



مقدمه ای بر برنامه نویسی Mobile

انواع گوشی های موبایل

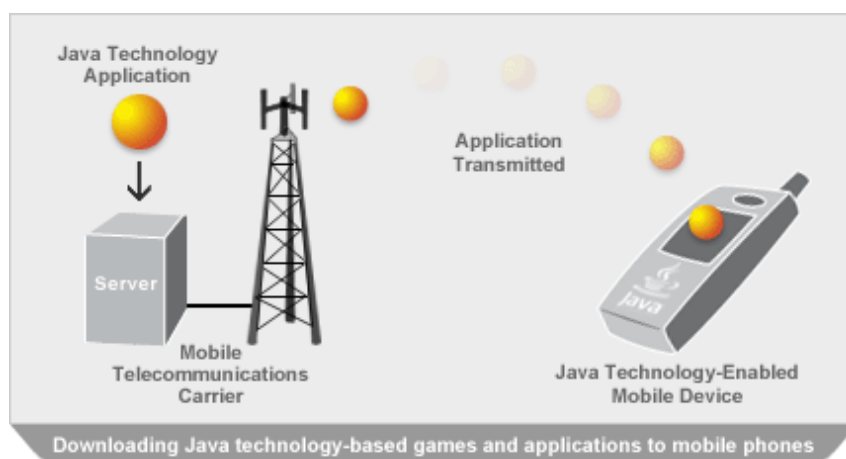
- موبایل های دارای سیستم عامل مانند nokia 6600, Sony Ericsson p900
- بدون سیستم عامل, Sony Ericsson k750

انواع سیستم عاملهای موبایل

- Palm OS
- Symbian OS
- Linux OS
- Windows OS
 - Windows CE
 - Windows CE.NET
 - Windows Mobile

جاوا بر روی موبایل

از جمله مهمترین زبانها که می توان بر روی موبایل برنامه نویسی کرد به زبان Java اشاره کرد انگیزه اصلی طراحان جاوا تولید نرم افزارهایی بود که بتوان از آن در وسایل مصرفی الکترونیکی خانگی دستگاه های کنترل از راه دور(بی سیم) موبایل ، PDA ، Pocket Pc و اتومبیل ها و استفاده کرد.



ویژگی های زبان جاوا

۱. شیء گرا
۲. انتقال پذیری(Portable) شعار جاوا : یک بار بنویسید و همه جا از آن استفاده کنید.(قابلیت اجرا بر روی کلیه سیستم عاملها را دارد)
۳. کارایی بالا (جاوا ۱۵ بار بهینه شده است)
۴. قابلیت Multithread
۵. قابلیت نصب بر روی موبایل های بدون سیستم عامل مانند Sony Ericsson k750
۶. قابلیت پشتیبانی کامل از Unicode
۷. قابلیت پشتیبانی از xml
۸. و در نهایت جاوا مجانی است(اما Open Source نیست)
۹. ساده
۱۰. پویا
۱۱. ایمن(Secure)

جاوا از نظر ساختار بسیار شبیه زبان C/C++

جاوا از نظر ساختار بسیار شبیه زبان C/C++ و این به هیچ وجه تصادفی نیست. C زبانی است ساخته یافته و ++C زبانی شی گرا و مهمتر از همه قسمت اعظم برنامه نویسان دنیا از C/C++ استفاده می کنند. و از سوی دیگر این حرکت به طرف جاوا را برای این قبیل افراد ساده خواهد کرد.

جاوا با دور انداختن نشانگرها (Pointers) و بر دوش کشیدن بار مدیریت حافظه برنامه نویسان C/C++ را برای همیشه از این کابوس رهایی بخشیده است. جاوا همچون C/C++ به بزرگی و کوچکی حروف حساس است و برنامه نوشته شده باید دارای متد main باشد.

مثال از زبان c

```
void main(void){
    for(int i=0;i<10;i++)
        printf("\n Hello From C++");
}
```

مثال از زبان جاوا

```
Class ClassMain(){
    public static void main(){
        for (int i=0;i<10;i++)
            System.Out.PrintLine("Hello From JAVA");
    }
}
```

کلا جاوا بر روی سه پلتفرم مجزا ارائه شده است:

J2EE: Java 2 Enterprise Edition

که برای سیستم های سطح بالا (Enterprise)، سرورها (Server Applications) و web

J2SE: Java 2 Standard Edition

J2SE برای توسعه client/desktop applications بکار می رود

J2ME: Java 2 Micro Edition

برای موبایلها و PDA ها و pocket pc و غیره.....

