

ФГАОУ ВО «Санкт-Петербургский политехнический университет Петра Великого» Институт компьютерных наук и технологий

Высшая школа киберфизических систем и управления

по дисциплине Практикум по программированию

Выполнил студент группы 23533/2

Руководитель

ассистент

В.Э. Ковалевский

Разработаем программу, в которой обработаем часто встречающиеся на практике исключения, а также создадим собственное исключение.

Санкт-Петербург 2019

30.05.2019

А.А. Тарасов

Отчет по лабораторной работе

Обработка исключительных ситуаций

Программа – простой калькулятор, делящий числа. Число вводится как string, а затем переводится в double. Первое, что нужно обработать – попытку перевода string в double. Для этого код защищается блоком try. Если при выполнении try возникает исключение, то выполняется код из блока catch. Catch должен идти сразу за блоком try или другим catch.

cout << "Делим число 1 на 2" << endl;

cout << "Введите число 1: ";

cin >> input1;

while (cin.get() != '\n');

cout << "Введите число 2: ";

cin >> input2;

while (cin.get() != '\n');

double d1, d2, res;

NewException a;

try

{

d1 = stod(input1);

if (input2 == "Easteregg")

throw a;

else

d2 = stod(input2);

}

catch (NewException)

{

cout << a << endl;

continue;

}

catch (...)

{

cout << "Возникло исключение. Попробуйте снова." << endl;

continue;

}

В данном коде в блоке try используется функция stod – string to double. Если при её выполнение возникает исключение, например, был введён текст, то управление передаётся блоку catch(…). Catch(NewException) в данном случае не работает, так как тип исключения был не NewException. catch(…) ловит все возможные исключения, если они не были пойманы блоками выше.

Возможно генерировать исключение после какого-то события. Например, при делении не 0 выбрасывается 1. Catch ловит исключение по его типу, и возвращает сообщение.

try

{

if (d2 == 0)

throw 1;

else

{

res = d1 / d2; cout << res << endl;

}

}

catch (int i)

{

cout << "Было выполнено деление на 0" << endl;

}

Создадим собственное исключение. На самом деле исключением может быть любой класс (хотя бы базовый тип, как показано выше), ведь исключение ловят по его типу.

class NewException

{

const string msg = "An easter egg, haha";

friend ostream& operator<<(ostream& os, NewException a)

{

os << a.msg;

return os;

}

};

Это исключение хранит в себе сообщение. Также объект класса может быть использован с объектами ostream.

Если пользователь при вводе второго числа введёт «Easteregg», то сгенерируется исключение типа NewException, которое будет обработано в catch(NewException).

Итак, подводя итог: try – блок выполнения опасного кода. Если в try будет сгенерировано исключение, то его обработает catch. Исключения можно генерировать самому программисту. Такие исключения также обрабатываются в catch.

Заключение

В данной программе были разобраны основы обработки исключений. Было создано новое исключение, а затем успешно обработано. Исключения помогли защитить код от небезопасных действий.