

**UNIVERSITATEA DIN ORADEA  
FACULTATEA IMT**

# **PROGRAMAREA CALCULATORULUI II**

## **MAPA 3P**



**Student: Student X  
Grupa: 211  
Program de studii: IA**

**2010-2011**

## CURRICULUM VITAE

### STUDENT X

Adresă: *str. Universității, nr. 1, bl. C1, et. I, ap.15, Oradea, Bihor, România*  
Telefon: 0744- 345678;  
e-mail: stud.x@yahoo.com  
Naționalitate: *română*  
Data nașterii: 17.12.1989



### Obiectiv

Promovarea examenului la PCII, anul I, semestrul II, Facultatea de IMT, Universitatea Oradea.

### Experiență Profesională

-

### Educație și Formare

2005 – 2009: Liceul Traian Vuia, Profilul Tehnician CAD  
Diploma de bacalaureat

2010-2011: Facultatea IMT, Universitatea din Oradea, Specializarea IA  
anul I

-

### Abilități și Competențe

#### Limbi străine

- limba engleză nivel mediu

-

#### Aptitudini și competențe în domeniul tehnic

- MS Office

- AutoCAD

#### Altele

- permis de conducere categoria B

### Recomandări

**Sunt disponibile la cerere.**

**CUPRINS**

Pag.

1. Programul 1: CALCULE REPER TRIUNGHILAR .....	4
2. Programul 2: MEDIA SEMESTRIALĂ .....	9
3. Programul 3: DURITĂȚI .....	14
4. Bibliografie .....	20

## PROGRAMUL 1 CALCULE REPER TRIUNGHIULAR

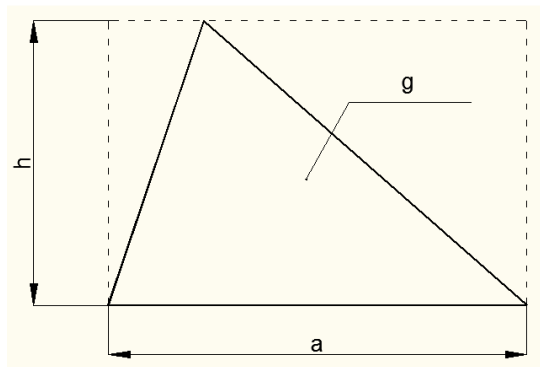
### 1. Formularea problemei

Se consideră o piesă din tablă de oțel, de formă triunghiulară, obținută dintr-un semifabricat dreptunghiular (figura de mai jos).

Se cunosc următoarele date:

- **a**, baza triunghiului, în intervalul [50,200] mm
- **h**, înălțimea triunghiului, în [mm] în intervalul [50,200] mm
- **g**, grosimea tablei, în intervalul [1,10] mm
- **gs**, greutatea specifică a oțelului  $gs=7,8$  [kg/dmc]

Să se scrie un program, în limbajul de programare C/C++, prin care se calculează **S** aria triunghiului, **V** volumul piesei, **G** greutatea piesei și **cu** coeficientul de utilizare a materialului (aria piesei/aria semifabricatului\*100 [%]).



