import javax.microedition.lcdui.\*;  
import javax.microedition.midlet.MIDlet;

public final class ItMIDlet extends MIDlet implements CommandListener  
{  
private static final int NUM\_SIZE = 20;

private final Command exitCmd = new Command(“Exit”, Command.EXIT, 2);

private final Command calcCmd = new Command(“Calc”, Command.SCREEN, 1);

private final TextField t1 = new TextField(“Age”, “”, NUM\_SIZE, TextField.DECIMAL);

private final TextField t2 = new TextField(“Monthly Income”, “”, NUM\_SIZE, TextField.DECIMAL);

private final TextField t3 = new TextField(“Investment under 80C”, “”, NUM\_SIZE, TextField.DECIMAL);

private final TextField t4 = new TextField(“Hometown Int/Rent”, “”, NUM\_SIZE, TextField.DECIMAL);

private final TextField t5 = new TextField(“MediClaim Premiums”, “”, NUM\_SIZE, TextField.DECIMAL);

private final TextField t6 = new TextField(“Donations”, “”, NUM\_SIZE, TextField.DECIMAL);

private final TextField ti = new TextField(“Taxable Income”, “”, NUM\_SIZE, TextField.UNEDITABLE);

private final TextField tt = new TextField(“Payable Tax”, “”, NUM\_SIZE, TextField.UNEDITABLE);

private final ChoiceGroup cg =  
new ChoiceGroup(“Sex”, ChoiceGroup.POPUP,  
new String[] { “Male”, “Female” }, null);

private final Alert alert = new Alert(“Error”, “”, null, AlertType.ERROR);

private boolean isInitialized = false;

protected void startApp()  
{  
if (isInitialized)  
return;

Form f = new Form(“Income Tax Calculator”);  
f.append(t1);  
f.append(cg);  
f.append(t2);  
f.append(t3);  
f.append(t4);  
f.append(t5);  
f.append(t6);  
f.append(ti);  
f.append(tt);  
f.addCommand(exitCmd);  
f.addCommand(calcCmd);  
f.setCommandListener(this);  
Display.getDisplay(this).setCurrent(f);  
alert.addCommand(new Command(“Back”, Command.SCREEN, 1));  
isInitialized = true;  
}

protected void destroyApp(boolean unconditional)  
{  
}

protected void pauseApp()  
{  
}

public void commandAction(Command c, Displayable d)  
{  
if (c == exitCmd)  
{  
destroyApp(false);  
notifyDestroyed();  
return;  
}

double res = 0.0;  
double pt = 0.0;

try {  
double age=getNumber(t1, “First”);  
double income = getNumber(t2, “Second”);  
double invst = getNumber(t3, “Third”);  
double hometown = getNumber(t4, “Fourth”);  
double mediclaim = getNumber(t5, “Fifth”);  
double donation = getNumber(t6, “Sixth”);

if(invst<=150000) //exemption of 1.5 lakhs under investment 80C  
res=(income\*12)-invst-hometown-mediclaim-donation;  
else  
res=(income\*12)+invst-150000-hometown-mediclaim-donation;  
if(age<=60)  
{  
switch (cg.getSelectedIndex())  
{  
case 0:if(res<=200000) //for males  
pt=0.0;  
if(res>200000 && res<=300000)  
pt=(res\*10)/100;  
if (res>300000 && res<=500000)  
pt=(res\*20)/100;  
if (res>500000)  
pt=(res\*30)/100;  
break;  
case 1:if(res<=250000) //for females  
pt=0.0;  
if(res>250000 && res<=350000)  
pt=(res\*10)/100;  
if (res>350000 && res<=500000)  
pt=(res\*20)/100;  
if (res>500000)  
pt=(res\*30)/100;  
break;  
default:  
}  
}  
else  
{  
if(age<=80) //senior citizen  
{  
if(res<=300000)  
pt=0.0;  
if(res>300000 && res<=400000)  
pt=(res\*10)/100;  
if (res>400000 && res<=500000)  
pt=(res\*20)/100;  
if (res>500000)  
pt=(res\*30)/100;  
}  
else //super senior citizen  
{  
if(res<=400000)  
pt=0.0;  
if (res>400000 && res<=500000)  
pt=(res\*20)/100;  
if (res>500000)  
pt=(res\*30)/100;  
}  
}  
} catch (NumberFormatException e)  
{  
return;  
}  
catch (ArithmeticException e)  
{  
alert.setString(“Divide by zero.”);  
Display.getDisplay(this).setCurrent(alert);  
return;  
}

String res\_str = Double.toString(res); //Taxable Income

if (res\_str.length() > ti.getMaxSize()) {  
ti.setMaxSize(res\_str.length());  
}

ti.setString(res\_str);

String pt\_str; //Payable Tax

if(pt==0.0)  
pt\_str = “Tax not applicable”;  
else  
pt\_str = Double.toString(pt);

if (pt\_str.length() > tt.getMaxSize()) {  
tt.setMaxSize(pt\_str.length());  
}

tt.setString(pt\_str);  
}

private double getNumber(TextField t, String type)throws NumberFormatException  
{  
String s = t.getString();

if (s.length() == 0)  
{  
alert.setString(“No ” + type + ” Argument”);  
Display.getDisplay(this).setCurrent(alert);  
throw new NumberFormatException();  
}

double n;

try  
{  
n = Double.parseDouble(s);  
}  
catch (NumberFormatException e)  
{  
alert.setString(type + ” argument is out of range.”);  
Display.getDisplay(this).setCurrent(alert);  
throw e;  
}

return n;  
}  
} // end of class ‘ItMIDlet’ definition