

Исключения и их обработка

Работа с исключениями в языке Си отсутствует. Потому мы начинаем плавный переход в язык C++.

Задача. Дать пользователю ввести два числа. Попробовать поделить их, если они поделились – вывести результат на экран, иначе сообщить об ошибке и дать ввести новые числа. Операция деления должна быть отдельной функцией, возвращающей результат.

Решение с использованием языка Си без исключений.

С теми знаниями, которые у нас есть сейчас можно решить данную задачу, например следующим образом:

```
// библиотеки
#include <stdio.h>
#include <locale.h>
#include <math.h>

double div(double a, double b) {
    if (b == 0) {
        printf("Ошибка в функции 'div': деление на ноль\n");
        return NAN;
    }
    return a / b;
}

// объявление главной функции
int main() {
    // настройки
    setlocale(LC_ALL, "rus");

    // объявление переменных
    double result, num1, num2;

    while (1) {
        // инициализация переменных
        printf("Введите два числа: ");
        scanf_s("%lf %lf", &num1, &num2);

        // код
        result = div(num1, num2);
        if (isnan(result)) continue;
        else break;
    }

    // вывод результата
    printf("Result %.2lf / %.2lf = %.2lf\n",
        num1, num2, result);

    system("pause");
    return 0;
}
```

Нам пришлось добавить `return NAN`; так как функция должна что-то возвращать, а любое число 0, -1, ... может быть правильным ответом. Потому в качестве однозначно неправильного мы добавили not a number. Но на практике никто так не делает. Если вы обязаны писать код на Си, то придётся самим реализовывать исключения, однако в языке C++ они уже реализованы и воспользуемся мы именно ими.

Решение с использованием языка C++ с исключениями.

```
// библиотеки
#include <iostream>
#include <locale>
#include <exception>

// библиотеки
#include <iostream>
#include <locale>
#include <exception>

double div(double a, double b) {
    if (b == 0) {
        // кидаем исключение и прерываем им выполнение функции
        throw std::logic_error("Ошибка в функции 'div': деление на ноль\n");
    }
    return a / b;
}

// объявление главной функции
int main() {
    // настройки
    setlocale(LC_ALL, "rus");

    // объявление переменных
    double result, num1, num2;

    while (1) {
        // инициализация переменных
        printf("Введите два числа: ");
        scanf_s("%lf %lf", &num1, &num2);

        // код который пытаемся выполнить
        try {
            result = div(num1, num2);
        }
        catch (const std::exception& e) {
            // если выпало исключение, попадаем в этот блок и пишем текст ошибки
            printf("%s", e.what());
            continue;
        }
        break;
    }

    // вывод результата
    printf("Result %.2lf / %.2lf = %.2lf\n",
        num1, num2, result);

    system("pause");
    return 0;
}
```

Мы в данном примере использовали так называемую логическую ошибку. Вообще существуют разные виды исключений, класс логических ошибок включает более конкретные, например, выход за границы. Ознакомиться со всеми классами ошибок, которые вы можете использовать, можно тут: https://en.cppreference.com/w/cpp/error/logic_error

Если затрудняетесь что за вид ошибки в данном случае – можно всегда использовать `std::logic_error`.