

ДОДАТКИ

ДОДАТОК А
ПУБЛІКАЦІЯ ЗА ТЕМОЮ РОБОТИ

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**ХАРКІВСЬКИЙ
НАЦІОНАЛЬНИЙ
УНІВЕРСИТЕТ
РАДІОЕЛЕКТРОНІКИ**

Матеріали XXVIII Міжнародного
молодіжного форуму

«Радіоелектроніка та молодь у XXI столітті»

ТОМ 7

«Комп'ютерний зір, системний
аналіз та математичне
моделювання»

Харків 2024

Гаронін В.В.

Науковий керівник – ст. викл. Штих І.А.

Харківський національний університет радіоелектроніки, каф. РТІКС,
м. Харків, Україна

e-mail: vladyslav.haronin@nure.ua

Work is devoted to creation and usage of programs that utilize geometric calculations in three dimensions. This type of programs is widely used in modern day building industry as well as game design, reason for that being that both buildings and three-dimensional game space are based on numberless combinations of simpler geometrical bodies with different sets of characteristics.

У сучасному світі все більша роль відводиться комп'ютерам і техніці в першу чергу в тій галузі, для якої перші комп'ютери були створені – обчислення.

Вибираючи більш конкретні приклади, не можна залишити без уваги геометрію і фізику, за допомогою комбінації яких колись були створені базові принципи архітектурного мистецтва.

Програми зі створення моделей, симуляції та обчислення характеристик об'єктів у тривимірному просторі при заданих їм властивостях є бажаними, а також постійно допрацьовуються різними фірмами за гроші будівельних компаній, що хочуть швидше отримувати красиві, дешеві та надійні креслення, моделі майбутніх проєктів. Прикладом є програма на рис. 2, що використовує безліч простих тривимірних об'єктів, основні види яких наведені на рис. 1. Технології на тій же основі використовують і для більш комплексних проєктів [1].

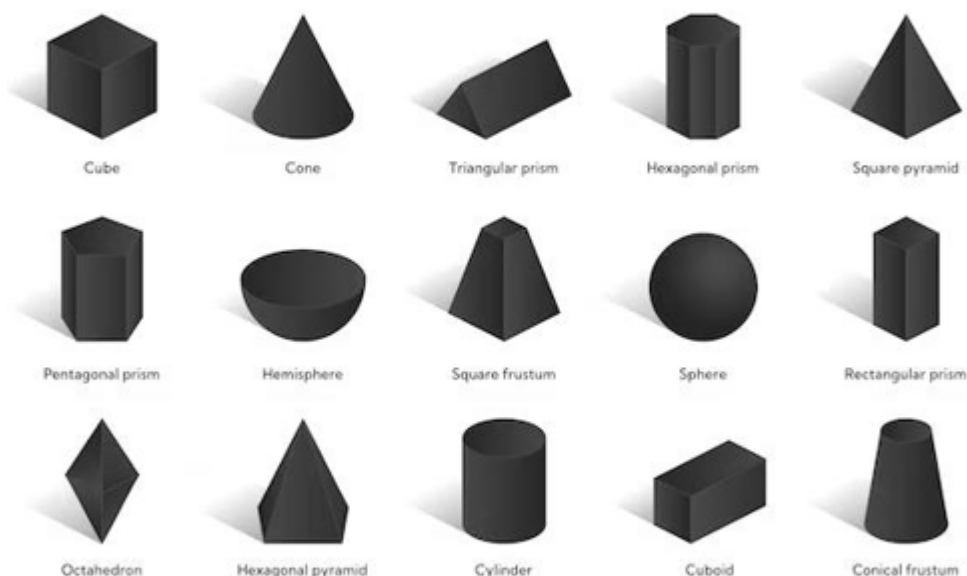


Рисунок 1 – Основні види тривимірних фігур



Г

Гаронін В.В., 176
Гвоздев М.І., 178

Кит М.О., 208
Коваль Ю.І., 210
Ковальова В.Ю., 212
Ковальова Т.Ю., 214

ДОДАТОК Б

ПОВНИЙ КОД ПРОГРАМИ

```

// Libraries
#include <iostream>
#include <math.h>

// Classes + functions

using namespace std;

class Sphere {

public:
    double MpV; /// Mass per Volume
    double R;
    double Cv;
    double Cm;
    double Csa;

    double CSa (double R){ /// Calculate Surface area
        Csa=(4.0*3.0)*3.14*pow(R,2);
        cout<<"Surface area of the Sphere="<<Csa<<endl;
        return Csa;
    }

    double CV (double R){ /// Calculate Volume
        Cv=3.14*pow(R,3);
        cout<<"Volume of the Sphere="<<Cv<<endl;
        return Cv;
    }

    double CM (double R, double MpV){ /// Calculate Mass
        Cm=3.14*pow(R,3)*MpV;
        cout<<"Mass of the Sphere="<<Cm<<endl;
        return Cm;
    }
};

class Cone {

public:

    double R;
    double H;
    double L; /// Side line
    double MpV;
    double Cv;
    double Cm;
    double Csa;

    double CSa(double L,double R){
        Csa=3.14*(L*R)+3.14*(pow(R,2));
    }
};

```

```

        cout<<"Surface area of the Cone="<<Csa<<endl;
        return Csa;
    }

    double CV (double R, double H){
        Cv=0.33*3.14*(pow(R,2))*H;
        cout<<"Volume of the Cone="<<Cv<<endl;
        return Cv;
    }

    double CM (double R, double H,double MpV){
        Cm=0.33*3.14*(pow(R,2))*H*MpV;
        cout<<"Mass of the Cone="<<endl;
        return Cm;
    }
};
class Cylindber {

    public:

    double R;
    double H;
    double MpV;
    double Cv;
    double Cm;
    double Csa;

    double CSa(double R,double H){
        Csa=2*((3.14*R*R)+3.14*R*H);
        cout<<"Surface area of the Cylinder="<<Csa<<endl;
        return Csa;
    }

    double CV (double R,double H){
        Cv=3.14*R*R*H;
        cout<<"Volume of the Cylinder="<<Cv<<endl;
        return Cv;
    }

    double CM (double R,double H,double MpV){
        Cm=3.14*R*R*H*MpV;
        cout<<"Mass of the Cylinder="<<Cm<<endl;
        return Cm;
    }
};
class Cuboid {

    public:

    double S1;

```

```

double S2;
double S3;
double MpV;
double Cv;
double Cm;
double Csa;

double CSa(double S1,double S2,double S3){
    Csa=2*((S1*S2)+(S2*S3)+(S1*S3));
    cout<<"Surface area of the a Cuboid="<<Csa<<endl;
    return Csa;
}

double CV (double S1,double S2,double S3){
    Cv=S1*S2*S3;
    cout<<"Volume of the Cuboid="<<Cv<<endl;
    return Cv;
}

double CM (double S1,double S2,double S3,double MpV){
    Cm=S1*S2*S3*MpV;
    cout<<"Mass of the Cuboid="<<Cm<<endl;
    return Cm;
}
};
class Cube {

public:

double S1;
double MpV;
double Cv;
double Cm;
double Csa;

double CSa(double S1){
    Csa=6*S1*S1;
    cout<<"Surface are of the Cube="<<Csa<<endl;
    return Csa;
}

double CV (double S1){
    Cv=S1*S1*S1;
    cout<<"Volume of the Cube="<<Cv<<endl;
    return Cv;
}

double CM (double S1,double MpV){
    Cm=S1*S1*S1*MpV;

```

```

        cout<<"Mass of the Cube="<<Cm<<endl;
        return Cm;
    }
};
class Pyramid {

    public:

    double S1;
    double H;
    double S2;
    double MpV;
    double Cv;
    double Cm;
    double Csa;

    double CSA(double S1,double S2,double H){
        Csa=2*S1*S2+S1*S1;
        cout<<"Surface area of the Pyramid="<<Csa<<endl;
        return Csa;
    }

    double CV (double S1,double H){
        Cv=0.33*S1*S1*H;
        cout<<"Volume of the Pyramid="<<Cv<<endl;
        return Cv;
    }

    double CM (double S1,double H,double MpV){
        Cm=0.33*S1*S1*H*MpV;
        cout<<"Mass of the Pyramid="<<Cm<<endl;
        return Cm;
    }
};
class Prism {

    public:

    double S1;
    double S2;
    double S3;
    double H;
    double MpV;
    double Cv;
    double Cm;
    double Csa;

    double CSA(double S1,double S2,double S3,double H){
        Csa=(S1*H)+(S2*S3*2)+(S1*S2);
        cout<<"Surface area of the Prism ="<<Csa<<endl;
    }
};

