Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное учреждение высшего образования  
«Пермский национальный исследовательский политехнический университет»

Электротехнический факультет  
Кафедра «Информационные технологии и автоматизированные системы»

**О Т Ч Ё Т  
по лабораторной работе**

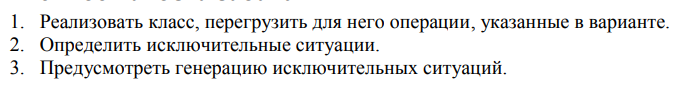
Дисциплина: «Основы алгоритмизации и программирования»

Выполнил  
студент группы ИВТ-21-1б  
Мифтахов И.Г

Проверил  
ст. п. кафедры ИТАС  
Яруллин Д.В.

Пермь 2022

Постановка задачи:



Анализ задачи:

1. int max\_size – максимальный размер множества, int size – текущий размер множества, int \*array указатель для дальнейшего создания динамического массива. (для второго случая доп.переменная в классе error – string msg, отвечающий за сообщение об ошибке).

2. В классе host в обоих случаях реализованы одинаковые конструкторы, деструкторы, методы вывода, перегруженные операторы. Также класс error отвечает за вывод конкретного сообщения при исключении.

3. В первом случае «ловим» исключения в ф-ии main() через параметр int кода ошибки, во втором создаем объект класса error, где и будем выводить сообщение об ошибке.

Ответы на вопросы:

* Исключение – аварийное или непредвиденное событие.
* Обнаружение и обработка. Удобно использовать в программе из нескольких модулей, не требуется возвращать значение в вызывающую ф-ию.
* Часть программы, которая будет выполняться до возникновения исключения.
* Также часть программы, которая «отлавливает» исключения и обрабатывает их.
* По значению, по ссылке, по константной ссылке, по указателю. Лучше всего передавать объект по ссылке, так как при этом не создается временный объект-исключение.
* Exception()
* Унаследовав стандартную.
* Int и double
* Любые
* По идее, в любой, может встретиться в блоке try в самом начале, но обрабатывается catch в конце.

#include<iostream>

#include<cmath>

#include<string>

using namespace std;

class error

{

private:

string msg;

public:

error(string msg)

{

this->msg = msg;

}

void what()

{

cout << "Произошла ошибка: " << msg << endl;

}

};

class triangle

{

protected:

int first; int second; int third;

float s; int p;

public:

triangle(int first, int second, int third)

{

this->first = first;

this->second = second;

this->third = third;

}

~triangle()

{}

void cur\_s\_4()

{

if (first <= 0 or second <= 0 or third <= 0) { throw error("Ошибочкааа"); }

p = first + second + third;

s = sqrt(p\*(p - first)\*(p - second)\*(p - third));

}

void cur\_s\_3()

{

int num = 2;

if (first <= 0 or second <= 0 or third <= 0) { throw num; }

p = first + second + third;

s = sqrt(p\*(p - first)\*(p - second)\*(p - third));

}

void cur\_s\_2()

{

if (first <= 0 or second <= 0 or third <= 0) { throw(1); }

p = first + second + third;

s = sqrt(p\*(p - first)\*(p - second)\*(p - third));

}

void cur\_s\_1()

{

if (first <= 0 or second <= 0 or third <= 0) { throw; }

p = first + second + third;

s = sqrt(p\*(p - first)\*(p - second)\*(p - third));

}

void show()

{

cout << "Площадь - " << s << endl;

}

};

int main()

{

setlocale(LC\_ALL, "rus");

int n;

cout << "Введите вариант исключения: ";

cin >> n;

cout << endl;

switch (n)

{

case 1:

try

{

triangle one(-5, 10, 15);

one.cur\_s\_1();

one.show();

}

catch (...)

