



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Балтийский государственный технический университет «ВОЕНМЕХ» им. Д.Ф. Устинова»
(БГТУ «ВОЕНМЕХ» им. Д.Ф. Устинова)

Факультет

О

Естественнонаучный

шифр

наименование

Кафедра

О7

Информационные системы и программная инженерия

шифр

наименование

Дисциплина

Информационные технологии и программирование

ПРАКТИЧЕСКАЯ РАБОТА №1

Классы

Вариант №6

Выполнил студент группы

О722Б

Вяткин Н.А.

Фамилия И.О.

ПРЕПОДАВАТЕЛЬ

Гладевич А.А.

Фамилия И.О.

Подпись

« 17 »

февраля

2023 г.

Цель работы

Изучить понятие класса, механизмы работы с классами, научиться обеспечивать вызов методов и обращение к полям классов, изменять видимость компонентов в определении класса с использованием спецификаторов доступа, перегружать операции и использовать дружественные функции.

Постановка задачи

Описать класс в соответствии с индивидуальным вариантом задания и реализовать все его методы. Каждый класс должен содержать, помимо указанных в варианте методов, конструктор с параметрами, конструктор копирования, деструктор, методы ввода с клавиатуры, установки и получения значений полей, вывода этих значений на экран. В каждом методе класса, включая конструкторы и деструктор, предусмотреть отладочную печать сообщения, содержащего имя метода. Написать программу для тестирования всех методов класса, выбор метода должен осуществляться с помощью меню.

Вариант 6

Класс Треугольник.

Поля: длина одной из сторон и величины прилежащих к ней углов.

Методы: вычисление высоты, проведенной из большего угла, вычисление медианы, проведенной к большей стороне, вычисление длин всех сторон, вычисление площади.

Операторы: перегрузка операции сравнения на равенство $==$.

Текст программы

Файл main.cpp:

```
#include <iostream>
#include "class.h"

using namespace std;

int main()
{
    int menu, variant;
    bool flag;
    Triangle first, second(5, 90, 45), third(second);
    string yes;
    do
    {
        system("pause");
        system("cls");
        cout << "1. Processing of a triangle created by the constructor without
parameters" << endl;
        cout << "2. Processing of a triangle created by the constructor with
parameters" << endl;
        cout << "3. Processing of a triangle created by the constructor with
copy" << endl;
        cout << "4. Compare triangle" << endl;
        cout << "5. Exit" << endl;
        cin >> menu;
        switch (menu)
        {
            case 1:
            {
                system("cls");
                do
                {
                    system("cls");
                    cout << first << endl;
                    flag = first.CorrectTriangle(); // проверяем треугольник на
корректность
                    if (flag == false) // если треугольник не
корректный
                    {
                        cout << "Enter correct value of corners or side" << endl;
                        cout << "1. Enter new values of triangle" << endl
                        << "2. Input and set value of lenght side" << endl
                        << "3. Input and set value of first corner" << endl
                        << "4. Input and set value of second corner" << endl
                        << "For return to menu press 9" << endl;
                        cin >> variant;
                        switch (variant)
                        {
                            case 1: // заполнение новыми значениями
                            {
                                system("cls");
                                cin >> first;
                                cout << "For return to the menu, press e or E" <<
endl;
                                do
                                {
                                    cin >> yes;
                                } while (yes != "e" && yes != "E");

                                break;
                            }
                        }
                    }
                } while (!flag);
            }
        }
    } while (menu != 5);
}
```

```

    }
    case 2:
    {
        system("cls");
        cout << "Changing the value of the length of the side
of the triangle" << endl;
        first.SetLenght(first.InputNewLenght());
        cout << "For return to the menu, press e or E" <<
endl;

        do
        {
            cin >> yes;
        } while (yes != "e" && yes != "E");

        break;
    }
    case 3:
    {
        system("cls");
        cout << "Changing the value of the first corner of the
triangle" << endl;
        first.SetFirstCorner(first.InputNewCorner());
        cout << "For return to the menu, press e or E" <<
endl;

        do
        {
            cin >> yes;
        } while (yes != "e" && yes != "E");

        break;
    }
    case 4:
    {
        system("cls");
        cout << "Changing the value of the second corner of
the triangle" << endl;
        first.SetSecondCorner(first.InputNewCorner());
        cout << "For return to the menu, press e or E" <<
endl;

        do
        {
            cin >> yes;
        } while (yes != "e" && yes != "E");

        break;
    }
}
else // если треугольник существует
{
    cout << "1. Enter new values" << endl;
    cout << "2. Input and set value of lenght side" << endl;
    cout << "3. Input and set value of first corner" << endl;
    cout << "4. Input and set value of second corner" << endl;
    cout << "5. Calculate the lengths of the sides" << endl;
    cout << "6. Calculate the area of tringle" << endl;
    cout << "7. Calculate the height of tringle " << endl;
    cout << "8. Calculate the median drawn to the larger side"
<< endl;

    cout << "9. For return to menu press 9" << endl;
    cin >> variant;
    switch (variant)
    {

