Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ МОРСКОЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ  
(СПБГМТУ)

Факультет Морского Приборостроения

Кафедра систем автоматического управления и бортовой вычислительной техники

**Пояснительная записка**

К курсовой работе

«Программа табулирования функции»

Вариант 16

По дисциплине «Объектно-ориентированное программирование»

Специальность: Вычислительные машины, комплексы, системы и сети

Выполнил:

Студент 2 курса группы № 3270

Сергиенко Д.К.

Подпись:

Проверил:

Доцент

Сакович С. Ю.

Подпись:

Дата выполнения отчёта:

Дата сдачи отчёта:

СОДЕРЖАНИЕ

[**ВВЕДЕНИЕ 3**](#_Toc90240464)

[**1 Анализ решаемой задачи 4**](#_Toc90240465)

[**2 Блок-схема алгоритма Программы 1 4**](#_Toc90240466)

[**3 Диаграмма и описание механизма взаимодействия классов Программы 2 4**](#_Toc90240467)

[**4 Диаграмма и описание механизма взаимодействия классов Программы 3 4**](#_Toc90240468)

[**ПРИЛОЖЕНИЕ А 5**](#_Toc90240469)

[**ПРИЛОЖЕНИЕ Б 8**](#_Toc90240470)

[**ПРИЛОЖЕНИЕ В 12**](#_Toc90240471)

# ВВЕДЕНИЕ

Задача

Составить программы табулирования заданной функции в заданном интервале с заданным шагом:

1. Программа 1  
   Используя парадигму структурного программирования в терминах языка C++ с функцией записи в файл.
2. Программа 2  
   Используя ООП: исходные данные читаются из файла в формате txt, программа рассчитывает значение заданной функции с помощью иерархии классов и записывает точки табуляции в другой файл.
3. Программа 3  
   Разработка графического интерфейса пользователя с использованием инструментария Windows Forms и реализации функции табулирования в виде иерархии классов в рамках интерфейса Windows Forms с выводом в график (инструмент chart).

# Анализ решаемой задачи

Анализ функции

Функция 16 варианта:

Область Допустимых Значений функции:

ОДЗ функции обусловлена присутствием аргумента функции *x* :

* под знаком корня
* под знаком корня в знаменателе дроби в функции натурального логарифма

Поскольку подкоренное выражение не может быть отрицательно, то значение знаменателя подкоренного выражения строго положительно (). Значение знаменателя подкоренного выражения при , следовательно :

В совокупности (2) и (3):

Область Определения Функции:

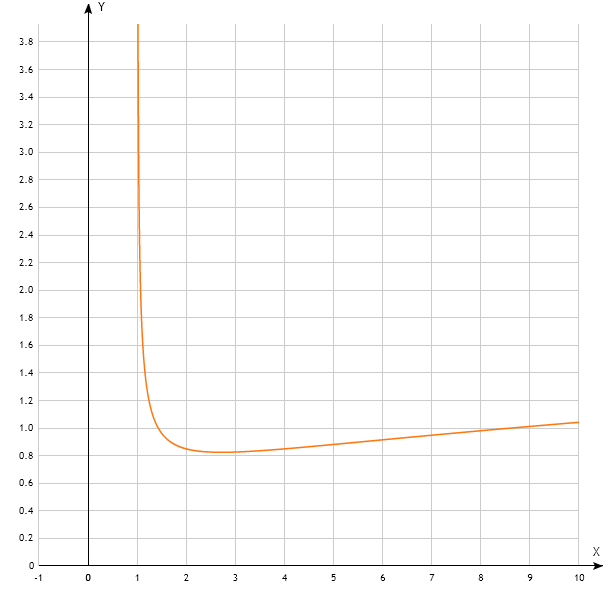
Стационарные точки:

*– точка минимума.*

Минимальное значение функция принимает в:

Следовательно:

График функции:



*Рисунок 1 – график функции 16 варианта*

Детали реализации

Сообщения

В поставленной задаче существуют вводные данные при которых вывода быть не может. Следовательно программа должна уведомить об этом пользователя.

Шаг как модуль

Термин «шаг табуляции функции» можно интерпретировать, применимо к интервалу табуляции функции, как его абсолютная часть, и, при вводе «отрицательного» (от большей точки к меньшей) интервала можно табулировать с шагом отрицательного значения но положительного модуля относительно интервала. Следовательно при вводе положительного значения шага программа при выполнении табуляции меняет его знак на отрицательный (и наоборот), если интервал задан от большего значения к меньшему, т.е. при вводе шага имеет значение только его модуль.