```

```

        return Csa;
    }

    double CV (double S1,double S2,double H){
        Cv=S1*H*S2*0.5;
        cout<<"Volume of the Prism ="<<Cv<<endl;
        return Cv;
    }

    double CM (double S1,double S2,double H,double MpV){
        Cm=S1*H*S2*0.5*MpV;
        cout<<"Mass of the Prism ="<<Cm<<endl;
        return Cm;
    }
};

class Hemisphere {

    public:

    double R;
    double MpV;
    double Cv;
    double Cm;
    double Csa;

    double CSa(double R){
        Csa=3*(3.14)*R*R;
        cout<<"Surface area of the Hemisphere="<<Csa<<endl;
        return Csa;
    }

    double CV (double R){
        Cv=0.66*(3.14)*R*R*R;
        cout<<"Volume of the Hemisphere="<<Cv<<endl;
        return Cv;
    }

    double CM (double R, double MpV){
        Cm=0.66*3.14*R*R*R*MpV;
        cout<<"Mass of the Hemisphere="<<Cm<<endl;
        return Cm;
    }
};

double comp2 (double n, double n1){
    if ( n>n1 ){
        return n;
    }
    if ( n1>n ){
        return n1;
    }
}

```

```

    }
};
double comp3 (double n, double n1, double n2){
    if ( n>n1 && n>n2 ){
        return n;
    }
    if ( n1>n && n1>n2 ){
        return n1;
    }
    if ( n2>n && n2>n1 ){
        return n2;
    }
};
double comp4 (double n, double n1, double n2, double n3){
    if ( n>n1 && n>n2 && n>n3){
        return n;
    }
    if ( n1>n && n1>n2 && n1>n3){
        return n1;
    }
    if ( n2>n && n2>n1 && n2>n3){
        return n2;
    }
    if ( n3>n && n3>n1 && n3>n2){
        return n3;
    }
};
double En (double n){
    double m=n;
    double c=(3*pow(10,8));
    double E=m*pow(c,2);
    cout<<"Mass of the object ="<<m<<endl;
    cout<<"Energy of the object ="<<E<<endl;
    return E;
}
void restart (){
    cout<<"Restarting the program"<<endl;
    std::exit(42);
}

int main (){

    /// Setting of

    int C1=0;
    int C2=0;
    cout<<"Choice of the number of figures created - "<<endl;
    cout<<"0 to 3, to create 1-4 figures"<<endl;
    cin>>C2; /// from 0 to 3
    cout<<endl;

```

```

cout<<"Figure type options:"<<endl;
cout<<"0->Sphere"<<endl;
cout<<"1->Cube"<<endl;
cout<<"2->Cuboid"<<endl;
cout<<"3->Prism"<<endl;
cout<<"4->Hemisphere"<<endl;
cout<<"5->Pyramid"<<endl;
cout<<"6->Cone"<<endl;
cout<<"7->Cylinder"<<endl;
cout<<endl;

/// Object creation segment

double Csa0,Csa1,Csa2,Csa3;
double Cv0,Cv1,Cv2,Cv3;
double Cm0,Cm1,Cm2,Cm3;
double Ch0,Ch1,Ch2,Ch3;
double Cw0,Cw1,Cw2,Cw3;

for (int i=0; i<=C2; i++){
    cout<<"Choice of the figure type created - "<<endl;
    cin>>C1; /// from 0 to 7
    if (C1==0 && i==0){
        Sphere T;
        cout<<"Radius of the Sphere=";
        cin>>T.R;
        cout<<endl;
        cout<<"Mass per volume of the Sphere=";
        cin>>T.MpV;
        cout<<endl;
        cout<<"Charactersitics of the Sphere"<<endl;
        Csa0=T.CSa(T.R);
        Cv0=T.CV(T.R);
        Cm0=T.CM(T.R,T.MpV);
        Ch0=T.R*2;
        Cw0=T.R*2;
    }
    if (C1==0 && i==1){ /// sphere 2
        Sphere T1;
        cout<<"Radius of the Sphere=";
        cin>>T1.R;
        cout<<endl;
        cout<<"Mass per volume of the Sphere=";
        cin>>T1.MpV;
        cout<<endl;
        cout<<"Charactersitics of the Sphere"<<endl;
        Csa1=T1.CSa(T1.R);
        Cv1=T1.CV(T1.R);
        Cm1=T1.CM(T1.R,T1.MpV);
        Ch1=T1.R*2;
    }
}

```

```

        Cw1=T1.R*2;
    }
    if (C1==0 && i==2){ /// sphere 3
        Sphere T2;
        cout<<"Radius of the Sphere=";
        cin>>T2.R;
        cout<<endl;
        cout<<"Mass per volume of the Sphere=";
        cin>>T2.MpV;
        cout<<endl;
        cout<<"Charactersitics of the Sphere"<<endl;
        Csa2=T2.CSa(T2.R);
        Cv2=T2.CV(T2.R);
        Cm2=T2.CM(T2.R,T2.MpV);
        Ch2=T2.R*2;
        Cw2=T2.R*2;
    }
    if (C1==0 && i==3){ /// sphere 4
        Sphere T3;
        cout<<"Radius of the Sphere=";
        cin>>T3.R;
        cout<<endl;
        cout<<"Mass per volume of the Sphere=";
        cin>>T3.MpV;
        cout<<endl;
        cout<<"Charactersitics of the Sphere"<<endl;
        Csa3=T3.CSa(T3.R);
        Cv3=T3.CV(T3.R);
        Cm3=T3.CM(T3.R,T3.MpV);
        Ch3=T3.R*2;
        Cw3=T3.R*2;
    }
    if (C1==1 && i==0){
        Cube T;
        cout<<"Side of the Cube=";
        cin>>T.S1;
        cout<<endl;
        cout<<"Mass per volume of the Cube=";
        cin>>T.MpV;
        cout<<endl;
        cout<<"Charactersitics of the Cube"<<endl;
        Csa0=T.CSa(T.S1);
        Cv0=T.CV(T.S1);
        Cm0=T.CM(T.S1,T.MpV);
        Ch0=T.S1;
        Cw0=T.S1;
    }
    if (C1==1 && i==1){
        Cube T1;
        cout<<"Side of the Cube=";

```

```

cin>>T1.S1;
cout<<endl;
cout<<"Mass per volume of the Cube=";
cin>>T1.MpV;
cout<<endl;
cout<<"Charactersitics of the Cube"<<endl;
Csa1=T1.CSa(T1.S1);
Cv1=T1.CV(T1.S1);
Cm1=T1.CM(T1.S1,T1.MpV);
Ch1=T1.S1;
Cw1=T1.S1;
}
if (C1==1 && i==2){
    Cube T2;
    cout<<"Side of the Cube=";
    cin>>T2.S1;
    cout<<endl;
    cout<<"Mass per volume of the Cube=";
    cin>>T2.MpV;
    cout<<endl;
    cout<<"Charactersitics of the Cube"<<endl;
    Csa2=T2.CSa(T2.S1);
    Cv2=T2.CV(T2.S1);
    Cm2=T2.CM(T2.S1,T2.MpV);
    Ch2=T2.S1;
    Cw2=T2.S1;
}
if (C1==1 && i==3){
    Cube T3;
    cout<<"Side of the Cube=";
    cin>>T3.S1;
    cout<<endl;
    cout<<"Mass per volume of the Cube=";
    cin>>T3.MpV;
    cout<<endl;
    cout<<"Charactersitics of the Cube"<<endl;
    Csa3=T3.CSa(T3.S1);
    Cv3=T3.CV(T3.S1);
    Cm3=T3.CM(T3.S1,T3.MpV);
    Ch3=T3.S1;
    Cw3=T3.S1;
}
if (C1==2 && i==0){
    Cuboid T;
    cout<<"First side of the Cuboid=";
    cin>>T.S1;
    cout<<endl;
    cout<<"Second side of the Cuboid=";
    cin>>T.S2;
    cout<<endl;

```

```

        cout<<"Third side of the Cuboid=";
        cin>>T.S3;
        cout<<endl;
        cout<<"Mass per volume of the Cuboid=";
        cin>>T.MpV;
        cout<<endl;
        cout<<"Charactersitics of the Cuboid"<<endl;
        Csa0=T.CSa(T.S1,T.S2,T.S3);
        Cv0=T.CV(T.S1,T.S2,T.S3);
        Cm0=T.CM(T.S1,T.S2,T.S3,T.MpV);
        Ch0=T.S1;
        Cw0=T.S2;
    }
    if (C1==2 && i==1){
        Cuboid T1;
        cout<<"First side of the Cuboid=";
        cin>>T1.S1;
        cout<<endl;
        cout<<"Second side of the Cuboid=";
        cin>>T1.S2;
        cout<<endl;
        cout<<"Third side of the Cuboid=";
        cin>>T1.S3;
        cout<<endl;
        cout<<"Mass per volume of the Cuboid=";
        cin>>T1.MpV;
        cout<<endl;
        cout<<"Charactersitics of the Cuboid"<<endl;
        Csa1=T1.CSa(T1.S1,T1.S2,T1.S3);
        Cv1=T1.CV(T1.S1,T1.S2,T1.S3);
        Cm1=T1.CM(T1.S1,T1.S2,T1.S3,T1.MpV);
        Ch1=T1.S1;
        Cw1=T1.S2;
    }
    if (C1==2 && i==2){
        Cuboid T2;
        cout<<"First side of the Cuboid=";
        cin>>T2.S1;
        cout<<endl;
        cout<<"Second side of the Cuboid=";
        cin>>T2.S2;
        cout<<endl;
        cout<<"Third side of the Cuboid=";
        cin>>T2.S3;
        cout<<endl;
        cout<<"Mass per volume of the Cuboid=";
        cin>>T2.MpV;
        cout<<endl;
        cout<<"Charactersitics of the Cuboid"<<endl;
        Csa2=T2.CSa(T2.S1,T2.S2,T2.S3);
    }
}

