

Программирование на C++



Минцифры
России

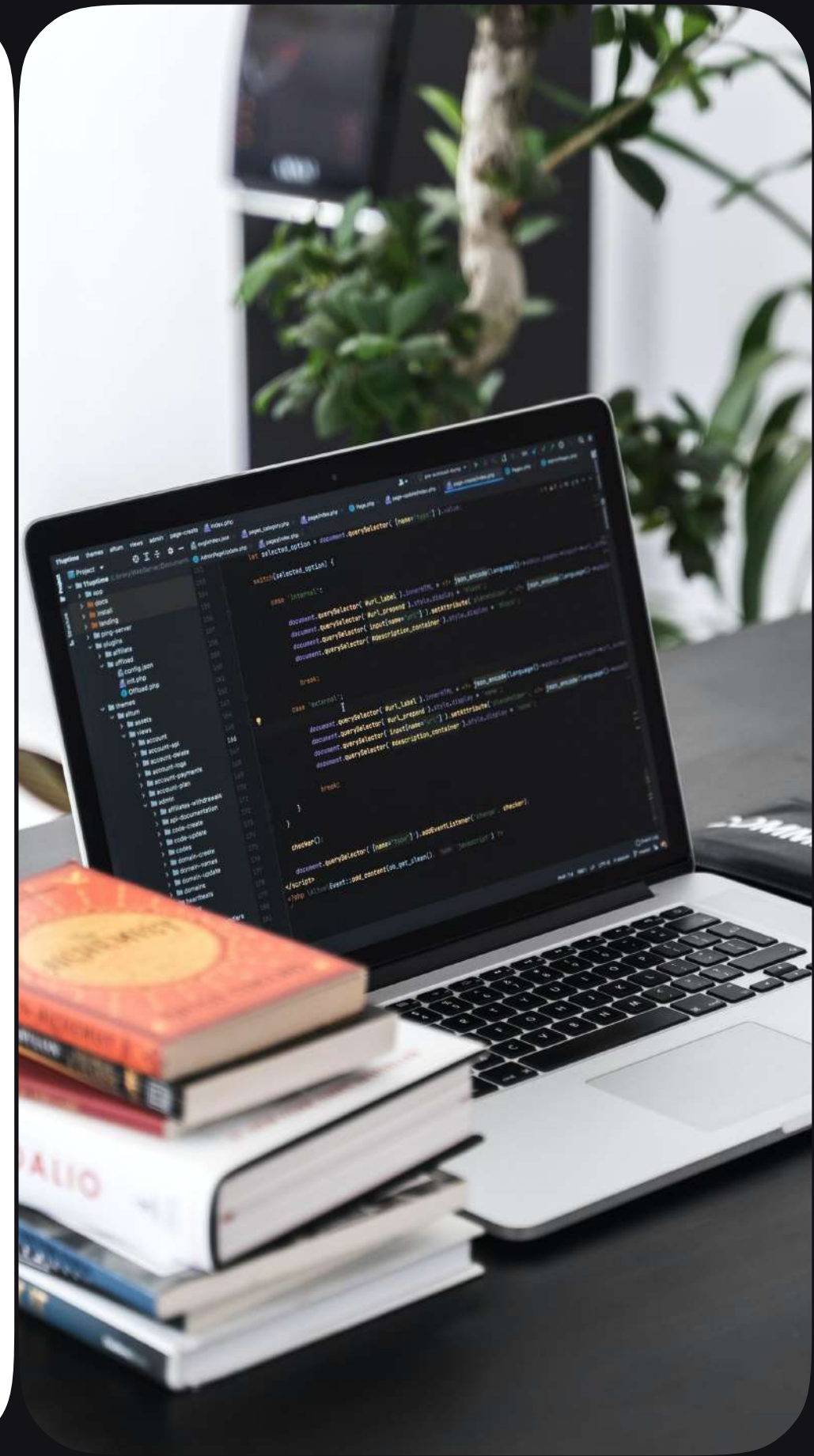
UCHi **DOMA**

20.35
УНИВЕРСИТЕТ

Урок 11 Модуль 3

Исключения

Полезные материалы



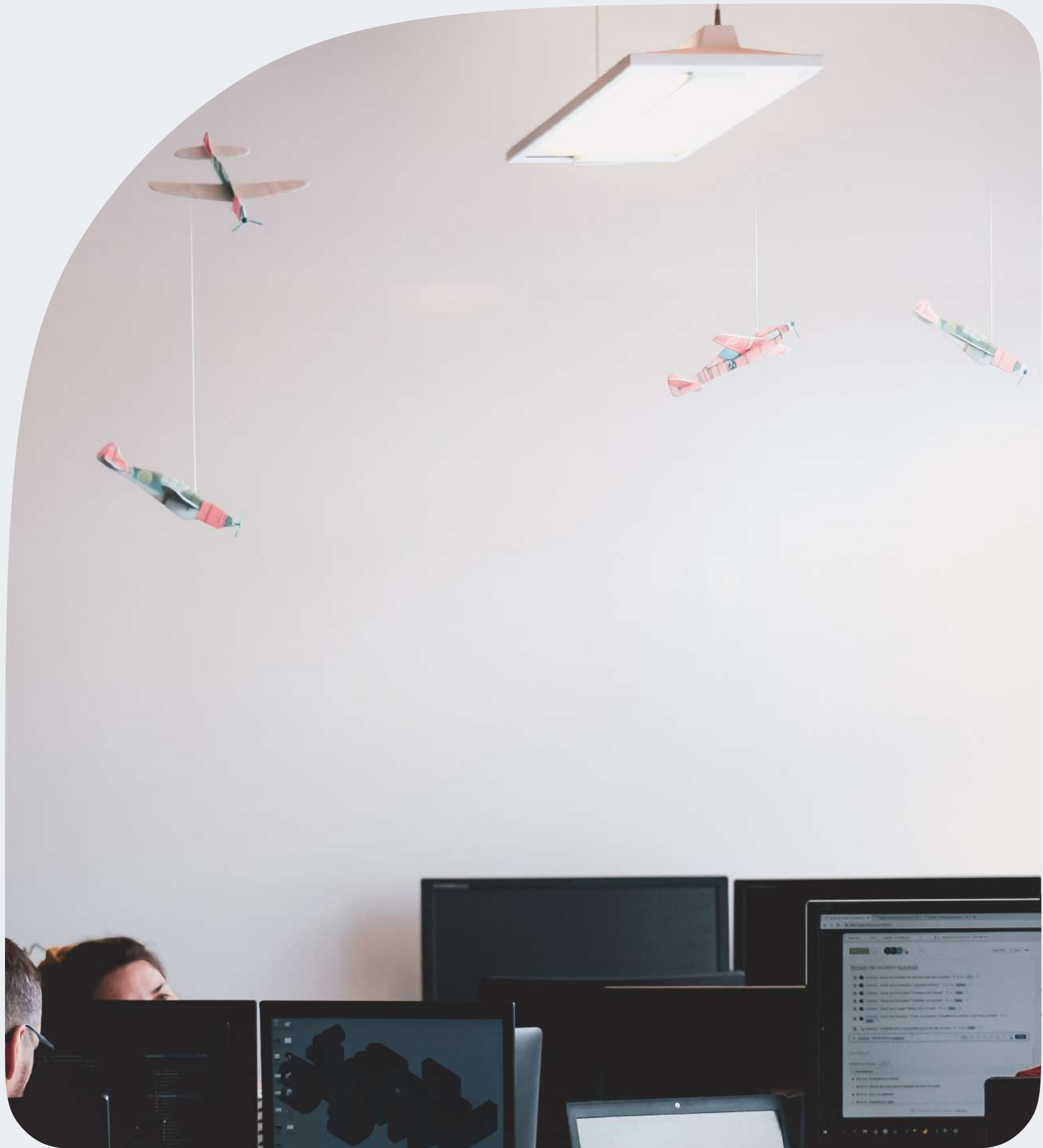
Цели урока



изучить обработку исключений



отработать на практике
написание алгоритмов
с обработкой исключений
на C++



Исключение



Исключение — это событие при выполнении программы, которое приводит к её ненормальному или неправильному поведению.

Исключения

Два вида исключений:

Аппаратные (структурные, SE-Structured Exception), которые генерируются процессором. К ним относятся, например:

- ✦ деление на 0
- ✦ выход за границы массива
- ✦ обращение к невыделенной памяти
- ✦ переполнение разрядной сетки

Программные, генерируемые операционной системой и прикладными программами — возникают тогда, когда программа их явно инициирует. Когда встречается аномальная ситуация, та часть программы, которая ее обнаружила, может сгенерировать, или **возбудить**, исключение.

Обработка исключений



Механизм структурной обработки исключений позволяет однотипно обрабатывать как программные, так и аппаратные исключения.

В блок после ключевого слова `try` помещается код, который потенциально может сгенерировать исключение.

После ключевого слова `catch` в скобках идет параметр, который передает информацию об исключении. Затем в блоке производится собственно обработка исключения.

```
try
{
