

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ**

федеральное государственное автономное образовательное  
учреждение высшего образования

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

Инженерная школа информационных технологий и робототехники  
Отделение информационных технологий  
Направление информатика и вычислительная техника

**Отчет**

по лабораторной работе № 5

по дисциплине

«ПРОГРАММИРОВАНИЕ»

Вариант № 26

**Прототипы функций. Многофайловые проекты. Конфликты имён.  
Обработка исключений try-catch**

Выполнил:

Студент группы 8В32

\_\_\_\_\_

Д.О.Карташов

Проверил:

Ассистент ОИТ ИШИТР

\_\_\_\_\_

А.Ю.Малкин

Томск 2024

## Цель работы

Получить навыки работы с прототипами функций, реализации многофайловых проектов, обработки исключений try-catch с применением системы автоматизации сборки программного обеспечения CMake в среде разработки Visual Studio Code для создания программы, написанной на языке программирования C++.

## Задание

Реализовать с применением связки технологий CMake, GCC, VS Code. Допускается создание различных проектов для различных заданий. Очередные проекты выполнять как многофайловые: логика выполнения выносится в отдельный файл, в main.cpp остаётся линковка с нужными заголовочными файлами, определение параметров для функций, их вызов.

1. Модернизировать проект с лабораторной работы про длинную арифметику: функции расчёта чисел с применением контейнера vector вынести в отдельный cpp-файл, в main.cpp осуществить линковку с заголовочными файлами, вызов функций.

2. Реализовать функцию, которая возвращает частное от деления двух знаковых целых чисел (signed integer). Если ответ выходит с дробной частью, то округляем его в сторону нуля. Запрещается использование операторов умножения (\*), деления (/), взятия остатка от деления (%), вычитания (-) при взятии разности делимого непосредственно от делителя. Предписывается использование любых типов данных, размерность которых не превышает 32 бит. Если частное выходит за диапазон значений типа signed int, вернуть ближайшее к этому диапазону число.

3. Реализовать случай, когда в стеке вызовов находятся 6 функций, первая снизу - функция main. Реализовать обработку исключений типа int для функций, которые находятся третьей и четвёртой снизу в стеке вызовов. Реализовать обработку исключений типа double для функции, которая находится третьей снизу в стеке вызовов. Осуществить вызов исключения типа double в верхней (шестой) функции. Каждая функция, начиная со второй снизу и вверх по стеку вызовов, должны осуществлять простые действия (например, печать строки или вычисление выражения 1+2), которые будут выполняться до блока catch и в конце функции.

## Ход работы

На рисунке 1 демонстрируется результат работы программы из задания 1. Действия программы идентичны программе в лабораторной №2.

```
55
1269640335365827592596510084756651695958032105144943676227584000000000000
64
18446744073709551616
32 64
18446744078004518912
100 50
1267650600228228275596796362752
555
43516638122555047989641805373140394725407202037260729735885664398655775748034950972577909265605502785297675867877570
```

Рисунок 1 – Результаты работы программы с длинной арифметикой

На рисунке 2 демонстрируются результаты деления 50 на 10, 50 на -10, 5 на 2, -50 на 100, 1 на 0 и MIN\_INT на -1 соответственно. В случае деления на 0 исключение обработано.

```
5
-5
2
0
You cannot divide by zero!
2147483647
```

Рисунок 2 – Результаты деления

На рисунке 3 демонстрируются вызовы и выходы из функций. Функции 3 – 6 были прерваны исключением, поэтому их конец никогда не был достигнут. В функции 2 исключение было обработано, о чём выведено соответствующее сообщение, и дальнейшее выполнение произошло без исключений.

```
1st function start
2nd function start
3rd function start
4th function start
5th function start
6th function start
In 2nd function exception of type double was caught (value = 3.14)
2nd function end
1st function end
```

Рисунок 3 – Демонстрация стека вызова функций

## **Вывод**

Получены навыки работы с прототипами функций, реализации много-файловых проектов, обработки исключений try-catch с применением системы автоматизации сборки программного обеспечения CMake в среде разработки Visual Studio Code для создания программы, написанной на языке программирования C++.

## **Приложение 1 – Ссылка на репозиторий с исходными кодами программ**

Ссылка на репозиторий: *<https://github.com/The-Aozzi/Lab5>*