```

```

case 1: // ВВОД НОВЫХ ЗНАЧЕНИЙ
{
    system("cls");
    cin >> first;
    cout << "For return to the menu, press e or E" <<
endl;

    do
    {
        cin >> yes;
    } while (yes != "e" && yes != "E");

    break;
}
case 2:
{
    system("cls");
    cout << "Changing the value of the length of the side
of the triangle" << endl;

    first.SetLenght(first.InputNewLenght());
    cout << "For return to the menu, press e or E" <<
endl;

    do
    {
        cin >> yes;
    } while (yes != "e" && yes != "E");

    break;
}
case 3:
{
    system("cls");
    cout << "Changing the value of the first corner of the
triangle" << endl;

    first.SetFirstCorner(first.InputNewCorner());
    cout << "For return to the menu, press e or E" <<
endl;

    do
    {
        cin >> yes;
    } while (yes != "e" && yes != "E");

    break;
}
case 4:
{
    system("cls");
    cout << "Changing the value of the second corner of
the triangle" << endl;

    first.SetSecondCorner(first.InputNewCorner());
    cout << "For return to the menu, press e or E" <<
endl;

    do
    {
        cin >> yes;
    } while (yes != "e" && yes != "E");

    break;
}
case 5: // ВЫВОД СТОРОН НА ЭКРАН
{
    system("cls");
    cout << "Lenghts of sides" << endl;
    cout << "Lenght of side a: " << endl

```

```

        << first.GetLenghtSide() << endl;
    cout << "Lenght of side b: " << endl
        << first.SearchSideB() << endl;
    cout << "Lenght of side c: " << endl
        << first.SearchSideC() << endl;
        cout << "For return to the menu, press e or E" <<
endl;

    do
    {
        cin >> yes;
    } while (yes != "e" && yes != "E");

    break;
}
case 6: // ВЫВОД ПЛОЩАДИ
{
    system("cls");
    cout << "Area of tringle" << endl;
    cout << "Area this tringle: " << endl
        << first.SearchArea() << endl;
        cout << "For return to the menu, press e or E" <<
endl;

    do
    {
        cin >> yes;
    } while (yes != "e" && yes != "E");

    break;
}
case 7: // ВЫВОД ВЫСОТЫ ПРОВЕДЕННОЙ ИЗ БОЛЬШЕГО УГЛА
{
    system("cls");
    cout << "Height of tringle drawn from lager corner: "
<< endl
        << first.SearchHeight() << endl;
        cout << "For return to the menu, press e or E" <<
endl;

    do
    {
        cin >> yes;
    } while (yes != "e" && yes != "E");

    break;
}
case 8: // ВЫВОД МЕДИАНЫ ПРОВЕДЕННОЙ К БОЛЬШЕЙ СТОРОНЕ
{
    system("cls");
    cout << "Mediana of tringle drawn to lager side: " <<
first.SearchMedian() << endl;
        cout << "For return to the menu, press e or E" <<
endl;

    do
    {
        cin >> yes;
    } while (yes != "e" && yes != "E");

    break;
}
}
} while (variant != 9);
break;
}

```

```

case 2: // работа с треугольником созданным с параметрами
{
    system("cls");
    do
    {
        system("cls");
        cout << second << endl;
        flag = second.CorrectTriangle();
        if (flag == false) // если не существует, просим ввести новые
значения
        {
            cout << "Enter correct value of corners or side" << endl;
            cout << "1. Enter new values of triangle" << endl
                << "2. Input and set value of lenght side" << endl
                << "3. Input and set value of first corner" << endl
                << "4. Input and set value of second corner" << endl
                << "For return to menu press 9" << endl;
            cin >> variant;
            switch (variant)
            {
                case 1: // заполнение новыми значениями
                {
                    system("cls");
                    cin >> second;
                    cout << "For return to the menu, press e or E" <<
endl;

                    do
                    {
                        cin >> yes;
                    } while (yes != "e" && yes != "E");

                    break;
                }
                case 2:
                {
                    system("cls");
                    cout << "Changing the value of the length of the side
of the triangle" << endl;

                    second.SetLenght(second.InputNewLenght());
                    cout << "For return to the menu, press e or E" <<
endl;

                    do
                    {
                        cin >> yes;
                    } while (yes != "e" && yes != "E");

                    break;
                }
                case 3:
                {
                    system("cls");
                    cout << "Changing the value of the first corner of the
triangle" << endl;

                    second.SetFirstCorner(second.InputNewCorner());
                    cout << "For return to the menu, press e or E" <<
endl;

                    do
                    {
                        cin >> yes;
                    } while (yes != "e" && yes != "E");

                    break;
                }
            }
        }
    }
}

