Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное  
учреждение высшего образования

Новосибирский государственный технический университет

Кафедра Автоматизированных систем управления

**ОТЧЁТ**

**по лабораторной работе № 6**

**Тестирование по методу «Белого ящика»**

**по дисциплине: «Технология программирования»**

Выполнили: Проверил:

Студенты гр. «АВТ-813», «АВТФ» Лукоянычева Ольга Викторовна

Букова Анна

Ланг Татьяна

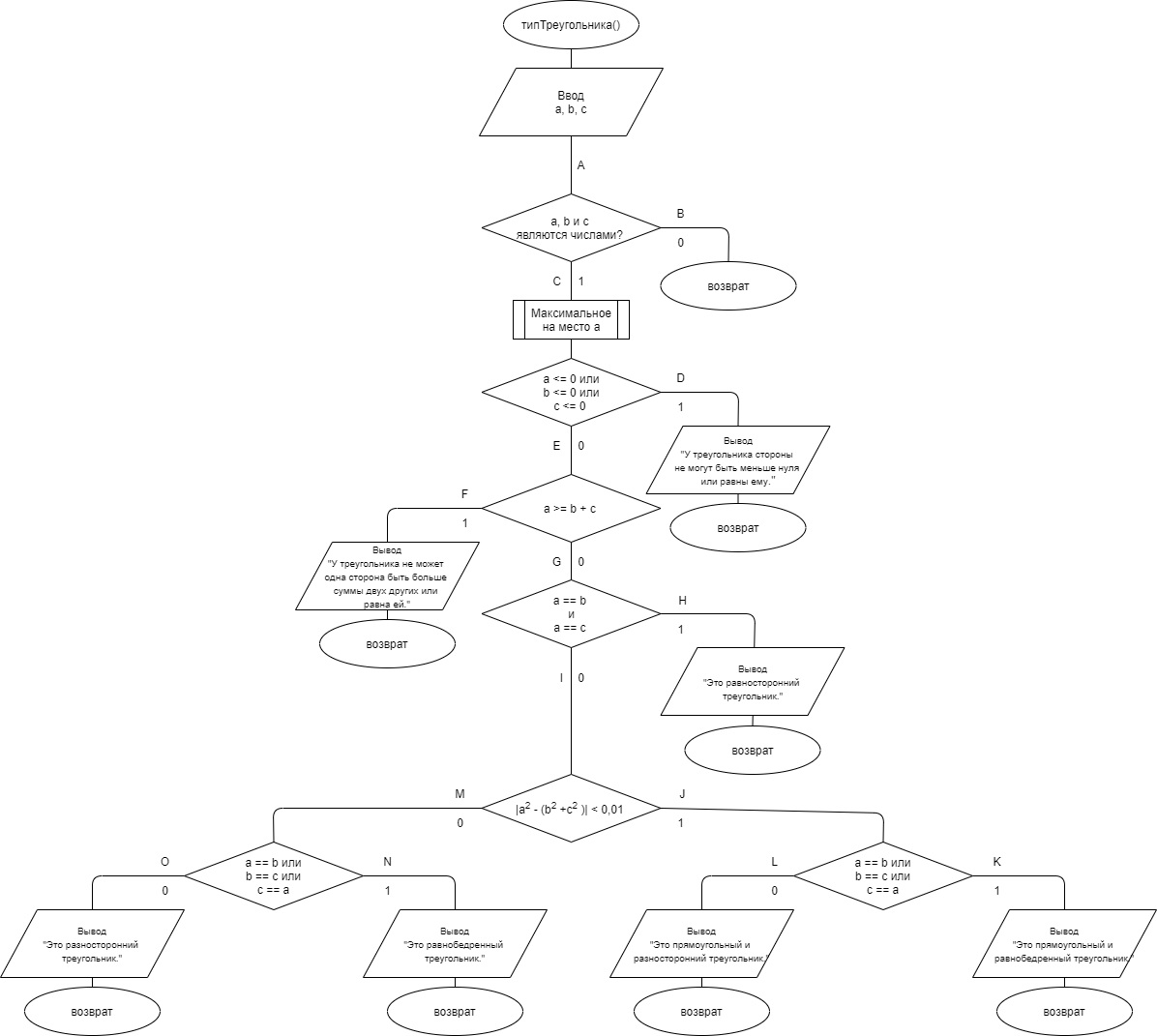
Пустовских Дмитрий

Новосибирск 2020

# Цель работы: Получить навыки тестирования программных продуктов при помощи метода “белого ящика”.

# Задание:

Разработать программу определения вида треугольника, заданного длинами его сторон: равносторонний, равнобедренный, прямоугольный, разносторонний.

**Разработка тестов по методике “белого ящика”  
Алгоритм программы:**

# Тесты:

**Метод покрытия операторов**

Целью этого метода тестирования является выполнение каждого оператора программы хотя бы один раз.

*Исходные данные*

|  |  |  |
| --- | --- | --- |
| Входное условие | Допустимые значения | Недопустимые значения |
| Тип вводимых данных | 3 вещественных числа в диапазоне  0 < X < 2147483648 | Нецифровой символ,  X<=0,  X>=2147483648, Пустой ввод,  Некорректный ввод вещественного,  Одна сторона больше суммы двух других |
| 3 целых числа в диапазоне  0 < X < 2147483648 |
| Комбинация вещественных и целых чисел в диапазоне  0 < X < 2147483648 |

*Тесты*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Тест | Ожидаемый результат | Фактический результат | Результат тестирования |