```

```

    Cv2=T2.CV(T2.S1,T2.S2,T2.S3);
    Cm2=T2.CM(T2.S1,T2.S2,T2.S3,T2.MpV);
    Ch2=T2.S1;
    Cw2=T2.S2;
}
if (C1==2 && i==3){
    Cuboid T3;
    cout<<"First side of the Cuboid=";
    cin>>T3.S1;
    cout<<endl;
    cout<<"Second side of the Cuboid=";
    cin>>T3.S2;
    cout<<endl;
    cout<<"Third side of the Cuboid=";
    cin>>T3.S3;
    cout<<endl;
    cout<<"Mass per volume of the Cuboid=";
    cin>>T3.MpV;
    cout<<endl;
    cout<<"Charactersitics of the Cuboid"<<endl;
    Csa3=T3.CSa(T3.S1,T3.S2,T3.S3);
    Cv3=T3.CV(T3.S1,T3.S2,T3.S3);
    Cm3=T3.CM(T3.S1,T3.S2,T3.S3,T3.MpV);
    Ch3=T3.S1;
    Cw3=T3.S2;
}
if (C1==3 && i==0){
    Prism T;
    cout<<"Side of the Prism=";
    cin>>T.S1;
    cout<<endl;
    cout<<"Second side of the Prism=";
    cin>>T.S2;
    cout<<endl;
    cout<<"Third side of the Prism=";
    cin>>T.S3;
    cout<<endl;
    cout<<"Height of the Prism=";
    cin>>T.H;
    cout<<endl;
    cout<<"Mass per volume of the Prism=";
    cin>>T.MpV;
    cout<<"Charactersitics of the Prism"<<endl;
    Csa0=T.CSa(T.S1,T.S2,T.S3,T.H);
    Cv0=T.CV(T.S1,T.S2,T.H);
    Cm0=T.CM(T.S1,T.S2,T.H,T.MpV);
    Ch0=T.H;
    Cw0=T.S1;
}
if (C1==3 && i==1){

```

```

Prism T1;
cout<<"Side of the Prism=";
cin>>T1.S1;
cout<<endl;
cout<<"Second side of the Prism=";
cin>>T1.S2;
cout<<endl;
cout<<"Third side of the Prism=";
cin>>T1.S3;
cout<<endl;
cout<<"Height of the Prism=";
cin>>T1.H;
cout<<endl;
cout<<"Mass per volume of the Prism=";
cin>>T1.MpV;
cout<<"Charactersitics of the Prism"<<endl;
Csa1=T1.CSa(T1.S1,T1.S2,T1.S3,T1.H);
Cv1=T1.CV(T1.S1,T1.S2,T1.H);
Cm1=T1.CM(T1.S1,T1.S2,T1.H,T1.MpV);
Ch1=T1.H;
Cw1=T1.S1;
}
if (C1==3 && i==2){
    Prism T2;
    cout<<"Side of the Prism=";
    cin>>T2.S1;
    cout<<endl;
    cout<<"Second side of the Prism=";
    cin>>T2.S2;
    cout<<endl;
    cout<<"Third side of the Prism=";
    cin>>T2.S3;
    cout<<endl;
    cout<<"Height of the Prism=";
    cin>>T2.H;
    cout<<endl;
    cout<<"Mass per volume of the Prism=";
    cin>>T2.MpV;
    cout<<"Charactersitics of the Prism"<<endl;
    Csa2=T2.CSa(T2.S1,T2.S2,T2.S3,T2.H);
    Cv2=T2.CV(T2.S1,T2.S2,T2.H);
    Cm2=T2.CM(T2.S1,T2.S2,T2.H,T2.MpV);
    Ch2=T2.H;
    Cw2=T2.S1;
}
if (C1==3 && i==3){
    Prism T3;
    cout<<"Side of the Prism=";
    cin>>T3.S1;
    cout<<endl;

```

```

        cout<<"Second side of the Prism=";
        cin>>T3.S2;
        cout<<endl;
        cout<<"Third side of the Prism=";
        cin>>T3.S3;
        cout<<endl;
        cout<<"Height of the Prism=";
        cin>>T3.H;
        cout<<endl;
        cout<<"Mass per volume of the Prism=";
        cin>>T3.MpV;
        cout<<"Charactersitics of the Prism"<<endl;
        Csa3=T3.CSa(T3.S1,T3.S2,T3.S3,T3.H);
        Cv3=T3.CV(T3.S1,T3.S2,T3.H);
        Cm3=T3.CM(T3.S1,T3.S2,T3.H,T3.MpV);
        Ch3=T3.H;
        Cw3=T3.S1;
    }

    if (C1==4 && i==0){
        Hemisphere T;
        cout<<"Radius of the Hemisphere=";
        cin>>T.R;
        cout<<endl;
        cout<<"Mass per volume of the Hemisphere=";
        cin>>T.MpV;
        cout<<endl;
        cout<<"Charactersitics of the Hemisphere"<<endl;
        Csa0=T.CSa(T.R);
        Cv0=T.CV(T.R);
        Cm0=T.CM(T.R,T.MpV);
        Ch0=T.R;
        Cw0=T.R*2;
    }

    if (C1==4 && i==1){
        Hemisphere T1;
        cout<<"Radius of the Hemisphere=";
        cin>>T1.R;
        cout<<endl;
        cout<<"Mass per volume of the Hemisphere=";
        cin>>T1.MpV;
        cout<<endl;
        cout<<"Charactersitics of the Hemisphere"<<endl;
        Csa1=T1.CSa(T1.R);
        Cv1=T1.CV(T1.R);
        Cm1=T1.CM(T1.R,T1.MpV);
        Ch1=T1.R;
        Cw1=T1.R*2;
    }

    if (C1==4 && i==2){

```

```

Hemisphere T2;
cout<<"Radius of the Hemisphere=";
cin>>T2.R;
cout<<endl;
cout<<"Mass per volume of the Hemisphere=";
cin>>T2.MpV;
cout<<endl;
cout<<"Charactersitics of the Hemisphere"<<endl;
Csa2=T2.CSa(T2.R);
Cv2=T2.CV(T2.R);
Cm2=T2.CM(T2.R,T2.MpV);
Ch2=T2.R;
Cw2=T2.R*2;
}
if (C1==4 && i==3){
    Hemisphere T3;
    cout<<"Radius of the Hemisphere=";
    cin>>T3.R;
    cout<<endl;
    cout<<"Mass per volume of the Hemisphere=";
    cin>>T3.MpV;
    cout<<endl;
    cout<<"Charactersitics of the Hemisphere"<<endl;
    Csa3=T3.CSa(T3.R);
    Cv3=T3.CV(T3.R);
    Cm3=T3.CM(T3.R,T3.MpV);
    Ch3=T3.R;
    Cw3=T3.R*2;
}

if (C1==5 && i==0){
    Pyramid T;
    cout<<"Side of the Pyramid=";
    cin>>T.S1;
    cout<<endl;
    cout<<"Second side of the Pyramid=";
    cin>>T.S2;
    cout<<endl;
    cout<<"Height of the Pyramid=";
    cin>>T.H;
    cout<<endl;
    cout<<"Mass per volume of the Pyramid=";
    cin>>T.MpV;
    cout<<"Charactersitics of the Pyramid"<<endl;
    Csa0=T.CSa(T.S1,T.S2,T.H);
    Cv0=T.CV(T.S1,T.H);
    Cm0=T.CM(T.S1,T.H,T.MpV);
    Ch0=T.H;
    Cw0=T.S1;
}

