Міністерство освіти і науки України

Національний технічний університет України „КПІ”

Факультет інформатики та обчислювальної техніки

Кафедра автоматизованих систем обробки

інформації та управління

**ЗВІТ**

з комп’ютерного практикуму № 3

на тему :

„ ПОЛИМОРФИЗМ С++”

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Виконав**  **студент** |  | *ІП-63 Дроздович В.О.* |  |  |
|  |  | (№ групи, прізвище, ім’я, по батькові ) |  |  |
|  |  |  |  |  |
| **Прийняв** |  | *ас. Головченко М.М.* |  |  |
|  |  | (посада, прізвище, ім’я, по батькові ) |  |  |

Київ 2017

ЗМІСТ

[1 Мета роботи 3](#_Toc501592162)

[2 Постановка задачі 4](#_Toc501592163)

[3 Аналітичні викладки 5](#_Toc501592164)

[4 uml діаграмма класу 6](#_Toc501592165)

[5 Тексти програмного коду 7](#_Toc501592166)

[6 Результат роботи програми 9](#_Toc501592167)

[Висновки 10](#_Toc501592168)

# Мета роботи

Цель работы – изучить основные концепции объектно-ориентированного программирования. Изучить особенности виртуальных функций, абстрактных классов и полиморфизма.

# Постановка задачі

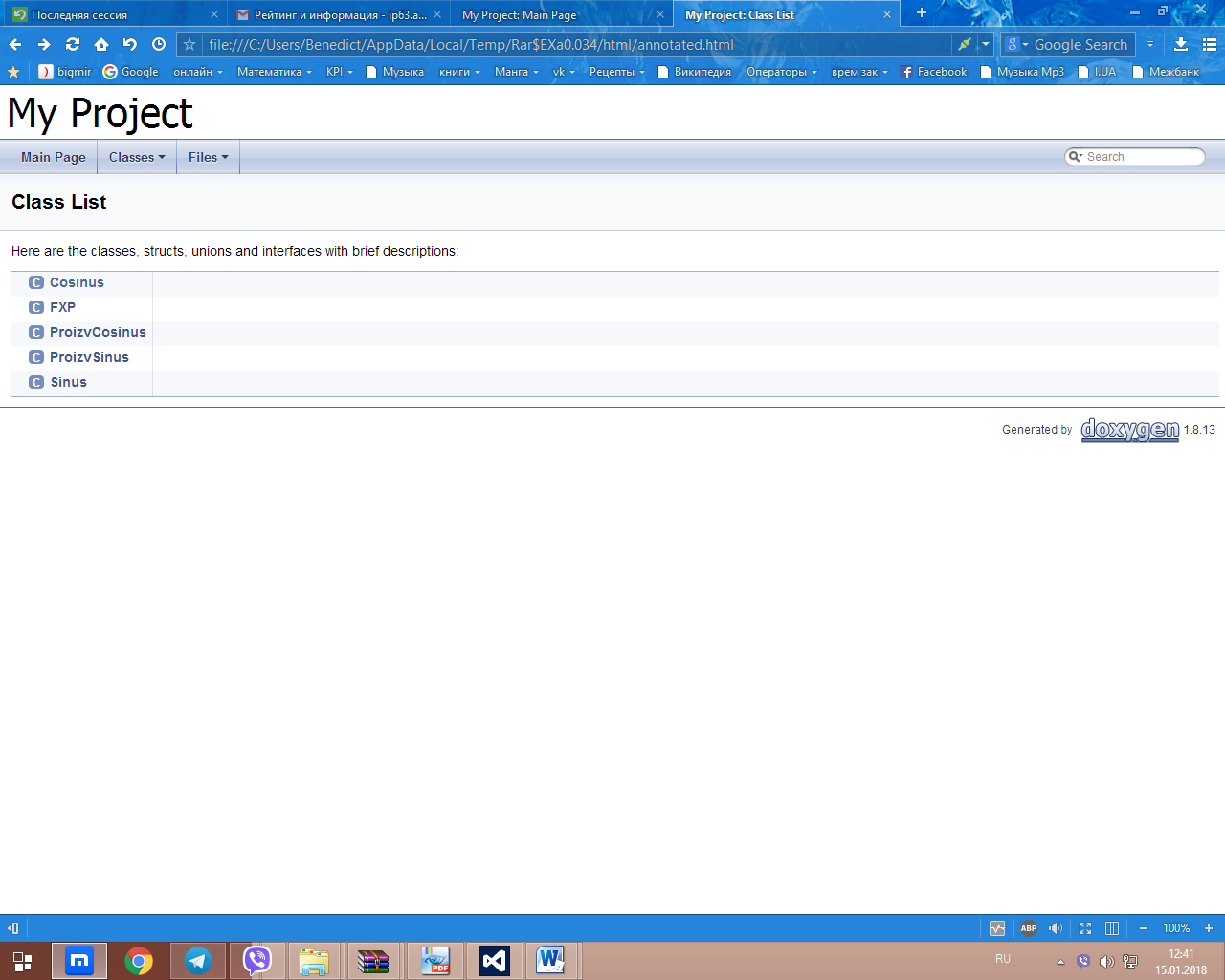
Спроектировать иерархию классов: класс функция от одной