### 2. Descrierea algoritmului

#### Variabile

**a, h, g** – de tip **float** (reale în simplă precizie), date de intrare

**gama** – constantă de tip **float**

**S, V, G, cu** – de tip **float**, date de ieșire

**cont** – variabila de tip **char**, pentru repetarea calculului (răspuns la întrebarea *Continuati?* ['D'a, 'N'u])

#### Formule

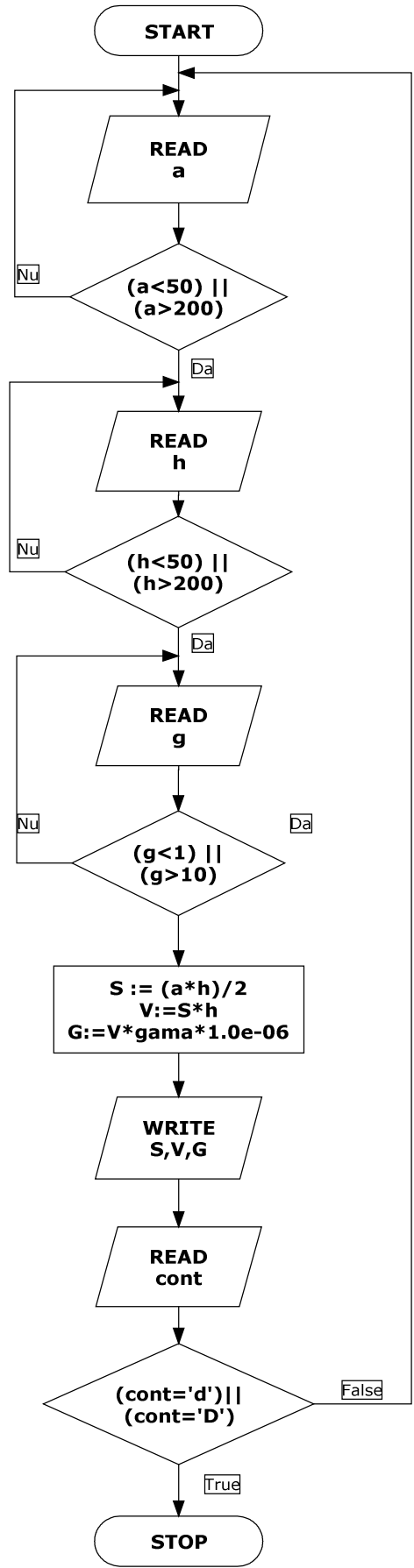
$$S = (a \cdot h) / 2$$

$$V = S \cdot g$$

$$G = V \cdot \text{gama} \cdot 1E-06$$

$$\text{cu} = S / (a \cdot h)$$

#### Schema logică



### 3. Programul sursă

```

/*
    Programul 1 (Model pentru Mapa 3P)
    Calcule reper triunghiular
    Autor: Student X
    Data: 26.04.2011
*/

#include "stdafx.h"
#include <iostream>
#include <conio.h>

using namespace std;

int main()
{
    float a,h,g;           // in [mm]
    float S,V,G,cu;
    //aria,volumul,greutatea,coeficient
    const float gama=7.8; //greutatea specifica
    [kg/dmc]
    char cont='D';        // raspuns la Continuati ?
    [Da,Nu]

    cout<<"CALCULE PENTRU UN REPER
    TRIUNGHIULAR"<<endl<<endl;    // titlu
do
{
    do
    {
        cout<<endl<<"Latura a in [50,200] = ";
        cin>>a;
        if ((a<50)|| (a>200))cout<<"Eroare! Introduceti
a din intervalul [50,100]"<<endl;
    }while ((a<50)|| (a>200)); // a este in intervalul
[50,200] mm

    do
    {
        cout<<endl<<"Inaltimea h in [50,200] = ";
        cin>>h;
        if ((h<50)|| (h>200))cout<<"Eroare! Introduceti
h din intervalul [50,100]"<<endl;
    }while ((h<50)|| (h>200)); // h este in intervalul
[50,200] mm

    do

```

```

{
    cout<<endl<<"Grosimea g in [1,10] = ";
    cin>>g;
    if ((g<1)|| (g>10))cout<<"Eroare! Introduceti g
din intervalul [1,10]"<<endl;
    }while ((g<1)|| (g>10)); // g este in intervalul
[1,10] mm

    S=(a*h)/2;
    V=S*g;
    G=V*gama*1.0e-06;
    cu=S/(a*h)*100;

    cout<<endl<<"Suprafata= "<<S;
    cout<<endl<<"Volumul= "<<V;
    cout<<endl<<"Greutatea = "<<G;
    cout<<endl<<"Coeficientul cu= "<<cu<<endl;

    cout<<endl<<"Continuati ? [Da,Nu]";
    //reluam problema ?
    cont=getch();
    cout<<endl;

}while ((cont=='d') || (cont=='D'));

    cout<<endl<<"Apasati orice tasta pentru a
termina...";
    getch();
    return 0;
}

```

#### 4. Exemplu de rulare:

CALCULE PENTRU UN REPER TRIUNGHIULAR

Latura a in [50,200] = 100

Inaltimea h in [50,200] = 150

Grosimea g in [1,10] = 4

Suprafata= 7500

Volumul= 30000

Greutatea = 0.234

Coeficientul cu= 50

Continuati ? [Da,Nu]

Apasati orice tasta pentru a termina...