```

```

if (C1==5 && i==1){
    Pyramid T1;
    cout<<"Side of the Pyramid=";
    cin>>T1.S1;
    cout<<endl;
    cout<<"Second side of the Pyramid=";
    cin>>T1.S2;
    cout<<endl;
    cout<<"Height of the Pyramid=";
    cin>>T1.H;
    cout<<endl;
    cout<<"Mass per volume of the Pyramid=";
    cin>>T1.MpV;
    cout<<"Charactersitics of the Pyramid"<<endl;
    Csa1=T1.CSa(T1.S1,T1.S2,T1.H);
    Cv1=T1.CV(T1.S1,T1.H);
    Cm1=T1.CM(T1.S1,T1.H,T1.MpV);
    Ch1=T1.H;
    Cw1=T1.S1;
}
if (C1==5 && i==2){
    Pyramid T2;
    cout<<"Side of the Pyramid=";
    cin>>T2.S1;
    cout<<endl;
    cout<<"Second side of the Pyramid=";
    cin>>T2.S2;
    cout<<endl;
    cout<<"Height of the Pyramid=";
    cin>>T2.H;
    cout<<endl;
    cout<<"Mass per volume of the Pyramid=";
    cin>>T2.MpV;
    cout<<"Charactersitics of the Pyramid"<<endl;
    Csa2=T2.CSa(T2.S1,T2.S2,T2.H);
    Cv2=T2.CV(T2.S1,T2.H);
    Cm2=T2.CM(T2.S1,T2.H,T2.MpV);
    Ch2=T2.H;
    Cw2=T2.S1;
}
if (C1==5 && i==3){
    Pyramid T3;
    cout<<"Side of the Pyramid=";
    cin>>T3.S1;
    cout<<endl;
    cout<<"Second side of the Pyramid=";
    cin>>T3.S2;
    cout<<endl;
    cout<<"Height of the Pyramid=";
    cin>>T3.H;
}

```

```

        cout<<endl;
        cout<<"Mass per volume of the Pyramid=";
        cin>>T3.MpV;
        cout<<"Charactersitics of the Pyramid"<<endl;
        Csa3=T3.CSa(T3.S1,T3.S2,T3.H);
        Cv3=T3.CV(T3.S1,T3.H);
        Cm3=T3.CM(T3.S1,T3.H,T3.MpV);
        Ch3=T3.H;
        Cw3=T3.S1;
    }

    if (C1==6 && i==0){
        Cone T;
        cout<<"Side of the Cone=";
        cin>>T.L;
        cout<<endl;
        cout<<"Radius of the Cone=";
        cin>>T.R;
        cout<<endl;
        cout<<"Height of the Cone=";
        cin>>T.H;
        cout<<endl;
        cout<<"Mass per volume of the Cone=";
        cin>>T.MpV;
        cout<<endl;
        cout<<"Charactersitics of the Cone"<<endl;
        Csa0=T.CSa(T.L,T.H);
        Cv0=T.CV(T.R,T.H);
        Cm0=T.CM(T.R,T.H,T.MpV);
        Ch0=T.H;
        Cw0=T.R*2;
    }
    if (C1==6 && i==1){
        Cone T1;
        cout<<"Side of the Cone=";
        cin>>T1.L;
        cout<<endl;
        cout<<"Radius of the Cone=";
        cin>>T1.R;
        cout<<endl;
        cout<<"Height of the Cone=";
        cin>>T1.H;
        cout<<endl;
        cout<<"Mass per volume of the Cone=";
        cin>>T1.MpV;
        cout<<endl;
        cout<<"Charactersitics of the Cone"<<endl;
        Csa1=T1.CSa(T1.L,T1.H);
        Cv1=T1.CV(T1.R,T1.H);
        Cm1=T1.CM(T1.R,T1.H,T1.MpV);
    }

```

```

        Ch1=T1.H;
        Cw1=T1.R*2;
    }
    if (C1==6 && i==2){
        Cone T2;
        cout<<"Side of the Cone=";
        cin>>T2.L;
        cout<<endl;
        cout<<"Radius of the Cone=";
        cin>>T2.R;
        cout<<endl;
        cout<<"Height of the Cone=";
        cin>>T2.H;
        cout<<endl;
        cout<<"Mass per volume of the Cone=";
        cin>>T2.MpV;
        cout<<endl;
        cout<<"Charactersitics of the Cone"<<endl;
        Csa2=T2.CSa(T2.L,T2.H);
        Cv2=T2.CV(T2.R,T2.H);
        Cm2=T2.CM(T2.R,T2.H,T2.MpV);
        Ch2=T2.H;
        Cw2=T2.R*2;
    }
    if (C1==6 && i==3){
        Cone T3;
        cout<<"Side of the Cone=";
        cin>>T3.L;
        cout<<endl;
        cout<<"Radius of the Cone=";
        cin>>T3.R;
        cout<<endl;
        cout<<"Height of the Cone=";
        cin>>T3.H;
        cout<<endl;
        cout<<"Mass per volume of the Cone=";
        cin>>T3.MpV;
        cout<<endl;
        cout<<"Charactersitics of the Cone"<<endl;
        Csa3=T3.CSa(T3.L,T3.H);
        Cv3=T3.CV(T3.R,T3.H);
        Cm3=T3.CM(T3.R,T3.H,T3.MpV);
        Ch3=T3.H;
        Cw3=T3.R*2;
    }

    if (C1==7 && i==0){
        Cylinbder T;
        cout<<"Radius of the Cylinder=";
        cin>>T.R;
    }

```

```

cout<<endl;
cout<<"Height of the Cylinder=";
cin>>T.H;
cout<<endl;
cout<<"Mass per volume of the Cylinder=";
cin>>T.MpV;
cout<<endl;
cout<<"Charactersitics of the Cylinder"<<endl;
Csa0=T.CSa(T.R,T.H);
Cv0=T.CV(T.R,T.H);
Cm0=T.CM(T.R,T.H,T.MpV);
Ch0=T.H;
Cw0=T.R*2;
}
if (C1==7 && i==1){
Cylindber T1;
cout<<"Radius of the Cylinder=";
cin>>T1.R;
cout<<endl;
cout<<"Height of the Cylinder=";
cin>>T1.H;
cout<<endl;
cout<<"Mass per volume of the Cylinder=";
cin>>T1.MpV;
cout<<endl;
cout<<"Charactersitics of the Cylinder"<<endl;
Csa1=T1.CSa(T1.R,T1.H);
Cv1=T1.CV(T1.R,T1.H);
Cm1=T1.CM(T1.R,T1.H,T1.MpV);
Ch1=T1.H;
Cw1=T1.R*2;
}
if (C1==7 && i==2){
Cylindber T2;
cout<<"Radius of the Cylinder=";
cin>>T2.R;
cout<<endl;
cout<<"Height of the Cylinder=";
cin>>T2.H;
cout<<endl;
cout<<"Mass per volume of the Cylinder=";
cin>>T2.MpV;
cout<<endl;
cout<<"Charactersitics of the Cylinder"<<endl;
Csa2=T2.CSa(T2.R,T2.H);
Cv2=T2.CV(T2.R,T2.H);
Cm2=T2.CM(T2.R,T2.H,T2.MpV);
Ch2=T2.H;
Cw2=T2.R*2;
}

```

```

        if (C1==7 && i==3){
            Cylindber T3;
            cout<<"Radius of the Cylinder=";
            cin>>T3.R;
            cout<<endl;
            cout<<"Height of the Cylinder=";
            cin>>T3.H;
            cout<<endl;
            cout<<"Mass per volume of the Cylinder=";
            cin>>T3.MpV;
            cout<<endl;
            cout<<"Charactersitics of the Cylinder"<<endl;
            Csa3=T3.CSa(T3.R,T3.H);
            Cv3=T3.CV(T3.R,T3.H);
            Cm3=T3.CM(T3.R,T3.H,T3.MpV);
            Ch3=T3.H;
            Cw3=T3.R*2;
        }

        if (C1>7){
            cout<<"No more figure types are currently evaliable in the
programm"<<endl;
        }
        if (C1<0){
            cout<<"Negative numbers cannot be used in this segment of the
programm"<<endl;
        }

        cout<<endl;

        double Chs [C2]={Ch0,Ch1,Ch2,Ch3};
        double Cws [C2]={Cw0,Cw1,Cw2,Cw3};
        double Csas [C2]={Csa0,Csa1,Csa2,Csa3};
        double Cvs [C2]={Cv0,Cv1,Cv2,Cv3};
        double Cms [C2]={Cm0,Cm1,Cm2,Cm3};

        /// to read N=Csas/Cvs/Cms [number of the needed characteristic];
    }

    /// Input of the coordinates segment

    int x,x1,x2,x3;
    int y,y1,y2,y3;

    if (C2>=0){
        cout<<"Input of coordinates for first object"<<endl;
        cout<<"X=";
        cin>>x;
        cout<<endl;
        cout<<"Y=";

