Министерство науки и высшего образования РФ

 Пензенский государственный университет Кафедра "Вычислительная техника"

Отчет по лабораторной работе №5

Работу выполнили студенты группы 23ВВВ3 бригады №9:

Полиневский Вадим

Кизым Иван

Проверили:

Кандидат т.н. Гудков А.А

Кандидат т.н. Евсеева И. Е

Пенза 2025

**Цель работы**: Изучить механизмы наследования в языке C++

**Задаие :** 9.

1. Доработать шаблонный класс из предыдущей лабораторной работы таким образом, чтобы в программе осуществлялась генерация и обработка исключительных ситуаций.

Исключительные ситуации генерируются:

1. В конструкторе с параметром при попытке создать контейнер больше максимального

размера;

1. В операции []
2. При попытке обратиться к элементу с номером меньше 0 или больше текущего размера контейнера ;
3. В операции +
4. При попытке добавить элемент с номером больше максимального размера;
5. В операции –
6. При попытке удалить элемент из пустого контейнера.
7. Создать программу, использующую механизм обработки исключительных ситуаций, позволяющую выполнять перевод вещественного числа из одной системы счисления в другую.

Варианты заданий:

1. Перевод числа из восемнадцатеричной системы счисления в двенадцатеричную.

**Описание практической части**:

Для выполнения программы было использовано два модуля: **lab5.cpp** с функцией main и файл заголовка **header.h.**

Генерация и обработка исключений выполнена для передачи пустого вектора в функии Remove\_Duplicates после перегрузки оператора + в классе Set; при условии превышения предусмотренного размера вектора в цикле при пользовательском вводе; а также при перегрузке оператора () при передаче пустого вектора.

Листинг:

**lab5.cpp:**

#include <iostream>

#include <vector>

#include "header.h"

using namespace std;

int main (){

cout << "lab4" << endl;

vector <int> plenty\_1;

vector <int> plenty\_2;

vector <int> plenty\_3;

//vector for exception withou initialization

vector <int> plenty\_4;

// Check exception for iterator > max\_size\_vector;

try {Enter\_Plenty(plenty\_1, 1);}

catch(const exception& ex){cerr << ex.what() << '\n';}

Enter\_Plenty(plenty\_2, 2);

Enter\_Plenty(plenty\_3, 3);

Operation\_Planty(plenty\_1, plenty\_2, plenty\_3, plenty\_4);

Conversion\_eighteen\_digit\_systeme\_to\_duodecimal();

}

**header.h:**

#pragma once

#include <string>

#include <algorithm>

#include <unordered\_set>

#include <stdexcept>

//#include <exception>

using namespace std;

template <typename Type>

class Set{

private:

vector <Type> plenty;

void Remove\_Duplicates(vector <Type>& this\_vector) {

// check zero

if (this\_vector.empty()) {throw runtime\_error("Exception: vector passed to the funtion <Remove\_Duplicates> is empty\n");}

//

unordered\_set <Type> s;

vector <Type> result;

for (Type value : this\_vector){

if(s.find(value) == s.end()){

s.insert(value);

result.push\_back(value);

}

}

this\_vector.clear();

this\_vector = result;

}

public:

void Print\_Vector(string Out\_Info) {

cout << Out\_Info<< endl;

for (int num : plenty) {

cout << num << "\t";

}

cout << endl;

}

void operator () (const vector <Type>& temp\_vector) {

if (temp\_vector.empty()) throw runtime\_error ("Exception: this vector is empty\n");

this->plenty = temp\_vector;

}

void operator + (const Set& \_Set) {

this->plenty.insert(plenty.end(), \_Set.plenty.begin(), \_Set.plenty.end());

try { this->Remove\_Duplicates(this->plenty);}

catch (const exception & ex){cout << ex.what();}

//

}

bool operator <= (const Set& \_Set) {

if (this->plenty.size() > \_Set.plenty.size()) return false;

else return true;

}

};

void Print\_Plenty (Set<int> set\_1, Set<int> set\_2, Set<int> set\_3) {

set\_1.Print\_Vector("SET1 :");

set\_2.Print\_Vector("SET2 :");

set\_3.Print\_Vector("SET3 :");

}

void Enter\_Plenty(vector <int>& temp\_vector, int number) {

const int count = 5;

int temp\_value;

cout << " ENTER " << number << " PLANTY:" << endl;

// size + 1 in for

if (number == 1){

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

if (i >= count) {throw runtime\_error("Exception: iterator >= size your vector");}

cout << i << ": ";

cin >> temp\_value;

temp\_vector.push\_back(temp\_value);

}

}

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

cout << i << ": ";

cin >> temp\_value;

temp\_vector.push\_back(temp\_value);

