

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
**«ПЕРМСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

ЭЛЕКТРОТЕХНИЧЕСКИЙ ФАКУЛЬТЕТ

КАФЕДРА «ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И
АВТОМАТИЗИРОВАННЫХ СИСТЕМ»

ОТЧЁТ

**«ЛАБОРАТОРНАЯ №9:
ОБРАБОТКА ИСКЛЮЧЕНИЙ»**

Дисциплина: «Программирование»

Выполнил:

Студент группы ИВТ-21-26

Безух Владимир Сергеевич

Проверил:

Доцент кафедры ИТАС

Полякова Ольга Андреевна

Пермь, 2022

Содержание

1.	Постановка задачи	3
2.	Контрольные вопросы	4
3.	Анализ результатов.....	7

1. Постановка задачи

Сделать класс-обработчик исключений `Error`. Продемонстрировать его работу на примере класса `SinglyList`.

2. Контрольные вопросы

1. Что представляет собой исключение C++?

Исключение — непредвиденное или аварийное событие. В C++ исключение — это объект, который система должна генерировать при возникновении исключительной ситуации. Генерация такого объекта и создаёт исключительную ситуацию.

2. На какие части исключения позволяют разделить вычислительный процесс?

Исключения позволяют разделить вычислительный процесс на 2 части:
— обнаружение аварийной ситуации (неизвестно как обрабатывать);
— обработка аварийной ситуации (неизвестно, где она возникла).

3. Какой оператор используется для генерации исключительной ситуации?

`throw <выражение>`, где `<выражение>`:
— либо константа;
— либо переменная некоторого типа;
— либо выражение некоторого типа.

4. Что представляет собой контролируемый блок?

```
try {  
    // some code  
}
```

Контролирующий блок проверяет возникновение исключения, с которым связана одна или несколько секций-ловушек `catch`.

5. Что представляет собой секция-ловушка?

catch (спецификация исключения), где спецификация исключения может иметь три формы:

- (тип имя);
- (тип);
- (...).

Тип — это встроенный тип или тип, определенный программистом.

6. Какие формы может иметь спецификация исключения в секции ловушке?

Формы 1 и 2 обрабатывают конкретные исключения, а форма 3 перехватывает все исключения, такую ловушку надо помещать последней, тогда она будет обрабатывать все исключения, которые ещё не были обработаны. Форма 1 означает, что объект передаётся в блок обработки, чтобы его каким-то образом там использовать, например, для вывода информации в сообщении об ошибке.

7. Какой стандартный класс можно использовать для создания собственной иерархии исключений?

Класс exception.

8. Каким образом можно создать собственную иерархию исключений?

Для создания собственной иерархии исключений надо объявить свой базовый класс-исключение, например:

```
class BaseException{ };
```

Остальные классы будут наследниками этого класса, аналогично тому, как это сделано в иерархии стандартных исключений:

```
class ChildException1 : public BaseException{ };
```

```
class ChildException : public BaseException{ };
```

Класс BaseException можно унаследовать от стандартного класса exception, это позволит использовать метод what для вывода сообщений об ошибках.

3. Анализ результатов

Результаты выполнения программы (рис. 1). Если попытаться в список типа `string` вставить число, то будет брошено исключение.

```
numbers.pushBack(1);
strings.pushBack("string");

/*try
{
    setInteger(numbers, 0, 21);
}*/

try
{
    setInteger(strings, 0, 21);
}

catch (Error& error)
{
    error.printMessage();
}
```

```
SinglyList<int> numbers;
SinglyList<std::string> strings;

This is non-numbers type container!
1
string

template <typename T>
void setInteger(T& data, const int& index, const int& integer)
{
    if (std::is_arithmetic_v<decltype(data[0])>())
        data.insert(index, std::to_string(integer));
    else
        throw
        Error("This is non-numbers type container!"); // throw 1;
}
```

Рисунок 1 — Результаты