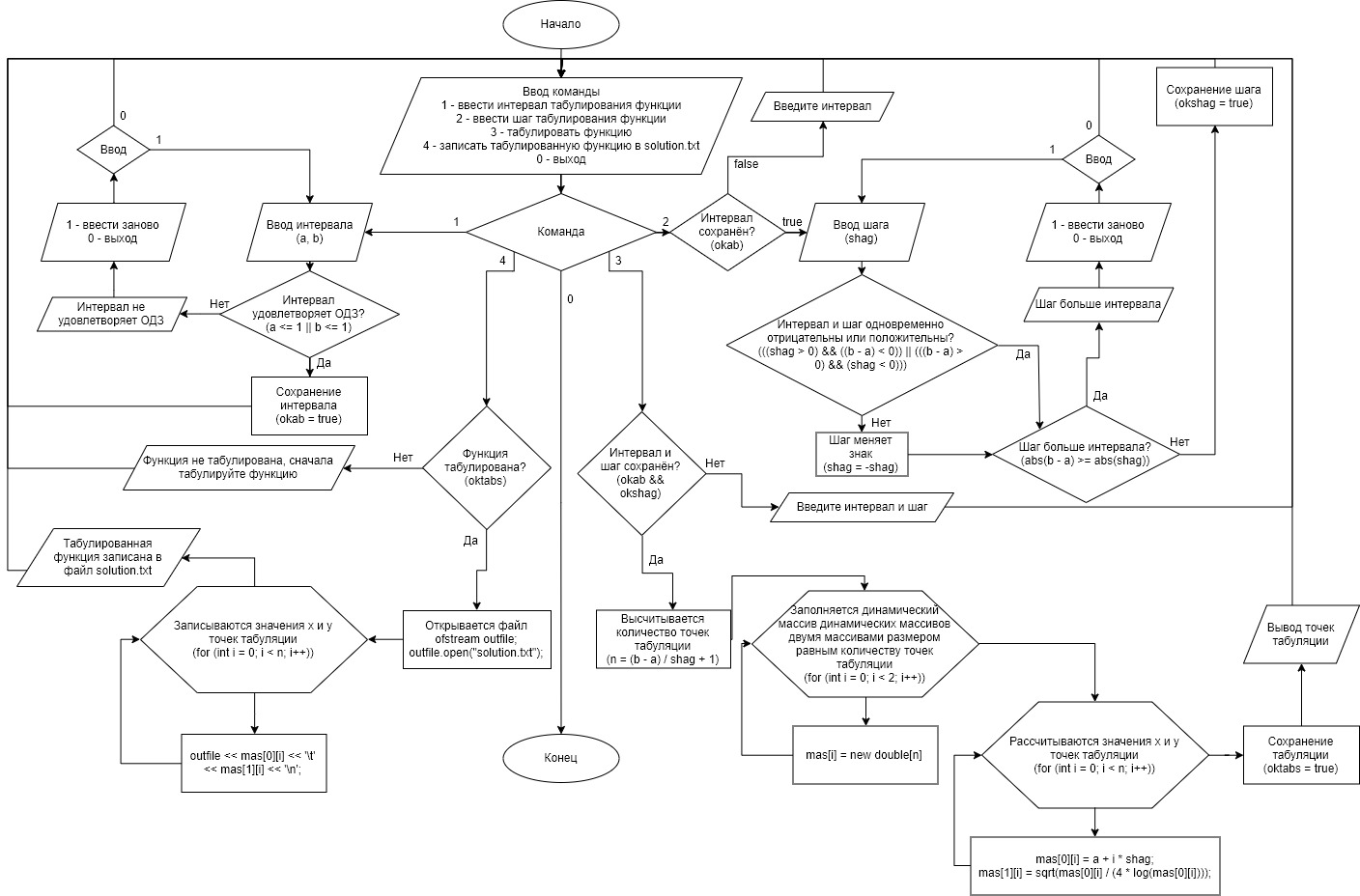
Пустой datain.txt

Во второй программе требуется реализовать чтение из фала. При попытке чтения файл datain.txt не создаётся, поскольку пользователь в состоянии справится с этой тривиальной задачей, а создание пустого файла программой не обосновано.

Ввод в консоль запятой и большого количества непричастных символов

В программах 1 и 2 присутствуют строки кода «cin.ignore(std::cin.rdbuf()->in\_avail());» и «cin.clear();», которые игнорируют все введённые до этого момента символы и чистят буфер потока перед каждым вводом переменной. Причиной их появления в программах служит наличие цикла while и вероятность ошибки пользователя при вводе переменных. При вводе значений с плавающей точкой через консоль следует использовать исключительно точку. Поскольку при вводе переменных, например: «1,5», в консоли, на частный ввод переменной уходил символ «1», а на все последующие – «5». Таким образом консоль выполняла только одну цепочку событий и выйти из этого бесконечного цикла или остановить его было нельзя.

# Блок-схема алгоритма Программы 1



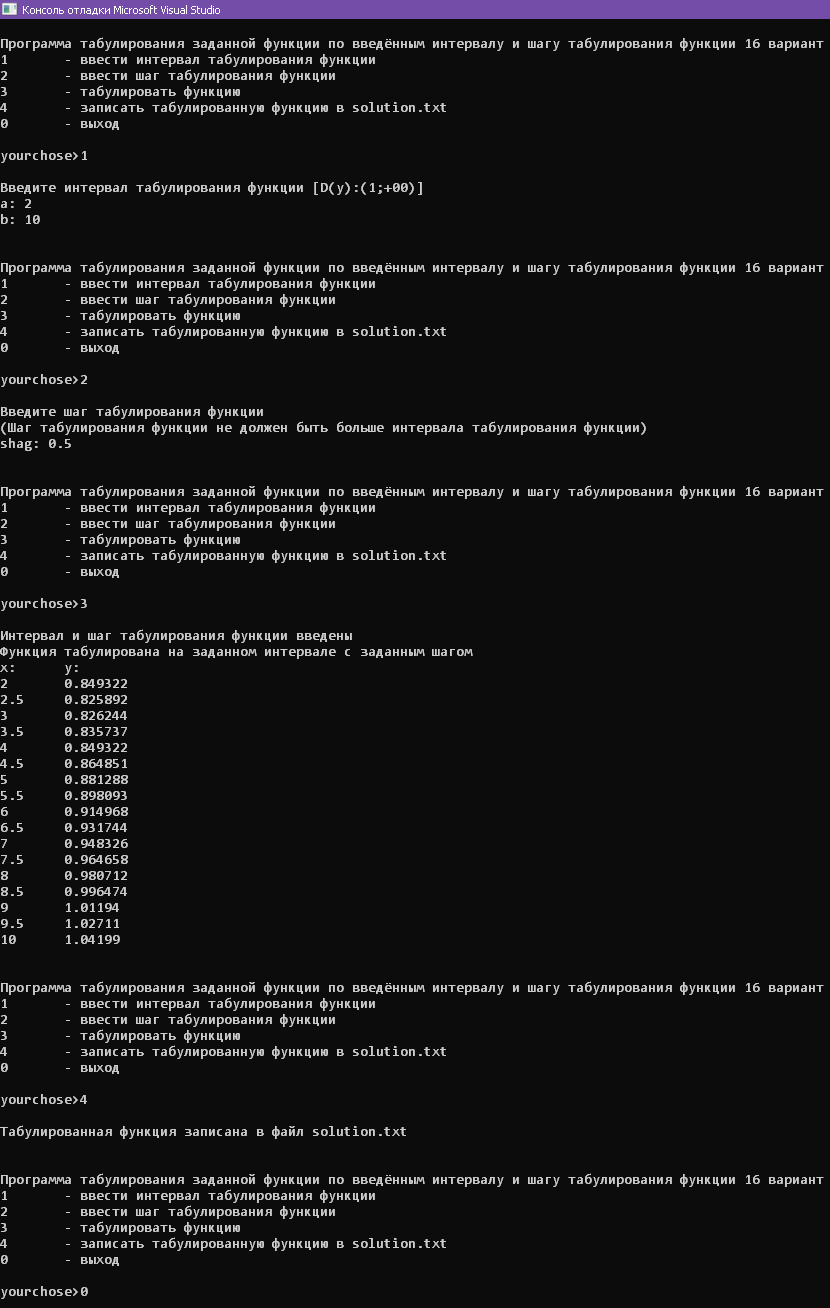
*Рисунок 2 – Блок-схема алгоритма Программы 1*