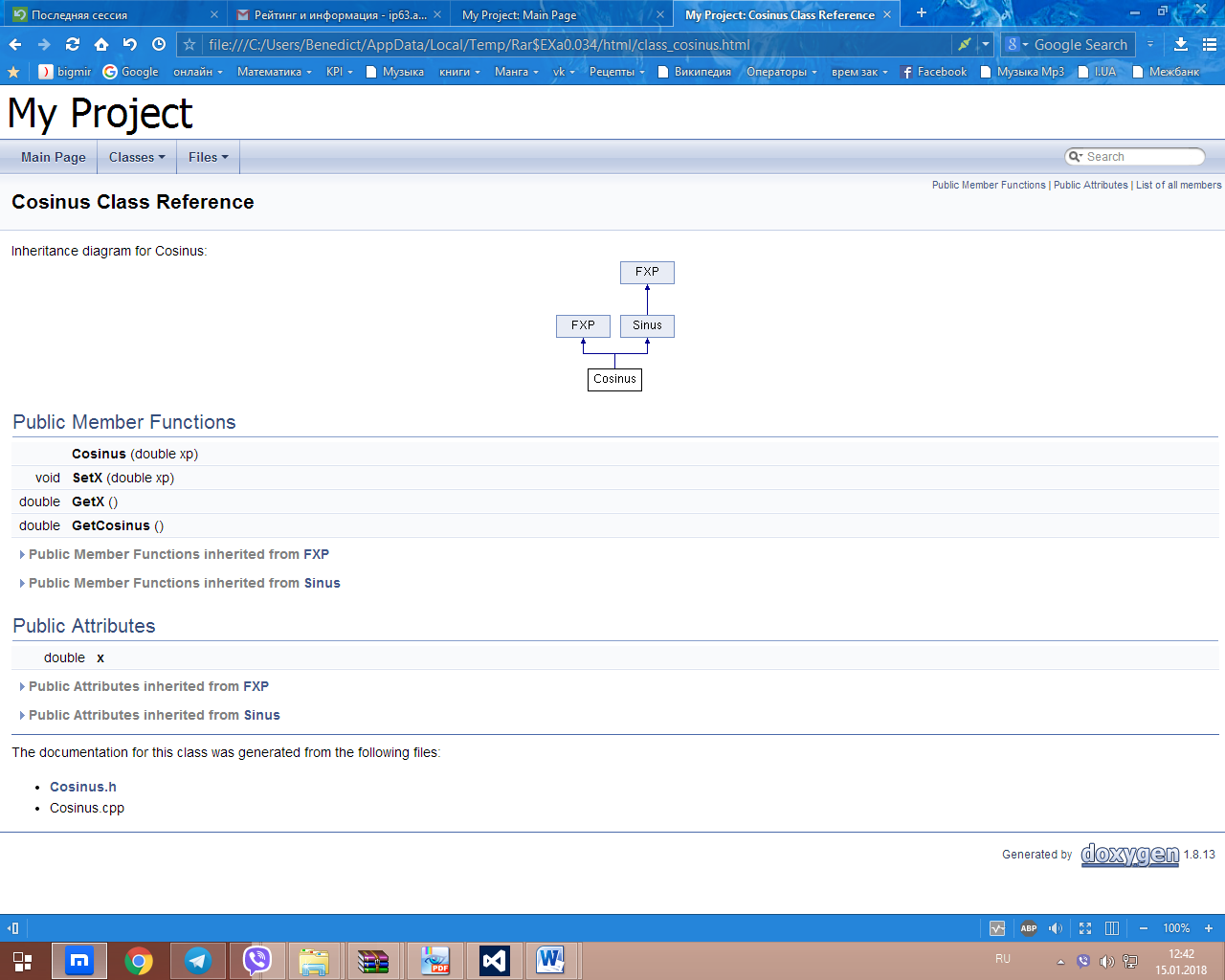
переменной и его наследники: синус и его производная, косинус и его производная. Определить в базовом классе и переопределить в наследниках методы вычисления значения функции для заданного значения переменной. Элементы-данные объявляются в базовом классе, а инициализируются в наследниках (элементы данные: значение переменной).

