ГУАП

КАФЕДРА № 14

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

| ассистент | | Н.Ю. Чумакова |
|---|---|---|
| должность, уч. степень, звание | подпись, дата | инициалы, фамилия |

# ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ

# ВИРТУАЛЬНЫЕ ФУНКЦИИ

по курсу: ТЕХНОЛОГИЯ ПРОГРАММИРОВАНИЯ

РАБОТУ ВЫПОЛНИЛ

| СТУДЕНТ ГР. № 1042 | | Д.А. Васейко |
|---|---|---|
| | подпись, дата | инициалы, фамилия |

Санкт-Петербург 2022

## 1. Постановка задачи

Вариант 3: Создать абстрактный класс «Кривые» для вычисления координаты y для некоторой x. Создать производные классы «Прямая», «Эллипс», «Гипербола» со своими функциями вычисления y в зависимости от входного параметра x.

## 2. Формализация задачи

Данная программа разбита на восемь файлов: файл родительского абстрактного класса (Parr.h), два файла дочернего класса «Прямая» (Straight.h и Straight.cpp), два файла дочернего класса «Эллипс» (Ellips.h и Ellips.cpp), два файла дочернего класса «Гипербола» (Hyperbola.h и Hyperbola.cpp), и один главный (main.cpp) – это файлы с объявлением, определением методов класса соответственно, а также файл с управляющей функцией.

В каждом дочернем классе есть свои приватные переменные – это координаты и коэффициенты для каждой кривой соответственно. Все они объявлены типом float. Для прямой: координаты X, У, коэффициенты К и В: координата У находится по уравнению «y = kx+b». Для эллипса: координаты X, У, полуоси А и В (причем А и В должны быть больше нуля): координата У находится по уравнению «y = b*sqrt((1-x/a)(1+1/a))». Для гиперболы: координаты X, У, коэффициенты К и В: координата У находится по уравнению «y = k/x + b» (причем X не должен быть равен нулю). Каждое исключение в виде недопустимых значений обрабатывается, и вместо введенного значения, переменной присваивается единица.

Пользователю доступно выбрать количество классов наследников.

В абстрактном классе Parr наличествует чистая виртуальная функция float findY, которая переопределяется в каждом дочернем классе в зависимости от нужного результата. Все дочерние классы наследуют Parr публично. .

## 3. Исходный код

Файл Curve.h

```
#pragma once

class Parr {
public:
	Parr() {};
	~Parr() {};
	virtual float findY() = 0;

};
```

Файл Straight.h

```
#pragma once
#include"Parr.h"
//y=kx+b
class Straight final : public Parr {
private:
	float y = 0.0;
	float x;
	float k;
	float b;
public:
	Straight();
	Straight(float x_var, float k_var = 1.0, float b_var = 0.0);
	Straight(const Straight& copy_straight);
	virtual ~Straight();

	float findY() override;
```

}; Файл Straight.cpp

```
#include <iostream>
#include <cmath>
#include <locale>
#include "Straight.h"

using namespace std;

Straight::Straight() : x(0.0), y(0.0), k(1.0), b(0.0) {}
Straight::~Straight() {}

Straight::Straight(float x_var, float k_var, float b_var)
{
	x = x_var;
	k = k_var;
	b = b_var;
}

Straight::Straight(const Straight& copy_straight)
{
	x = copy_straight.x;
	y = copy_straight.y;
	k = copy_straight.k;
	b = copy_straight.b;
}

float Straight::findY()
{
	y = k * x + b;
	return y;
}
```

Файл Ellips.h

```
#pragma once
```

```
#include"Parr.h"
//y=sqrt((1-(x^x)/(a^a))(b^b))
class Ellips final : public Parr {
private:
	float x;
	float y = 0.0;
	float a;
	float b;
public:
	Ellips();
	Ellips(float x_var, float a_var = 1.0, float b_var = 1.0); //круг по умолчанию
	Ellips(const Ellips& copy_ellips);
	virtual ~Ellips();

	float findY() override;
};
```