```

```

cin>>y;
cout<<endl;

if ( x<0 || x>4 ){
    cout<<"Error, coordinates incorrect"<<endl;
}
if ( y<0 || y>4 ){
    cout<<"Error, coordinates incorrect"<<endl;
}
}
if (C2>=1){
    cout<<"Input of coordinates for second object"<<endl;
    cout<<"X=";
    cin>>x1;
    cout<<endl;
    cout<<"Y=";
    cin>>y1;
    cout<<endl;
    if (x==x1 && y==y1){
        cout<<"Error, objects cannot occupy same coordinates"<<endl;
    }
    if ( x1<0 || x1>4 ){
        cout<<"Error, coordinates incorrect"<<endl;
    }
    if ( y1<0 || y1>4 ){
        cout<<"Error, coordinates incorrect"<<endl;
    }
}
if (C2>=2){
    cout<<"Input of coordinates for third object"<<endl;
    cout<<"X=";
    cin>>x2;
    cout<<endl;
    cout<<"Y=";
    cin>>y2;
    cout<<endl;
    if (x==x2 && y==y2){
        cout<<"Error, objects cannot occupy same coordinates"<<endl;
    }
    if (x1==x2 && y1==y2){
        cout<<"Error, objects cannot occupy same coordinates"<<endl;
    }
    if ( x2<0 || x2>4 ){
        cout<<"Error, coordinates incorrect"<<endl;
    }
    if ( y2<0 || y2>4 ){
        cout<<"Error, coordinates incorrect"<<endl;
    }
}
if (C2>=3){

```

```

cout<<"Input of coordinates for fourth object"<<endl;
cout<<"X=";
cin>>x3;
cout<<endl;
cout<<"Y=";
cin>>y3;
cout<<endl;
if (x==x3 && y==y3){
    cout<<"Error, objects cannot occupy same coordinates"<<endl;
}
if (x1==x3 && y1==y3){
    cout<<"Error, objects cannot occupy same coordinates"<<endl;
}
if (x2==x3 && y2==y3){
    cout<<"Error, objects cannot occupy same coordinates"<<endl;
}
if ( x3<0 || x3>4 ){
    cout<<"Error, coordinates incorrect"<<endl;
}
if ( y3<0 || y3>4 ){
    cout<<"Error, coordinates incorrect"<<endl;
}
}

```

/// Summ of width and height, summ of mass's, average mass of the object, which object has smallest and biggest mass/volume/surface area

```

int C3;
cout<<"Comparison of created figures by:"<<endl;
cout<<"1)Surface area"<<endl;
cout<<"2)Volume"<<endl;
cout<<"3)Mass"<<endl;
cout<<"4)Restart the program"<<endl;
cout<<"Choose - ";
cin>>C3;
cout<<endl;

if (C3==4){
    restart();
}

if (C2==0){
    if (C3==1){
        cout<<"Surface area of the first figure is the largest and ="<<Csa0<<endl;
    }
    if (C3==2){
        cout<<"Volume of the first figure is the largest and ="<<Cv0<<endl;
    }
    if (C3==3){
        cout<<"Mass of the first figure is the largest and ="<<Cm0<<endl;
    }
}

```

```

    }
}
if (C2==1){
    if (C3==1){
        double CsaMax= comp2(Csa0,Csa1);
        if (Csa0==CsaMax){
            cout<<"Surface area of the first figure is the largest and
="<<Csa0<<endl;
        }
        if (Csa1==CsaMax){
            cout<<"Surface area of the second figure is the largest and
="<<Csa1<<endl;
        }
    }
    if (C3==2){
        double CvMax= comp2(Cv0,Cv1);
        if (Cv0==CvMax){
            cout<<"Volume of the first figure is the largest and ="<<Cv0<<endl;
        }
        if (Cv1==CvMax){
            cout<<"Volume of the second figure is the largest and ="<<Cv1<<endl;
        }
    }
    if (C3==3){
        double CmMax= comp2(Cm0,Cm1);
        if (Cm0==CmMax){
            cout<<"Mass of the first figure is the largest and ="<<Cm0<<endl;
        }
        if (Cm1==CmMax){
            cout<<"Mass of the second figure is the largest and ="<<Cm1<<endl;
        }
    }
}
if (C2==2){
    if (C3==1){
        double CsaMax= comp3(Csa0,Csa1,Csa2);
        if (Csa0==CsaMax){
            cout<<"Surface area of the first figure is the largest and
="<<Csa0<<endl;
        }
        if (Csa1==CsaMax){
            cout<<"Surface area of the second figure is the largest and
="<<Csa1<<endl;
        }
        if (Csa2==CsaMax){
            cout<<"Surface area of the third figure is the largest and
="<<Csa2<<endl;
        }
    }
    if (C3==2){

```



```

    }
    if (Cv1==CvMax){
        cout<<"Volume of the second figure is the largest and ="<<Cv1<<endl;
    }
    if (Cv2==CvMax){
        cout<<"Volume of the third figure is the largest and ="<<Cv2<<endl;
    }
    if (Cv3==CvMax){
        cout<<"Volume of the fourth figure is the largest and ="<<Cv3<<endl;
    }
}

if (C3==3){
    double CmMax= comp4(Cm0,Cm1,Cm2,Cm3);
    if (Cm0==CmMax){
        cout<<"Mass of the first figure is the largest and ="<<Cm0<<endl;
    }
    if (Cm1==CmMax){
        cout<<"Mass of the second figure is the largest and ="<<Cm1<<endl;
    }
    if (Cm2==CmMax){
        cout<<"Mass of the third figure is the largest and ="<<Cm2<<endl;
    }
    if (Cm3==CmMax){
        cout<<"Mass of the fourth figure is the largest and ="<<Cm3<<endl;
    }
}
if (C3>3 || C3<1){
    cout<<"Error, no choice of comparison bound to this number"<<endl;
}
}

int C4;
cout<<"Further information of created figures or structures formed"<<endl;
cout<<"1)Energy of the objects created"<<endl;
cout<<"2)Summary od width and length of the structure created by figures"<<endl;
cout<<"3)Restart the program"<<endl;
cout<<"Choose - ";
cin>>C4;
cout<<endl;
double wsm,hsm;

if (C4==3){
    restart();
}

if ( C4<1 || C4>3 ){
    cout<<"Invalid number used for choice of action"<<endl;
}
if (C2==0 && C4==1){

```

```

        double e1=En(Cm0);
        cout<<"Energy of the only object created ="<<e1<<endl;
    }
    if (C2==1 && C4==1){
        double e1=En(Cw0);
        double e2=En(Cw1);
        cout<<"Energy of the first object ="<<e1<<endl;
        cout<<"Energy of the second object ="<<e2<<endl;
    }
    if (C2==2 && C4==1){
        double e1=En(Cm0);
        double e2=En(Cm1);
        double e3=En(Cm2);
        cout<<"Energy of the first object ="<<e1<<endl;
        cout<<"Energy of the second object ="<<e2<<endl;
        cout<<"Energy of the third object ="<<e3<<endl;
    }
    if (C2==3 && C4==1){
        double e1=En(Cm0);
        double e2=En(Cm1);
        double e3=En(Cm2);
        double e4=En(Cm3);
        cout<<"Energy of the first object ="<<e1<<endl;
        cout<<"Energy of the second object ="<<e2<<endl;
        cout<<"Energy of the third object ="<<e3<<endl;
        cout<<"Energy of the fourth object="<<e4<<endl;
    }
}

if (C2==0 && C4==2){
    wsm=Cw0;
    cout<<"There is only one figure, so combined width ="<<wsm<<endl;
    hsm=Ch0;
    cout<<"There is only one figure, so combined height ="<<hsm<<endl;
}
if (C2==1 && C4==2){
    if ( x==x1 && (y1==(y-1) || y1==(y+1)) ){
        wsm=comp2(Cw0,Cw1);
        hsm=Ch0+Ch1;
        cout<<"Width summ="<<wsm<<endl;
        cout<<"Height summ="<<hsm<<endl;
        cout<<"Object form a vertical line"<<endl;
    }
    if ( y==y1 && (x1==(x-1) || x1==(x+1)) ){
        hsm=comp2(Ch0,Ch1);
        wsm=Cw0+Cw1;
        cout<<"Width summ="<<wsm<<endl;
        cout<<"Height summ="<<hsm<<endl;
        cout<<"Objects form a horizontal line"<<endl;
    }
}

