씨가 따뜻해지고 있습니다.

올해 들어 두번째 강좌가 되겠네요. 제가 모바일 프로그램을 시작한지 이제 언 5년차가 되네요..  
세월도 많이 흘렀는데...그동안 멀 했는지..ㅋㅋㅋ

이번엔 컴포넌트를 정리해서 설명드리기로 했는데..귀차니즘에 빠져있습니다. 그놈의 정리는  
말로만 매번 하고 있습니다. 먼 미래에는 생각만 하거나 혹은 말로 하면 자동으로 코딩되는 툴  
이 좀 나왔으면 좋겠습니다..^^b

모든 컴포넌트를 한번에 소개해 드릴수는 없으니 가장 핵심이되는 컴포넌트인 List에 대해서 말  
씀드리겠습니다. 아마 List에 강좌는 이번이 3번째일겁니다. 그렇게 List는 탈도 많고 고칠부분  
도 많고 향상될부분도 많은가 봅니다. 예전에 어떻게 설명들렸는지는 까먹었습니다. 다시 강좌  
내용을 찾기도 귀찮고 해서 새로게 시작한다는 마음으로 설명드리겠습니다.

List에 대한 정의는 너무나 다양합니다. 어디까지 요구 수준을 들어주어야 할지도 문제가 되겠  
지요. 그럼 약간 평균적인 요구사항을 설명을 드리면

1. 당근 리스트 내용이 보여야 겠지요.
2. 리스트 영역 보다 리스트 내용이 많으면 스크롤바가 생기면서 찢리겠쥬
3. 리스트의 이동은 맨 하단에 도착하면 하단에서 하나씩 내려가게 되쥬.
4. 맨 마지막 리스트를 선택하면 첫번째 리스트로 순환되고..

옵션

숫자키를 누르면 해당 리스트가 선택되기(한자리 숫자는 쉬움)

- 문제점은 숫자가 두자리 숫자면 표현은 가능한데..귀차니즘 발동

대충 이렇쥬. 근데 가끔씩 어떤곳은 가운데를 포커스 중심으로 움직여서 맨 하단이 나올때까지  
계속 가운데 있기도 하는데 그런 특수한 UI구조는 나름대로 구현하셔야 할 듯.

그리고 여기서 핵심 포인트는 리스트가 내용잉 영역보다 많을 경우 하단에 어떻게 보이게 할것



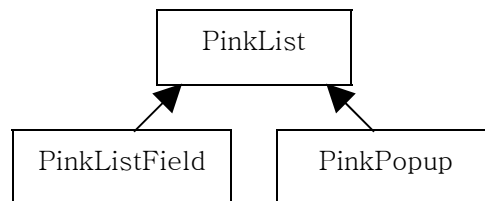
인지에 대한 이슈도 많습니다. 그러나  
저의 멋진 UI는 걸치기 기법으로 멋지  
게 해결했습니다. 핑크레드 강좌 20편  
에 이미 다 설명되어있군요..ㅏㅏ 20  
편을 안보신분을 위해서..다시 그림을  
보여드립니다...ㅋㅋㅋ

예전 그림이라서 색상 조화가 독특합  
니다. 왼쪽 그림을 보시면 7번이 살짝  
걸치게 되어있쥬. 7번이 선택되면 1번

이 살짝쿵 위로 올라갑니다. 그래서 힘들게 리스트 줄간격을 글자 크기에 정확하게 맞추려고 힘들게 하지 않아도 자연스러운 분위기가 나는 것이지요. 머 다들 저 정도야 하시겠지만..그래도 나름대로 자부심을 느끼기도 했지만..ㅋㅋㅋ 구현은 누구나 다 할수 있다는..ㅠㅠ

우측의 그림을 보여준 이유는 무엇인지 궁금하실겁니다. 단순한 팝업인데 왜 보여주시죠. 라고 하시겠죠. 그렇습니다. 팝업이 이제 그냥 팝업이 된 것이 아니죠. 아시다시피 리스트의 기본 골격은 팝업의 기본골격과 비슷한 부분이 존재하게 되는 것입니다. 그래서 리스트의 기본 골격에 티커와 스크롤바가 생기면 리스트가 되는 것이죠. 팝업은 음... 왼쪽 오른쪽인지와 별거 없네요.. ㅎㅎㅎ

그래서 구조가 상속구조 입니다.  
UML의 상속용 화살표를 구하지 못해서..ㅠㅠ  
UML에서 상속을 뜻하는 화살표는 화살표 안이 비어있습니다.



여튼..저런식으로 구현되어 있습니다. 그래서 실제로 ListField에서는 리스트의 역할이 구현되어있고 Popup에는 팝업의 역할이 있습니다. 또한 리스트에는 영역을 나타내는 setBounds와 영역내의 여백을 줄수 있는 setMargin이 있어서 리스트 영역내에서도 여백을 주어서 더 세밀하게 리스트를 표현할 수 있도록 했습니다.