В Программе 1 (Приложение А) пользователю предлагается: ввести интервал табулирования функции, ввести шаг табулирования функции, табулировать функцию, записать табулированную функцию в solution.txt, выйти из программы. После действия (такого как ввод интервала или запись табулированной функции) программа возвращается в меню, чтобы продолжить работу. При правильном последовательном вводе интервала и шага с помощью программы можно табулировать функцию, а после записать данные в файл.

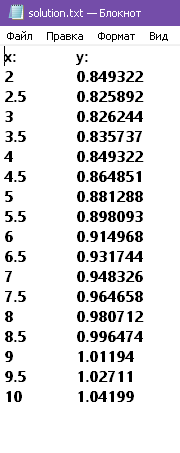
В случае если пользователь:

* при вводе интервала табулирования функции, выйдет за пределы ОДЗ функции, то интервал не будет считаться введённым, и, пользователю, после уведомления о том, что заданный интервал не удовлетворяет ОДЗ функции, или заданный шаг больше интервала, будет предложено вернуться в меню или ввести интервал заново.
* введёт шаг табулирования функции меньше интервала, то шаг не будет считаться введённым, и, пользователю, после уведомления о том, что заданный шаг больше интервала, будет предложено вернуться в меню или ввести шаг заново.
* перейдёт к табуляции функции, не задав предварительно интервал и шаг, то будет выведено сообщение что функция не может быть табулирована.
* перейдёт к записи табулированной функции в файл, не табулировав предварительно функцию, то программа выведет сообщение о том, что функцию следует табулировать перед тем, как записать в файл.

Пример работы

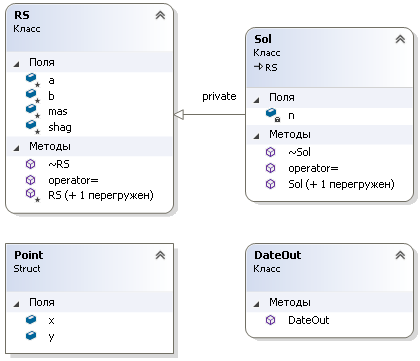


*Рисунок 3 – Скриншот консоли Программы 1*



*Рисунок 4 – Скриншот записи в файл Программы 1*

# Диаграмма и описание механизма взаимодействия классов Программы 2



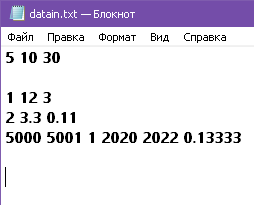
*Рисунок 5 – Диаграмма классов Программы 2*

В программе реализованы классы и структура:

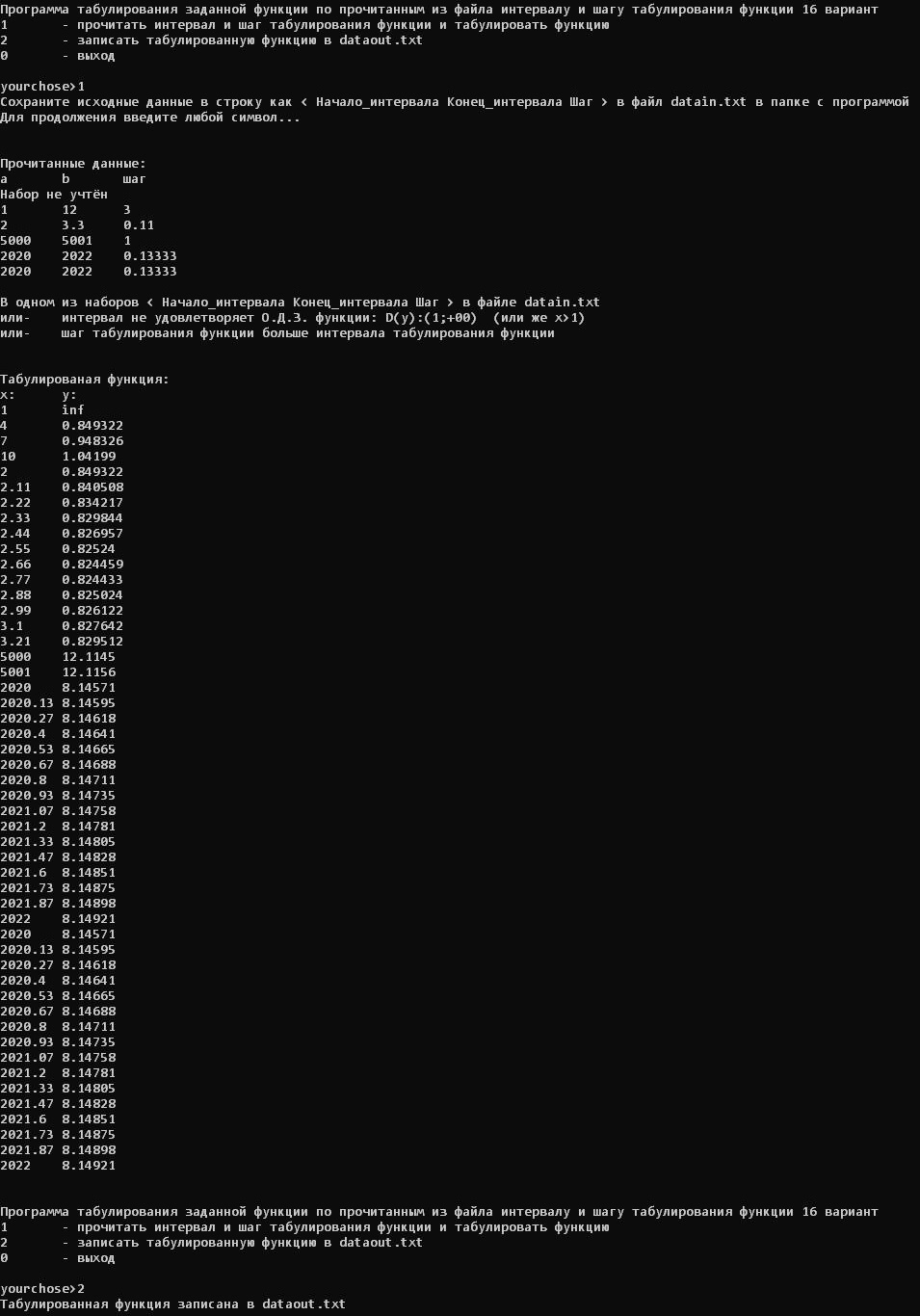
* структура Point (описывает точку табуляции функции), хранящая в себе аргумент функции и её вычисленное значение
* класс RS (Read and Store) читает данные из файла datain.txt для табуляции функции и хранит их (для подкласса Sol), родительский класс для наследования (экземпляр этого класса в программе не используется)
* класс Sol решает поставленную задачу – табуляция функции, дочерний класс для наследования
* класс DateOut выводит данные (принимает в конструктор аргументом экземпляр класса Sol) в файл dataout.txt

Класс RS и структура Point находятся в отношении типа has-a (композиция), т.е. класс Point относится к классу RS как его часть.  
Классы RS и Sol находятся в отношении типа генерализация (наследование), т.е. все, что справедливо для класса RS справедливо и для класса Sol.  
Классы DateOut и Sol находятся в отношении типа зависимость, т.е. класс DateOut ссылается на класс Sol , который в свою очередь не знает о существовании первого. Класс DateOut не хранит указатель на экземпляр класса Sol, как и наоборот, поэтому постоянной связи для ассоциации между ними нет.

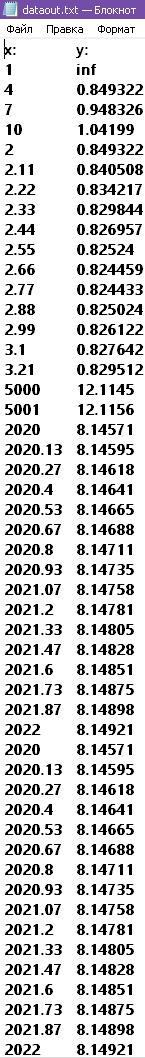
Пример работы

**

*Рисунок 6 – Скриншот файла datain.txt Программы 2*

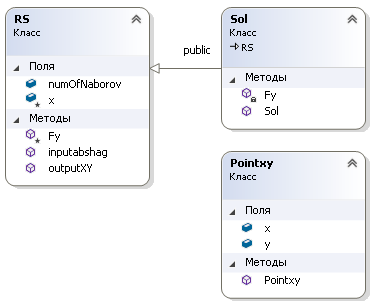


*Рисунок 7 – Скриншот консоли Программы 2*

**

*Рисунок 8 – Скриншот файла dataout.txt Программы 2*

# Диаграмма и описание механизма взаимодействия классов Программы 3



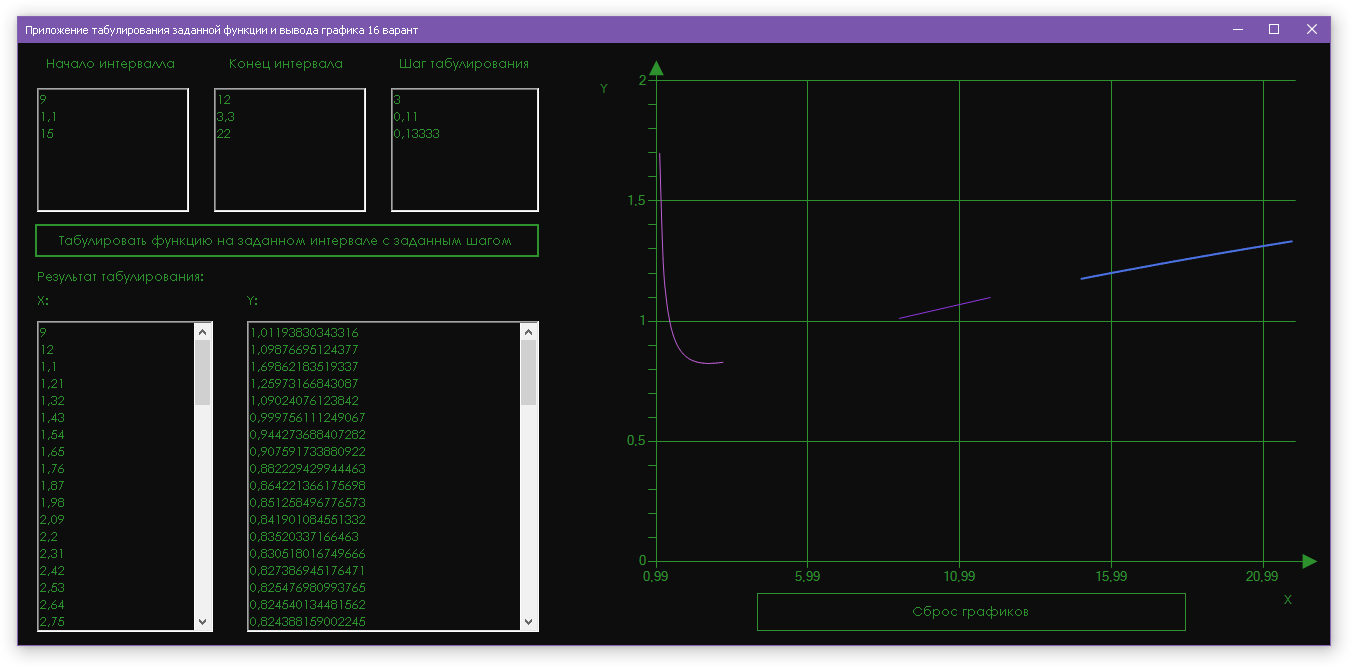
*Рисунок 9 – Диаграмма классов Программы 3*