```

```

        if (((x1==(x-1) && y1==(y-1)) || (x1==(x+1) && y1==(y-1)) || (x1==(x-1) &&
y1==(y+1)) || (x1==(x+1) && y1==(y+1)))){
            hsm=Ch0+Ch1;
            wsm=Cw0+Cw1;
            cout<<"Width summ="<<wsm<<endl;
            cout<<"Height summ="<<hsm<<endl;
            cout<<"Objects form a diagonal line"<<endl;
        }
        else {
            cout<<"Objects are too far away to form united structure"<<endl;
        }
    }
    if (C2==2 && C4==2){

        /// three in line variant
        if ((y==y1 || y1==y2) && ((x==x1+1 && x==x2-1) || (x==x2+1 && x==x1-1)
|| (x1==x+1 && x1==x2-1) || (x1==x2+1 && x1==x-1) || (x2==x+1 && x2==x1-1) || (x2==x+1
&& x2==x1-1))){

            wsm=comp3(Cw0,Cw1,Cw2);
            hsm=Ch0+Ch1+Ch2;
            cout<<"Width summ="<<wsm<<endl;
            cout<<"Height summ="<<hsm<<endl;
            cout<<"Object form a vertical line"<<endl;
        }
        if ((x==x1 || x1==x2) && ((y1==y+1 && y1==y2-1) || (y1==y2+1 && y1==y-
1) || (y2==y+1 && y2==y1-1) || (y2==y1+1 && y2==y-1) || (y==y1+1 && y==y2-1) || (y==y2+1
&& y==y1-1))){

            hsm=comp3(Ch0,Ch1,Ch2);
            wsm=Cw0+Cw1+Cw2;
            cout<<"Width summ="<<wsm<<endl;
            cout<<"Height summ="<<hsm<<endl;
            cout<<"Objects form a horizontal line"<<endl;
        }
    }

    /// two lined one not variant
    if ((y==y1) && (x2==x1)){
        hsm=comp2(Ch0,Ch1)+Ch2;
        wsm=comp2(Cw2,Cw1)+Cw0;
        cout<<"Width summ="<<wsm<<endl;
        cout<<"Height summ="<<hsm<<endl;
        cout<<"Objects form an L shape"<<endl;
        /// 01| 2| 2 | 10
        /// 2| 01| 10| 2
    }
    if ((y==y1) && (x2==x)){
        hsm=comp2(Ch0,Ch1)+Ch2;
        wsm=comp2(Cw2,Cw0)+Cw1;
        cout<<"Width summ="<<wsm<<endl;
        cout<<"Height summ="<<hsm<<endl;
        cout<<"Objects form an L shape"<<endl;
    }
}

```

```

        /// 10| 2| 01| 2
        /// 2| 10| 2 | 10
    }

    if ((y2==y1) && (x==x1)){
        hsm=comp2(Ch2,Ch1)+Ch0;
        wsm=comp2(Cw0,Cw1)+Cw2;
        cout<<"Width summ="<<wsm<<endl;
        cout<<"Height summ="<<hsm<<endl;
        cout<<"Objects form an L shape"<<endl;
        /// 21| 0| 12| 0
        /// 0| 21| 0 | 12
    }
    if ((y2==y1) && (x==x2)){
        hsm=comp2(Ch2,Ch1)+Ch0;
        wsm=comp2(Cw2,Cw0)+Cw1;
        cout<<"Width summ="<<wsm<<endl;
        cout<<"Height summ="<<hsm<<endl;
        cout<<"Objects form an L shape"<<endl;
        /// 12| 0| 21| 0
        /// 0| 12| 0 | 21
    }

    if ((y2==y) && (x1==x2)){
        hsm=comp2(Ch2,Ch0)+Ch1;
        wsm=comp2(Cw2,Cw1)+Cw1;
        cout<<"Width summ="<<wsm<<endl;
        cout<<"Height summ="<<hsm<<endl;
        cout<<"Objects form an L shape"<<endl;
        /// 02| 1| 02| 1
        /// 1| 02| 1 | 20
    }
    if ((y2==y) && (x1==x)){
        hsm=comp2(Ch2,Ch0)+Ch1;
        wsm=comp2(Cw1,Cw0)+Cw2;
        cout<<"Width summ="<<wsm<<endl;
        cout<<"Height summ="<<hsm<<endl;
        cout<<"Objects form an L shape"<<endl;
        /// 20| 1| 02| 1
        /// 1| 20| 1 | 02
    }

    else {
        cout<<"Objects are too far away to form united structure"<<endl;
    }
}
if (C2==3 && C4==2){

    if (x==x1 && x1==x2 && ( x3>x1 || x3<x1 ) && (y3==y1 || y3==y2 ||
y3==y)){

```

```

double CCw;

if (y3==y){
    CCw=comp2(Cw0,Cw3);
    wsm=CCw+Cw1+Cw2;
    cout<<"Width summ="<<wsm<<endl;
}
if (y3==y1){
    CCw=comp2(Cw1,Cw3);
    wsm=Cw0+CCw+Cw2;
    cout<<"Width summ="<<wsm<<endl;
}
if (y3==y2){
    CCw=comp2(Cw2,Cw3);
    wsm=Cw0+Cw1+CCw;
    cout<<"Width summ="<<wsm<<endl;
}
hsm=(comp3(Ch0,Ch1,Ch2))+Ch3;
cout<<"Height summ="<<hsm<<endl;
}

if (y==y1 && y1==y2 && ( y3>y1 || y3<y1 ) && (x3==x1 || x3==x2 ||
x3==x)){

double CCh;

if (x3==x){
    CCh=comp2(Ch0,Ch3);
    wsm=CCh+Ch1+Ch2;
    cout<<"Height summ="<<hsm<<endl;
}
if (x3==x1){
    CCh=comp2(Ch1,Ch3);
    wsm=Ch0+CCh+Ch2;
    cout<<"Height summ="<<hsm<<endl;
}
if (x3==x2){
    CCh=comp2(Ch2,Ch3);
    wsm=Ch0+Ch1+CCh;
    cout<<"Height summ="<<hsm<<endl;
}
}
wsm=(comp3(Cw0,Cw1,Cw2))+Cw3;
cout<<"Width summ="<<wsm<<endl;

}
// Squares from blocks 12 / 13 / 14 / 21 / 23 / 24 / 31 / 32 / 34 / 41 / 42 / 43

```

```

// 24 | 13 | 31 | 42
// 13 | 24 | 42 | 31
if ( (x==x2 && x1==x3) && (x1==x+1 || x1==x-1) && (y3==y2 &&
y1==y)) {
    wsm=comp2(Cw0,Cw1)+comp2(Cw2,Cw3);
    hsm=comp2(Ch0,Ch2)+comp2(Ch1,Ch3);
    cout<<"Height summ="<<hsm<<endl;
    cout<<"Width summ="<<wsm<<endl;
    cout<<"Object's coordinates create a square line"<<endl;
}
// 24 | 31 | 13 | 42
// 31 | 24 | 42 | 13
if ( (x==x2 && x1==x3) && (x1==x+1 || x1==x-1) && (y3==y &&
y1==y2)) {
    wsm=comp2(Cw0,Cw3)+comp2(Cw2,Cw1);
    hsm=comp2(Ch0,Ch2)+comp2(Ch1,Ch3);
    cout<<"Height summ="<<hsm<<endl;
    cout<<"Width summ="<<wsm<<endl;
    cout<<"Object's coordinates create a square line"<<endl;
}
// 23 | 14 | 41 | 32
// 14 | 23 | 32 | 41
if ( (x==x1 && x2==x3) && (x2==x+1 || x2==x-1) && (y1==y2 &&
y==y3)) {
    wsm=comp2(Cw0,Cw1)+comp2(Cw2,Cw3);
    hsm=comp2(Ch0,Ch3)+comp2(Ch1,Ch2);
    cout<<"Height summ="<<hsm<<endl;
    cout<<"Width summ="<<wsm<<endl;
    cout<<"Object's coordinates create a square line"<<endl;
}
// 32 | 14 | 41 | 23
// 14 | 32 | 23 | 41
if ( (x==x2 && x1==x3) && (x2==x+1 || x2==x-1) && (y1==y2 &&
y==y3)) {
    wsm=comp2(Cw0,Cw2)+comp2(Cw1,Cw3);
    hsm=comp2(Ch0,Ch3)+comp2(Ch1,Ch2);
    cout<<"Height summ="<<hsm<<endl;
    cout<<"Width summ="<<wsm<<endl;
    cout<<"Object's coordinates create a square line"<<endl;
}
// 43 | 34 | 21 | 12
// 12 | 21 | 34 | 43
if ( (x==x3 && x2==x1) && (x2==x3+1 || x2==x3-1) && (y1==y &&
y2==y3)) {
    wsm=comp2(Cw0,Cw2)+comp2(Cw1,Cw3);
    hsm=comp2(Ch1,Ch3)+comp2(Ch0,Ch2);
    cout<<"Height summ="<<hsm<<endl;
    cout<<"Width summ="<<wsm<<endl;
    cout<<"Object's coordinates create a square line"<<endl;
}

