

**Proiect pentru Dezvoltarea de Aplicatii pentru Telefoane Mobile**

Scientific Calculator

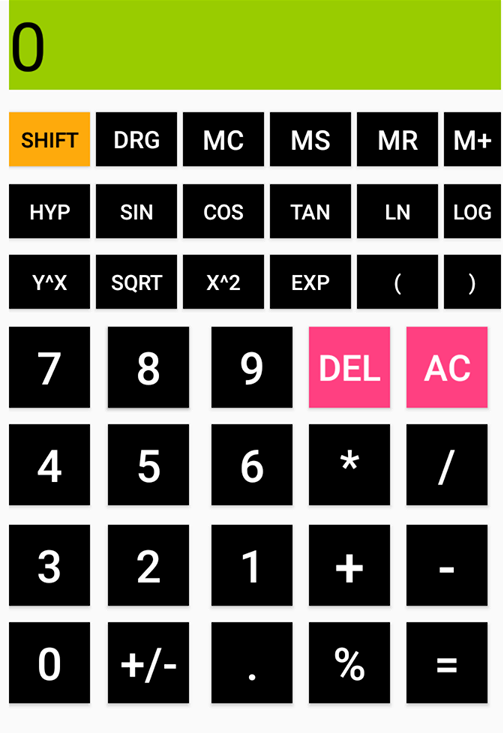
Profesor coordonator: Student:

As.dr.ing. Adriana STAN Sandy Endre

1. ***Tema proiectului***

Tema proiectului reprezinta realizarea unei aplicatii Android pentru un calculator stiintific obisnuit permitand utilizatorului sa foloseasca mai multe functii de calcul simple si complexe. Denumirea butoanelor este facut dupa modelul standard si este destul de explicit pentru utilizator. Precizia calculelor este limitata la dimensiunea ecranului fiind de 11 zecimale.

Utilizatorul vede interfata grafica la intrarea in aplicatie. Urmatoare captura de ecran include butoanele cu functii iar rezultatul final va fi afisat in dreptunghiul sus.



1. ***Secvente de cod importante***

Afisarea rezultatului in ecranul principal:

**public** **void** writeText (Double result) {

Integer integerResult;

String s;

**if**(result%1==0){

integerResult=result.intValue();

s=integerResult.toString();

}

**else**{

s = result.toString();

}

textView.setText(s.substring(0, Math.*min*(s.length(), 13)));

}

Procesarea valorii din ecranul principal:

**public** Double readText() {

**try**{

String a = textView.getText().toString();

initial = Double.*parseDouble*(a);

**return** initial;

}

**catch** (Exception e){

textView.setText("0");

paranthesisOpen=**false**;

**return** 0.0;

}

}

Tratarea evenimentelor legate de operatii de baza:

**public** **void** waitToTheNextOperand(String operationExpression){

aux=readText();

expression=operationExpression;

textView.setText("0");

}

plus.setOnClickListener(**new** View.OnClickListener() {

@Override

**public** **void** onClick(View v) {

**if**(!paranthesisOpen){

waitToTheNextOperand("plus");

dotActive=**false**;

}

**else**{

textView.setText(textView.getText()+"+");

}

}

});

Tratarea evenimentelor legate de butoane cu numere:

**public** **void** handleNumbers(**int** number){

**if**(!paranthesisOpen){

initial=readText();

text=initial.toString();

**if**(!dotActive){

result=10\*readText()+number;

}

**else**{

**if**(initial%1==0){

result=initial+number\*Math.*pow*(10, -1);

}

**else** {

text=text+String.*valueOf*(number);

result = Double.*parseDouble*(text);

}

}

writeText(result);

}

**else**{

textView.setText(textView.getText()+String.*valueOf*(number));

}

}

two.setOnClickListener(**new** View.OnClickListener() {

@Override

**public** **void** onClick(View v) {

handleNumbers(2);

}

});

Tratarea evenimentelui legate de butonul de egalizare:

**switch**(expression) {

**case** "plus":

**if**(!percentActive){

result=aux+readText();

}

**else** {

result=aux+aux\*readText()/100;

}

writeText(result);

**break**;

});

Tratarea evenimentelui inchiderii parantezei:

rightParanthesis.setOnClickListener(**new** View.OnClickListener() {

@Override

**public** **void** onClick(View v) {

**try**{

**if**(paranthesisOpen){

String complex = textView.getText().toString();

complex=complex.substring(1, complex.length());

result=*eval*(complex);

writeText(result);

Log.*i*("Value:", result.toString());

paranthesisOpen=**false**;

}

}

**catch**(Exception e) {

Log.*e*("Problem", e.getMessage());

}

}

});

Exista posibilitatea introducerii unui text mai lung care va fi dupa aia parsat si procesat de catre aplicatie. Captura de ecran cu un exemplu este prezentat mai jos.