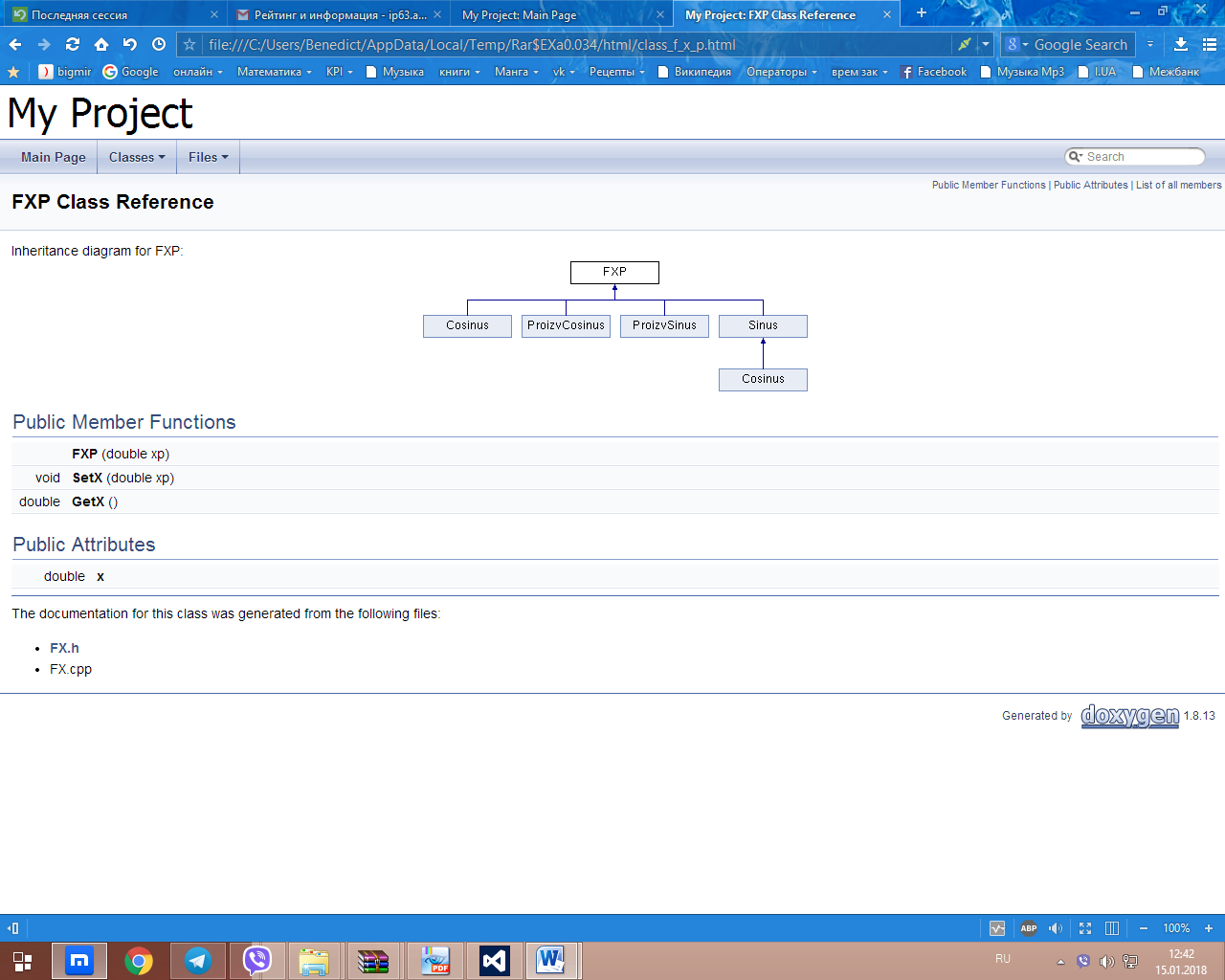
# Аналітичні викладки

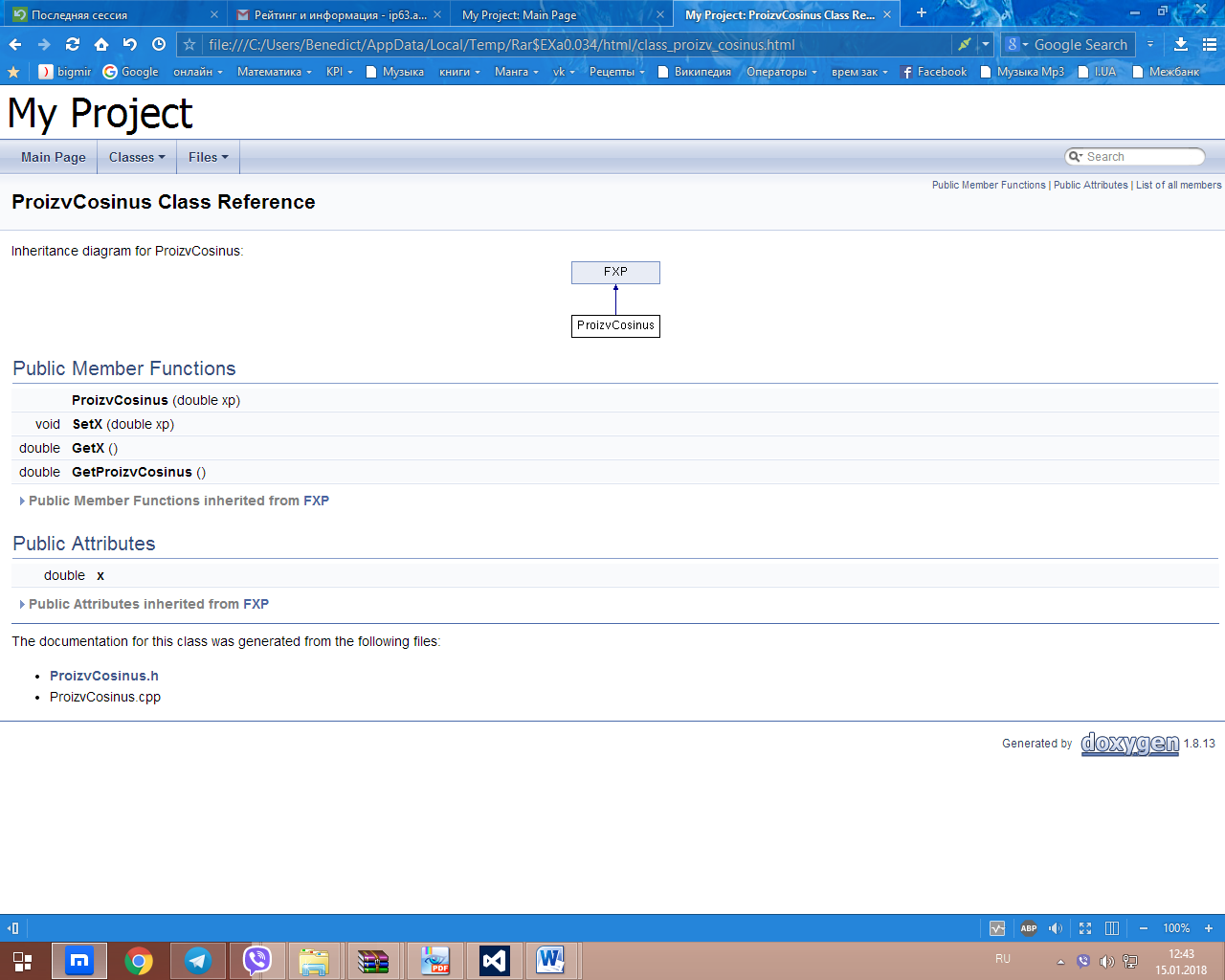
Полиморфизм – свойство кода С++ вести себя по-разному в зависимости от ситуации, возникающей в момент выполнения; это возможность объектов разных классов, связанных наследованием, реагировать различным образом при обращении к одной и той же функции-элементу с одинаковым именем и разным определением. Возникает проблема распознавания функций, решаемая при помощи раннего (статического) связывания или позднего (динамического).