В программе реализованы классы:

* Pointxy (описывает точку табуляции функции), хранящий в себе аргумент функции и её вычисленное значение для их вывода из класса
* RS (Read and Store) читает данные из элементов Windows Forms richTextBox`ов для табуляции функции и хранит их (для подкласса Sol), родительский класс для наследования (экземпляр этого класса в программе не используется)
* Sol решает поставленную задачу – табуляция функции, дочерний класс для наследования

Классы RS и Pointxy находятся в отношении типа has-a (композиция), т.е. класс Pointxy относится к классу RS как его часть.  
Классы RS и Sol находятся в отношении типа генерализация (наследование), т.е. все, что справедливо для класса RS справедливо и для класса Sol. Так же этим классам присуще свойство полиморфизм, которое используется в программе.

Пример работы

**

*Рисунок 10 – Пример работы Программы 3*

# ЗАКЛЮЧЕНИЕ

В курсовой работе были рассмотрены табулирование и исследование функции.

В ходе выполнения были решены все поставленные задачи.

Проведено математическое исследование функции, в ходе которого была найдена область допустимых значений функции и произведены исключения в вводных данных, что позволило исключить ошибки при табулировании и правильно вывести график.

Используя методы структурного программирования, написана программа для табулирования функции. Составлена блок-схема, которая схематично описывает алгоритм решения задачи. Увеличен навык работы со стандартными потоками ввода/вывода в C++.

С помощью иерархии классов была написаны программы табулирования функции в парадигме объектно-ориентированного программирования. Также составлены диаграммы используемых классов. Они наглядно отображают атрибуты классов, операции классов и ограничения, которые накладываются на связи между этими классами. Так же увеличен навык создания и работы с классами в C++.

В том числе, был разработан графический интерфейс пользователя с использованием инструментария Windows Forms с помощью использования компонента Chart, позволяющий выполнить задачу табулирования функции с выводом точек табуляции в график. Был также построен график исследуемой функции.

В процессе выполнения курсовой работы усовершенствованы навыки программирования на C++.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. orionll Чистая архитектура. Часть II — Парадигмы программирования / orionll. — Текст : электронный // Хабр : [сайт]. — URL: https://habr.com/ru/post/554474/ (дата обращения: 01.12.2021).
2. Юрий Уроки по С++ / Юрий. — Текст : электронный // Ravesli : [сайт]. — URL: https://ravesli.com/uroki-cpp/ (дата обращения: 03.12.2021).
3. Контроль правильности ввода в int. — Текст : электронный // Cyberforum.ru : [сайт]. — URL: https://www.cyberforum.ru/cpp-beginners/thread359236.html (дата обращения: 04.12.2021).
4. desperius cin\_buffer\_cleanup.cpp / desperius. — Текст : электронный // GitHubGist : [сайт]. — URL: https://gist.github.com/desperius/6704613 (дата обращения: 04.12.2021).
5. String (C++/CLI и C++/CX). — Текст : электронный // Техническая документация Майкрософт : [сайт]. — URL: https://docs.microsoft.com/ru-ru/cpp/extensions/string-cpp-component-extensions?view=msvc-160 (дата обращения: 05.12.2021).
6. ГОСТ 7.32-2017 СИБИД. — Текст : электронный // Техэксперт : [сайт]. — URL: https://docs.cntd.ru/document/1200157208 (дата обращения: 06.12.2021).
7. Генератор библиографического описания. — Текст : электронный // Библиотеки Тольятти : [сайт]. — URL: https://cls.tgl.ru/generator-bo/ (дата обращения: 16.12.2021).

# ПРИЛОЖЕНИЕ А

**Программа 1**

#include <iostream>

#include <cmath>

#include <fstream>

using namespace std;

int main()

{

setlocale(LC\_ALL, "rus");

bool okab = 0, okshag = 0, oktabs = 0;

char endOfProgram = 0;

int switcher, n;

double a, b, shag, \*\*mas = new double\*[2];

do

{

cout << "\nПрограмма табулирования заданной функции по введённым интервалу и шагу табулирования функции 16 вариант\n1\t- ввести интервал табулирования функции\n2\t- ввести шаг табулирования функции\n3\t- табулировать функцию\n4\t- записать табулированную функцию в solution.txt\n0\t- выход\n\nyourchose>";

cin.ignore(std::cin.rdbuf()->in\_avail());

cin.clear();

cin >> switcher;

cout << endl;

switch (switcher)

{

case 1:

{

bool nokab = 0;

do

{

cout << "Введите интервал табулирования функции [D(y):(1;+00)]\n";

cout << "a: ";

cin.ignore(std::cin.rdbuf()->in\_avail());

cin.clear();

cin >> a;

cout << "b: ";

cin >> b;

cout << endl;

if (a <= 1 || b <= 1)