{

cout << "АШИБАЧЬКА" << endl;

}

break;

case 2:

try

{

triangle two(-10, 15, 20);

two.cur\_s\_2();

two.show();

}

catch (int)

{

cout << "АШИБАЧЬКА "<< endl;

}

break;

case 3:

try

{

triangle three(-15, 20, 25);

three.cur\_s\_3();

three.show();

}

catch (int num)

{

cout << "АШИБАЧЬКА: " << num << endl;

}

break;

case 4:

try

{

triangle four(-20, 25, 30);

four.cur\_s\_4();

four.show();

}

catch (error er)

{

er.what();

}

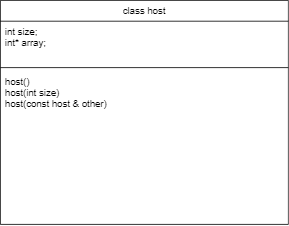
break;

}

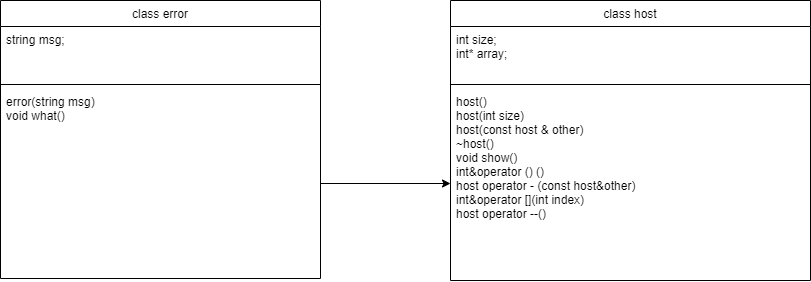
return 0;

* }

Блок-схема 1:



Блок-схема 2:



Код программы 1:

#include<iostream>

#include<ctime>

using namespace std;

int max\_size = 6;

class host

{

private:

int size;

int\* array;

public:

host()

{

size = 0;

array = 0;

}

host(int size)

{

this->size = size;

if (size > max\_size) { throw 1; }

array = new int[size];

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

{

array[i] = rand() % 100 + 1;

}

}

host(const host & other)

{

this->size = other.size;

array = new int[size];

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

{

array[i] = other.array[i];

}

}

~host()

{

}

void show()

{

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

{

cout << array[i] << " ";

}

cout << endl;

}

int&operator () () { return size; }

host operator - (const host&other)

{

host tmp(size);

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

{

tmp.array[i] = array[i] - other.array[i];

}

delete[]array;

return tmp;

}

int&operator [](int index)

{

if (index >= 0 && index < size)

{

return array[index];

}

else

{

cout << endl;

throw 2;

}

}

host operator --()

{

if (size == 0) { throw 3; }

if (size == 1)

{

size = 0;

delete[]array;

array = 0;

return\*this;

}

int tmp\_size = size--;

host tmp(tmp\_size);

for (int i = 0; i < tmp\_size; i++)

{

tmp.array[i] = 0;

}

for (int i = 0; i < tmp\_size; i++)

{

tmp.array[i] = array[i];

}

return tmp;

delete[]array;

}

};

int main()

{

setlocale(LC\_ALL, "rus");

srand(time(0));

try

{

host first(3);//host first(0) - ошибка при удалении. host first(8)-ошибка при создании

first.show();

--first;

first.show();

cout << "Элемент с избранным индексом: " << first[0] << endl;//first[5] - ошибка вывода, такого элемента нет.

cout << "Размер множества: " << first() << endl;

host second(2);

second.show();

host third(2);

third = first - second;

third.show();

}

catch (int num)

{

cout << "Ошибка с кодом: " << num << endl;

cout << "1 - выбранный размер множества недопустим. 2 - выбранный индекс недопустим. 3 - нельзя удалить элемент, размер равен 0" << endl;

}

return 0;

}

Код программы 2:

#include<iostream>

#include<ctime>

#include<string>

using namespace std;

int max\_size = 6;

class error

{

private:

string msg;

public:

error(string msg)

{

this->msg = msg;

}

void what()

{

cout << "Произошла ошибка: " << msg << endl;