تفاوت این نسخه های مختلف در کلاسها و پکیجهایشان باشد که این هم بخاطر کاربرد خاص هر یک می باشد

جاوا و تولید بازی برای موبایل

در جاوا کتابخانه ای به نام java2D,java3D وجود دارد. java3D مجموعه ای از چند کتابخانه است که برای ایجاد گرافیک سه بعدی، قابلیت های زیادی را در اختیار ما می گذارد. و امکاناتی از قبیل نور پردازی، اشکال هندسی، گرافها، و غیره ... را در اختیار ما قرار می دهد.

چگونه برنامه نویسی بر روی موبایل را آغاز کنیم ؟

در مرحله یک

برای نوشتن برنامه جاوا برای موبایل نیاز به J2ME Sdk موبایل می باشد که از سایت سازنده موبایل باید دانلود شود.

J2ME Sdk چیست ؟

شبیه سازهای موبایل برای نوشتن و توسعه برنامه و تست برنامه روی PC قبل از نصب برنامه بر روی موبایل می باشد. که این **Sdk** از وب سایت شرکت سازنده گوشی موبایل قابل دریافت است. مثلاً برای دریافت **Sdk** گوشی های Sony Ericsson به آدرس زیر مراجعه فرمائید :

http://developer.sonyericsson.com/site/global/home/p_home.jsp

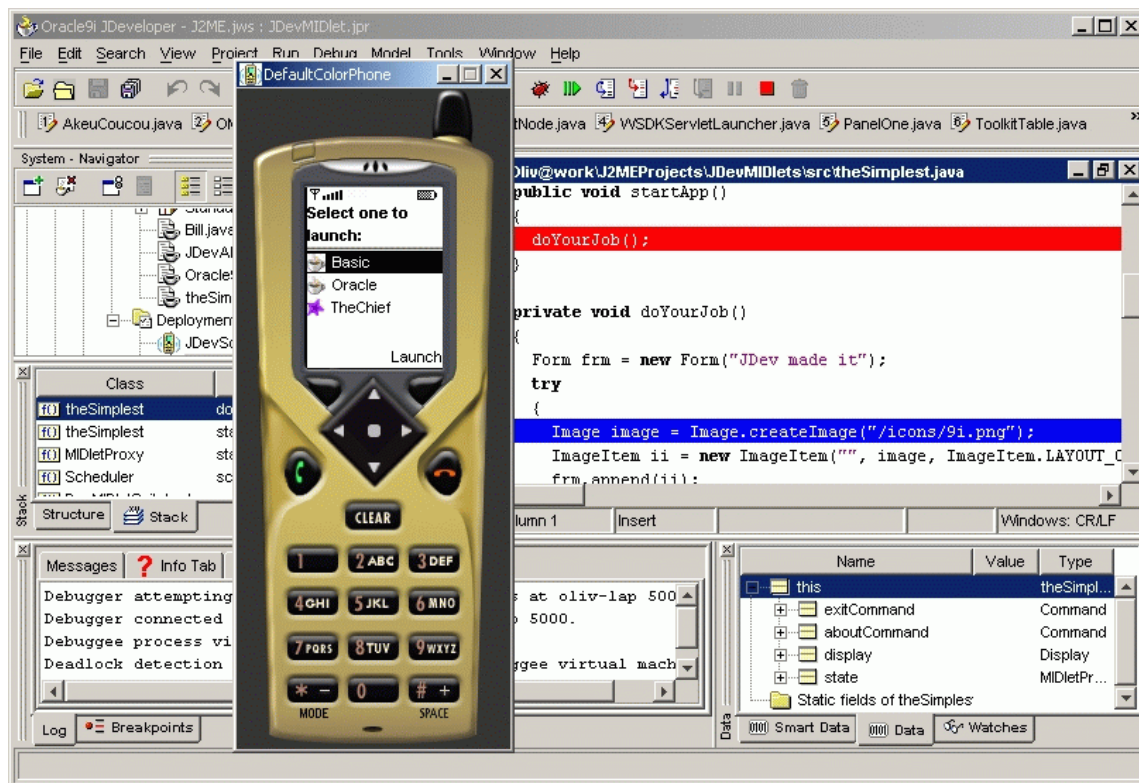
مرحله دوم :

نصب J2ME Sdk بر روی PC فایل هایی که از اینترنت دانلود می شود ابتدا باید نصب شود که شامل فایل هایی J2ME مربوط به موبایل و نیز Emulator (شبیه ساز) مربوطه می باشد.

مرحله سوم:

انتخاب IDE

بعد از نصب j2me مخصوص توسعه موبایل نوبت به محیط برنامه نویسی جهت نوشتن کد برنامه نویسی و توسعه می گردد .



