

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Пермский национальный исследовательский политехнический
университет»
Электротехнический факультет
Кафедра «Информационные технологии и автоматизированные
системы»

ЛАБОРАТОРНАЯ РАБОТА №9
«Обработка исключительных ситуаций»

Выполнила:
студентка группы ИВТ-23-26
Соловьева Екатерина
Александровна

Проверила:
доцент кафедры ИТАС
О. А. Полякова

2024 г.

Постановка задачи

1. Реализовать класс, перегрузить для него операции, указанные в варианте.
2. Определить исключительные ситуации.
3. Предусмотреть генерацию исключительных ситуаций.

11 Вариант:

Класс- контейнер СПИСОК с ключевыми значениями типа int. Реализовать операции: [] – доступа по индексу; int() – определение размера списка; + вектор – сложение элементов списков a[i]+b[i]; - n - переход влево к элементу с номером n.	2 , 3
---	-------

Код программы на C++

List.h

```
#pragma once
#include <iostream>

using namespace std;
const int MAX_SIZE = 20;

class List {
    int size, * beg;
public:
    List() { size = 0; beg = 0; }
    List(int L);
    List(int L, int* arr);
    List(const List& list);
    ~List();

    List& operator=(const List& list);
    int operator[](int index);

    List operator +(const List& other);
    int operator -(int n);

    friend ostream& operator << (ostream& out, const List& list);
    friend istream& operator >> (istream& in, List& list);
};
```

Error.h

```
#pragma once
#include <iostream>
#include <string>
```

```
using namespace std;

class Error {
    string str;
public:
    Error(string S) { str = S; }
    void what() { cout << str << endl; }
};
```

List.cpp

```
#include "List.h"
#include <iostream>
#include "Error.h"

using namespace std;

List::List(int l) {
    size = l;
    beg = new int[l];
    for (int i = 0; i < size; i++) { beg[i] = 5; }
}

List::List(int l, int* arr) {
    if (l > MAX_SIZE) throw Error("Length > 20\n");
    size = l;
    beg = new int[size];
    for (int i = 0; i < size; i++) { beg[i] = arr[i]; }
}

List::List(const List& list) {
    size = list.size;
    beg = new int[size];
    for (int i = 0; i < size; i++) {
        beg[i] = list.beg[i];
    }
}

List::~List() {
    if (beg != 0) { delete[] beg; }
}

List& List::operator=(const List& list) {
    if (this == &list) { return *this; }
    if (beg != 0) { delete[] beg; }
    size = list.size;
    beg = new int[size];
    for (int i = 0; i < size; i++) { beg[i] = list.beg[i]; }
    return *this;
}

int List::operator[] (int index) {
    if (index > size) { throw Error("index > size"); }
    if (index < 0) { throw Error("Index < 0 or > size"); }
    return beg[index];
}

List List::operator+(const List& other) {
    List result;
    if (size != other.size) { return result; }
    result.size = size;
    result.beg = new int[size];
    for (int i = 0; i < size; i++) { result.beg[i] = beg[i] + other.beg[i]; }
    return result;
}

int List::operator-(int n) {
```

```

        if (n >= 0 && n < size) { return beg[n]; }
        else { throw Error("Error n"); }
    }
ostream& operator << (ostream& out, const List& list) {
    if (list.size == 0) { out << "Empty" << endl; }
    else {
        for (int i = 0; i < list.size; i++) { out << list.beg[i] << " "; }
        out << endl;
    }
    return out;
}
istream& operator >> (istream& in, List& list) {
    for (int i = 0; i < list.size; i++) {
        cout << "> ";
        in >> list.beg[i];
    }
    return in;
}
}

```

Main.cpp

```

#include <iostream>
#include "Error.h"
#include "List.h"

using namespace std;

int main() {
    try {
        List x(15), y;
        int i;

        cout << "Result with parametres: " << x << endl;
        cout << "Number: "; cin >> i;

        cout << "Result: " << x[i] << endl;

        y = x + i;

        cout << "Result y = " << y;
    }
    catch (Error e) {
        e.what();
    }
    return 0;
}

```

Вывод:

```

}
Result with parametres: 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
}
Number: 3
Result: 5
Result y = Empty

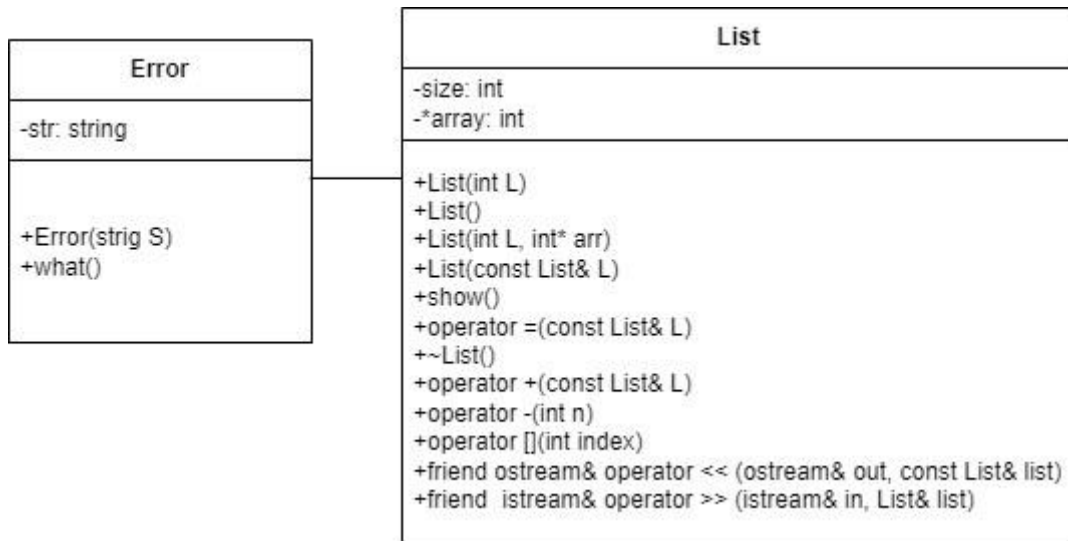
```

```

Result with parametres: 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
Number: 20
index > size

```

UML-диаграмма



Анализ результатов

Программа сработала корректно и вывела необходимые результаты.