    инструкции, которые могут вызвать исключение
}
catch(объявление_исключения)
{
    обработка исключения
}
```

Обработка исключений

```
catch (параметр) { // обработка исключения }  
catch (...) { // обработка остальных исключений }
```

Пример



```
1  #include <iostream>
2  using namespace std;
3  int main() {
4      int a = 0, b = 10;
5      try {
6          cout << b / a << endl;
7      }
8      catch (...)
9      {
10         cout << "Error";
11     }
12     cin.get();
13     return 0;
14 }
```


Обработка исключений

Но деление на 0 — не стандартное исключение, его необходимо генерировать самостоятельно с помощью `throw`.

Оператор **`throw`** генерирует исключение. Через оператор `throw` можно передать информацию об ошибке

Пример



```
1  #include <iostream>
2  using namespace std;
3  int main()
4  {
5      try
6      {
7          int a=5,b=0;
8          if (b==0) throw 1;
9          cout<<a/b<<endl;
10     }
11     catch(int x) {
12         cout<<"Error!"<< endl;
13     }
14     return 0;
15 }
```

Пример



```
1  #include <iostream>
2  using namespace std;
3  int main()
4  {
5      try
6      {
7          int a=5,b=0;
8          if (b==0) throw "division by zero";// выбрасываем исключение типа const char*
9          cout<<a/b<<endl;
10     }
11     catch (const char* exception) // ловим исключения типа const char*
12     {
13         cout<<exception<< endl;
14     }
15     return 0;
16 }
```