IDE های موجود جهت توسعه:

- Net Beans
- JBuilder
- Oracle JDeveloper
-

چرا جاوا تکنولوژی ؟

- تنها راه برای برنامه نویسی بر روی چندین وسیله متعدد
- به اشتراک قراردادن دستگاه ها با سرور
- برنامه کاربردی پویا

چرا J2ME ؟

یک میلیارد دستگاه و زیر مجموعه های مجهز به J2ME می باشد
۳۰ الی ۵۰ درصد از دستگاه قابلیت ارتباط با اینترنت در ۲ الی ۴ سال آینده می باشد.
نرم افزار قابل نصب بر روی دستگاه های بی سیم .
فعال کردن خدمات و سفارشی کردن و تنوع بر روی دستگاه های بی سیم
J2ME یک محیط استاندارد و فارغ از تنوع سخت افزاری برای توسعه برنامه بر روی دستگاه های بی سیم
قابلیت برنامه نویسی شخصی ، بلوتوث ، بازی

پیکربندی مشخصات J2ME ؟

سایز J2ME برای تمامی گوشی های یکسان نیست
دارای حداقل کتابخانه های JVM می باشد .
J2ME پیکربندی شده است از :

- CLDC (Connected Limited Device Configuration)

- CDC (Connected Device Configuration)

J2ME نمایه ای از MIDP ()

MIDP : (Mobile Information Device Profile)

کتابخانه های اضافی برای وسیله های بی سیم (، منبع ذخیره سازی ، و غیره GUI)

:CLDC

کتابخانه های سطح پائین برای وسیله های کوچک

:KVM

موتور اجرا

اکنون دیگر میانی را به اتمام می رسانیم و کد نویسی را برای موبایل با استفاده از فایل جاوا شروع می کنیم

به چه چیزهایی برای برنامه نویسی بر روی موبایل نیاز دارید:

۱. دانلود Sdk Sony Ericsson و نصب آن بر روی PC

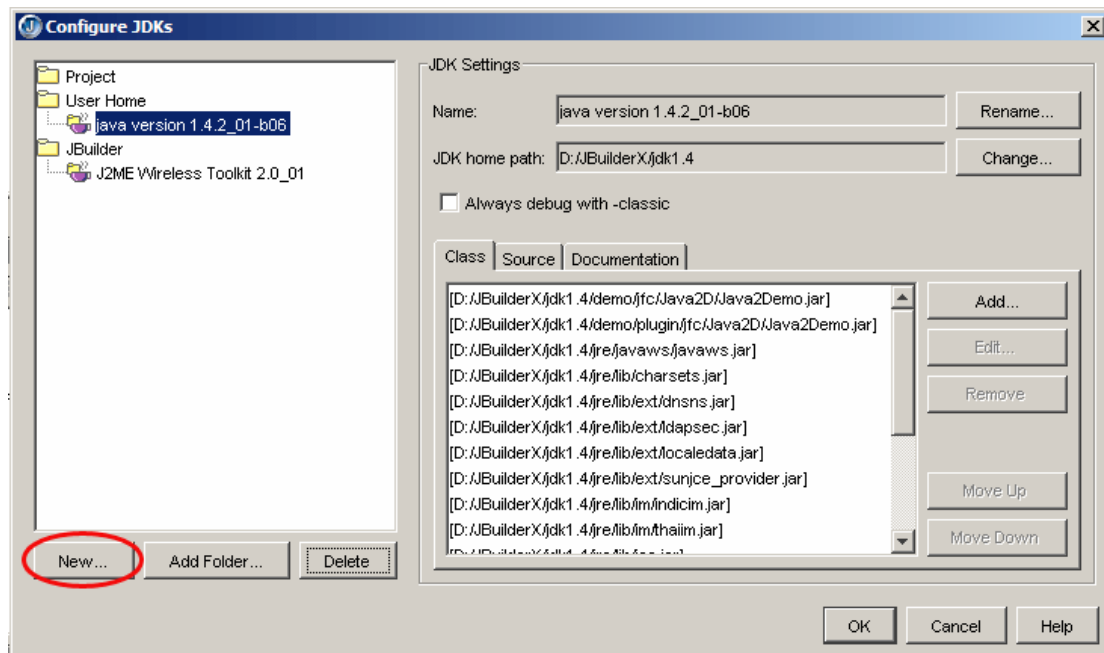
۲. نصب IDE JBuilder X یا ویرایش بالاتر

۳. آشنایی مقدماتی با برنامه نویسی به زبان جاوا و مفاهیم شی گرا

خوب من برای شروع این ۲ ابزار را جهت توسعه برنامه نویسی انتخاب کرده ام دوستان دیگر می توانند sdk موبایل خودشان را دانلود کنند. و از JBuilder جهت سهولت در برنامه نویسی

