ЦЕЛЬ РАБОТЫ

Продемонстрировать механизм обработки исключительных ситуаций.

ПРАКТИЧЕСКОЕ ЗАДАНИЕ

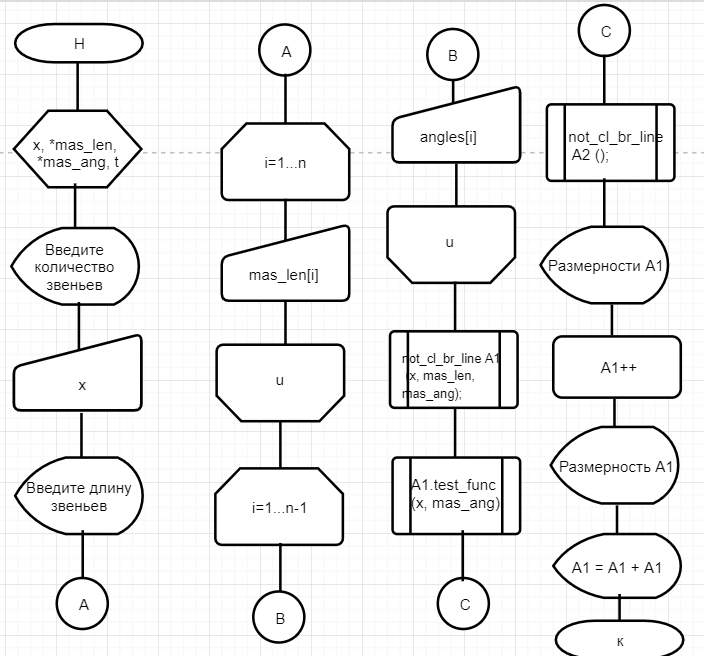
Продемонстрировать обработку исключительных ситуаций в конструкторе класса из 8 лабораторной работы, применив условия на значения параметров.

ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ

Вариант 10. Ломаная линия

БЛОК-СХЕМА АЛГОРИТМА

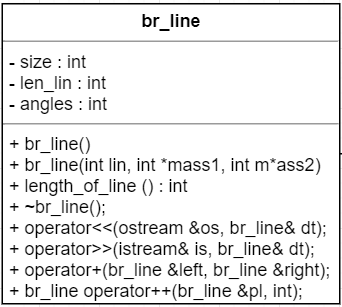
Блок-схема алгоритма представлена на рисунке 1.



*Рисунок 1 Блок-схема программы*

ДИАГРАММА КЛАССОВ

Диаграмма классов представлена на рисунке 2.



*Рисунок 2 Диаграмма классов*

ЛИСТИНГ

#include <iostream>

#include <exception>

using namespace std;

class br\_line {

private:

int size;//количество звеньев

int \*len\_lin;//массив из длин звеньев

int \*angles;//массив из углов между звеньями

public:

br\_line() { //конструктор по умолчанию

size = 3;

len\_lin = new int[size];

len\_lin[0] = 1; len\_lin[1] = 1; len\_lin[2] = 1;

angles = new int[size];

angles[0] = 1; angles[1] = 1;

}

br\_line(int lin, int \*mass1, int \*mass2) { //параметризированный конструктор

size = lin;

len\_lin = new int[size];

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

try {

len\_lin[i] = mass1[i];

if (len\_lin[i] == 0)

{

string exe = "Введите ненулевую длину для данного звена";

throw runtime\_error(exe);

}

}

catch (runtime\_error& r) {

cout << r.what() << endl;

cin >> len\_lin[i];

}

}

angles = new int[size - 1];

for (int i = 0; i < (size - 1); i++) {

try {

angles[i] = mass2[i];

if (angles[i] == 0)

{

string exex = "Введите ненулевую длину для данного звена";

throw runtime\_error(exex);

}

}

catch (runtime\_error& r) {

cout << r.what() << endl;

cin >> len\_lin[i];

}

}

}//lin\_k-количество звеньев

~br\_line() {

delete[] len\_lin;

delete[] angles;

};

int length\_of\_line()

{

int sum = 0;

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

{

sum = sum + len\_lin[n];

}

return sum;

}//функция высчитывает длину ломанной

};

int main(void) {

setlocale(LC\_ALL, "Russian");

int n, \*mass0, \*mass00;

cout<<"Введите количество звеньев"<< endl;

try{

cin >> n;

if (n == 0) {

string ex = "Нулевое количество звеньев, переопределите их количество";

throw runtime\_error(ex);

}

}

catch (runtime\_error& r) {

cout << r.what() << endl;

cin >> n;

}

cout << "Введите длины звеньев" << endl;

mass0 = new int[n];

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

cin >> mass0[i];

}

cout << "Введите размерности углов" << endl;

mass00 = new int[n - 1];

for (int i = 0; i < (n - 1); i++) {

cin >> mass00[i];

}

br\_line (n, mass0, mass00);

delete[] mass0;

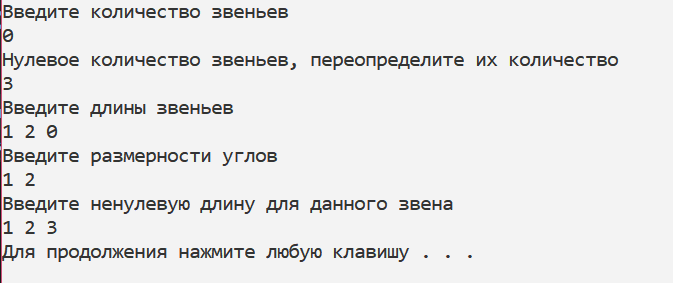
delete[] mass00;

system("pause");

}

ПРОТОКОЛ ВЫПОЛНЕНИЯ ПРОГРАММЫ

Протокол выполнения программы представлен на рис.3.



*Рисунок 3 Протокол программы*