| 1 | А = 0,002  В =  С = 1,82 | Пустота не лучшее решение. | Пустота не лучшее решение. | Тест пройден успешно |
| 2 | A = 0 B = 0 C = -1 | У треугольника стороны не могут быть меньше нуля или равны ему. | У треугольника стороны не могут быть меньше нуля или равны ему. | Тест пройден успешно |
| 3 | A = 22 B = 1 C = 2 | У треугольника не может одна сторона быть больше суммы двух других или равна ей. | У треугольника не может одна сторона быть больше суммы двух других или равна ей. | Тест пройден успешно |
| 4 | A = 100 B = 100 C = 100 | Это равносторонний треугольник. | Это равносторонний треугольник. | Тест пройден успешно |
| 5 | A = 5 B = 7,071 C = 5 | Это равнобедренный и прямоугольный треугольник. | Это равнобедренный и прямоугольный треугольник. | Тест пройден успешно |
| 6 | A = 3 B = 4 C = 5 | Это разносторонний и прямоугольный треугольник. | Это разносторонний и прямоугольный треугольник. | Тест пройден успешно | |
| 7 | A = 4 B = 4 C = 6 | Это равнобедренный треугольник. | Это равнобедренный треугольник. | Тест пройден успешно |
| 8 | A = 4 B = 5 C = 6 | Это разносторонний треугольник. | Это разносторонний треугольник. | Тест пройден успешно |

**Метод покрытия решений (покрытия переходов)**

Для программы покрытие решений может быть выполнено восемью тестами, свой для каждого пути {*AB, ACD, ACEF, ACEGH, ACEGIJK, ACEGIJL, ACEGIMN, ACEGIMO*}. Пути {*AB, ACD, ACEF, ACEGH, ACEGIJK, ACEGIJL, ACEGIMN, ACEGIMO*} покроим, выбрав следующие исходные данные: {A=1, B=2, С= }, {A=5, B=3, С=0}, {A=4, B=40, С=400}, {A=3, B=3, С=3}, {A=8,485, B=6, С=6}, {A=8, B=6, С=10}, {A=80, B=80, С=100} и {A=15, B=30, С=20}.