지금 소스를 보니깐 메소드만 보기도 빠듯하네요..어떻게 설명을 시작해야 할지..ㅋㅋㅋ  
처음부터 차근차근 시작해 봅시다...ㅠㅠ 이리 복잡해서야...  
일단 머릿속에 필요한 클래스를 생각해 보시면..참고로 왜이리 친절히 하나하나 설명을 드리는 이유는 프로그램시에 어떻게 리팩토링을 하는지 소프트웨어의 개발론을 참고 하여 구현 하려고 합니다. 하지만 그게 그렇게 쉽게 되면 제가 여기 있겠습니까..ㅠㅠ 그냥 한번 따라해볼라고 합니다.ㅠㅠ 일단 아래 정도면 아주 기본적으로 필요한 메소드만 만들었습니다.

setChars(), setString(), setBounds(), setMargin()

getChars(), getString(), getSelect()

paint()

keyPressed(), keyReleased()

벌써 다 짠 것 같지 않습니까..뿌듯합니다.

근데 위의 내용들 어디서 많이 본 것 같습니다. ㅜㅜ

이제 필요한 변수를 생각해 봅시다.

```
final int LIST_ITEM_WIDTH = 16; // 리스트의 숫자와 글자와의 간격

int list_x, list_y, list_width, list_height; // List X, Y, Width, Height

int margin_left, margin_right, margin_top, margin_bottom; // Margin 영역

int lineFirst; // 현재 화면에 보이는 상단 Line
int lineCur; // 현재 선택된 Line
int lineMax; // 리스트 전체 라인 개수

int maxLinePerHeight; // LIST_HEIGHT에 들어가 최대 줄 개수

char[][] list_text; // 리스트 문자 배열
char[] list_num = {'①', '②', '③', '④', '⑤', '⑥', '⑦', '⑧', '⑨', '○'}; // List 숫자 배열

int colorFont; // 폰트 칼라
int colorSelFont; // 선택된 폰트 칼라
int colorSelBar; // 선택바 칼라
int colorBack; // 배경 색상.

setChars(), setString(), setBounds(), setMargin()

getChars(), getString(), getSelect()

paint()

keyPressed(), keyReleased()
```

대략 필요하다고 생각되는 것만 뽑아도..무지 많군요..실제로 더 필요한 변수가 있는데..조금 제외했습니다. ㅜㅜ 언제나 변수명을 만들때는 이름을 어떻게 해야할지 가장 고민하기도 합니다. 도대체 어떻게 해야지 자바스런 변수명이 될까..아직도 이 부분은 제가 내공이 많이 부족한듯 합니다. 함수명은 대략 기존의 자바 API를 따라하면 되는데 이념의 변수는 째나는군요..여튼.. 변수 명은 누구나 다 보더라도 저게 머하는 변수인지 이해할수 있는 변수명으로..하는게 좋죠..

^^ 메소드도 마찬가지로..ㅋㅋㅋ

이제 슬슬 메소드의 내용을 채워 봅시다.

```
/**
 * 리스트에 필요한 내용을 넣는다.
 * @param list 내용(이중 배열로 받습니다.)
 * @param count 리스트의 개수
 * @param anchor 옵션(나중에 리스트에도 옵션을 주어서 서로 다르게 구현)
 */
public void setChars(char[][] list, int count, int anchor)
{
    // 예외처리 더 많은 예외처리가 있는데요..
    // 전 예외 처리를 별로 좋아하지 않아서 문제라는..
    if(list == null || list.length < count || count <= 0)
        throw new ArrayIndexOutOfBoundsException();

    // 일단 보여준 메소드는 아니죠..기존의 리스트 정보를 초기화 시키는 것입니다.
    removeAllElements();

    // 변수 초기화
    lineFirst = 0;
    lineCur = 0;
    lineMax = count;

    // 위에는 정의 하지않았지만 위의 그림을 보셨으면 리스트의 핵심 구현이
    // 살짝 올라가기 기능을 위한 부분입니다.
    gapHeightOfListAndReal = false;
    // 만일을 대비한 옵션이죠..리스트의 종류가 여러가지 일경우에 사용.
    this.list_anchor = anchor;
    // 리스트의 내용을 그냥 받아도 되고 새로 생성해도 되는데..
    // 예전에는 메모리가 부족할 것 같아서 그냥 넘겨 받았는데요..
    // 나중에 처리하기가 복잡해서 이제는 리스트는 리스트대로
    // 목록을 카피해서 이용합니다. 이게 더 구조적이더라구요.
    list_text = new char[maxLine][];

    for(int i = 0; i < maxLine; i++)
```

```

    {
        list_text[i] = new char[list[i].length];
        System.arraycopy(list[i], 0, list_text[i], 0, list[i].length);
    }
}

/**
 *  가능하면 String를 쓰지 말자. 제가 제일 싫어하는 String입니다.
 *  @param list
 *  @param count
 *  @param anchor
 */
public void setString(String[] list, int count, int anchor)
{
    // 예외처리
    if(list == null || list.length < count || count <= 0)
        throw new ArrayIndexOutOfBoundsException();

    removeAllElements();

    // 변수 초기화
    lineFirst = 0;
    lineCur = 0;
    lineMax = count;
    gapHeightOfListAndReal = false;
    this.list_anchor = anchor;

    list_text = new char[lineMax][];

    for(int i = 0; i < lineMax; i++)
        list_text[i] = list[i].toCharArray();
}