```

```

case 4:
{
    system("cls");
    cout << "Changing the value of the second corner of
the triangle" << endl;
    second.SetSecondCorner(second.InputNewCorner());
    cout << "For return to the menu, press e or E" <<
endl;

    do
    {
        cin >> yes;
    } while (yes != "e" && yes != "E");

    break;
}
}
else // если треугольник существует
{
    cout << "1. Enter new values" << endl;
    cout << "2. Input and set value of lenght side" << endl;
    cout << "3. Input and set value of first corner" << endl;
    cout << "4. Input and set value of second corner" << endl;
    cout << "5. Calculate the lengths of the sides" << endl;
    cout << "6. Calculate the area of tringle" << endl;
    cout << "7. Calculate the height of tringle " << endl;
    cout << "8. Calculate the median drawn to the larger side"
<< endl;

    cout << "9. For return to menu press 9" << endl;
    cin >> variant;
    switch (variant)
    {
    case 1: // новые значения
    {
        system("cls");
        cin >> second;
        cout << "For return to the menu, press e or E" <<
endl;

        do
        {
            cin >> yes;
        } while (yes != "e" && yes != "E");

        break;
    }
    case 2:
    {
        system("cls");
        cout << "Changing the value of the length of the side
of the triangle" << endl;
        second.SetLenght(second.InputNewLenght());
        cout << "For return to the menu, press e or E" <<
endl;

        do
        {
            cin >> yes;
        } while (yes != "e" && yes != "E");

        break;
    }
    case 3:
    {
        system("cls");

```



```

triangle" << endl;

endl;

cout << "Changing the value of the first corner of the
second.SetFirstCorner(second.InputNewCorner());
cout << "For return to the menu, press e or E" <<

do
{
    cin >> yes;
} while (yes != "e" && yes != "E");

break;
}
case 4:
{
    system("cls");
    cout << "Changing the value of the second corner of
the triangle" << endl;

second.SetSecondCorner(second.InputNewCorner());
cout << "For return to the menu, press e or E" <<

do
{
    cin >> yes;
} while (yes != "e" && yes != "E");

break;
}
case 5: // ВЫВОД СТОРОН
{
    system("cls");
    cout << "Lenghts of sides" << endl;
    cout << "Lenght of side a: " << endl
        << second.GetLenghtSide() << endl;
    cout << "Lenght of side b: " << endl
        << second.SearchSideB() << endl;
    cout << "Lenght of side c: " << endl
        << second.SearchSideC() << endl;
    cout << "For return to the menu, press e or E" <<

endl;

do
{
    cin >> yes;
} while (yes != "e" && yes != "E");

break;
}
case 6: // ВЫВОД ПЛОЩАДИ
{
    system("cls");
    cout << "Area of tringle" << endl;
    cout << "Area this tringle: " << endl
        << second.SearchArea() << endl;
    cout << "For return to the menu, press e or E" <<

endl;

do
{
    cin >> yes;
} while (yes != "e" && yes != "E");

break;
}
case 7: // ВЫВОД ВЫСОТЫ
{