*Тесты*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Тест | Ожидаемый результат | Фактический результат | Результат тестирования |
| 1 | А = 1  В = 2  С = | Пустота не лучшее решение. | Пустота не лучшее решение. | Тест пройден успешно |
| 2 | A = 5 B = 3 C = 0 | У треугольника стороны не могут быть меньше нуля или равны ему. | У треугольника стороны не могут быть меньше нуля или равны ему. | Тест пройден успешно |
| 3 | A = 4 B = 40 C = 400 | У треугольника не может одна сторона быть больше суммы двух других или равна ей. | У треугольника не может одна сторона быть больше суммы двух других или равна ей. | Тест пройден успешно |
| 4 | A = 3 B = 3 C = 3 | Это равносторонний треугольник. | Это равносторонний треугольник. | Тест пройден успешно |
| 5 | A = 8,485 B = 6 C = 6 | Это равнобедренный и прямоугольный треугольник. | Это равнобедренный и прямоугольный треугольник. | Тест пройден успешно |
| 6 | A = 8 B = 6 C = 10 | Это разносторонний и прямоугольный треугольник. | Это разносторонний и прямоугольный треугольник. | Тест пройден успешно | |
| 7 | A = 80 B = 80 C = 100 | Это равнобедренный треугольник. | Это равнобедренный треугольник. | Тест пройден успешно |
| 8 | A = 15 B = 30 C = 20 | Это разносторонний треугольник. | Это разносторонний треугольник. | Тест пройден успешно |

**Метод покрытия условий**

Лучшие результаты по сравнению с предыдущими методами может дать метод покрытия условий. В этом случае записывается число тестов, достаточное для того, чтобы все возможные результаты каждого условия в решении выполнялись по крайней мере один раз.

В нашей программе мы имеем десять условий: {A=\*число\*, B=\*число\*, C=\*число\*}, {A<=0, B<=0, C<=0}, {A>=(B+C), B, C}, {A=B, A=C, C }, {|A^2-B^2+C^2|<0,01, B, C}, {A=B или B=С или C=A, B, C}. При этом мы уменьшили число возможных условий за счет функции, которая перед заходов в эти условия выполняет перестановку, подходящую под условие A>B и A>C, т.е. А – самая длинная сторона. Для реализации метода требуется достаточное число тестов, такое, чтобы реализовать ситуации, где

A=\*число\*, A!=\*число\*, B=\*число\*, B!=\*число\*, C=\*число\*, C!=\*число\*, в точке A;

A>0, A≤0, B>0, B≤0, C>0, C≤0, в точке C;

A<(B+C), A>=(B+C), в точке E*;*

A=B и A=C, A ≠ B и A ≠ C в точке G;

|A^2 – (B^2+C^2)| < 0.01, |A^2 – (B^2+C^2)| >= 0.01 в точке I;

A≠B, A≠C, B≠C, A=B, A=C, B=C в точке J;

A≠B, A≠C, B≠C, A=B, A=C, B=C в точке M;

Тесты, удовлетворяющие критерию покрытия условий и соответствующие им пути:

1. A=g, B=2, C=3 *AB*
2. A=1, B=-2, C=3 *ACD*
3. A=10, B=2, C=3 *ACEF*
4. A=13, B=13, C=13 *ACEGH*
5. A=5, B=5, C=7,071 *ACEGIJK*
6. A=3, B=4, C=5 *ACEGIJL*
7. A=4, B=4, C=6 *ACEGIMN*
8. A=11, B=20, C=30 *ACEGIMO*