```

setString()은 setChars()와 거진 비슷합니다. 그러나 가능하시면 String을 사용하지 말아주십시오. 스트링 사용시에는 언제나 속도와 메모리에 많은 피해가 갑니다. 근데 가끔씩 보면 String을

사용하지 말라고 했다고 일단 String으로 하시고 toCharArray()를 하셔서 전부 char로 바꾸고 나신 후에 사용하는데..이건 그냥 똑같습니다. 결국 Byte->Char로 바꾸는 변수가 없기 때문에 Byte->String->Char로 바꾸어서 사용하시는데 이는 그냥 String으로 사용하시는 것이 좋습니다. 어차피 생성시켜놓았는데..바꾸실 필요까지는 없습니다.

그럼 어떻게 사용해야 할까요? Byte->Char로 바꾸어서 사용 API를 이용합니다. 중간에 String을 거치지 않기 때문에 사용이 가능하죠..^^ 해당 함수는 MIDP API 아니기 때문에 모든 곳에서는 사용할 수는 없고 SKVM에서만 가능합니다. 다른 곳에서 사용을 원하시면 직접 구현도 가능하데..이는 조금 번잡스럽고 속도가 많이 저하될수도 있을 것 같습니다.

그리고 요즘에는 폰들이 좋아서 그냥 String으로 사용하셔도 무방합니다.

```
/**
 * 리스트의 영역을 설정.
 *
 * @param x
 * @param y
 * @param width
 * @param height
 */
public void setBounds(int x, int y, int width, int height)
{
    list_x = x;
    list_y = y;
    list_width = width;
    list_height = height;

    // 해당 메소드는 리스트의 높이에 얼마나 많은 라인이 들어갈것인가에 대한 메소드
    setMaxLinePerHeight();
}

/**
 * margin의 크기를 잡는다.
 * margin은 setBounds 이전 혹은 다음에 설정해도 되지만 가능하면
 * 보기 좋게 setBounds 다음에 설정하도록 하자.
 *
 * @param top
```

```

* @param right
* @param bottom
* @param left
*/
public void setMargin(int top, int right, int bottom, int left)
{
    margin_top      = top;
    margin_right     = right;
    margin_bottom    = bottom;
    margin_left      = left;

    setMaxLinePerHeight();
}

/**
 * 리스트의 실제높이(리스트 높이에서 마진의 크기를 뺀) 들어갈 라인수.
 *
 */
void setMaxLinePerHeight()
{
    // Height에 들어갈 최대 줄개수를 구합니다.
    // 뒤부분은 화면에 조금 보이더라도 줄로 인정하는 것입니다.(이부분이 줄개수의 핵심이죠)
    maxLinePerHeight = getRealHeight() / LIST_PER_LINEHEIGHT +
        ((getRealHeight() % LIST_PER_LINEHEIGHT > 0)?1:0);
}

```

위에서 Real의 메소드들은 실제적인 것을 의미하여 마진을 뺀 실제 높이라고 생각되시면 됩니다. 그런데 의구심이 생기실 겁니다. 왜 똑 같은 메소드를 두번이나 써서 사용하는가에 대해서.. 실제로 우리가 코딩을 한다면..

1. 실제로 해당 값을 가지고 있는 전역 변수를 이용하거나
2. 변수를 설정해서 해당 값을 받아오거나
3. 변수를 설정해서 그자리에서 계산하거나..

등등의 방법이 있습니다. 저 역시 무엇이 맞다고 할 수는 없습니다. 그렇지만 가능하면 변수를 설정하지 않고 프로그램을 짜면 코드가 깨끗해진다는 것을 배웠습니다. 그래서 한번 그렇게 구현해 보았습니다. 깔끔해 보이시나요? 실제적으로 단말에 코딩하시더라도 속도 저하는 전혀 느

까지 못했습니다. 오히려 빠른 경우가 있었는데..의심스럽기는 하죠..ㅋㅋㅋ

아~~ 첫번째 강좌 이후로 거진 1달 반이 흘렀습니다. 조금씩 조금씩 하다 보니..진도가 잘 안나가네요..일단 이번 강좌는 여기까지 하고 다음강좌에 이어서 구현하도록 하겠습니다.

너무 강좌가 안올라와도 식상할지 모르니깐요..벌써 식상해지셨을 듯..

요즘들어서 슬슬 바쁠는지...강좌 쓸시간이 없다는...ㅎㅎㅎ

닉네임 : 핑크레드

성격 : 왕소심, 왕빼짐, 이중인격, 수다쟁이

이메일 : [pinkred@hanafos.com](mailto:pinkred@hanafos.com)

추신 : 강좌를 다른 사이트에 올리시는 것은 자유입니다..단지 저한테 메일 한 통 이라도 보내주시고 올리시면 감사하겠습니다. 언제나 좋은 하루 되십시오.