```

```

        system("cls");
        cout << "Height of tringle drawn from lager corner: "
<< endl
        << second.SearchHeight() << endl;
        cout << "For return to the menu, press e or E" <<
endl;

        do
        {
            cin >> yes;
        } while (yes != "e" && yes != "E");

        break;
    }
    case 8: // ВЫВОД МЕДИАНЫ
    {
        system("cls");
        cout << "Mediana of tringle drawn to lager side: " <<
endl
        << second.SearchMedian() << endl;
        cout << "For return to the menu, press e or E" <<
endl;

        do
        {
            cin >> yes;
        } while (yes != "e" && yes != "E");

        break;
    }
}
} while (variant != 9);
break;
}
case 3: // обработка треугольника построенного конструктором
копирования
{
    system("cls");
    do
    {
        system("cls");
        cout << third << endl;
        flag = third.CorrectTriangle();
        if (flag == false)
        {
            cout << "Enter correct value of corners or side" << endl;
            cout << "1. Enter new values of triangle" << endl
            << "2. Input and set value of lenght side" << endl
            << "3. Input and set value of first corner" << endl
            << "4. Input and set value of second corner" << endl
            << "For return to menu press 9" << endl;
            cin >> variant;
            switch (variant)
            {
                case 1: // заполнение новыми значениями
                {
                    system("cls");
                    cin >> third;
                    cout << "For return to the menu, press e or E" <<
endl;

                    do
                    {
                        cin >> yes;

```

```

        } while (yes != "e" && yes != "E");

        break;
    }
    case 2:
    {
        system("cls");
        cout << "Changing the value of the length of the side
of the triangle" << endl;

        third.SetLenght(third.InputNewLenght());
        cout << "For return to the menu, press e or E" <<
endl;

        do
        {
            cin >> yes;
        } while (yes != "e" && yes != "E");

        break;
    }
    case 3:
    {
        system("cls");
        cout << "Changing the value of the first corner of the
triangle" << endl;

        third.SetFirstCorner(third.InputNewCorner());
        cout << "For return to the menu, press e or E" <<
endl;

        do
        {
            cin >> yes;
        } while (yes != "e" && yes != "E");

        break;
    }
    case 4:
    {
        system("cls");
        cout << "Changing the value of the second corner of
the triangle" << endl;

        third.SetSecondCorner(third.InputNewCorner());
        cout << "For return to the menu, press e or E" <<
endl;

        do
        {
            cin >> yes;
        } while (yes != "e" && yes != "E");

        break;
    }
    }
    else
    {
        cout << "1. Enter new values" << endl;
        cout << "2. Input and set value of lenght side" << endl;
        cout << "3. Input and set value of first corner" << endl;
        cout << "4. Input and set value of second corner" << endl;
        cout << "5. Calculate the lengths of the sides" << endl;
        cout << "6. Calculate the area of tringle" << endl;
        cout << "7. Calculate the height of tringle " << endl;
        cout << "8. Calculate the median drawn to the larger side"
<< endl;

        cout << "9. For return to menu press 9" << endl;

```

```

        cin >> variant;
        switch (variant)
        {
        case 1:
        {
            system("cls");
            cin >> third;
            cout << "For return to the menu, press e or E" <<
endl;

            do
            {
                cin >> yes;
            } while (yes != "e" && yes != "E");

            break;
        }
        case 2:
        {
            system("cls");
            cout << "Changing the value of the length of the side
of the triangle" << endl;

            third.SetLenght(third.InputNewLenght());
            cout << "For return to the menu, press e or E" <<
endl;

            do
            {
                cin >> yes;
            } while (yes != "e" && yes != "E");

            break;
        }
        case 3:
        {
            system("cls");
            cout << "Changing the value of the first corner of the
triangle" << endl;

            third.SetFirstCorner(third.InputNewCorner());
            cout << "For return to the menu, press e or E" <<
endl;

            do
            {
                cin >> yes;
            } while (yes != "e" && yes != "E");

            break;
        }
        case 4:
        {
            system("cls");
            cout << "Changing the value of the second corner of
the triangle" << endl;

            third.SetSecondCorner(third.InputNewCorner());
            cout << "For return to the menu, press e or E" <<
endl;

            do
            {
                cin >> yes;
            } while (yes != "e" && yes != "E");

            break;
        }
        case 5:
        {

```

```

        system("cls");
        cout << "Lenghts of sides" << endl;
        cout << "Lenght of side a: " << endl
              << third.GetLenghtSide() << endl;
        cout << "Lenght of side b: " << endl
              << third.SearchSideB() << endl;
        cout << "Lenght of side c: " << endl
              << third.SearchSideC() << endl;
        cout << "For return to the menu, press e or E" <<
endl;

        do
        {
            cin >> yes;
        } while (yes != "e" && yes != "E");

        break;
    }
    case 6:
    {
        system("cls");
        cout << "Area of tringle" << endl;
        cout << "Area this tringle: " << endl
              << third.SearchArea() << endl;
        cout << "For return to the menu, press e or E" <<
endl;

        do
        {
            cin >> yes;
        } while (yes != "e" && yes != "E");

        break;
    }
    case 7:
    {
        system("cls");
        cout << "Height of tringle drawn from lager corner: "
endl;

        << third.SearchHeight() << endl;
        cout << "For return to the menu, press e or E" <<
endl;

        do
        {
            cin >> yes;
        } while (yes != "e" && yes != "E");

        break;
    }
    case 8:
    {
        system("cls");
        cout << "Mediana of tringle drawn to lager side: " <<
endl;

        << third.SearchMedian() << endl;
        cout << "For return to the menu, press e or E" <<
endl;

        do
        {
            cin >> yes;
        } while (yes != "e" && yes != "E");

        break;
    }
}
}