## PROGRAMUL 2 MEDIA SEMESTRIALĂ

### 1. Formularea problemei

Să se întocmească programul sursă prin care se calculează media semestrială a unui student, cu următoarele cerințe:

- notele sunt introduse de la tastatură în variabila tablou **note[N]**, iar creditele fiecărei discipline sunt date în program, prin initializare la declararea variabilei tablou **credite[N]**. Numărul notelor **N** este o constantă de tip întreg;
- programul determină și afișează pe ecran media notelor ponderată cu creditele numai pentru examene promovate, nota maximă, nota minimă și numărul de examene nepromovate (note  $\leq 4$ );
- la introducerea notelor se verifică dacă valoarea este în intervalul admis  $[1,10]$ , în caz contrar semnaleză eroarea;
- se utilizează funcții utilizator pentru introducerea datelor, prelucrarea acestora și afișarea rezultatelor;
- calculele se pot relua opțional pentru alt set de note, după cum se răspunde la întrebarea "Continuați ? [d,n]".

### 2. Descrierea algoritmului

#### Constante

**N**, de tip **int**, numărul notelor

#### Variabile, funcții

**note, credite** – tablouri monodimensionale cu elemente de tip **int** (notele respectiv creditele), date de intrare

**min, max, restante** – variabile de tip **int** (nota minimă, maximă, număr de restante), date de ieșire

**media**, – de tip **float** (media semestrială), dată de ieșire

**i, suma, nr\_credite** - variabile intermediare

**cont** – variabila de tip **char**, pentru repetarea calculului (răspuns la întrebarea *Continuati?* ['D'a, 'N'u])

#### Formule

$suma = \sum note[i]*credite[i]$

$nr\_credite = \sum credite[i]$

$media = suma/nr\_credite$

### 3. Programul sursă

```
/*
    Programul 2 (Model pentru Mapa 3P)
    Calculul mediei semestriale
    Autor: Student Boboc
    Data: 5.05.2011
*/

#include "stdafx.h"
#include <stdio.h>
#include <conio.h>

#define N 9          //dimensiunea tabloului note

int note[N];        //tablou cu valorile notelor
int credite[N]={7,6,3,3,3,3,2,1,1};    //tablou cu
creditele disciplinelor
int min, max, suma=0, nr_credite=0, restante=0;
    //variabile globale
float media;

void citeste_date(void);    //prototipuri functii
utilizator
void scrie_rezultate(void);
void calculeaza_valori(void);

int main()
{
    char cont='D';    //variabila pentru reluarea
problemei

    printf("\nCALCULUL MEDIEI SEMESTRIALE\n");

do
{
    citeste_date();
    calculeaza_valori();
    scrie_rezultate();

    printf("\n\nContinuati ? [D,N]: ");
    cont=getche();

}while ((cont=='D') || (cont=='d')); //repetă problema
da/nu

    printf("\n\nApasati orice tasta pentru a
termina...");
    getch();
    return 0;
}
```

```
void citeste_date(void) // functie pentru introducerea
datelor
{
    int i;

    printf("\nIntroduceti notele:\n");

    for (i=0;i<N;i++)
    {
        do
        {
            printf("\n nota[%d]=",i+1);
            scanf("%d",&note[i]);
            if ((note[i]<0)|| (note[i]>10))
                printf("\nEroare! Nota trebuie sa fie
in intervalul [1,10]!");
            }while((note[i]<0)|| (note[i]>10)); //reia
introducerea petru valori incorecte
        }
    }

void scrie_rezultate(void) //functie pentru afisarea
rezultatelor
{
    int i;
    printf("\nNotele semestrului I:  ");
    for (i=0;i<N;i++)
    {
        printf(" %d,",note[i]);
    }

    printf("\nCreditele semestrului I:");
    for (i=0;i<N;i++)
    {
        printf(" %d,",credite[i]);
    }

    printf("\n Media=%f \n Max=%d \n Min=%d \n
Restante=%d \n" ,media,max,min,restante);
}

void calculeaza_valori(void) //functie pentru calculul
valorilor
{
    int i;

    max=note[0];
```

```
min=note[0];  
  
for (i=0;i<N;i++)  
{  
    if (note[i]>=5)  
    {  
        suma=suma+note[i]*credite[i];  
        nr_credite=nr_credite+credite[i];  
    }  
    else  
        restante=restante+1;  
  
    if (note[i]>max) (int)max=note[i];  
    if (note[i]<min) (int)min=note[i];  
}  
  
media=(float) suma/nr_credite;  
}
```