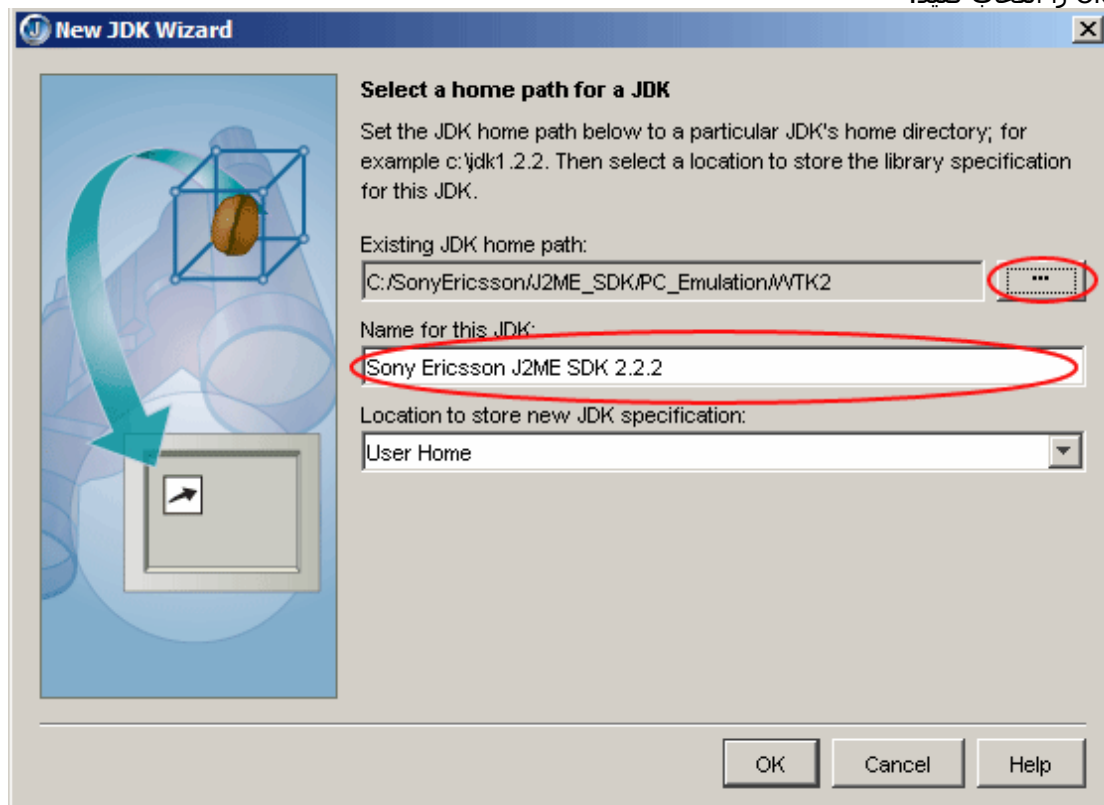
خوب اکنون به سراغ JBuilder می رویم و برای معرفی J2ME Sdk Sony Ericsson به JBuilder جهت ترجمه و اجرا به صورت زیر عمل می کنیم :

بروی منوی tools از منو بار رفته و منوی Configure jdks.. را انتخاب نمائید پنجره ای Configure jdks بر روی صفحه باز می شود برای معرفی J2ME Sdk Sony Ericsson بر روی کلید new کلیک کنید شکل ۱



شکل ۱

سپس پنجره ای به نام New JDK Wizard که از شما مسیر sdk را می خواهد. شما برای معرفی sdk جدید بر روی دکمه ... کلیک کرده و پنجره ای باز می شود که از شما مسیر sdk را می خواهد چون مسیر نصب sdk را در درایو C:\ بوده است مسیر زیر را انتخاب می کنیم:
C:/SonyEricsson/J2ME_SDK/PC_Emulation/WTK2
اگر مسیر sdk مورد نظر را درست داده باشیم در قسمت Name For this JDK نام مورد نظر را می نویسد و اگر مسیر به اشتباه یا فایل های J2ME Sdk وجود نداشته باشد یا ناقص باشد به جای اسم sdk پیغام java Unknown version ظاهر می گردد. اکنون برای بستن این پنجره کلیک OK را انتخاب کنید.



شکل ۲

نکته :
ممکت است در J2ME Sdk که نصب می کنید انواع گوشی ها دسته بندی می شود مثلا در Sdk J2ME Sony Ericsson فایلها در دو مسیر گروه بندی شده اند :

C:/SonyEricsson/J2ME_SDK/PC_Emulation/WTk1

در این مسیر مدل های گوشی :