}

};

class host

{

private:

int size;

int\* array;

public:

host()

{

size = 0;

array = 0;

}

host(int size)

{

this->size = size;

if (size > max\_size) { throw error("Недопустимый размер множества!"); }

array = new int[size];

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

{

array[i] = rand() % 100 + 1;

}

}

host(int size, int\*array)

{

this->size = size;

array = new int[size];

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

{

this->array[i] = array[i];

}

}

host(const host & other)

{

this->size = other.size;

array = new int[size];

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

{

array[i] = other.array[i];

}

}

~host()

{

}

void show()

{

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

{

cout << array[i] << " ";

}

cout << endl;

}

int&operator () () { return size; }

host operator - (const host&other)

{

host tmp(size);

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

{

tmp.array[i] = array[i] - other.array[i];

}

delete[]array;

return tmp;

}

int&operator [](int index)

{

if (index >= 0 && index < size)

{

return array[index];

}

else

{

cout << endl;

throw error("Недопустимый индекс");

}

}

host operator --()

{

if (size == 0) { throw error("Невозможно удалить элемент, размер множества равен 0"); }

if (size == 1)

{

size = 0;

delete[]array;

array = 0;

return\*this;

}

int tmp\_size = size--;

host tmp(tmp\_size);

for (int i = 0; i < tmp\_size; i++)

{

tmp.array[i] = 0;

}

for (int i = 0; i < tmp\_size; i++)

{

tmp.array[i] = array[i];

}

return tmp;

delete[]array;

}

};

int main()

{

setlocale(LC\_ALL, "rus");

srand(time(0));

try

{

host first(3);//host first(0) - ошибка при удалении. host first(8)-ошибка при создании

first.show();

--first;

first.show();

cout << "Элемент с избранным индексом: " << first[0] << endl;//first[5] - ошибка вывода, такого элемента нет.

cout << "Размер множества: " << first() << endl;

host second(2);

second.show();

host third(2);

third = first - second;

third.show();

}

catch (error ER)

{

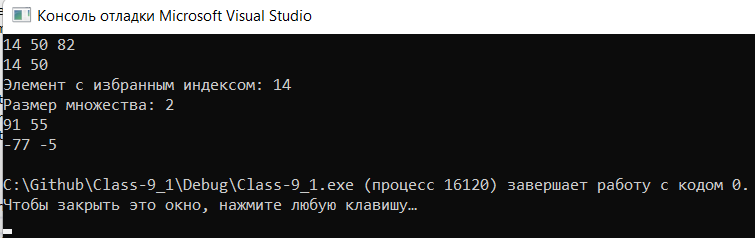
ER.what();

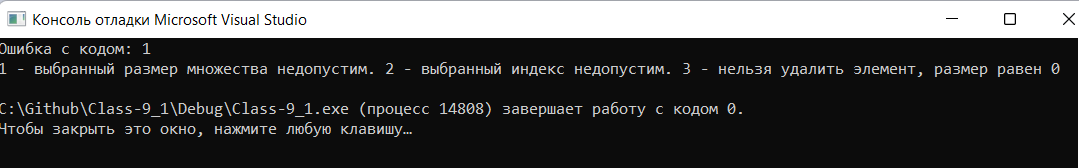
}

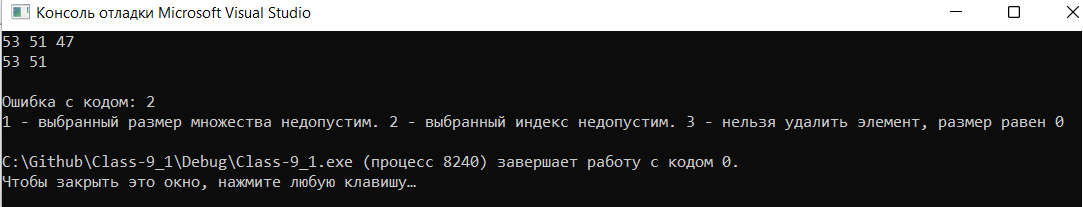
return 0;