```

```

    }
    } while (variant != 9);
    break;
}
case 4: // сравнение треугольников
{
    system("cls");
    do
    {
        system("cls");
        cout << "1. Compare first and second triangles" << endl;
        cout << "2. Compare first and third triangles" << endl;
        cout << "3. Compare third and second triangles" << endl;
        cout << "4. For return to the menu, press 4" << endl;
        cin >> variant;
        switch (variant) // выбор какие треугольники сравнивать
        {
            case 1: // сравнение первого и второго
            {
                system("cls");
                if ((first == second) == true)
                    cout << "Triangles first and second are equal" <<
endl;

                else
                    cout << "Triangles first and second are not equal" <<
endl;

                cout << "For return to the menu, press e or E" << endl;
                do
                {
                    cin >> yes;
                } while (yes != "e" && yes != "E");

                break;
            }
            case 2: // сравнение первого и третьего
            {
                system("cls");
                if ((first == third) == true)
                    cout << "Triangles first and third are equal" << endl;
                else
                    cout << "Triangles first and third are not equal" <<
endl;

                cout << "For return to the menu, press e or E" << endl;
                do
                {
                    cin >> yes;
                } while (yes != "e" && yes != "E");

                break;
            }
            case 3: // сравнение второго и третьего
            {
                system("cls");
                if ((third == second) == true)
                    cout << "Triangles second and third are equal" <<
endl;

                else
                    cout << "Triangles second and third are not equal" <<
endl;

                cout << "For return to the menu, press e or E" << endl;
                do
                {
                    cin >> yes;

```

```

        } while (yes != "e" && yes != "E");

        break;
    }
}
} while (variant != 4);
break;
}
}
} while (menu != 5);
return 0;
}

```

Файл class.h:

```

#ifndef CLASS_H
#define CLASS_H
#include "iostream"

using namespace std;

class Triangle
{
private:
    double LenghtSide;    // длины стороны
    double FirstCorner;    // первый прилежащий угол
    double SecondCorner;  // второй прилежащий угол

public:
    Triangle();            // конструктор без параметров
    Triangle(double, double, double); // конструктор с параметрами
    Triangle(const Triangle &); // конструктор копирования
    ~Triangle();           // деструктор

    double GetLenghtSide(); // получение значения поля стороны
    double GetFirstCorner(); // получение значения первого угла
    double GetSecondCorner(); // получение значения второго угла

    void SetLenght(double); // установить значение поля сторона
    void SetFirstCorner(double); // установить значение поля первого угла
    void SetSecondCorner(double); // установить значение поля второго угла

    double InputNewLenght(); // ввод нового значения для стороны
    double InputNewCorner(); // ввод нового значения для угла

    bool CorrectTriangle(); // проверка треугольника на
    корректность
    double GetRadian(double); // перевод угла в радианы
    double SearchMax(double, double, double); // поиск максимума из трёх
    элементов

    double SearchThirdCorner(); // поиск третьего угла (угла напротив известной
    стороны)
    double SearchSideB(); // поиск стороны напротив первого угла
    double SearchSideC(); // поиск стороны напротив второго угла

    double SearchArea(); // поиск площади треугольника
    double SearchHeight(); // поиск высоты проведенной из большего угла
    double SearchMedian(); // поиск медианы проведенной к большей стороне

    friend ostream &operator<<(ostream &, const Triangle &); // перегрузка
    оператора вывода
    friend istream &operator>>(istream &, Triangle &); // перегрузка
    оператора ввода