#### 4. Exemplu de rulare

##### CALCULUL MEDIEI SEMESTRIALE

Introduceti notele:

nota[1]=8

nota[2]=7

nota[3]=0

nota[4]=9

nota[5]=10

nota[6]=-1

Eroare! Nota trebuie sa fie in intervalul [1,10]!

nota[6]=7

nota[7]=4

nota[8]=8

nota[9]=5

Notele semestrului I: 8, 7, 0, 9, 10, 7, 4, 8, 5,

Creditele semestrului I: 7, 6, 3, 3, 3, 3, 2, 1, 1,

Media=7.875000

Max=10

Min=0

Restante=2

Continuati ? [D,N]: n

Apasati orice tasta pentru a termina...

## PROGRAMUL 3 DURITĂȚI

### 1. Formularea problemei

Să se întocmească programul cu interfață grafică Windows pentru calculul durității Brinell și Vickers.



### 2. Descrierea algoritmului

#### Constante

**PI** constanta 3,14...

**D** – diametrul bilei

#### Variabile, funcții

**d** – dimensiunea amprenteii pe proba de material

**float HB(float d)** – formula pentru duritatea Brinell

**float HV(float d)** – formula pentru duritatea Vickers

#### Formule

$$HB = 0,102 * \frac{2 * F}{\pi * D * (D - \sqrt{D^2 - d^2})}$$

$$F = 3000 N$$

$$D = 10 mm$$

$$HV = \frac{0,102 * F * \sin\left(\frac{136^\circ}{2}\right)}{d^2}$$

$$F = 3000 N$$

### 3. Programul sursă

#### În Form1.h

```
#pragma once

#include "h1.h"
#include "w1.h"
#include "w2.h"

namespace Programul3 {

    using namespace System;

    ...

    Form1(void)
    {
        InitializeComponent();
        //
        //TODO: Add the constructor code here
        //
        f1 = gnew w1;
        f2 = gnew h1;
        f3 = gnew w2;

    }

    ...

    private:
        /// <summary>
        /// Required designer variable.
        /// </summary>
        System::ComponentModel::Container ^components;

        private: Programul3::w1 ^ f1;
        private: Programul3::h1 ^ f2;
        private: Programul3::w2 ^ f3;

    ...

#pragma endregion
    private: System::Void
    iesireToolStripMenuItem_Click(System::Object^ sender,
    System::EventArgs^ e)
    {
        Windows::Forms::DialogResult Raspuns;

        Raspuns= MessageBox::Show("Esti sigur?",
        "Iesire",
```

```

        MessageBoxButtons::YesNo,
        MessageBoxIcon::Question,
        MessageBoxDefaultButton::Button1 );

        if( Raspuns==
Windows::Forms::DialogResult::Yes )
            this->Close();
    }
private: System::Void
despreToolStripMenuItem_Click(System::Object^ sender,
System::EventArgs^ e)
    {
        MessageBox::Show( "Programul 3\nAutor:
Student X\n(c) 2011",
                            "Despre" );
    }
private: System::Void
toolStripButton2_Click(System::Object^ sender,
System::EventArgs^ e)
    {
        Windows::Forms::DialogResult Raspuns;
        Raspuns= MessageBox::Show("Esti sigur?",
"Iesire",
        MessageBoxButtons::YesNo,
        MessageBoxIcon::Question,
        MessageBoxDefaultButton::Button1 );

        if( Raspuns==
Windows::Forms::DialogResult::Yes )
            this->Close();
    }
private: System::Void
calcule1ToolStripMenuItem_Click(System::Object^ sender,
System::EventArgs^ e)
    {
        f1->ShowDialog(this);
    }
private: System::Void
ajutorToolStripMenuItem_Click(System::Object^ sender,
System::EventArgs^ e)
    {