Файл Ellips.cpp

```
#include"Ellips.h"
#include <iostream>
#include <cmath>
#include <locale>

using namespace std;

Ellips::Ellips() : x(1.0), y(0.0), a(1.0), b(1.0) {}
Ellips::~Ellips() {}

Ellips::Ellips(float x_var, float a_var, float b_var)
{
	a = a_var;
	b = b_var;
	x = x_var;
}

Ellips::Ellips(const Ellips& copy_ellips)
{
	x = copy_ellips.x;
	y = copy_ellips.y;
	a = copy_ellips.a;
	b = copy_ellips.b;
}

float Ellips::findY()
{
	float sqrt_num = (1 - x / a) * (1 + x / a);
	if (sqrt_num < 0)
		return y = -1;
	y = b * sqrt(sqrt_num);
	return y;
}
```

Файл Hyperbola.h

```
#pragma once
#include "Parr.h"

class Hyperbola final : public Parr {
private:
	float x;
	float y = 0.0;
	float k;
	float b;
public:
	Hyperbola();
```

```
        Hyperbola(float x_var, float k_var = 1.0, float b_var = 0);
        Hyperbola(const Hyperbola& copy_hyper);
        virtual ~Hyperbola();

        float findY() override;
```

}; Файл Hyperbola.cpp

```
#include "Hyperbola.h”
#include <iostream>
#include <cmath>
#include <locale>

using namespace std;

Hyperbola::Hyperbola() : x(1.0), y(0.0), k(1.0), b(0.0) {}
Hyperbola::~Hyperbola() {}

Hyperbola::Hyperbola(float x_var, float k_var, float b_var)
{
        x = x_var;
        k = k_var;
        b = b_var;
}

Hyperbola::Hyperbola(const Hyperbola& copy_hyper)
{
        x = copy_hyper.x;
        y = copy_hyper.y;
        k = copy_hyper.k;
        b = copy_hyper.b;
}

float Hyperbola::findY()
{
        y = k / x + b;
        return y;
}
```

Файл main.cpp

```
#include <iostream>
#include <cmath>
#include <locale>
#include <string>
#include "Parr.h"
#include "Straight.h"
#include "Ellips.h"
#include "Hyperbola.h"

using namespace std;

/*Вариант 3 - Создать абстрактный класс "Кривые" для вычисления координаты y для некоторой
x.
Создать производные классы "Прямая", "Эллипс", "Гипербола" со своими функциями вычисления y
в зависимости от входного параметра x*/



int main()
{
        setlocale(LC_ALL, "Rus");

        int c, countY = 0, exit_equ = 0, i = 0, N;
```

```
	float x_ell, a_ell, b_ell; //ellips
	float x_str, k_str, b_str; //straight

	float x_hyp, k_hyp, b_hyp; //hyperbola

	cout << "Как много наследников вы хотите??" << endl;
	cout << "--> ";
	cin >> N;

	Parr** parr = new Parr * [N];

	while (exit_equ != 1 && N > 0)
	{
		cout << "С чем вы хотите работать" << endl;
		cout << "1 - Прямая" << endl;
		cout << "2 - Эллипс" << endl;
		cout << "3 - Гипербола" << endl;
		cout << "0 - Выход" << endl;
		cout << "--> ";
		cin >> c;

		if (c == 1)
		{
			system("cls");
			cout << "Введите коэфициенты для прямой через пробел (x, k, b): ";
			cin >> x_str >> k_str >> b_str;
			parr[i] = new Straight;
			Straight straight(x_str, k_str, b_str);
			parr[i] = &straight;
			cout << "Координата Y прямой: " << parr[i]->findY() << ", при введенных
X = " << x_str << ", K = " << k_str << ", B = " << b_str << endl;
			i++;
			N--;
		}

		if (c == 2)
		{
			system("cls");
			cout << "Введите коэфициенты для эллипса через пробел (x, a, b): ";
				cin >> x_ell >> a_ell >> b_ell;
				if (a_ell == 0)
				{
					cout << "Значение A для эллипса не может быть
нулевым, поэтому A будет равен единице" << endl;
					a_ell = 1;
				}
				if (b_ell == 0)
				{
					cout << "Значение B для эллипса не может быть
нулевым, поэтому B будет равен единице" << endl;
					b_ell = 1;
				}
			parr[i] = new Ellips;
			Ellips ellips(x_ell, a_ell, b_ell);
			parr[i] = &ellips;
			cout << "Координата Y эллипса: " << parr[i]->findY() << ", при
введенных X = " << x_ell << ", A = " << a_ell << ", B = " << b_ell << endl;
			i++;
			N--;
		}
		if (c == 3)
		{
			system("cls");
			cout << "Введите коэффициенты для гиперболы через пробел (x, k, b): ";
```

```
                        cin >> x_hyp >> k_hyp >> b_hyp;
                        if (x_hyp == 0)
                        {
                                cout << "Значение X для гиперболы не может быть нулевым, поэтому
X будет равен единице" << endl;
                                x_hyp = 1;
                        }
                        parr[i] = new Hyperbola;
                        Hyperbola hyperbola(x_hyp, k_hyp, b_hyp);
                        parr[i] = &hyperbola;
                        cout << "Координата Y гиперболы: " << parr[i]->findY() << ", при
введенных X = " << x_hyp << ", K = " << k_hyp << ", B = " << b_hyp << endl;
                }
                if (c == 0)
                {
                        exit_equ = 1;
                }
        }

        return 0;

}
```