```

```

        bool operator==(const Triangle &);
оператора равно
    };

#endif

    Файл methods.cpp:

#include "class.h"
#include "cmath"
#include "limits"

// конструктор без параметров
Triangle::Triangle()
{
    cout << "---Constructor without parametr---" << endl;
    cout << "-----Create Triangle-----" << endl;
    cout << endl;
    LenghtSide = 0;
    FirstCorner = 0;
    SecondCorner = 0;
}

// конструктор с параметрами
Triangle::Triangle(double a, double b, double c) : LenghtSide(a),
FirstCorner(b), SecondCorner(c)
{
    cout << "---Constructor with parametr---" << endl;
    cout << "-----Create Tringle-----" << endl;
    cout << endl;
}

// конструктор копирования
Triangle::Triangle(const Triangle &triangle)
{
    cout << "-----Copy constructor-----" << endl;
    cout << "-----Create copy-----" << endl;
    cout << endl;
    LenghtSide = triangle.LenghtSide;
    FirstCorner = triangle.FirstCorner;
    SecondCorner = triangle.SecondCorner;
}

// деструктор
Triangle::~Triangle()
{
    cout << "-----Destructor-----" << endl;
}

// перегрузка оператора вывода
ostream &operator<<(ostream &out, const Triangle &output)
{
    cout << "Outputing triangle values" << endl;
    out << "Lenght of side: " << output.LenghtSide << "\n"
        << "First corner: " << output.FirstCorner << "\n"
        << "Second corner: " << output.SecondCorner << endl;

    return out;
}

```



```

// перегрузка оператора ввода
istream &operator>>(istream &stream, Triangle &triangle)
{
    cout << "Entering triangle values" << endl;
    cout << "Input lenght of side: "; // ввод стороны
    while (1)
    {
        stream >> triangle.LenghtSide;
        if (triangle.LenghtSide > 0)
            break;
        else // если введено отрицательное число или непонятный набор
СИМВОЛОВ
        {
            cout << "Input correct lenght of side: ";
            cin.clear(); // очищаем поле
            cin.ignore(numeric_limits<streamsize>::max(), '\n');
        }
    }
    cout << "Input value of first corner: "; // ввод значение первого
прилежащего угла
    while (1)
    {
        stream >> triangle.FirstCorner;
        if (triangle.FirstCorner > 0 && triangle.FirstCorner < 180)
            break;
        else
        {
            cout << "Input correct value of first corner: "; // если
введено некоректное значение числа
            cin.clear(); //
очищаем поле
            cin.ignore(numeric_limits<streamsize>::max(), '\n');
        }
    }
    cout << "Input value of second corner: "; // ввод второго
прилежащего угла
    while (1)
    {
        stream >> triangle.SecondCorner;
        if (triangle.SecondCorner > 0 && triangle.SecondCorner < 180)
            break;
        else // если введено некоректное значение
        {
            cout << "Input correct value of second corner: ";
            cin.clear();
            cin.ignore(numeric_limits<streamsize>::max(), '\n');
        }
    }
    return stream;
}

// получить значения из поля длина стороны
double Triangle::GetLenghtSide()
{
    cout << "Return of the side length value" << endl;
    return LenghtSide;
}

```

```

}

// получить значение из поля первого угла
double Triangle::GetFirstCorner()
{
    cout << "Return the value of the first angle" << endl;
    return FirstCorner;
}

// получить значение из поля второго угла
double Triangle::GetSecondCorner()
{
    cout << "Return the value of the second angle" << endl;
    return SecondCorner;
}

// установка нового значения для стороны
void Triangle::SetLenght(double side)
{
    cout << "Setting the side length value" << endl;
    LenghtSide = side;
}

// установка нового значения для первого угла
void Triangle::SetFirstCorner(double first)
{
    cout << "Setting the value of the first angle" << endl;
    FirstCorner = first;
}

// установка нового значения для второго угла
void Triangle::SetSecondCorner(double second)
{
    cout << "Setting the value of the second angle" << endl;
    SecondCorner = second;
}

// проверка треугольника на корректность
bool Triangle::CorrectTriangle()
{
    cout << "Checking the triangle for existence" << endl;
    if (FirstCorner + SecondCorner >= 180 || FirstCorner <= 0 ||
        SecondCorner <= 0 || LenghtSide <= 0)
        return false;
    else
        return true;
}

// перевод угла в радианы
double Triangle::GetRadian(double a)
{
    cout << "Converting an angle to radians" << endl;
    double m = 3.14 / 180;
    return m * a;
}

// поиск третьего угла (напротив известного угла)