```



```

        f2->ShowDialog(this);
    }
private: System::Void
toolStripButton1_Click(System::Object^ sender,
System::EventArgs^ e)
    {
        f1->ShowDialog(this);
    }
private: System::Void
calcule2ToolStripMenuItem_Click(System::Object^ sender,
System::EventArgs^ e)
    {
        f3->ShowDialog(this);
    }
private: System::Void
toolStripButton3_Click(System::Object^ sender,
System::EventArgs^ e)
    {
        f3->ShowDialog(this);
    }
};
}

```

### În w1.h

```

#pragma once

#include <math.h>
#define PI 3.14159
...
#pragma endregion

private: System::Void button1_Click(System::Object^
sender, System::EventArgs^ e)
    {
        const int F=3000,D=10;
        double d,HB;
        String^ Mesaj;

        if( !Double::TryParse( textBox1->Text,d
)||
            (d<=0)|| (d>=10))
        {
            MessageBox::Show( "Trebuie introduce
valori numerice pentru d in intervalul (0,10)! ", "Eroare"
);
            textBox1->Text="";
            textBox2->Text="";
            return;
        }
    }

```

```

    }

    HB=0.102*(2*F)/(PI*D*(D-sqrt(D*D-d*d)));

    Mesaj=Convert::ToString(HB);
    textBox2->Text=Mesaj;
    }
};
}

```

### În w2.h

```

#pragma once

#include <math.h>
#define PI 3.14159
...
#pragma endregion
private: System::Void button1_Click(System::Object^
sender, System::EventArgs^ e)
{
    const int F=3000,D=10;
    double d,HV;
    String^ Mesaj;

    if( !Double::TryParse( textBox1->Text,d
)||
        (d<=0)||(d>=10))
    {
        MessageBox::Show( "Trebuie introduse
valori numerice pentru d in intervalul (0,10)! ", "Eroare"
);
        textBox1->Text="";
        textBox2->Text="";
        return;
    }

    HV=0.102*F*sin(PI*136/360)/(d*d);

    Mesaj=Convert::ToString(HV);
    textBox2->Text=Mesaj;
    }
};
}

```

## 4. Exemplu de rulare

**Duritatea Brinell**

d =       HB =

**$$HB = 0,102 * \frac{2 * F}{\pi * D * (D - \sqrt{D^2 - d^2})}$$**

**F = 3000 N**  
**D = 10 mm**

**Duritatea Vickers**

d =       HV =

**$$HV = \frac{0,102 * F * \sin\left(\frac{136^\circ}{2}\right)}{d^2}$$**

**F = 3000 N**

**Ajutor**

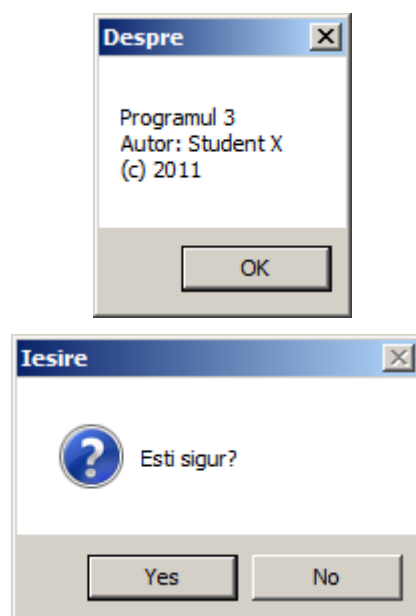
Descrierea programului

PROGRAMUL 3

Duritati->Brinell  
 Calculeaza duritatea Brinell. Se introduce d (diametrul amprenteii) si se calculeaza duritatea cu fomula afisata!

Duritati->Vickers  
 Calculeaza duritatea Vickers. Se introduce d (diagonala amprenteii) si se calculeaza duritatea cu fomula afisata!

Fisier->Iesire



## Bibliografie

1. Pele, Alexandru-Viorel, *Cursurile și aplicațiile pentru laborator*, <http://distance.iduoradea.ro>, 2011.
2. Kris Jamsa, Lars Klander, *Totul despre C și C++. Manual fundamental de programare în C și C++*, Editura Teora, București, 2004.
3. \*\*\*, *Microsoft Visual C++ 2008 Express Edition Online Help*