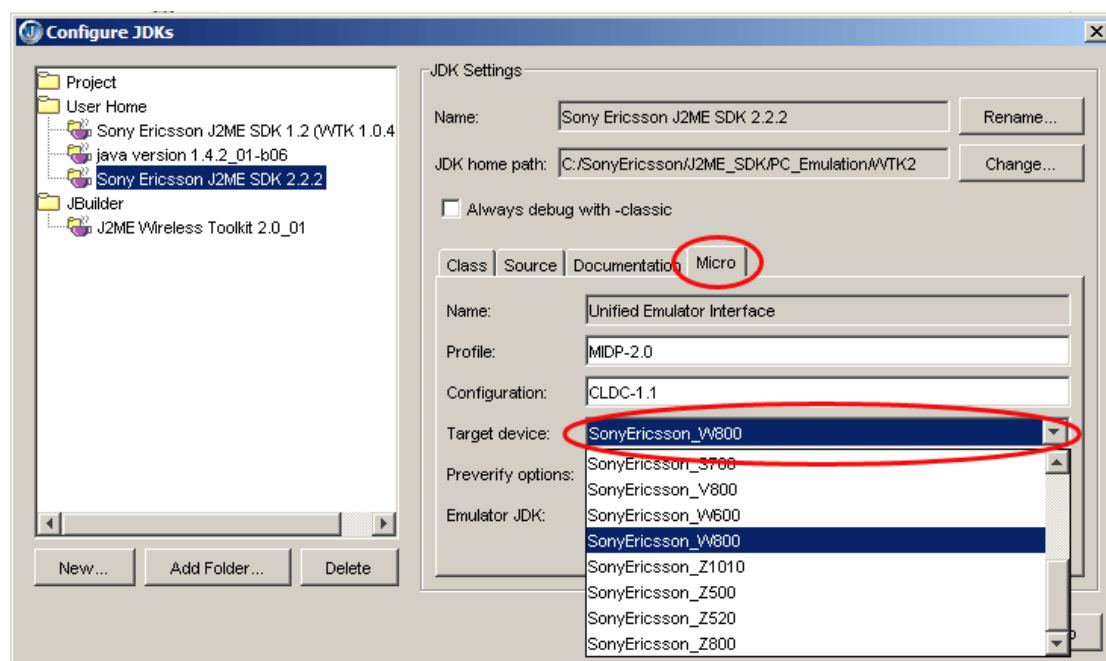
SonyEricsson_P800
SonyEricsson_T610
SonyEricsson_T616
SonyEricsson_T630
.....

و در این مسیر مدل های گوشی :

C:/SonyEricsson/J2ME_SDK/PC_Emulation/WTk2

SonyEricsson_F500i
SonyEricsson_K300
SonyEricsson_K750
.....

اکنون به قسمت آخر یعنی انتخاب شبیه ساز (emulator) می رسیم همانطور که در شکل ۳ می بینید دو sdk دیگر در قسمت چپ پنجره اضافه شده است اکنون برای انتخاب مدل گوشی مورد نظر از درخت سمت چپ J2ME مورد نظر را انتخاب کنید و از قسمت JDK Settings بر روی Micro کلیک کرده و از گزینه Target Device مدل گوشی خود را انتخاب کنید (مثلا مدل SonyEricsson_w800 را انتخاب می کنیم و در پایان بر روی کلید Ok کلیک کنید.



شکل ۳

خوب اکنون برای نوشتن برنامه HelloWorld بر روی موبایل کلیه پروژه های قبلی را ببندید و بر روی منوی File رفته و گزینه New را انتخاب کنید پنجره ای به نام Object Gallery باز می شود از قسمت چپ پنجره Tree View گزینه Project و آیکون سمت راست Project را انتخاب کنید و بر روی دکمه Ok انتخاب کنید سپس از پنجره Project Wizard در قسمت Name نام پروژه خود را انتخاب کنید و بر روی کلید Next کلیک کنید در پنجره بعدی برای انتخاب Sdk موبایل بر روی ... که

در مقابل کلمه JDK قرار گرفته کلیک کرده پنجره کوچکی به نام Select Jdks باز می شود که انواع JDK نصب شده ظاهر می گردد Sony Ericsson J2ME را انتخاب کنید و بروی کلید Next کلیک کرده در پنجره بعد از قسمت Encoding آخرین ردیف UTF8 را جهت پشتیبانی از فونت فارسی انتخاب کرده و بر روی گزینه Finish کلیک کرده اکنون پروژه ای جدید باز شده است اکنون کلاس جدیدی به نام HelloMIDlet.java ایجاد کنید. (برای ایجاد کلاس جدید بر روی منوی File و سپس گزینه new را انتخاب کنید آنگاه از پنجره Object Gallery و از سمت چپ پنجره گزینه General و از سمت راست آیکن Class را انتخاب کنید. و در قسمت Class Name نام کلاس را وارد کنید و قطعه کد زیر را درون HelloMIDlet.java بنویسید

```
1. import javax.microedition.midlet.MIDlet;
2. import javax.microedition.lcdui.Display;
3. import javax.microedition.lcdui.TextBox;

4. public class HelloMIDlet extends MIDlet {

5.     private Display display;
6.     public HelloMIDlet() {
7.         display=Display.getDisplay(this)
8.     }

9.     public void startApp() {
10.         TextBox t=new TextBox("Hello MIDlet",
11.             "Hello J2MECamp!", 256, 0);
12.         display.setCurrent(t);
13.     }

14.     public void pauseApp() {}

15.     protected void destroyApp(boolean u) {}
16. }
```

شرح کد :

سه کتابخانه در خط ۱ و ۲ و ۳ را بارگزاری کرده ایم جهت تعریف کلاس MIDlet ، Display ، TextBox

کلاس HelloMIDlet را از کلاس مجرد MIDlet به ارث می گیریم و سه متد startApp، pauseApp، destroyApp را تعریف می کنیم .

