МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**Липецкий Государственный Технический Университет**

Факультет автоматизации и информатики

Кафедра автоматизированных систем управления

Лабораторная работа

по программированию №3

“Наследование и полиморфизм”

Студент \_\_\_\_\_\_\_\_\_\_\_\_ Станиславчук С. М.

(подпись, дата)

Группа АС-21-1

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_ Ведищев В. В.

(подпись, дата)

Липецк 2022 г.

Содержание

2. Задание

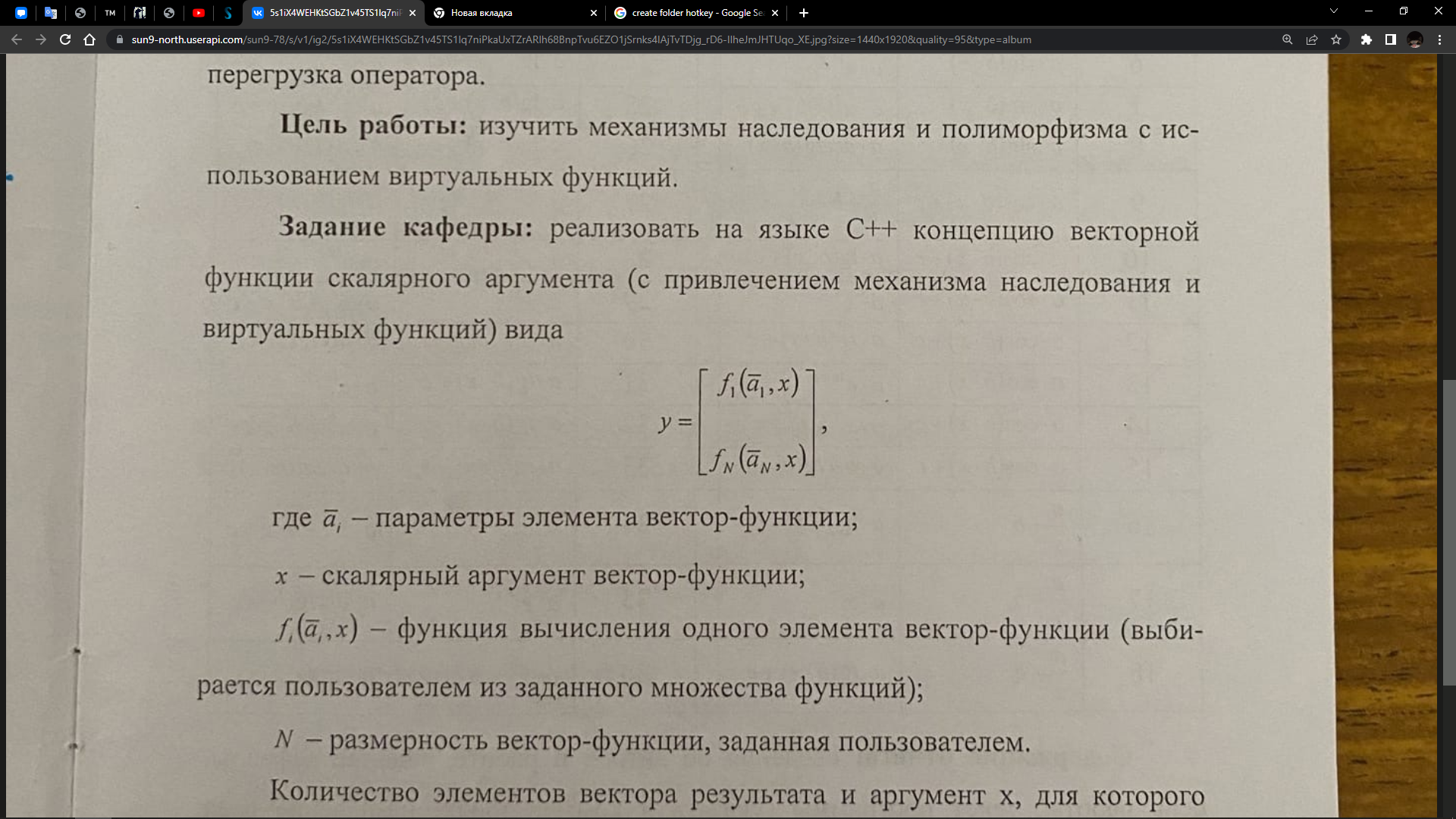
3. Цель

4. Код программы

5. Вывод

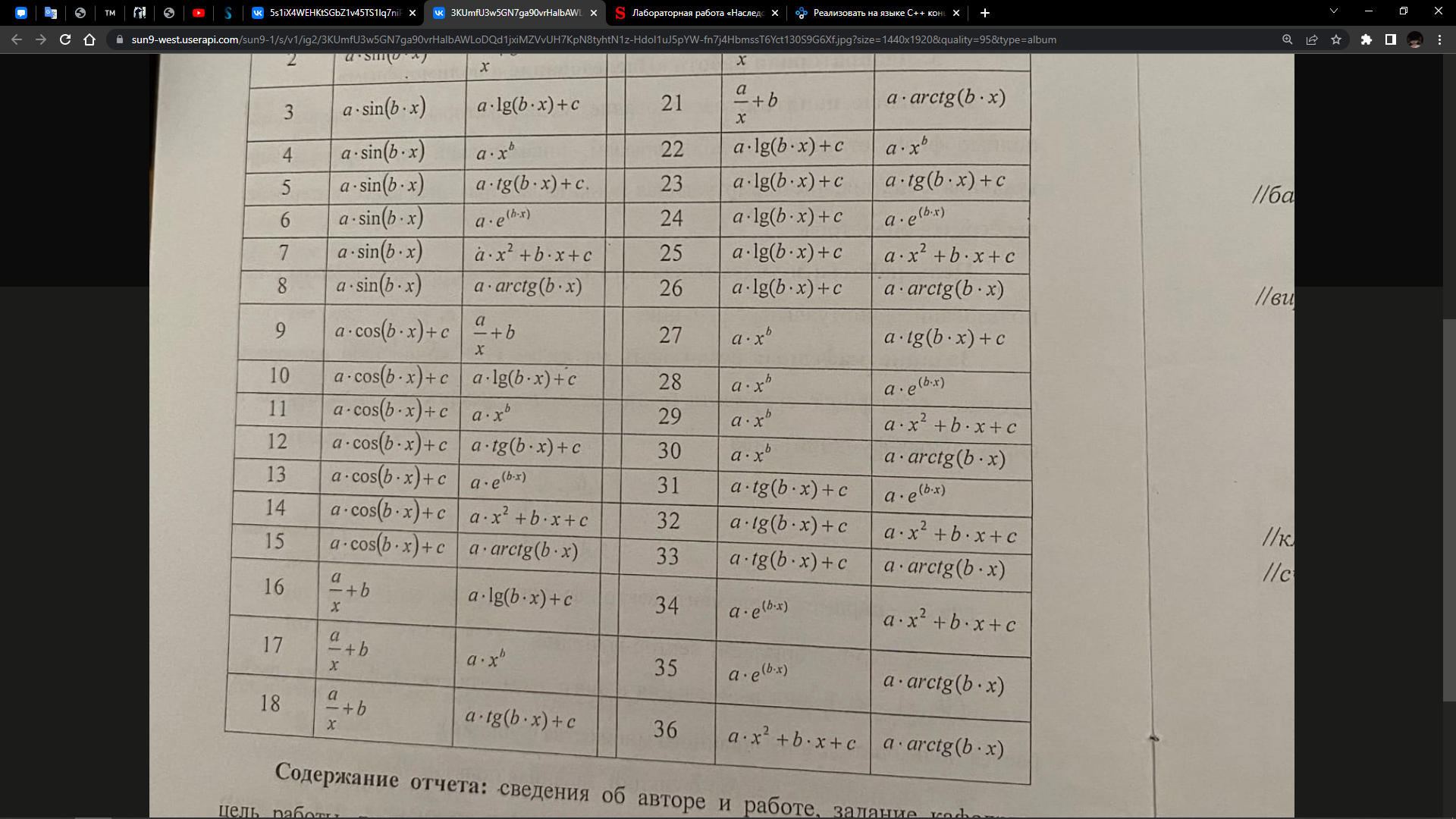
2. Задание

Реализовать на языке C++ концепцию векторной функции скалярного аргумента (с привлечением механизма наследования и виртуальных функций) вида



Количество элементов вектора результата и аргумент x, для которого производится вычисление, задается пользователем. Вид функции, по которому вычисляется значение каждого элемента вектора результата, указывается пользователем из заданного множества параметризированных функций. Параметры функции вводятся пользователем для каждого элемента вектора результата отдельно, при формировании вектора функции.

Вариант 11:



3. Цель работы.