*Тесты*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Тест | Ожидаемый результат | Фактический результат | Результат тестирования |
| 1 | А = g  В = 2  С = 3 | А может лучше ввести цифры? | А может лучше ввести цифры? | Тест пройден успешно |
| 2 | A = 1 B = -2 C = 3 | У треугольника стороны не могут быть меньше нуля или равны ему. | У треугольника стороны не могут быть меньше нуля или равны ему. | Тест пройден успешно |
| 3 | A = 10 B = 2 C = 3 | У треугольника не может одна сторона быть больше суммы двух других или равна ей. | У треугольника не может одна сторона быть больше суммы двух других или равна ей. | Тест пройден успешно |
| 4 | A = 13 B = 13 C = 13 | Это равносторонний треугольник. | Это равносторонний треугольник. | Тест пройден успешно |
| 5 | A = 5 B = 5 C = 7,071 | Это равнобедренный и прямоугольный треугольник. | Это равнобедренный и прямоугольный треугольник. | Тест пройден успешно |
| 6 | A = 3 B = 4 C = 5 | Это разносторонний и прямоугольный треугольник. | Это разносторонний и прямоугольный треугольник. | Тест пройден успешно | |
| 7 | A = 4 B = 4 C = 6 | Это равнобедренный треугольник. | Это равнобедренный треугольник. | Тест пройден успешно |
| 8 | A = 11 B = 20 C = 30 | Это разносторонний треугольник. | Это разносторонний треугольник. | Тест пройден успешно |

# Демонстрация работы программы:

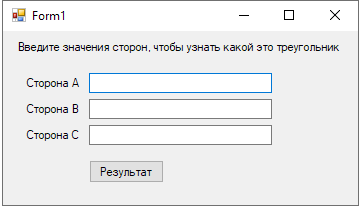


Рисунок 1 – Окно программы.

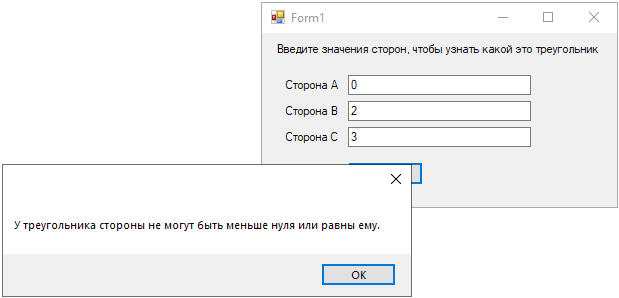


Рисунок 2 – Работа программы с значениями A=0, B=2, C=3.

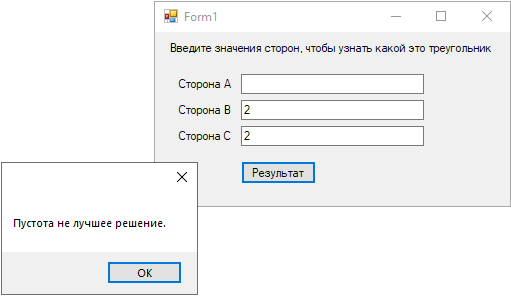


Рисунок 3 – Работа программы с значениями A= , B=2, C=2.

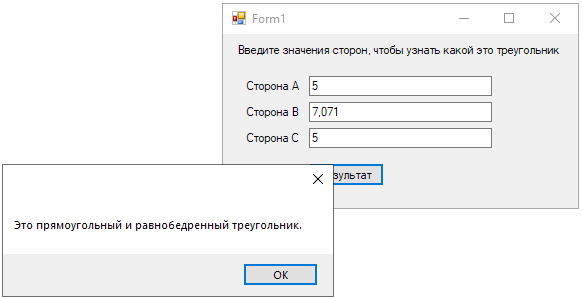


Рисунок 4 – Работа программы с значениями A=5, B=7,071, C=5.

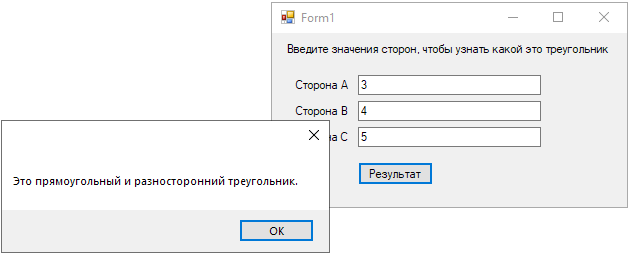


Рисунок 5 – Работа программы с значениями A=3, B=4, C=5.