هنگامی که برنامه اجرا می شود ابتدا متد startApp فراخوانی می شود و نقطه شروع برنامه موبایل است در خط ۵ متغیری از نوع کلاس جهت نمایش تعریف شده است و در خط ۶ سازنده ای برای کلاس تعریف شده است در خط ۷ جهت ارتباط کلاس HelloMIDlet را به متغیر display ارتباط می دهد. در خط ۱۰ یک متغیر از TextBox تعریف می شود و در خط ۱۱ متغیر TextBox به گزینه display اضافه می شود

اکنون برای اجرای برنامه کلید F9 را بزنید

پنجره ای به نام Runtime Configurations باز می شود جهت معرفی کلاس اجرایی برنامه کلید New را انتخاب کنید

پنجره ای به نام New Runtime Configurations باز می شود در قسمت Name می توانید نام اجرایی برنامه را بنویسید در قسمت Type نوع MIDlet را انتخاب کنید . و در قسمت Main Class بر روی ... کلیک کنید تا پنجره ای به نام Select Main Class For Project بر روی گزینه Browse کلیک کنید نام پروژه خود را از درون Tree View پیدا کنید و بر روی علامت + در کنار نام پروژه خود کلیک کنید تا کلاس خود جهت اجرا ظاهر شود و کلاس مورد نظر را انتخاب کنید همانطور که مشاهده می کنید در قسمت Class Name نام پروژه و کلاس مورد نظر نوشته می شود mobilehello.HelloMobile و شما برای تأیید نام کلاس بر روی دکمه Ok کلیک کنید تا به پنجره

قبلی باز گردید همانطور که مشاهده می کنید در قسمت Main Class نام کلاس جهت اجرا نوشته شده است سپس دکمه ok را انتخاب کنید. اکنون در پنجره Runtime Configurations سطری اضافه شده است جهت اجرای برنامه بر روی دکمه Ok کلیک کنید . سپس یکبار دیگر جهت اجرای برنامه کلید F9 را انتخاب کنید تا برنامه اجرا گردد خوب این ساده ترین برنامه بر روی موبایل می باشد.

برنامه helloWord همراه با دکمه Exit :

اکنون پروژه دیگری اجرا نمائید و کلاس HelloWorld.java را به وجود آورید .

```
1. import javax.microedition.midlet.MIDlet;
2. import javax.microedition.lcdui.Command;
3. import javax.microedition.lcdui.CommandListener;
4. import javax.microedition.lcdui.Display;
5. import javax.microedition.lcdui.Displayable;
6. import javax.microedition.lcdui.Form;

7. public class HelloWorld extends MIDlet implements
   CommandListener{
8. private Form form;
9. public HelloWorld() {
10. // Create a new form on which to display our text
11. form = new Form("Test App");

12. // Add the text "Hello World!" to the form
13. form.append("Hello World!");
14. // Add a command button labeled "Exit"
15. form.addCommand(new Command("Exit", Command.EXIT, 1));
16. // Register this object as a commandListener
17. form.setCommandListener(this);
18. }

19. public void startApp() {
20. // Get a reference to the display, and show the form
21. Display display = Display.getDisplay(this);
22. display.setCurrent(form);
23. }

24. public void pauseApp() {}

25. public void destroyApp(boolean unconditional) {
26.     form = null;
27. }

28. public void commandAction(Command c, Displayable d) {
29. // Destroy this MIDlet
30. destroyApp(true);
31. // Notify the application management software that this MIDlet
32. // has entered the destroyed state
33. notifyDestroyed();
34. }
35. }
```

خوب اکنون می خواهیم نگاهی دقیق تر به کد بالا ببینیم در خط ۱ تا ۶ کتابخانه های مورد نظر را جهت تعریف کلاسها در برنامه نوشته شده اند. در خط ۷ کلاس HelloWorld از کلاس MIDlet ارث گرفته است و سه متد destroyApp، startApp، pauseApp، را پیاده سازی کرده است آنگاه جهت شنیدن رویداد دکمه Exit و پاسخ برنامه interface CommandListener را پیاده سازی می کنیم .