## 4. Результаты работы программы

Для демонстрации работы введем три примера: с отрицательными значениями, с нулем у эллипса, с нулем у гиперболы. Отрицательные числа введем в файл, а нули будем вводить вручную. Одновременно с этим продемонстрируем возможность перезаписи файла.

Пример первый. Значения прямой (X, K, B): -10, 5.5, 8. Значения эллипса (X, A, B): -1, 4.33, 6.99. Значения гиперболы (X, K, B): 1, 2, 3.

Программа встречает нас следующим экраном (рис. 1)

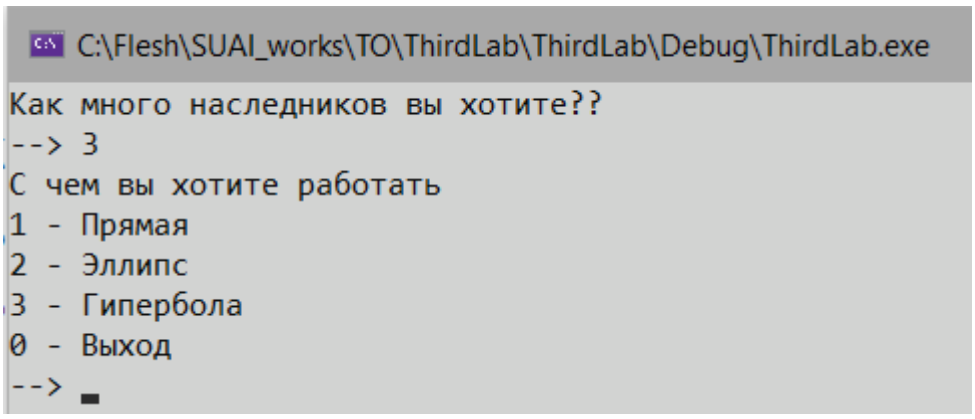


Рисунок 1 – Активация программы

Выбираем прямую. (рис. 2)

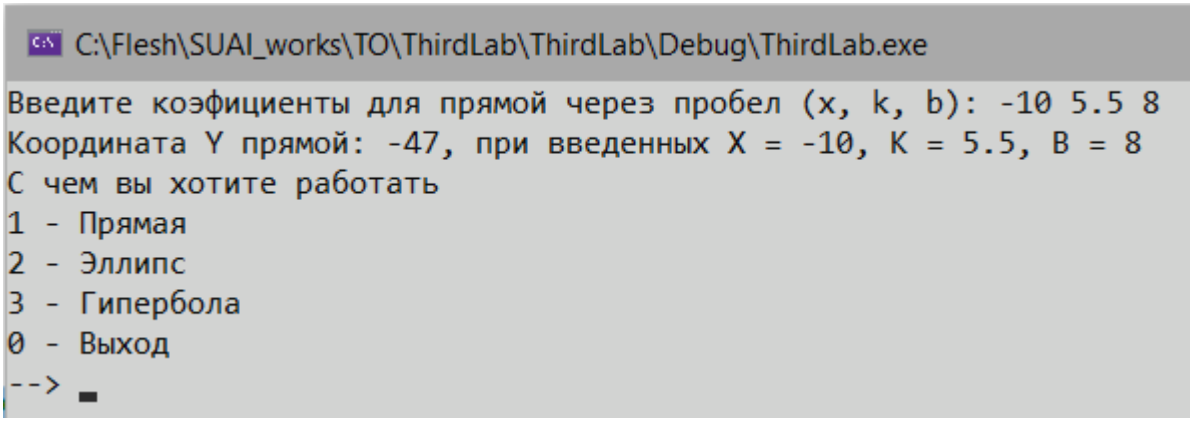


Рисунок 2 – Выбираем прямую

Выбираем эллипс. (рис. 3)