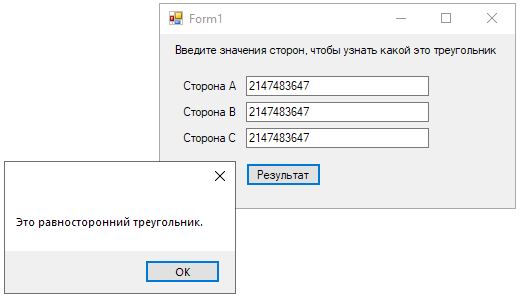


Рисунок 6 – Работа программы с значениями A=4 B=3 C=5.

# Код программы:

1) class Triangle

public float a { get; set; }

public float b { get; set; }

public float c { get; set; }

public void TestTang()

{

MaxA();

if (a <= 0 || b <= 0 || c <= 0)

{

MessageBox.Show("У треугольника стороны не могут быть меньше нуля или равны ему.");

return;

}

if (a >= b + c)

{

MessageBox.Show("У треугольника не может одна сторона быть больше суммы двух других или равна ей.");

return;

}

if (a == b && b == c && c == a)

{

MessageBox.Show("Это равносторонний треугольник.");

}

else

{

if (Math.Abs(a \* a - (b \* b + c \* c)) < 0.01)

{

if (a == b || b == c || c == a)

MessageBox.Show("Это прямоугольный и равнобедренный треугольник.");

else

MessageBox.Show("Это прямоугольный и разносторонний треугольник.");

}

else

{

if (a == b || b == c || c == a)

MessageBox.Show("Это равнобедренный треугольник.");

else

MessageBox.Show("Это разносторонний треугольник.");

}

}

}

private void MaxA()

{

float temp;

if (b >= a)

{

if (b >= c)

{

temp = a;

a = b;

b = temp;

}

else

{

temp = a;

a = c;

c = temp;

}

}

else

if (c >= a)

{

temp = a;

a = c;

c = temp;

}

}

2) class TestNumber

public string num;

public TestNumber(string n)

{

num = n;

}

public bool TestNum()

{

return (CheckChar() && CheckIntOrFloat());

}

//все ли символы числа

private bool CheckChar()

{

if (num.Length == 0)

{

MessageBox.Show("Пустота не лучшее решение.");

return false;

}

//проверяем являются ли все символы числами для положительного

if (num[0] != '-')

{

for (int i = 0; i < num.Length; i++)

if (num[i] < ',' || num[i] == '/' || num[i] == '.' || num[i] == '-' || num[i] > '9')

{

MessageBox.Show("А может лучше ввести цифры?");

return false;

}

}

//проверяем являются ли все символы числами для отрицательного

else

{

for (int i = 1; i < num.Length; i++)

{

if (num[i] < ',' || num[i] == '/' || num[i] == '.' || num[i] == '-' || num[i] > '9')

{

MessageBox.Show("А может лучше ввести цифры?");

return false;

}

}

}

return true;

}

//в зависимости от того, с кем работаем, в тот метод и попадаем

private bool CheckIntOrFloat()

{

int count = 0;

if (num[0] != '-')

{

if (num[0] == ',' || num[num.Length - 1] == ',')

{

MessageBox.Show("У дробного не может быть в начале или в конце запятая. Либо в начале 0, либо пишите целое число.");

return false;

}

}

else

{

if (num[1] == ',' || num[num.Length - 1] == ',')

{

MessageBox.Show("У дробного не может быть в начале или в конце запятая. Либо в начале 0, либо пишите целое число.");

return false;

}

}

for (int i = 1; i < num.Length; i++)

{

if (num[i] == ',')

{

count++;

}

}

switch (count)

{

case 0:

return CheckInt();

case 1:

return CheckFloat();

default:

MessageBox.Show("У дробного не может быть больше одной запятой.");

return false;

}

}

//проверка для int

private bool CheckInt()