}

}

void Operation\_Planty(vector <int> temp\_1, vector<int> temp\_2, vector <int> temp\_3, vector <int> temp\_4){

Set <int> Set\_1, Set\_2, Set\_3, Set\_4;

Set\_1(temp\_1);

Set\_2(temp\_2);

Set\_3(temp\_3);

// Check exception with empty vector

try{Set\_4(temp\_4); }

catch(const std::exception& ex){std::cerr << ex.what() << '\n';}

Print\_Plenty(Set\_1, Set\_2, Set\_3);

cout << "\n" << endl;

Set\_1 + Set\_2;

//operator with exception

try {Set\_4 + Set\_4;}

catch(const std::exception& e){std::cerr << e.what() << '\n';}

if (Set\_1 <= Set\_2) cout << "Set\_1 <= Set\_2"<<endl;

else cout << "Set\_1 > Set\_2";

cout << "\n" << endl;

Print\_Plenty(Set\_1, Set\_2, Set\_3);

}

int eighteen\_digit\_to\_decimal(const string& eig){return stoi(eig, nullptr, 18);}

string decimal\_To\_doudecimal (int decimal) {

string base\_12 = "";

while (decimal > 0) {

int remainder = decimal % 12;

if (remainder < 10) base\_12 += (char)('0' +remainder);

else base\_12 += char('A' + (remainder - 10));

decimal /=12;

}

reverse(base\_12.begin(), base\_12.end());

if (base\_12.empty()) return "0";

else return base\_12;

}

void Conversion\_eighteen\_digit\_systeme\_to\_duodecimal(){

cout << endl;

string temp\_value;

cout << "ENTER VALUE IN HEXANDEMICAL SYSTEM: ";

cin >> temp\_value;

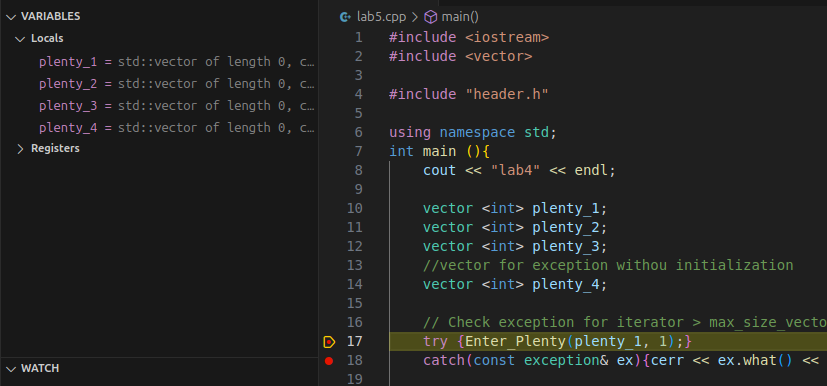
temp\_value = decimal\_To\_doudecimal(eighteen\_digit\_to\_decimal(temp\_value));

cout << "DUOCIMAL VALUE: " << temp\_value << endl;

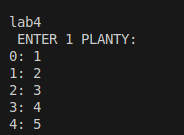
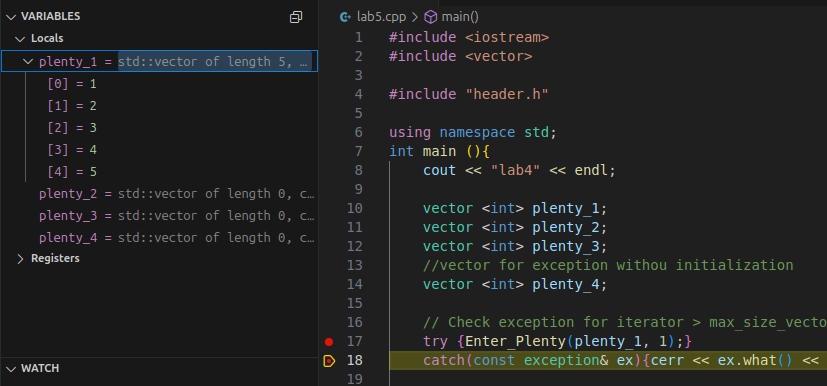
}

**Отладка:**

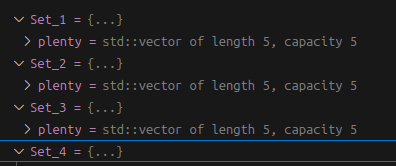
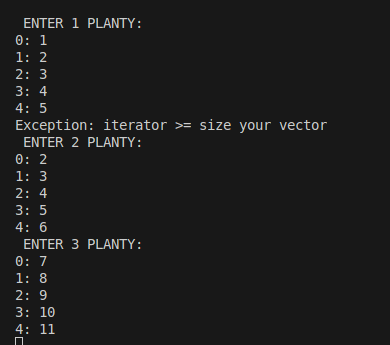
Первая точка остановки: векторы до передачи в множества.