در خط ۸ متغیری از کلاس Form تعریف کردیم (همانند فرم های در برنامه کاربردی بر روی کامپیوترها هستند) و در خط ۱۲ رشته "Hello World!" را به فرم اضافه کردیم در خط ۱۴ یک دکمه به نام Exit و از نوع Command.EXIT تعریف شده است و در خط ۱۷ کلاس جاری را جهت دریافت و شنیدن رویداد دریافت کلید معرفی می نماییم. در خط ۲۸ متد commandAction جهت پاسخ به رویداد کلید Exit پیاده سازی شده است.

اکنون می خواهیم برنامه ای بنویسیم که نام و نام خانوادگی را دریافت کرده و هنگامی که بر روی دکمه Hello دکمه شود نام و نام خانوادگی همراه کلمه Hello نمایش دهد. و هنگامی که دکمه Exit انتخاب شود از برنامه خارج شود.

کلاس Form

فرم همانند یک پنجره چهارگوش است که می توان اشیای دیگری را در آن قرار داد.

کلاس Command

پر استفاده ترین کنترلرها دکمه فشاری است مولفه ای است که حاوی برجسب است و هر وقت فشار داده شود رویدادی ایجاد می کند.

کلاس TextField

کلاس TextField یک ناحیه یک سطر را برای ورود متن ایجاد می کند. این ناحیه را کنترل ویرایش می نامند. در فیلدهای متنی می توان متنی را وارد کرد. و با کلید های مکان نما در آن حرکت نمود و تغییراتی را اعمال کرد. در این کلاس نیز می توان تعریف کرد که فقط اعداد دریافت کرد.

کلاس StringItem

برجسب یک رشته است که عناصر مزیت برجسب آن است که در درون طراحی پانل می نشیند و نیازی نیست هربار آن را دوباره ترسیم کنیم.

```
import javax.microedition.midlet.MIDlet;
import javax.microedition.lcdui.Command;
import javax.microedition.lcdui.CommandListener;
import javax.microedition.lcdui.Display;
import javax.microedition.lcdui.Displayable;
import javax.microedition.lcdui.Form;
import javax.microedition.lcdui.StringItem;
import javax.microedition.lcdui.TextField;

public class HelloWorld extends MIDlet implements CommandListener{
    private Form form;
    private Command cmdExit;
    private Command cmdHello;
    private StringItem showFamily=new StringItem("", "");
    private TextField myTxt;

    public HelloWorld() {

        form = new Form("Hello Family :");
        myTxt=new TextField("Family : ", "", 15, TextField.ANY);
        form.append(myTxt);
        cmdExit =new Command("Exit" , Command.EXIT, 1);
        cmdHello=new Command("Hello", Command.OK, 1);
        form.addCommand(cmdExit);
        form.addCommand(cmdHello);
    }
}
```

```

        form.setCommandListener(this);
    }

    public void startApp() {
// Get a reference to the display, and show the form
        Display display = Display.getDisplay(this);
        display.setCurrent(form);
    }

    public void pauseApp() {}

    public void destroyApp(boolean unconditional) {
        form = null;
    }

    public void commandAction(Command c, Displayable d) {

        if (c.getCommandType()==Command.OK){
            showFamily.setText("Hello : "+myTxt.getString());
            try {
                form.append(showFamily);
            }catch(Exception ex){

            }
        }

        if (c.getCommandType()==Command.EXIT){
            destroyApp(true);
            notifyDestroyed();
        }
    }
}

```

کلاس Choice:

دکمه رادیویی نوعی جعبه CheckBox و RadioBox است که در هر لحظه فقط یکی از آنها می تواند انتخاب شده باشد.

```

public class ChoiceTest extends MIDlet
    implements CommandListener{
    private Form form=new Form("Hello Family");;
    private Command cmdExit;
    private Command cmdChoice;
    private StringItem showItem=new StringItem("", "");
    private TextField myTxt;
    private ChoiceGroup myChoice = null;

    public ChoiceTest () {
        String choices[] = {"Man", "Woman"};
        myChoice = new ChoiceGroup("Type", Choice.EXCLUSIVE,choices, null);
// myTxt=new TextField("Family : ", "",15,TextField.ANY);
// form.append(myTxt);
        cmdExit=new Command("Exit ", Command.EXIT,1);
        cmdChoice=new Command("Choice ?", Command.OK, 1);
        form.append(myChoice);
        form.addCommand(cmdExit);
    }
}

```

```

        form.addCommand(cmdHello);
        form.setCommandListener(this);
        form.append(showItem);
    }

    public void startApp() {
// Get a reference to the display, and show the form
        Display display = Display.getDisplay(this);
        display.setCurrent(form);
    }

    public void pauseApp() {}

    public void destroyApp(boolean unconditional) {
        form = null;
    }

    public void commandAction(Command c, Displayable d) {

        if (c.getCommandType()==Command.OK){
            if (myChoice.getSelectedIndex()==0)
                showItem.setText("Your choice Man  ");
            else
                showItem.setText("Your choice Woman");
        }

        if (c.getCommandType()==Command.EXIT){
            destroyApp(true);
            notifyDestroyed();
        }
    }
}