```

```

    }
    /// 34 | 43 | 21 | 12
    /// 12 | 21 | 43 | 34
    if ( (x==x2 && x3==x1) && (x2==x+1 || x2==x-1) && (y1==y3 &&
y2==y)) {
        wsm=comp2(Cw0,Cw2)+comp2(Cw1,Cw3);
        hsm=comp2(Ch2,Ch3)+comp2(Ch0,Ch1);
        cout<<"Height summ="<<hsm<<endl;
        cout<<"Width summ="<<wsm<<endl;
        cout<<"Object's coordinates create a square line"<<endl;
    }

    if (x==x1 && x1==x2 && x2==x3 ) { /// vertical line
        hsm=Ch0+Ch1+Ch2+Ch3;
        cout<<"Object's coordinates create a vertical line"<<endl;
        cout<<"Height summ="<<hsm<<endl;
        wsm=comp4(Cw0,Cw1,Cw2,Cw3);
        cout<<"Biggest width is used as structure width here and
="<<wsm<<endl;
    }

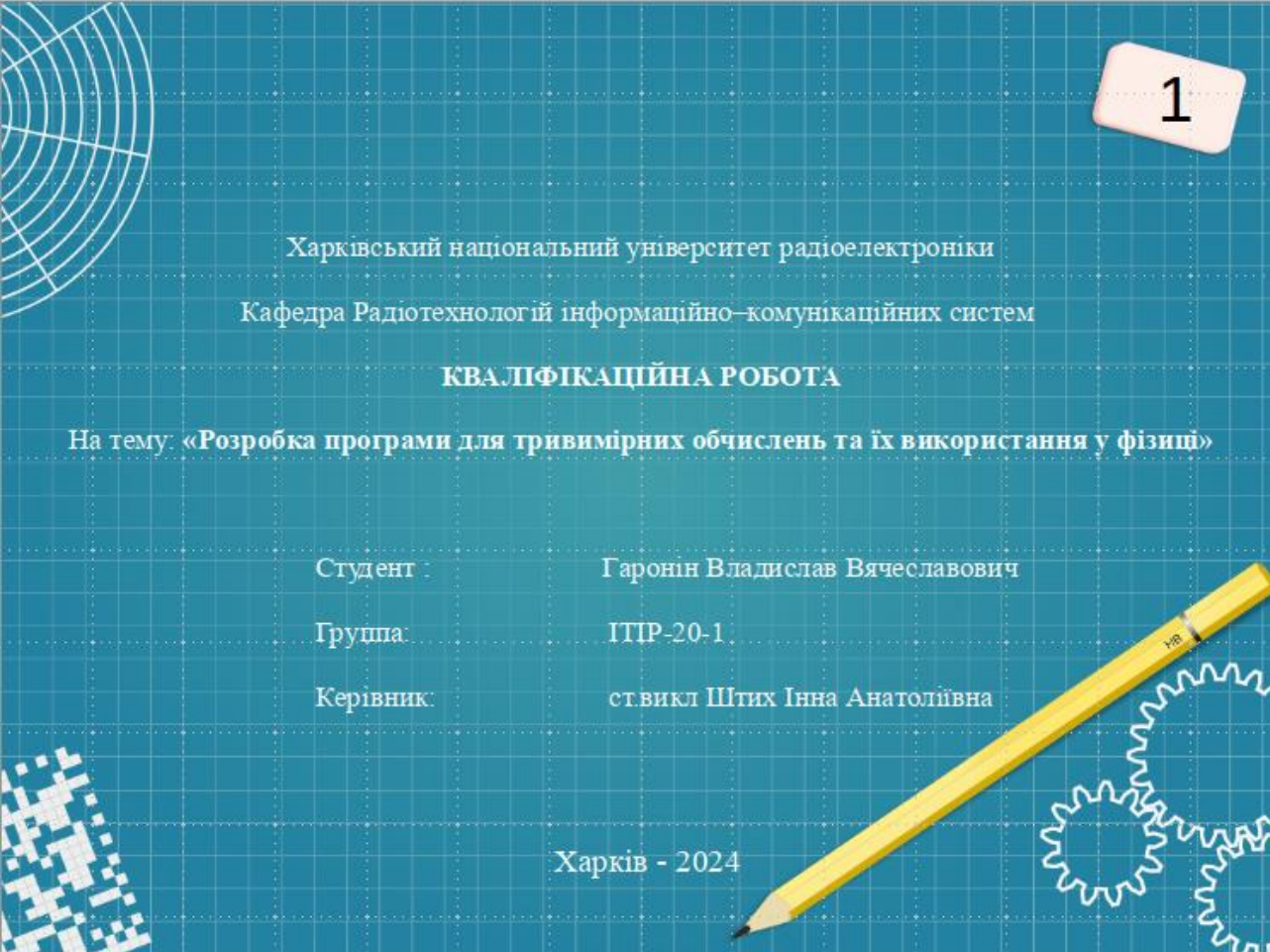
    if (y==y1 && y1==y2 && y2==y3 ) { /// horizontal line
        wsm=Cw0+Cw1+Cw2+Cw3;
        cout<<"Object's coordinates create a horizontal line"<<endl;
        cout<<"Width summ="<<wsm<<endl;
        hsm=comp4(Ch0,Ch1,Ch2,Ch3);
        cout<<"Biggest height is used as structure height here and
="<<hsm<<endl;
    }
}

return 0;
}

```

ДОДАТОК В

СЛАЙДИ ПРЕЗЕНТАЦІЇ



1

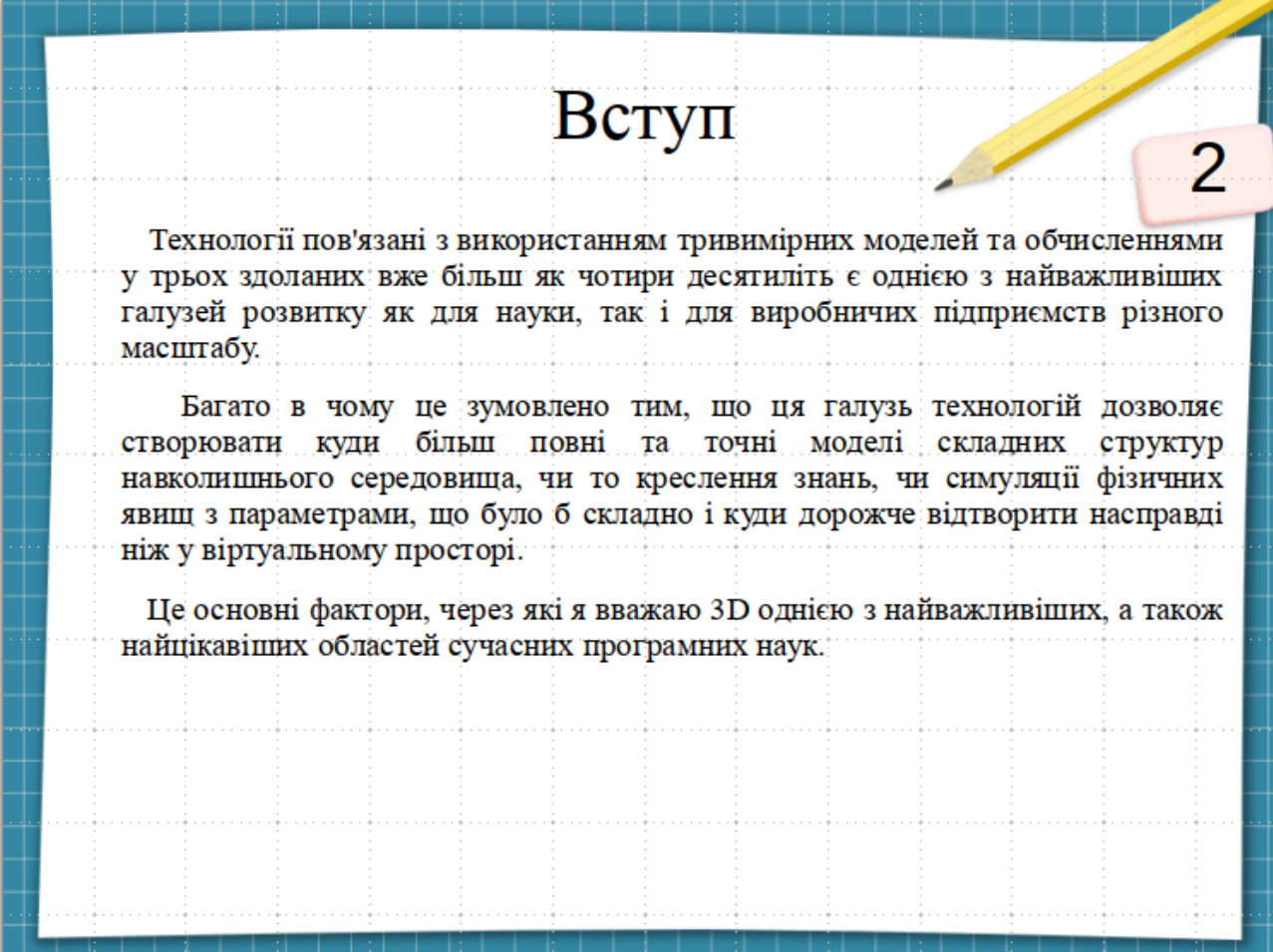
Харківський національний університет радіоелектроніки
Кафедра Радіотехнологій інформаційно–комунікаційних систем

КВАЛІФІКАЦІЙНА РОБОТА

На тему: «Розробка програми для тривимірних обчислень та їх використання у фізиці»

Студент :	Гаронін Владислав Вячеславович
Група:	ІТІР-20-1
Керівник:	ст.викл Штих Інна Анатоліївна

Харків - 2024



2

Вступ

Технології пов'язані з використанням тривимірних моделей та обчисленнями у трьох здоланих вже більш як чотири десятиліття є однією з найважливіших галузей розвитку як для науки, так і для виробничих підприємств різного масштабу.