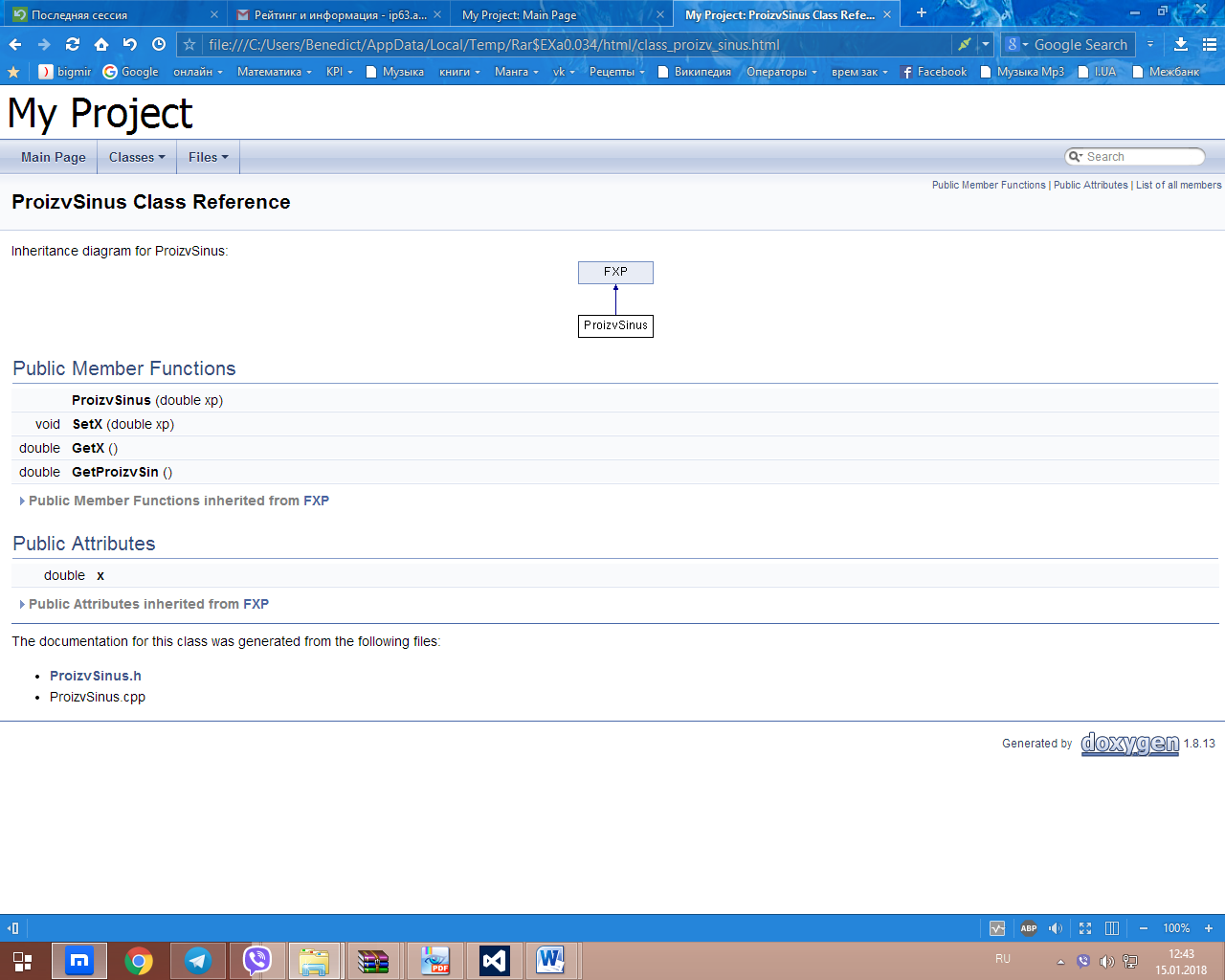
# uml діаграмма класу

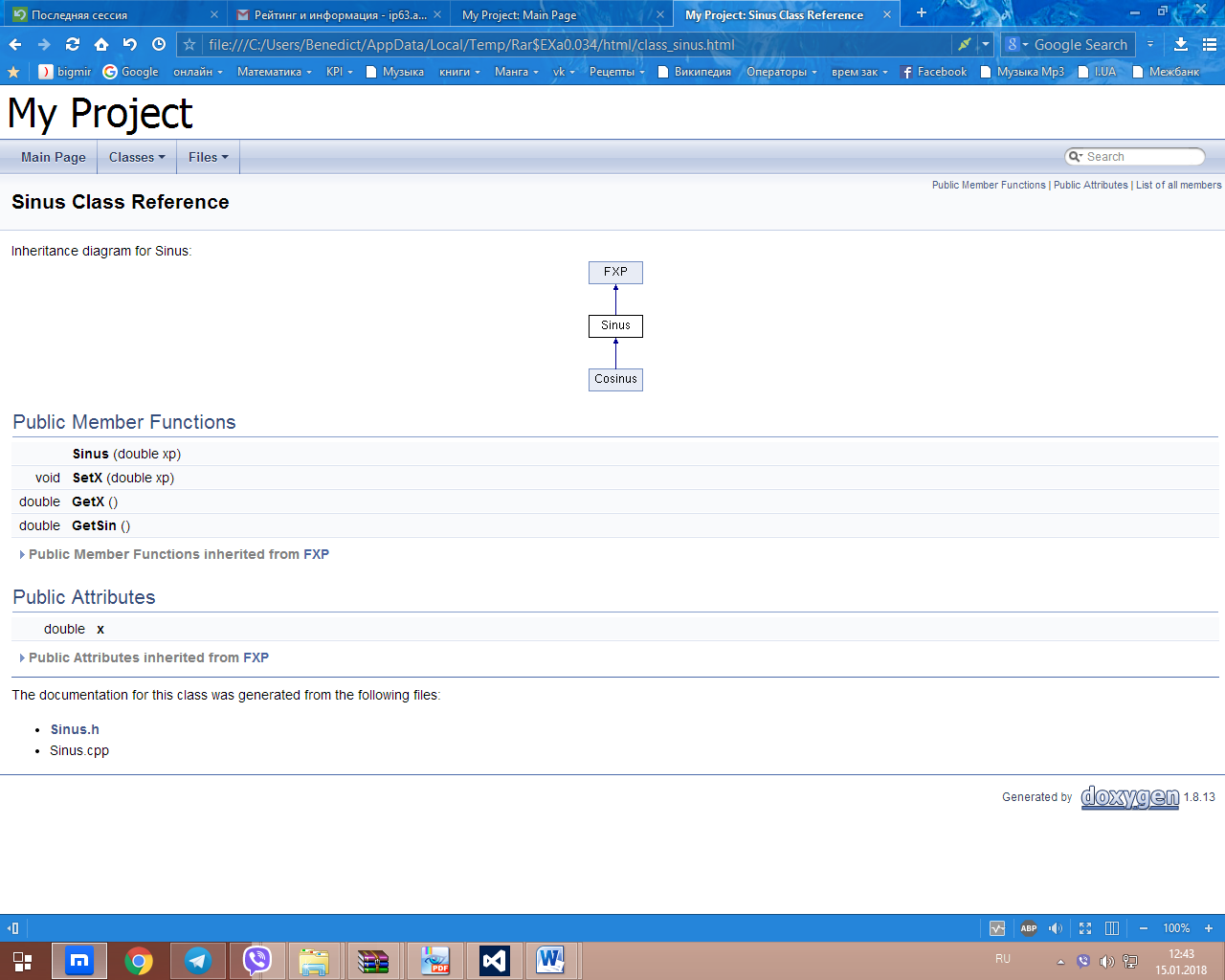












# Тексти програмного коду

#pragma once

#include<iostream>

class FXP {

public:

double x;

FXP ();

FXP(double xp);

void SetX(double xp);

double GetX();

~FXP();

};

#include"FX.h"

FXP::FXP(){

x=60;

}

FXP::FXP(double xp){

x=xp;

}

void FXP::SetX(double xp){

x=xp;

}

double FXP::GetX(){

return x;

}

FXP::~FXP(){

x=NULL;

}

#pragma once

#include"FX.h"

#include<iostream>

class Sinus

{

public:

double x;

Sinus ();

Sinus(double xp);

void SetX(double xp);

double GetX();

double GetSin();

~Sinus();

};

#include"Sinus.h"

#include<math.h>

Sinus::Sinus(){

x=60;

}

Sinus::Sinus(double xp){

x=xp;

}

void Sinus::SetX(double xp){

x=xp;

}

double Sinus::GetX(){

return x;

}