{

okab = 0;

cout << "Вы ввели интервал не удовлетворяющий О.Д.З. функции: D(y):(1;+00)\t(или же x>1)\n1\t- ввести интервал табулирования функции заново\n0\t- выход\n\nyourchose>";

cin.ignore(std::cin.rdbuf()->in\_avail());

cin.clear();

cin >> switcher;

cout << endl;

switch (switcher)

{

default:

nokab = 1;

case 1:

switcher = 42;

break;

}

}

else

okab = 1;

} while (!(okab || nokab));

}

break;

case 2:

{

if (!okab) {

cout << "Сначала введите интервал табулирования функции\n0\t- выход\n\nyouhavenochose>";

cin.ignore(std::cin.rdbuf()->in\_avail());

cin.clear();

cin.get();

break;

}

bool nokshag = 0;

do {

cout << "Введите шаг табулирования функции\n(Шаг табулирования функции не должен быть больше интервала табулирования функции)\nshag: ";

cin.ignore(std::cin.rdbuf()->in\_avail());

cin.clear();

cin >> shag;

cout << endl;

if (((shag > 0) && ((b - a) < 0)) || (((b - a) > 0) && (shag < 0)))

shag = -shag;

if (abs(b - a) >= abs(shag) && abs(shag) > 0)

okshag = 1;

else

{

okshag = 0;

cout << "Вы ввели шаг табулирования функции больше интервала табулирования функции\n1\t- ввести шаг табулирования функции заново\n0\t- выход\n\nyourchose>";

cin.ignore(std::cin.rdbuf()->in\_avail());

cin.clear();

cin >> switcher;

cout << endl;

switch (switcher)

{

default:

nokshag = 1;

case 1:

switcher = 42;

break;

}

}

} while (!(okshag || nokshag));

}

break;

case 3:

{

if (!(okab && okshag))

{

cout << "При вводе интервала и шага табулирования функции что-то пошло не так, функция не может быть табулирована\n0\t- выход\n\nyouhavenochose>";

cin.ignore(std::cin.rdbuf()->in\_avail());

cin.clear();

cin.get();

cout << endl;

switcher = 42;

}

else

{

cout << "Интервал и шаг табулирования функции введены\n";

n = static\_cast<int>((b - a) / shag + 1);

if (oktabs)

{

delete[] mas[0];

delete[] mas[1];

delete[] mas;

}

for (int i = 0; i < 2; i++)

mas[i] = new double[n];

for (int i = 0; i < n; i++)

{

mas[0][i] = a + i \* shag;

mas[1][i] = sqrt(mas[0][i] / (4 \* log(mas[0][i])));

}

cout << "Функция табулирована на заданном интервале с заданным шагом\n";

oktabs = 1;

// Проверка/вывод табуляции

cout << "x: " << '\t' << "y: " << '\n';

for (int i = 0; i < n; i++)

cout << mas[0][i] << '\t' << mas[1][i] << '\n';

cout << endl;

}

break;

}

case 4:

{

if (oktabs != 1)

{

cout << "Функция не табулирована, сначала табулируйте функцию\n0\t- выход\n\nyouhavenochose>";

cin.ignore(std::cin.rdbuf()->in\_avail());

cin.clear();

cin.get();

cout << endl;

}

else

{

ofstream outfile;

outfile.open("solution.txt");

outfile << "x: " << '\t' << "y: " << '\n';

for (int i = 0; i < n; i++)

outfile << mas[0][i] << '\t' << mas[1][i] << '\n';

outfile << endl;

outfile.close();

cout << "Табулированная функция записана в файл solution.txt" << '\n' << '\n';

}

break;

}

case 0:

{

endOfProgram = 1;

break;

}

default:

{

switcher = 42;

break;

}

}

} while (!endOfProgram);

if (oktabs)

{

delete[] mas[0];

delete[] mas[1];

delete[] mas;

}

return 0;

}

# ПРИЛОЖЕНИЕ Б

**Программа 2**

#include <iostream>

#include "Sol.h"

int main()

{

setlocale(LC\_ALL, "rus");

{

int switcher;

bool outwhile = 0;

Sol \*A = nullptr;

do

{

std::cout << "\nПрограмма табулирования заданной функции по прочитанным из файла интервалу и шагу табулирования функции 16 вариант\n1\t- прочитать интервал и шаг табулирования функции и табулировать функцию\n2\t- записать табулированную функцию в dataout.txt\n0\t- выход\n\nyourchose>";

std::cin.ignore(std::cin.rdbuf()->in\_avail());

std::cin.clear();

std::cin >> switcher;

switch (switcher)

{

case 1:

A = new Sol;

break;

case 2:

if ((A != nullptr) || (A))

DateOut B(A);

else

std::cout << "\nСначала следует прочитать интервал и шаг табулирования функции и табулировать функцию\n\n";

break;

case 0:

outwhile = 1;

default:

break;

}

} while (!outwhile);

}

return 0;

}

**Заголовочный файл “Sol.h”**

#pragma once

#include <vector>

struct Point

{

double x, y;

};

class RS //ReadandStore

{

protected:

std::vector <double> a, b, shag;

RS();

Point \*mas;

public:

RS(RS& A) { delete[] mas; this->a = A.a; this->b = A.b; this->shag = A.shag; delete &A; };

~RS() {};

RS operator=(RS& A) { this->a = A.a; this->b = A.b; this->shag = A.shag; };

};

class Sol : private RS

{

int \*n;

public:

Sol();

Sol(Sol& A) { for (int i = 0; i <= A.a.size(); i++) this->n[i] = A.n[i]; for (int j = 0; j < n[A.a.size()]; j++) this->mas[j] = A.mas[j]; this->a = A.a; this->b = A.b; this->shag = A.shag; delete &A; };

~Sol() { delete[] n; };

Sol operator=(Sol& A) { this->n = A.n; this->mas = A.mas; this->a = A.a; this->b = A.b; this->shag = A.shag; };

friend class DateOut;

};

class DateOut

{

public:

DateOut(Sol \*A);

};

**Файл кода “Sol.cpp”**

#include "Sol.h"

#include <cmath>

#include <iostream>

#include <fstream>

#include <vector>

using str = std::string;

double Fy(double x) { return sqrt(x / (4 \* log(x))); };

RS::RS()

{

std::ifstream datain;

{

std::cout << "Сохраните исходные данные в строку как « Начало\_интервала Конец\_интервала Шаг » в файл datain.txt в папке с программой\nДля продолжения введите любой символ...\n";

std::cin.ignore(std::cin.rdbuf()->in\_avail());

std::cin.clear();

std::cin.get();

}

{

int outwhile = 0;

do

{

try

{

datain.open("datain.txt", std::ios::\_Nocreate);

}

catch (...)