Багато в чому це зумовлено тим, що ця галузь технологій дозволяє створювати куди більш повні та точні моделі складних структур навколишнього середовища, чи то креслення знань, чи симуляції фізичних явищ з параметрами, що було б складно і куди дорожче відтворити насправді ніж у віртуальному просторі.

Це основні фактори, через які я вважаю 3D однією з найважливіших, а також найцікавіших областей сучасних програмних наук.

Основні галузі споживачі

3

Многие отрасли производства в современном мире в разной степени используют 3D в процессе работы будь то на стадии проектирования продукта или уже ближе к его полноценному созданию. Приведу несколько примеров.

У космічній промисловості та авіації використання 3D моделювання для створення деталей та компонентів космічних та авіаційних систем дозволяє інженерам швидко розробляти та тестувати нові конструкції.

У медицині 3D технології дозволяють створювати на замовлення імпланти, такі як кісткові гвинти, зубні коронки та протези, які точно відповідають анатомії пацієнта. 3D принтер застосовується у сфері моделювання органів та тканин.

СФЕРЫ ПРИМЕНЕНИЯ 3D-ПРИНТЕРОВ

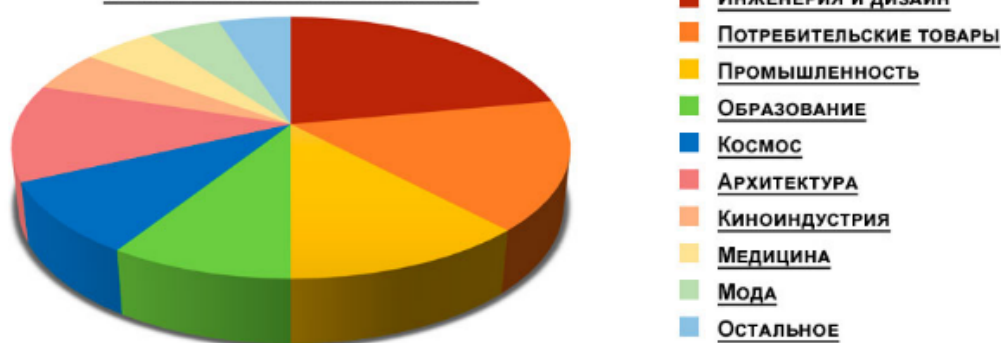


Рисунок 1 - Основні споживачі

Основні функції програми

4

```
Choice of the number of figures created -
0 to 3, to create 1-4 figures
1
```

```
Figure type options:
```

```
0->Sphere
1->Cube
2->Cuboid
3->Prism
4->Hemisphere
5->Pyramid
6->Cone
7->Cylinder
```

Рисунок 2 - Вибір числа фігур, список типів

```
Choice of the figure type created -
0
Radius of the Sphere=3
Mass per volume of the Sphere=1
Characteristics of the Sphere
Surface area of the Sphere=339.12
Volume of the Sphere=84.78
Mass of the Sphere=84.78
Choice of the figure type created -
1
Side of the Cube=3
Mass per volume of the Cube=2
Characteristics of the Cube
Surface are of the Cube=54
Volume of the Cube=27
Mass of the Cube=54
```

Рисунок 3 - Вибір типу фігур та введення характеристик

Вторинні функції програми

5

Comparison of created figures by:

- 1)Surface area
 - 2)Volume
 - 3)Mass
 - 4)Restart the program
- Choose - 3

Mass of the first figure is the largest and =84.78

Рисунок 5 - Вибір порівнюваних характеристик

Further information of created figures or structures formed

- 1)Energy of the objects created
- 2)Summary od width and length of the structure created by figures
- 3)Restart the program

Choose - 2

Height summ=14
Width summ=14
Object's coordinates create a square line

Рисунок 6 - Вибір подальших дій

Input of coordinates for first object
X=1
Y=2

Input of coordinates for second object
X=1
Y=1

Input of coordinates for third object
X=2
Y=1

Input of coordinates for fourth object
X=2
Y=2

Рисунок 4 - Введення координат

Висновки

У кваліфікаційній роботі було розглянуто роль та застосування технологій 3D у різних галузях науки і виробництва чи це процес моделювання конструкцій, симуляція умов середовища чи друк готових креслень за допомогою техніки.

У процесі виконання була розроблена програма, що дозволяє обчислювати основні характеристики кількох типів тривимірних об'єктів, їх фізичні характеристики, порівнювати і ті й інші між собою, а також отримувати дані в деяких сумарних характеристиках створених тіл, їх розташуванні в координатній сітці відносно один одного.

6

Публікація на тему роботи

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ



ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНИКИ

Матеріали XXVIII Міжнародного молодіжного форуму

«Радіоелектроніка та молодь у XXI столітті»

ТОМ 7 «Комп'ютерний зір, системний аналіз та математичне моделювання»

Харків 2024

УДК 517.927.517.58:519.6

ТРИВИМІРНІ ОБЧИСЛЕННЯ ТА ЇХ ЗАСТОСУВАННЯ
Гаронін В.В.
Науковий керівник – ст. викл. Шитих Л.А.
Харківський національний університет радіоелектроніки, каф. РТБКС,
м. Харків, Україна
e-mail: vladyslav.garonin@nure.ua

Work is devoted to creation and usage of programs that utilize geometric calculations in three dimensions. This type of programs is widely used in modern day building industry as well as game design, reason for that being that both buildings and three-dimensional game space are based on numberless combinations of simpler geometrical bodies with different sets of characteristics.

У сучасному світі все більшу роль відводиться комп'ютерам і техніці в першу чергу в тій галузі, для якої перші комп'ютери були створені – обчислення.

Вибравши більш конкретні приклади, не можна залишити без уваги геометрію і фізику, за допомогою комбінації яких колись були створені базові принципи архітектурного мистецтва.

Програми зі створення моделей, симуляції та обчислення характеристик об'єктів у тривимірному просторі при заданих їм властивостях є базовими, а також постійно доопрацьовуються різними фірмами за гроші будівельних компаній, що хочуть швидше створювати креслени, деталі та нагнітні аресленик, моделі майбутніх проектів. Прикладом є програма на рис. 2, що використовує базові простих тривимірних об'єктів, основні види яких наведені на рис. 1. Технології на тій же основі використовують і для більш комплексних проектів [1].



Рисунок 1 – Основні види тривимірних фігур

Крім цього подібні обчислення використовуються в галузі індустрії тривимірний простір вкресленні гор також складено з множини простих фігур, які можна побачити на рис. 1.

У роботі я розгляну створення калькулятора для тривимірних фігур різної складності, обчислення їхньої маси залежно від заданих параметрів, розміщення фігур щодо одне одного за допомогою зберігання їх об'єктів множини.

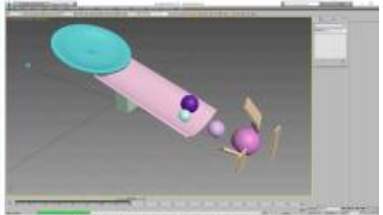


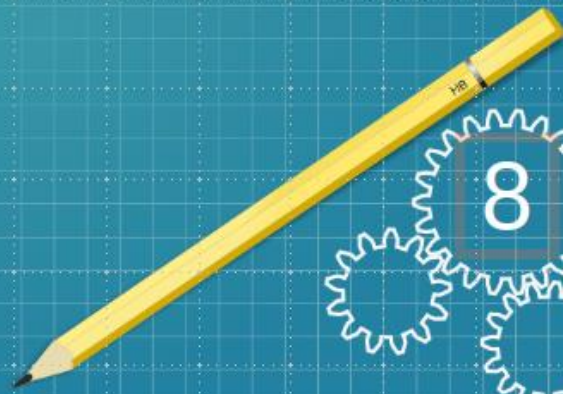
Рисунок 2 – Симулятори простих фізичних процесів

Це можна вважати створеною версією тих програм, які в сучасній галузі використовують програми для симуляції простих фізичних процесів та взаємодії тіл різної маси та обсягу.

Список використаних джерел:

1. Применіте 3D технології в производстві навісних фасадних систем: <https://asadny.pro/articles/primenenie-3d-tekhnologii-v-proizvodstvenavnesnykh-fasadnykh-sistem-bionichesky-fasad-2023-04-10>

Доповідь завершено Дякую за увагу



ДОДАТОК Г

ВІДОМІСТЬ КВАЛІФІКАЦІЙНОЇ РОБОТИ