{

if (num.Length < 10 || (num.Length < 11 && num[0] == '-')) // если число меньше нужного кол-ва символов, то оно уже точно подхоидмт

return true;

else

{

if (num[0] != '-') //проверяем больше ли максимального, учитывая длину

if (num[0] < '3' && num[1] < '2' && num[2] < '5' && num[3] < '8' && num[4] < '5' && num[5] < '9' && num[6] < '4' && num[7] < '7' && num[8] < '5' && num[9] < '8' && num.Length <= 10)

return true;

else

{

MessageBox.Show("Выход за диапазон типа int (-2 147 483 648 / 2 147 483 647). Слишком большое число.");

return false;

}

else //проверяем меньше ли минимального, учитывая длину

{

if (num[1] < '3' && num[2] < '2' && num[3] < '5' && num[4] < '8' && num[5] < '5' && num[6] < '9' && num[7] < '4' && num[8] < '7' && num[9] < '5' && num[10] < '9' && num.Length <= 11)

return true;

else

{

MessageBox.Show("Выход за диапазон типа int (-2 147 483 648 / 2 147 483 647). Слишком маленькое число.");

return false;

}

}

}

}

//проверка для float

private bool CheckFloat()

{

int count = 0;

for (int i = 1; i < num.Length - 1; i++)

if (num[i] == ',')

{

count = i;

break;

}

if (num[0] != '-') //проверяем больше ли максимального, учитывая длину

if (count > 10)

{

MessageBox.Show("Выход за диапазон типа int (-2 147 483 648.0 / 2 147 483 647.0). Слишком большое число.");

return false;

}

else

{

if (count < 10)

return true;

if (num[0] < '3' && num[1] < '2' && num[2] < '5' && num[3] < '8' && num[4] < '5' && num[5] < '9' && num[6] < '4' && num[7] < '7' && num[8] < '5' && num[9] < '8')

{

if (num[10] == ',')

{

if (num[0] == '2' && num[1] == '1' && num[2] == '4' && num[3] == '7' && num[4] == '4' && num[5] == '8' && num[6] == '3' && num[7] == '6' && num[8] == '4' && num[9] == '7')

{

for (int i = 11; i < num.Length; i++)

if (num[i] != '0')

{

MessageBox.Show("Выход за диапазон типа int (-2 147 483 648.0 / 2 147 483 647.0). Слишком большое число.");

return false;

}

return true;

}

else

return true;

}

else

{

MessageBox.Show("Выход за диапазон типа int (-2 147 483 648.0 / 2 147 483 647.0). Слишком большое число.");

return false;

}

}

else

{

MessageBox.Show("Выход за диапазон типа int (-2 147 483 648.0 / 2 147 483 647.0). Слишком большое число.");

return false;

}

}

else //проверяем меньше ли минимального, учитывая длину

{

if (count > 11)

{

MessageBox.Show("Выход за диапазон типа int (-2 147 483 648.0 / 2 147 483 647.0). Слишком большое число.");

return false;

}

else

{

if (count < 11)

return true;

if (num[1] < '3' && num[2] < '2' && num[3] < '5' && num[4] < '8' && num[5] < '5' && num[6] < '9' && num[7] < '4' && num[8] < '7' && num[9] < '5' && num[10] < '8')

{

if (num[11] == ',')

{

if (num[1] == '2' && num[2] == '1' && num[3] == '4' && num[4] == '7' && num[5] == '4' && num[6] == '8' && num[7] == '3' && num[8] == '6' && num[9] == '4' && num[10] == '7')

{

for (int i = 12; i < num.Length; i++)

if (num[i] != 0)

{

MessageBox.Show("Выход за диапазон типа int (-2 147 483 648.0 / 2 147 483 647.0). Слишком большое число.");

return false;

}

return true;

}

else

return true;

}

else

{

MessageBox.Show("Выход за диапазон типа int (-2 147 483 648.0 / 2 147 483 647.0). Слишком большое число.");

return false;

}

}

else

{

MessageBox.Show("Выход за диапазон типа int (-2 147 483 648 / 2 147 483 647). Слишком маленькое число.");

return false;

}

}

}

}