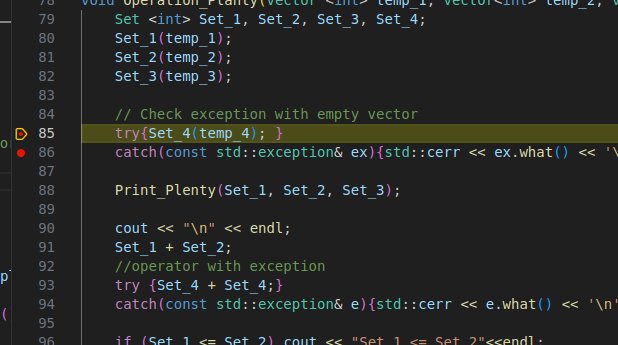


Вторая точка остановки: ввод переменных в первый вектор.

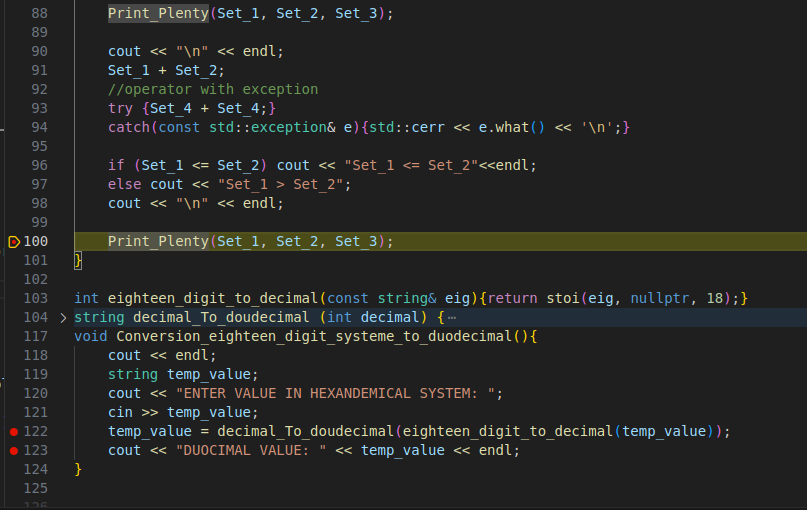


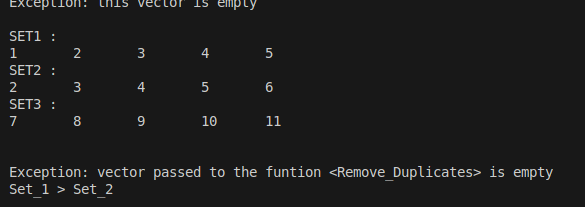
Третья точка остановки: вывод исключения. Ввод остальных векторов и передаче их в множества:



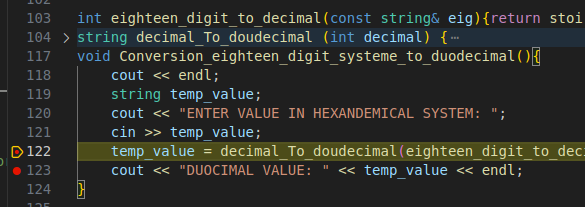


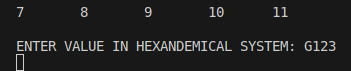
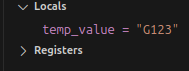
Пятая точка остновки: вывод 2 исключений:



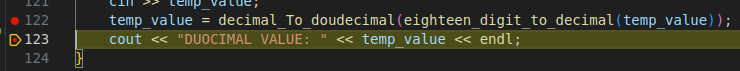
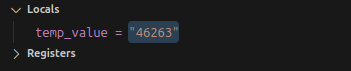


Шестая точка остановки: ввод 18-ричного числа.





Седьмая точка остановки: вывод 12-ричного числа:



**Terminal-Log:**

lab4

ENTER 1 PLANTY:

0: 1

1: 2

2: 3

3: 4

4: 5

Exception: iterator >= size your vector

ENTER 2 PLANTY:

0: 2

1: 3

2: 4

3: 5

4: 6

ENTER 3 PLANTY:

0: 7

1: 8

2: 9

3: 10

4: 11

Exception: this vector is empty

SET1 :

1 2 3 4 5

SET2 :

2 3 4 5 6

SET3 :

7 8 9 10 11

Exception: vector passed to the funtion <Remove\_Duplicates> is empty

Set\_1 > Set\_2

SET1 :

1 2 3 4 5 6

SET2 :

2 3 4 5 6

SET3 :

7 8 9 10 11

ENTER VALUE IN HEXANDEMICAL SYSTEM: G123

DUOCIMAL VALUE: 46263