double Sinus::GetSin()

{

return sin(x\*3.14/180);

}

Sinus::~Sinus(){

x=NULL;

}

#pragma once

#include"Sinus.h"

class ProizvSinus

{

public:

double x;

ProizvSinus ();

ProizvSinus(double xp);

void SetX(double xp);

double GetX();

double GetProizvSin();

~ProizvSinus();

};

#include"ProizvSinus.h"

ProizvSinus::ProizvSinus(){

x=60;

}

ProizvSinus::ProizvSinus(double xp){

x=xp;

}

void ProizvSinus::SetX(double xp){

x=xp;

}

double ProizvSinus::GetX(){

return x;

}

double ProizvSinus::GetProizvSin()

{

return cos(x\*3.14/180);

}

ProizvSinus::~ProizvSinus(){

x=NULL;

}

#pragma once

#include"FX.h"

#include<iostream>

class ProizvCosinus

{

public:

double x;

ProizvCosinus ();

ProizvCosinus(double xp);

void SetX(double xp);

double GetX();

double GetProizvCosinus();

~ProizvCosinus();

};

#include"ProizvCosinus.h"

ProizvCosinus::ProizvCosinus(){

x=60;

}

ProizvCosinus::ProizvCosinus(double xp){

x=xp;

}

void ProizvCosinus::SetX(double xp){

x=xp;

}

double ProizvCosinus::GetX(){

return x;

}

double ProizvCosinus::GetProizvCosinus()

{

return -sin(x\*3.14/180);

}

ProizvCosinus::~ProizvCosinus(){

x=NULL;

}

#include"ProizvCosinus.h"

#include"ProizvSinus.h"

#include"Sinus.h"

#include"Cosinus.h"

#include"ProizvSinus.h"

#include"ProizvCosinus.h"

using namespace std;

int main()

{ setlocale (LC\_ALL, "Russian");

cout<<"Введите x: \n";

double xpp;

cin>>xpp;

Sinus \*sinus = new Sinus();

sinus->SetX(xpp);

cout<<"Sin("<<xpp<<") = "<<sinus->GetSin()<<endl;

cout<<"Производная синуса Sin("<<xpp<<")= "<<cos(xpp\*3.14/180)<<endl;

cout<<"Cos("<<xpp<<")= "<<cos(xpp\*3.14/180)<<endl;