```

اکنون باهم برنامه دیگری را باهم بررسی می کنیم . برنامه زیر دو ورودی از نوع عددی دریافت می کند.(با پارامتر TextField.NUMERIC که به کلاس TextField ارسال می گردد. ورودی فقط از نوع عددی دریافت می شود). سپس با کمک کلاس ChoiceGroup دو گزینه برای برنامه تعریف می شود.که تعیین می کند برنامه این دو عدد را باهم ضرب کند یا به توان برساند.نکته دیگری که باید گفته شود این است که برای محاسبه توان از کلاسی به نام Power استفاده می کنیم که این کلاس را در برنامه تعریف می کنیم .سپس با استفاده از دکمه Calcute

```

public class Calculate extends MIDlet
    implements CommandListener{
    private Form form=new Form("Hello Family");;
    private Command cmdExit;
    private Command cmdChoice;
    private StringItem showItem=new StringItem("", "");
    private TextField numberOne;
    private TextField numberTwo;

    private ChoiceGroup myChoice = null;

    public Calculate () {
        String choices[] = {"Power", "Multiple"};
        myChoice = new ChoiceGroup("Type", Choice.EXCLUSIVE,choices, null);
        numberOne=new TextField("Number 1 : ", "",4,TextField.NUMERIC);
        numberTwo=new TextField("Number 2 : ", "",4,TextField.NUMERIC);
    }
}

```

```

// form.append(myTxt);
cmdExit =new Command("Exit " , Command.EXIT,1);
cmdChoice=new Command("Calcute", Command.OK, 1);
form.append(numberOne);
form.append(numberTwo);
form.append(myChoice);
form.addCommand(cmdExit);
form.addCommand(cmdChoice);
form.setCommandListener(this);
form.append(showItem);
}

public void startApp() {
// Get a reference to the display, and show the form
Display display = Display.getDisplay(this);
display.setCurrent(form);

}

public void pauseApp() {}

public void destroyApp(boolean unconditional) {
form = null;
}

public void commandAction(Command c, Displayable d) {

if (c.getCommandType()==Command.OK){
if (myChoice.getSelectedIndex()==0){
Power po=new Power(Integer.parseInt(numberOne.getString()),
Integer.parseInt(numberTwo.getString()));
showItem.setText("Your choice Power = "+po.pow());
}
else {
showItem.setText("Your choice Multiple =
"+(Integer.parseInt(numberOne.getString())*
Integer.parseInt(numberTwo.getString())));
}
}

if (c.getCommandType()==Command.EXIT){
destroyApp(true);
notifyDestroyed();
}
}
}

```

اکنون کلاس Power.java به شرح زیر به وجود آورید:

```

public class Power {
private int one;
private int two;

public Power() {}
public Power(int a,int b) {
one=a;
two=b;
}
}

```

```

}

public int pow(){
    int mul=1;
    for (int i=1;i<=two;i++)
        mul*=one;

    return mul;
}
}

```

گاهی ممکن است شما نیاز به ورود اطلاعات به چند سطر اطلاعات داشته باشیم جهت ورودی اطلاعات در چند سطر از کلاس TextBox استفاده می کنیم. مثال زیر چگونگی استفاده از کلاس TextBox را نمایش می دهد. نکته ای که در این برنامه باید به آن اشاره کرد این است که در این برنامه کلاس TextBox به طور مستقیم به عنوان پارامتر به کلاس Display فرستاده می شود.

```

public class TestTextBox
    extends MIDlet implements CommandListener
{
    private Display display; // Reference to Display object private
    TextBox tbMain; // Textbox to display a message
    private Command cmExit; // Button to exit the MIDlet
    public TestTextBox () {
        display = Display.getDisplay(this);
        cmExit = new Command("Exit", Command.SCREEN, 1);
        tbMain = new TextBox("Welcome", "Core J2ME", 256, 0);
        tbMain.addCommand(cmExit);
        tbMain.setCommandListener(this);
    }

    // Called by application manager to start the MIDlet.
    public void startApp() {
        display.setCurrent(tbMain);
    }

    // A required method
    public void pauseApp() {}

    // A required method
    public void destroyApp(boolean unconditional) {}

    // Check to see if Exit command was selected
    public void commandAction(Command c, Displayable s) {
        if (c == cmExit) {
            destroyApp(false);
            notifyDestroyed();
        }
    }
}
}
}

```

پایان قسمت اول

نوشته شده توسط: احسان شاهبختی
zehs_sha@yahoo.com