```

```

double Triangle::SearchThirdCorner()
{
    cout << "Search of third corner" << endl;
    return 180 - FirstCorner - SecondCorner;
}

// поиск стороны по теореме синусов стороны напротив первого угла
double Triangle::SearchSideB()
{
    cout << "Finding the side opposite the first corner" << endl;
    return LenghtSide * sin(GetRadian(FirstCorner)) /
sin(GetRadian(SearchThirdCorner()));
}

// поиск стороны напротив второго угла
double Triangle::SearchSideC()
{
    cout << "Finding the side opposite the second corner" << endl;
    return LenghtSide * sin(GetRadian(SecondCorner)) /
sin(GetRadian(SearchThirdCorner()));
}

// поиск площади треугольника
double Triangle::SearchArea()
{
    cout << "Finding the area of a triangle" << endl;
    return (SearchSideB() * SearchSideC() *
sin(GetRadian(SearchThirdCorner())) / 2;
}

// поиск высоты проведенной из наибольшего угла
double Triangle::SearchHeight()
{
    cout << "Finding the height drawn from a larger angle" << endl;
    double alfa = SearchThirdCorner(), area = SearchArea(), b =
SearchSideB(), c = SearchSideC();
    double max = SearchMax(FirstCorner, SecondCorner, alfa); // поиск
максимального угла
    if (max == FirstCorner) // если
максимальный угол первый ищем высоту через формулу площади
        return 2 * area / b;
    if (max == SecondCorner) // если второй угол максимальный
        return 2 * area / c;
    if (max == alfa)
        return 2 * area / LenghtSide;
}

// поиск максимального значения из трех
double Triangle::SearchMax(double a, double b, double c)
{
    cout << "Search for the maximum" << endl;
    double max = a >= b ? (a >= c ? a : c) : (b >= c ? b : c);
    return max;
}

// поиск медианы
double Triangle::SearchMedian()

```

```

{
    cout << "Median Search" << endl;
    double a = LenghtSide, b = SearchSideB(), c = SearchSideC();
    double max = SearchMax(a, b, c);
    if (max == a)
        return sqrt(2 * b * b + 2 * c * c - a * a) / 2;
    if (max == b)
        return sqrt(2 * a * a + 2 * c * c - b * b) / 2;
    if (max == c)
        return sqrt(2 * b * b + 2 * a * a - c * c) / 2;
}

// пергрузка операторов равно
bool Triangle::operator==(const Triangle &triangle)
{
    cout << "Comparison of triangles" << endl;
    if (this->LenghtSide == triangle.LenghtSide && this->FirstCorner
== triangle.FirstCorner && this->SecondCorner ==
triangle.SecondCorner)
        return true;
    else
        return false;
}

double Triangle::InputNewLenght()
{
    double a;
    cout << "Entering a new side value";
    cout << "Input value of side: ";
    while (1)
    {
        cin >> a;
        if (a > 0)
        {
            return a;
            break;
        }
        else // если введенно отрицательное число или непонятный набор
СИМВОЛОВ
        {
            cout << "Input correct lenght of side: ";
            cin.clear(); // очищаем поле
            cin.ignore(numeric_limits<streamsize>::max(), '\n');
        }
    }
}

double Triangle::InputNewCorner()
{
    double a;
    cout << "Entering a new angle value" << endl;
    cout << "Input value of corner: ";
    while (1)
    {
        cin >> a;
        if (a > 0 && a < 180)
        {

```

```

        return a;
        break;
    }
    else // если введено отрицательное число или непонятный набор
СИМВОЛОВ
    {
        cout << "Input correct value of corner: ";
        cin.clear(); // очищаем поле
        cin.ignore(numeric_limits<streamsize>::max(), '\n');
    }
}
}

```

Скриншоты

```

ПРОБЛЕМЫ    ВЫХОДНЫЕ ДАННЫЕ    КОНСОЛЬ ОТЛАДКИ    ТЕРМИНАЛ

PS C:\Informatics 2 sem\number 1.1> ./prog
---Constructor without parametrs---
-----Create Tringle-----

---Constructor with parametrs---
-----Create Tringle-----

-----Copy constructor-----
-----Create copy-----

Для продолжения нажмите любую клавишу . . .