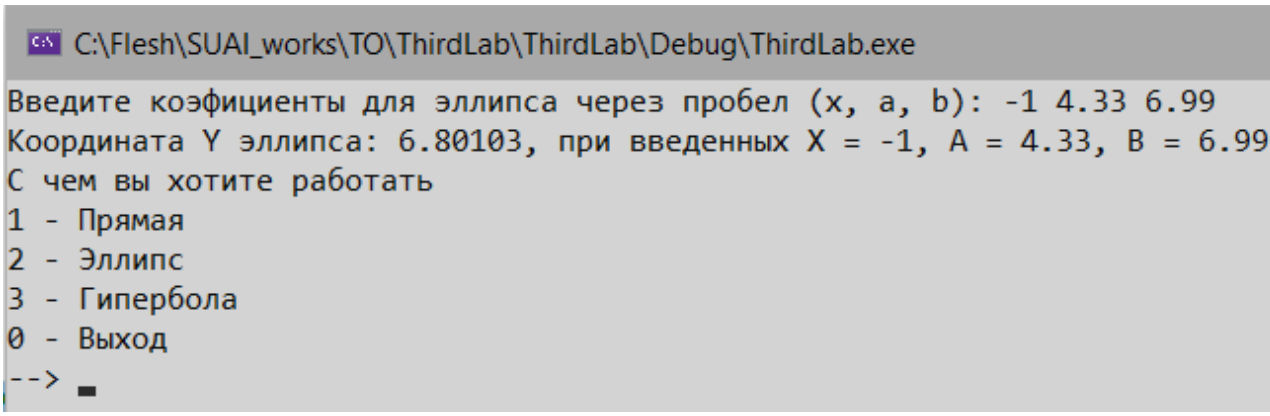


Рисунок 3 – Выбираем эллипс

Выбираем гиперболу (рис. 4)

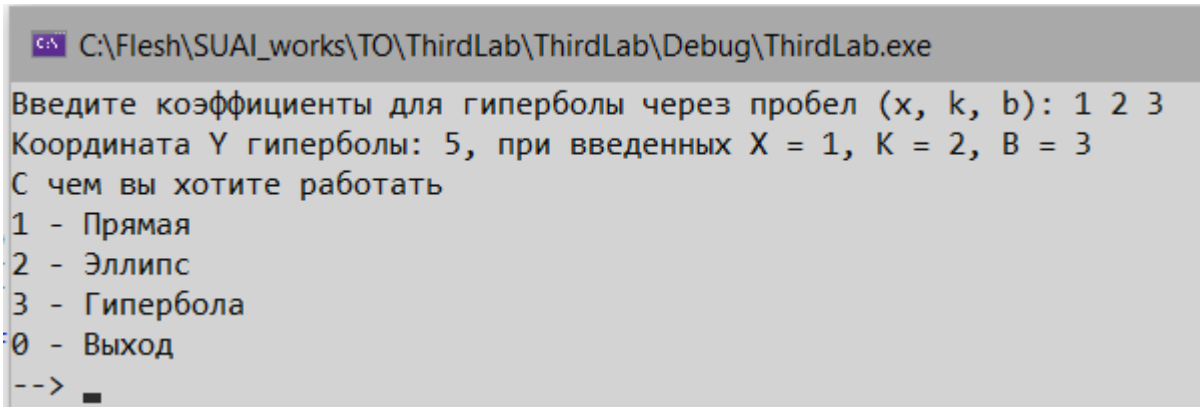


Рисунок 4 – Выбираем гиперболу

## 5. Выводы

В процессе выполнения лабораторной работы мы изучили основы применения виртуальных функций, вспомнили наследование классов. Изучили принципы работы виртуальных функций через таблицы виртуальных функций; продемонстрировали их применение через создание указателя родительского класса на объект дочернего класса.