Изучить механизмы наследования и полиморфизма с использованием виртуальных функций.

4. Код программы.

#include <iostream>

using namespace std;

class Base {

public:

virtual void Input\_Data() = 0;

virtual double Func(double x) = 0;

double result = 0;

protected:

double a = 0, b = 0;

};

class Cos : public Base {

private:

void Input\_Data() {

cout << "Input a, b, c: ";

while (!(cin >> a >> b >> c) || (cin.peek() != '\n')) {

cin.clear();

while (cin.get() != '\n');

cout << "Invalid values! " << endl;

}

}

double Func(double x) {

result = a \* cos(b \* x) + c;

return result;

}

double c = 0;

};

class Pow : public Base {

private:

void Input\_Data() {

cout << "Input a, b: ";

while (!(cin >> a >> b) || (cin.peek() != '\n')) {

cin.clear();

while (cin.get() != '\n');

cout << "Invalid values! " << endl;

}

}

double Func(double x) {

result = a \* pow(x, b);

return result;

}

};

int main() {

int N, choice;

double x; int choise;

cout << "Input number of elements of the vector-function: "; // #1

while (!(cin >> N) || (cin.peek() != '\n') || (N <= 0)) {

{

cin.clear();

while (cin.get() != '\n');

cout << "Invalid value! " << endl;

}

} // #1