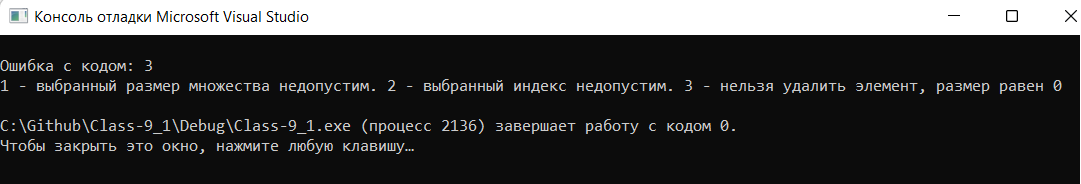
}

Скриншоты результатов 1:

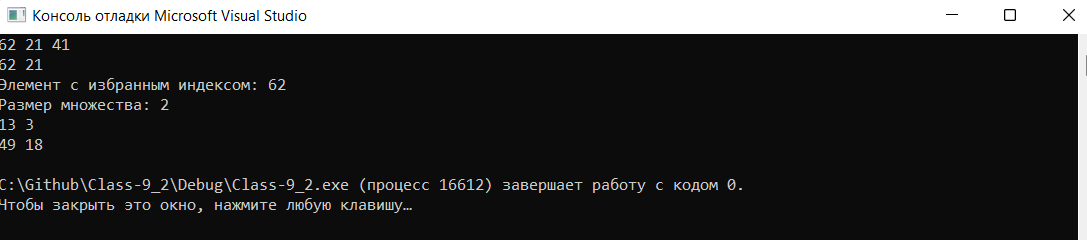


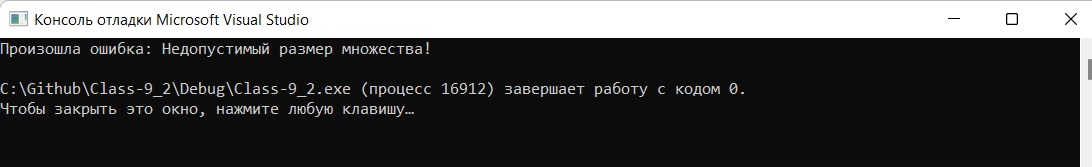


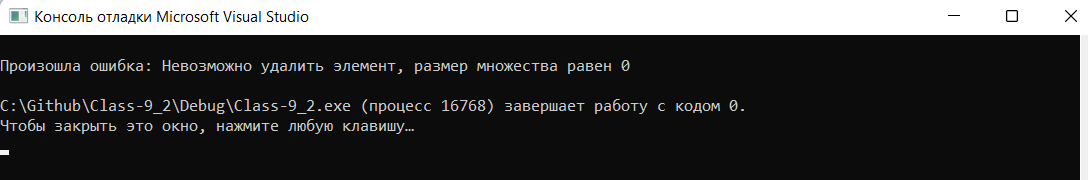


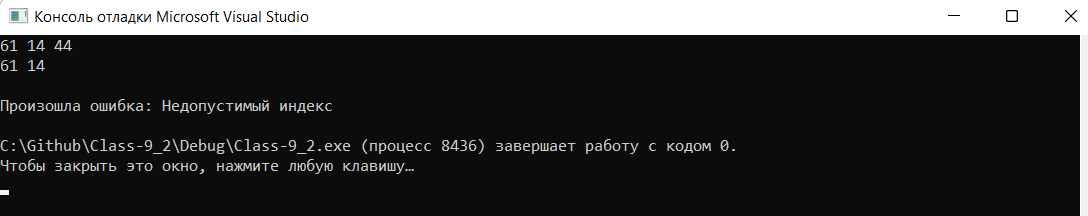


Скриншоты результатов 2:









Анализ результатов :

Программа работает корректно. Выводятся результаты без ошибок, а также продемонстрирована каждая ошибка в обоих вариантах.