{

int switcher;

std::cout << "Произошла ошибка при открытии файла\n1\t- попробовать открыть файл снова\n0\t- выход\n\nyourchose>";

std::cin.ignore(std::cin.rdbuf()->in\_avail());

std::cin >> switcher;

switch (switcher)

{

case 0:

outwhile = 1;

case 1:

default:

break;

}

}

if (!datain.is\_open())

{

int switcher;

std::cerr << "Вы не сохранили исходные данные в datain.txt\n1\t- попробовать открыть файл снова\n0\t- выход\n\nyourchose>";

std::cin.ignore(std::cin.rdbuf()->in\_avail());

std::cin.clear();

std::cin >> switcher;

switch (switcher)

{

case 0:

outwhile = 1;

case 1:

default:

break;

}

}

} while ((!datain.is\_open()) && (outwhile != 1));

}

if (datain.is\_open())

{

double Itera, Iterb, Itershag;

bool neych = 0;

int i = 0;

std::cout << "\nПрочитанные данные:\na\tb\tшаг\n";

while (!datain.eof())

{

datain >> Itera >> Iterb >> Itershag;

if (((Itera < 1) || (Iterb < 1)) || (abs(Itershag) > abs(Iterb - Itera)))

{

std::cout << "Набор не учтён\n";

neych = 1;

}

else {

if (((Itershag > 0) && ((Iterb - Itera) < 0)) || (((Iterb - Itera) > 0) && (Itershag < 0)))

Itershag = -Itershag;

a.push\_back(Itera); b.push\_back(Iterb); shag.push\_back(Itershag);

std::cout << a.at(i) << '\t' << b.at(i) << '\t' << shag.at(i) << '\n';

i++;

}

}

datain.close();

if (neych)

std::cout << "\nВ одном из наборов « Начало\_интервала Конец\_интервала Шаг » в файле datain.txt\nили-\tинтервал не удовлетворяет О.Д.З. функции: D(y):(1;+00)\t(или же x>1)\nили-\tшаг табулирования функции больше интервала табулирования функции\n\n";

else std::cout << '\n';

if (!(a.size() == b.size()) && (a.size() == shag.size()))

{

std::cout << "Наборы « Начало\_интервала Конец\_интервала Шаг » в файле datain.txt оказались не полные\n";

a.clear(); b.clear(); shag.clear();

std::cout << "Прочитанные данные удалены\n";

}

}

}

Sol::Sol()

{

if (!(a.empty()))

{

n = new int[a.size() + 1];

n[a.size()] = 0; //кол-во всех точек

for (int i = 0; i < a.size(); i++)

{

n[i] = static\_cast<int>((b.at(i) - a.at(i)) / shag.at(i) + 1);

n[a.size()] += n[i];

}

mas = new Point[n[a.size()]];

{

int iter = 0;

std::cout << "\nТабулированая функция:\nx: \ty: \n";

for (int i = 0; i < a.size(); i++) //Для каждой строки

for (int k = 0; k < n[i]; k++) //Для каждой точки интервала строки

{

mas[iter].x = a[i] + k \* shag[i];

mas[iter].y = Fy(mas[iter].x);

std::cout << mas[iter].x << '\t' << mas[iter].y << '\n';

iter++; //после каждой записи увеличивается, должен быть равен n[a.size()] после всех итераций

}

std::cout << "\n";

} // здесь уже массив должен быть заполнен точками

}

};

DateOut::DateOut(Sol \*A)

{

if (!A->a.empty()) {

std::ofstream outfile;

outfile.open("dataout.txt");

outfile << "x: " << '\t' << "y: " << '\n';

for (int i = 0; i < A->n[A->a.size()]; i++)

outfile << A->mas[i].x << '\t' << A->mas[i].y << '\n';

outfile << "\n";

outfile.close();

std::cout << "Табулированная функция записана в dataout.txt" << '\n' << '\n';

}

else

std::cout << "\nДанные не прочитаны\nСначала следует прочитать интервал и шаг табулирования функции и табулировать функцию\n\n";

}

# ПРИЛОЖЕНИЕ В

**Программа 3**

#include "Sol.h"

...

#pragma endregion

int ser = 0;

private: System::Void button1\_Click(System::Object^ sender, System::EventArgs^ e)

{

if ((richTextBox1->Lines->Length > 0) && (richTextBox2->Lines->Length > 0) && (richTextBox3->Lines->Length > 0))

{

Sol A;

RS \*B = &A;

B->inputabshag(richTextBox1, richTextBox2, richTextBox3);

for (int j = 0;j < B->numOfNaborov;j++)

{

DataVisualization::Charting::Series^ seriesser = (gcnew DataVisualization::Charting::Series());

seriesser->ChartArea = chart1->Series[ser]->ChartArea;

seriesser->ChartType = chart1->Series[ser]->ChartType;

seriesser->Font = chart1->Series[ser]->Font;

seriesser->LabelForeColor = chart1->Series[ser]->LabelForeColor;

seriesser->Name = L"Series" + Convert::ToString(ser + 2);

seriesser->BorderWidth = (ser % 2) + 1;

chart1->Series->Add(seriesser);

for (int i = 0;i < B->outputXY(j).size();i++)