Base\*\* Vector = new Base \*[N];

for (int i = 0; i < N; i++) {

cout << "Type of function is 1:(a\*cos(b\*x)+c) or 2:(a\*x^b)?: "; // #2

while (!(cin >> choice) || (cin.peek() != '\n') || (choice != 1) && (choice != 2)) {

cin.clear();

while (cin.get() != '\n');

cout << "Invalid value! " << endl;

} // #2

switch (choice) {

case 1:

Vector[i] = new Cos();

Vector[i]->Input\_Data();

break;

case 2:

Vector[i] = new Pow();

Vector[i]->Input\_Data();

break;

default:

i--;

cout << "Invalid choise! ";

break;

}

}

while (true) {

cout << "Input var x: ";

while (!(cin >> x) || (cin.peek() != '\n')) {

cin.clear();

while (cin.get() != '\n'); {

cout << "Invalid value! " << endl;

cout << "1:Retry again or 2:exit?" << endl;

cin >> choice;

switch (choice) {

case 1: break;

case 2: return 0;

default:

cout << "Invalid choise! ";

break;

}

}

}

for (int i = 0; i < N; i++) {

Vector[i]->Func(x);

cout << "Vector[" << i << "] = " << Vector[i]->result << endl;

}

}

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

delete Vector[i];

delete[] Vector;

cin.get();

return 0;

}  
Вывод: изучил механизмы наследования и полиморфизма с использованием виртуальных функций.