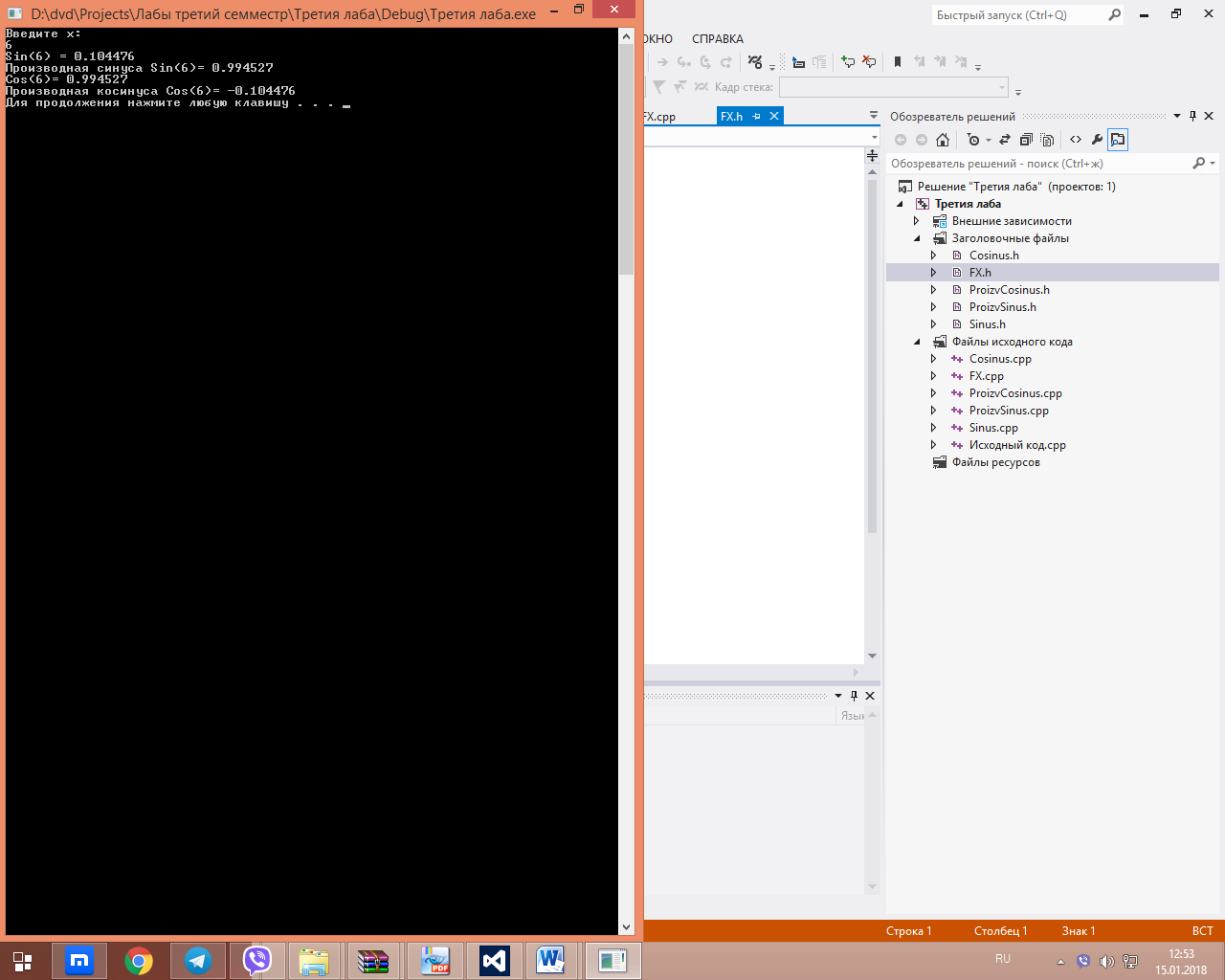
cout<<"Производная косинуса Cos("<<xpp<<")= "<<sinus->GetSin()\*(-1)<<endl;

system("pause");

return 0;

}

# Результат роботи програми



Висновки

В ході виконання лабораторної роботи я отримав навички складання програм із використанням віртуальних функцій, абстрактних класів та поліморфізму.