{

chart1->Series[ser]->Points->AddXY(B->outputXY(j)[i].x, B->outputXY(j)[i].y);

richTextBox4->AppendText(System::Convert::ToString(B->outputXY(j)[i].x) + "\n");

richTextBox5->AppendText(System::Convert::ToString(B->outputXY(j)[i].y) + "\n");

}

ser++;

}

}

else

{

MessageBox::Show("Вы не ввели интервал или шаг\nВведите данные", "Не считается", MessageBoxButtons::OK, MessageBoxIcon::Asterisk);

};

}

private: System::Void button2\_Click(System::Object^ sender, System::EventArgs^ e) {

while (ser > 0) {

chart1->Series->Remove(chart1->Series[ser]);

ser--;

}

chart1->Series[ser]->Points->Clear();

richTextBox4->Text = "";

richTextBox5->Text = "";

}

**Заголовочный файл “Sol.h”**

#pragma once

#include <cmath>

#include <set>

#include <vector>

class Pointxy

{

public:

double x, y;

Pointxy(double X = 0, double Y = 0) :x(X), y(Y) {};

};

class RS

{

protected:

virtual double Fy(double x) = 0;

std::vector<std::set<double>> x;

public:

int numOfNaborov;

virtual void inputabshag(System::Windows::Forms::RichTextBox ^ ai, System::Windows::Forms::RichTextBox ^ bi, System::Windows::Forms::RichTextBox ^ shagi);

virtual std::vector<Pointxy> outputXY(int i);

};

class Sol : public RS

{

double Fy(double x) { return sqrt(x / (4 \* log(x))); };

public:

Sol() {};

};

**Файл кода “Sol.cpp”**

#include "stdafx.h"

#include "Sol.h"

#include <vector>

#include <set>

#include <algorithm>

void RS::inputabshag(System::Windows::Forms::RichTextBox ^ ai, System::Windows::Forms::RichTextBox ^ bi, System::Windows::Forms::RichTextBox ^ shagi)

{

if ((ai->Lines->Length == bi->Lines->Length) && (ai->Lines->Length == shagi->Lines->Length))

{

std::vector<double> a;

std::vector<double> b;

std::vector<double> shag;

try

{

bool err = 0;

using namespace System::Windows::Forms;

for (int i = 0; i < ai->Lines->Length; i++)

if (ai->Lines[i] != "")

if (System::Convert::ToDouble(ai->Lines[i]) > 1)

a.push\_back(System::Convert::ToDouble(ai->Lines[i]));

else

{

MessageBox::Show("Вы ввели интервал не удовлетворяющий О.Д.З. функции: D(y):(1;+00)\t(или же x>1)\nВведите корректные данные", "Ошибка в первой границе интервала", MessageBoxButtons::OK, MessageBoxIcon::Asterisk);

err = 1;

}

for (int i = 0; i < bi->Lines->Length; i++)

if (bi->Lines[i] != "")

if (System::Convert::ToDouble(bi->Lines[i]) > 1)

b.push\_back(System::Convert::ToDouble(bi->Lines[i]));

else

{

MessageBox::Show("Вы ввели интервал не удовлетворяющий О.Д.З. функции: D(y):(1;+00)\t(или же x>1)\nВведите корректные данные", "Ошибка во второй границе интервала", MessageBoxButtons::OK, MessageBoxIcon::Error);

err = 1;

}

if (!err)

for (int i = 0; i < shagi->Lines->Length; i++)

if (shagi->Lines[i] != "")

if (abs(System::Convert::ToDouble(shagi->Lines[i])) <= abs(b[i] - a[i]))

if (((System::Convert::ToDouble(shagi->Lines[i]) > 0) && ((b[i] - a[i]) < 0)) || (((b[i] - a[i]) > 0) && (System::Convert::ToDouble(shagi->Lines[i]) < 0)))

shag.push\_back(-(System::Convert::ToDouble(shagi->Lines[i])));

else

shag.push\_back(System::Convert::ToDouble(shagi->Lines[i]));

else

{

MessageBox::Show("Вы ввели шаг табулирования функции больше интервала табулирования функции\nВведите корректные данные", "Ошибка в шаге", MessageBoxButtons::OK, MessageBoxIcon::Error);

err = 1;

}

if (err) {

a.clear();

b.clear();

MessageBox::Show("Количество интервалов и/или шагов не одинаковое, введите равное количество интервалов и/или шагов", "Ошибка во вводных данных", MessageBoxButtons::OK, MessageBoxIcon::Error);

return;

}

}

catch (System::FormatException ^e)

{

using namespace System::Windows::Forms;

a.clear();

b.clear();

MessageBox::Show(e->Message + "\n(Следует использовать запятую как десятичный разделитель)", "Ошибка во вводных данных", MessageBoxButtons::OK, MessageBoxIcon::Error);

}

int \* n; std::set<double> xi;

n = new int[a.size()];

for (int i = 0; i < a.size(); i++)

{

n[i] = (b[i] - a[i]) / shag[i] + 1;

xi.clear();

for (int j = 0; j < n[i]; j++)

xi.insert(a[i] + j \* shag[i]);

x.push\_back(xi);

};

numOfNaborov = static\_cast<int>(x.size());

}

else

{

using namespace System::Windows::Forms;

MessageBox::Show("Количество интервалов и/или шагов не одинаковое, введите равное количество интервалов и/или шагов", "Ошибка во вводных данных", MessageBoxButtons::OK, MessageBoxIcon::Asterisk);

}

}

std::vector<Pointxy> RS::outputXY(int i)

{

std::vector<Pointxy> \* XY = new std::vector<Pointxy>;

std::for\_each(x[i].begin(), x[i].end(), [XY, this](double i) mutable

{

Pointxy XYo(i, Fy(i));

XY->push\_back(XYo);

});

return \*XY;

}