```

Рисунок 1 – Работа конструкторов

```

ПРОБЛЕМЫ    ВЫХОДНЫЕ ДАННЫЕ    КОНСОЛЬ ОТЛАДКИ    ТЕРМИНАЛ

1. Processing of a triangle created by the constructor without parmeters
2. Processing of a triangle created by the constructor with parmeters
3. Processing of a triangle created by the constructor with copy
4. Compare triangle
5. Exit

```

Рисунок 2 – Основное меню

Меню обработки треугольников, если треугольник не существует (конструктор без параметров устанавливает значение нулей, или в конструктор с параметрами или копирования переданы значения при которых треугольник не существует). Это меню представлено на рисунке 3.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

Outputing triangle values
Lenght of side: 0
First corner: 0
Second corner: 0

Checking the triangle for existence
Enter correct value of corners or side
1. Enter new values of triangle
2. Input and set value of lenght side
3. Input and set value of first corner
4. Input and set value of second corner
For return to menu press 9
█
```

Рисунок 2 – Меню обработки треугольников, если треугольник не существует

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

Outputing triangle values
Lenght of side: 12
First corner: 20
Second corner: 60

Checking the triangle for existence
1. Enter new values
2. Input and set value of lenght side
3. Input and set value of first corner
4. Input and set value of second corner
5. Calculate the lengths of the sides
6. Calculate the area of tringle
7. Calculate the height of tringle
8. Calculate the median drawn to the larger side
9. For return to menu press 9
█
```

Рисунок 3 – Меню обработки если треугольник существует

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

Entering triangle values
Input lenght of side: 20
Input value of first corner: 20
Input value of second corner: 60
For return to the menu, press e or E
█
```

Рисунок 4 – Ввод новых значений треугольников

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

Changing the value of the length of the side of the triangle
Entering a new side valueInput value of side: 12
Setting the side length value
For return to the menu, press e or E
█
```

Рисунок 5 – Ввод и установка нового значения стороны

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  prog

Lenghts of sides
Lenght of side a:
Return of the side length value
5
Lenght of side b:
Finding the side opposite the first corner
Converting an angle to radians
Search of third corner
Converting an angle to radians
7.07388
Lenght of side c:
Finding the side opposite the second corner
Converting an angle to radians
Search of third corner
Converting an angle to radians
5
For return to the menu, press e or E
█
```

Рисунок 7 – Вывод длин всех сторон

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  prog

Area of tringle
Area this tringle:
Finding the area of a triangle
Finding the side opposite the first corner
Converting an angle to radians
Search of third corner
Converting an angle to radians
Finding the side opposite the second corner
Converting an angle to radians
Search of third corner
Converting an angle to radians
Search of third corner
Converting an angle to radians
12.5
For return to the menu, press e or E
█
```

Рисунок 6 – Вывод площади треугольника

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  prog

Height of tringle drawn from lager corner:
Finding the height drawn from a larger angle
Search of third corner
Finding the area of a triangle
Finding the side opposite the first corner
Converting an angle to radians
Search of third corner
Converting an angle to radians
Finding the side opposite the second corner
Converting an angle to radians
Search of third corner
Converting an angle to radians
Finding the side opposite the first corner
Converting an angle to radians
Search of third corner
Converting an angle to radians
Finding the side opposite the second corner
Converting an angle to radians
Search of third corner
Converting an angle to radians
Search for the maximum
3.53413
For return to the menu, press e or E
```

Рисунок 7 – Вывод высоты, проведенной из наибольшего угла

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

Mediana of tringle drawn to lager side:
Median Search
Finding the side opposite the first corner
Converting an angle to radians
Search of third corner
Converting an angle to radians
Finding the side opposite the second corner
Converting an angle to radians
Search of third corner
Converting an angle to radians
Search for the maximum
3.53413
For return to the menu, press e or E
```

Рисунок 8 – Вывод медианы, проведенной к большей стороне

```
ПРОБЛЕМЫ  ВЫХОДНЫЕ ДАННЫЕ  КОНСОЛЬ ОТЛАДКИ  ТЕРМИНАЛ

1. Compare first and second triangles
2. Compare first and third triangles
3. Compare third and second triangles
4. For return to the menu, press 4
```

Рисунок 9 – Меню сравнения треугольников


```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

Comparison of triangles
Triangles second and third are equal
For return to the menu, press e or E
█
```

Рисунок 10 – Если треугольники равны

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

Comparison of triangles
Triangles first and second are not equal
For return to the menu, press e or E
█
```

Рисунок 11 – Если треугольники не равны

```
ПРОБЛЕМЫ  ВЫХОДНЫЕ ДАННЫЕ  КОНСОЛЬ ОТЛАДКИ  ТЕРМИНАЛ

1. Processing of a triangle created by the constructor without parmeters
2. Processing of a triangle created by the constructor with parmeters
3. Processing of a triangle created by the constructor with copy
4. Compare triangle
5. Exit
5
-----Destructor-----
-----Destructor-----
-----Destructor-----
PS C:\Informatics 2 sem\number 1.1> █
```

Рисунок 12 – Деструктор